HITACHI
Inspire the Next

Hitachi Advanced Data Binder
システム構築・運用ガイド

3000-6-501-L0

前書き

■ 著作権

All Rights Reserved. Copyright (C) 2012, 2022, Hitachi, Ltd.

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI，HA モニタ，HiRDB，JP1 は，株式会社 日立製作所の商標または登録商標です。

Access は，マイクロソフト 企業グループの商標です。

AMD は，Advanced Micro Devices, Inc.の商標です。

Excel は，マイクロソフト 企業グループの商標です。

Intel は，Intel Corporation またはその子会社の商標です。

Linux は，Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は，マイクロソフト 企業グループの商標です。

Oracle および Java は，オラクルおよびその関連会社の登録商標です。

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat は，米国およびその他の国における Red Hat, Inc.の登録商標です。

Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat Enterprise Linux は，米国およびその他の国における Red Hat, Inc.の登録商標です。

UNIX は，The Open Group の登録商標です。

Visual C++は，マイクロソフト 企業グループの商標です。

Visual Studio は，マイクロソフト 企業グループの商標です。

Windows は，マイクロソフト 企業グループの商標です。

Windows Server は，マイクロソフト 企業グループの商標です。

その他記載の会社名，製品名などは，それぞれの会社の商標もしくは登録商標です。

1. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

2. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

3. This product includes software written by Tim Hudson (tjh@cryptsoft.com).

4. 本製品には OpenSSL Toolkit ソフトウェアを OpenSSL License および Original SSLeay License に従い使用しています。OpenSSL License および Original SSLeay License は以下の通りです。

LICENSE ISSUES

==========

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSD-style
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License

/* ==

* Copyright (c) 1998-2018 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in

* the documentation and/or other materials provided with the

* distribution.

*

* 3. All advertising materials mentioning features or use of this

* software must display the following acknowledgment:

* "This product includes software developed by the OpenSSL Project

* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

*

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to

* endorse or promote products derived from this software without

* prior written permission. For written permission, please contact

* openssl-core@openssl.org.

*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/
Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written

* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the rouines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*

* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

■Double precision SIMD-oriented Fast Mersenne Twister (dSFMT)

Copyright (c) 2007, 2008, 2009 Mutsuo Saito, Makoto Matsumoto
and Hiroshima University.
Copyright (c) 2011, 2002 Mutsuo Saito, Makoto Matsumoto, Hiroshima
University and The University of Tokyo.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided

with the distribution.
* Neither the name of the Hiroshima University nor the names of
its contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Hitachi Advanced Data Binder は，内閣府の最先端研究開発支援プログラム『超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価』（中心研究者：喜連川 東大特別教授／国立情報学研究所所長）の成果を利用して開発されました。

また，非順序型実行原理は，喜連川 東大特別教授／国立情報学研究所所長・合田 東大准教授が考案した原理です。

■ マイクロソフト製品のスクリーンショットの使用について

マイクロソフトの許可を得て使用しています。

■ マイクロソフト製品の表記について

このマニュアルでは，マイクロソフト製品の名称を次のように表記しています。

表記			製品名
Windows	Windows 7	Windows 7 x86	Windows(R) 7 Professional 日本語版(32 ビット版)
			Windows(R) 7 Enterprise 日本語版(32 ビット版)
			Windows(R) 7 Ultimate 日本語版(32 ビット版)
		Windows 7 x64	Windows(R) 7 Professional 日本語版(64 ビット版)
			Windows(R) 7 Enterprise 日本語版(64 ビット版)
			Windows(R) 7 Ultimate 日本語版(64 ビット版)

表記			製品名
	Windows 8.1	Windows 8.1 x86	Windows(R) 8.1 Pro 日本語版(32 ビット版)
			Windows(R) 8.1 Enterprise 日本語版(32 ビット版)
		Windows 8.1 x64	Windows(R) 8.1 Pro 日本語版(64 ビット版)
			Windows(R) 8.1 Enterprise 日本語版(64 ビット版)
	Windows 10	Windows 10 x86	Windows(R) 10 Pro 日本語版(32 ビット版)
			Windows(R) 10 Enterprise 日本語版(32 ビット版)
		Windows 10 x64	Windows(R) 10 Pro 日本語版(64 ビット版)
			Windows(R) 10 Enterprise 日本語版(64 ビット版)
	Windows Server 2008 R2		Windows Server(R) 2008 R2 Standard 日本語版
			Windows Server(R) 2008 R2 Enterprise 日本語版
			Windows Server(R) 2008 R2 Datacenter 日本語版
	Windows Server 2012		Windows Server(R) 2012 Standard 日本語版
			Windows Server(R) 2012 Datacenter 日本語版
	Windows Server 2012 R2		Windows Server(R) 2012 R2 Standard 日本語版
			Windows Server(R) 2012 R2 Datacenter 日本語版
	Windows Server 2016		Windows Server(R) 2016 Standard 日本語版
			Windows Server(R) 2016 Datacenter 日本語版
	Windows Server 2019		Windows Server(R) 2019 Standard 日本語版
			Windows Server(R) 2019 Datacenter 日本語版

■ 発行

2022 年 2 月

変更内容

変更内容(3000-6-501-L0)　Hitachi Advanced Data Binder　05-05

追加・変更内容		変更個所
ワーカーノードのサポート	マルチノード機能のノードの種類にワーカーノードを追加しました。ワーカーノードの HADB サーバでは，セカンダリノードの HADB サーバと同様に検索系 SQL（SELECT 文）の処理が実行されます。 ワーカーノードはプライマリノードの切り替え先にはならないため，HA モニタをインストールする必要はありません。 ワーカーノードの概要については，右記の変更個所で太字表示されている個所を参照してください。	2.19，2.20.2，7.1，7.2.1，7.2.2，7.2.3，7.2.5，7.2.8，7.2.10，7.2.11，12.9.2，12.9.3，12.10.2(3)，12.10.2(4)，16.1，16.2，16.2.1，16.2.2，16.2.3，16.2.4，16.2.4(1)，16.2.4(3)，16.3.1，16.3.5，16.3.7，16.3.8(1)，16.3.8(4)，16.4.1(1)，16.4.1(3)，16.4.1(4)，16.4.2(1)，16.4.2(2)(e)，16.4.2(2)(f)，16.4.2(3)，16.4.3，16.5.1，16.5.3，16.5.4，16.8.3，16.8.4(1)，16.9.1，16.10.2，16.10.3，16.10.6，16.11.2(1)，16.11.4(1)，16.11.5，16.11.6，16.11.7，16.15.1(1)，16.15.1(3)，16.15.2(1)，16.15.3(1)，16.16，16.17.1(1)，16.17.1(2)，16.17.3(2)，16.17.9，16.17.10(3)(b)，16.17.10(3)(c)，16.17.10(3)(d)，16.17.10(3)(e)，16.17.13，16.19，16.20，16.21.5，16.21.6，16.23.2，16.25，16.26.1(6)，16.26.2，16.26.5，16.27.1，16.27.3，付録 D.1
SQL パラレル実行機能のサポート	マルチノード機能の使用時，検索系 SQL（SELECT 文）を複数のノードで並列実行する機能（SQL パラレル実行機能）をサポートしました。 SQL パラレル実行機能の概要については，右記の変更個所で太字表示されている個所を参照してください。	2.19.8，2.20.2，5.6.1(3)，6.25.3，7.1，7.2.2，7.2.10，7.2.11，10.11.2，10.11.2(3)，10.11.2(7)，10.11.2(10)，10.11.2(13)，10.11.3(1)，10.11.4，10.11.11，12.9.1，12.9.2，16.3.5，16.3.6，16.5.3，16.5.4，16.9.7，16.11.2(1)，16.29，付録 D.2
見積もり式に関する変更	次に示すカーネルパラメタの見積もり式を変更しました。 • threads-max • aio-max-nr • nofile • nproc	6.2

追加・変更内容		変更個所
	HADB サーバ開始時のメモリ所要量の見積もり式を変更しました。	6.3.3(1)，6.3.3(3)(a)，6.3.3(3)(e)，6.3.3(4)
	通常運用時のメモリ所要量の見積もり式を変更しました。	6.3.4(1)(b)，6.3.4(2)(c)
	adbimport コマンド実行時のメモリ所要量の見積もり式を変更しました。	6.3.6(2)(d)
データ用 DB エリアの使用率の監視	データ用 DB エリアの使率量が，サーバ定義の adb_dbarea_wrn_pnt_use_rate オペランドの指定値に達したときに，警告メッセージKFAA41230-W を出力するようにしました。 詳細については，右記の変更個所で太字表示されている個所を参照してください。	7.1，7.2.4，10.5，**11.10.7**
バージョンアップに関する変更	バージョンアップ時の注意事項（バージョンアップ時に行われるビュー表の再作成）の説明を変更しました。	8.6.5(5)
	バージョンアップ時のリアルスレッド数の増加についての説明を追加しました。	8.6.5(9)
メッセージログファイルの切り替えを知らせるメッセージの出力	メッセージログファイルの切り替え時に，KFAA80099-I メッセージを出力するようにしました。	10.4.3，10.4.4，10.5
次に示す用語の説明を変更または追加しました。 • SQL サブノード • SQL パラレル実行機能 • SQL メインノード • ノード • ノード障害 • ノードの切り離し • ノードの復帰 • プライマリノード • マルチノード構成からのワーカーノードの削除 • マルチノード構成へのワーカーノードの追加 • ワーカーノード		用語解説

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは，Hitachi Advanced Data Binder のシステム設計，構築，および運用方法について説明したものです。

なお，このマニュアル中，および製品が出力する情報中（メッセージ，コマンドの出力結果など）では，Hitachi Advanced Data Binder を HADB と表記することがあります。

■ 対象製品

- P-9W62-C411 Hitachi Advanced Data Binder 05-05　（適用 OS：Red Hat Enterprise Linux Server 6（64-bit x86_64），Red Hat Enterprise Linux Server 7（64-bit x86_64），Red Hat Enterprise Linux Server 8（64-bit x86_64））
- P-9W62-C311 Hitachi Advanced Data Binder Client 05-05　（適用 OS：Red Hat Enterprise Linux Server 6（64-bit x86_64），Red Hat Enterprise Linux Server 7（64-bit x86_64），Red Hat Enterprise Linux Server 8（64-bit x86_64））
- P-2462-C114 Hitachi Advanced Data Binder Client 05-05　（適用 OS：Windows 7，Windows 8.1，Windows 10，Windows Server 2008 R2，Windows Server 2012，Windows Server 2012 R2，Windows Server 2016，Windows Server 2019）

■ 対象読者

このマニュアルは，次に示す方々を対象にしています。

- HADB のシステム設計・構築者およびシステム管理者
- AP 開発者

なお，このマニュアルは次に示す知識があることを前提に説明しています。

- Linux および Windows のシステム管理の基本的な知識
- RDBMS の運用管理の基本的な知識
- SQL の基本的な知識

■ マニュアルの構成

このマニュアルは，次に示す編，章と付録から構成されています。

第 1 編　解説編

第 1 章　概要

HADB の概要，特長，およびシステム構成について説明しています。

第 2 章　アーキテクチャ

HADB のデータベースの論理構造，DB エリアの概要，ユーザとスキーマの概要，ユーザ認証機能，権限，データベースのアクセス処理方式，トランザクション制御，排他制御，データベースの回復処理，クライアントグループ機能，クライアント定義の集中管理機能，バックグラウンドインポート機能，チャンクアーカイブ機能，CSV ファイル中のデータの検索，テキストデータの検索，監査証跡機能，マルチノード機能および HADB サーバのメモリ構成について説明しています。

第 2 編　体験編

第 3 章　体験環境構築ナビ

HADB の導入・操作を体験するために使用する，体験環境の構築手順について説明しています。

第 3 編　設計編

第 4 章　システム設計

HADB を構築するためのデータベースの設計，リソースの設計，およびサーバ定義の設計の流れについて説明しています。また，ディスク所要量，およびメモリ所要量の事前見積もりについて説明しています。

第 5 章　データベースの設計

表の設計，インデクスの設計，各 DB エリアの設計，および各 DB エリアの容量見積もりについて説明しています。

第 6 章　リソースの設計

ディスクの設計，カーネルパラメタの見積もり，メモリ所要量の見積もり，各ファイルの容量見積もり，各ディレクトリの容量見積もり，コマンド実行時に増加するデータ量の見積もり，コマンドを同時実行する際の考慮点，コマンドが使用する処理リアルスレッドの最大数の見積もり，クライアントグループ機能を適用する際の考慮点，および作業表に関する見積もりについて説明しています。

第 7 章　サーバ定義の設計

サーバ定義のオペランドの指定形式，サーバ定義のオペランドの内容，およびサーバ定義の文法規則について説明しています。

第 4 編　構築編

第 8 章　システム構築

HADB サーバのインストール，カーネルパラメタの設定，環境変数の設定，サーバ定義の作成，HADB サーバのバージョンアップ，HADB サーバのバージョンダウン，修正版 HADB サーバとの入れ替え，サーバマシンの OS の時刻変更，および HADB サーバのアンインストールについて説明しています。

第 9 章　データベースの作成

データベースの作成手順について説明しています。

第 5 編　運用編

第 10 章　定期運用

定期的に行う運用項目として，HADB サーバの開始と終了，データベースのバックアップ，およびモニタリングについて説明しています。

第 11 章　非定期運用

HADB サーバで非定期に実施する運用項目について説明しています。

第 12 章　監査証跡機能の運用

監査証跡機能の運用について説明します。

第 13 章　チューニング

HADB のチューニングについて説明しています。

第 14 章　障害運用

障害発生時の対処の流れ，トラブルシュート情報の取得方法，およびトラブルシュート情報の削除方法について説明しています。

第 15 章　トラブルシュート

トラブルが発生した場合の対処方法について説明しています。

第 6 編　マルチノード機能編

第 16 章　マルチノード機能の運用

マルチノード機能を使用したシステムの構築方法，およびマルチノード機能の運用方法について説明しています。

第 7 編　コールドスタンバイ構成編

第 17 章　コールドスタンバイ構成の運用

コールドスタンバイ構成でのシステムの構築方法，およびコールドスタンバイ構成の運用方法について説明しています。

第 8 編　Docker および Kubernetes 編

第 18 章　Docker および Kubernetes を使用した HADB の運用

Docker および Kubernetes を使用したシステムの構築方法，および Docker と Kubernetes を使用した HADB の運用方法について説明しています。

付録 A　HADB サーバのディレクトリの構成

HADB のサーバディレクトリ（インストール時），サーバディレクトリ（運用時），および DB ディレクトリの構成について説明しています。

付録 B　ディクショナリ表

ディクショナリ表に格納されている情報，ディクショナリ表の参照時に排他が取得される実表，ディクショナリ表に定義されているインデクス，およびディクショナリ表の検索方法について説明しています。

付録 C　システム表

システム表に格納されている情報，システム表の参照時に排他が取得される実表，システム表に定義されているインデクス，およびシステム表の検索方法について説明しています。

付録 D　HADB の最大値と最小値

システム構成およびデータベースに関する最大値と最小値について説明しています。

付録 E　HADB サーバで起動するプロセス

HADB サーバで起動するプロセスについて説明しています。

■ 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- 『Hitachi Advanced Data Binder AP 開発ガイド』（3000-6-502）
- 『Hitachi Advanced Data Binder コマンドリファレンス』（3000-6-503）
- 『Hitachi Advanced Data Binder SQL リファレンス』（3000-6-504）
- 『Hitachi Advanced Data Binder メッセージ』（3000-6-505）
- 『日立コード変換ユーザーズガイド』（3000-7-415）
- 『高信頼化システム監視機能 HA モニタ Linux(R)(x86)編』（3000-9-201）
- 『JP1 Version 11 JP1/Base 運用ガイド』（3021-3-A01）
- 『JP1 Version 11 JP1/Automatic Job Management System 3 設計ガイド（業務設計編）』（3021-3-B14）
- 『JP1 Version 11 JP1/Audit Management - Manager 構築・運用ガイド』（3021-3-A17）
- 『JP1 Version 12 JP1/Base 運用ガイド』（3021-3-D65）
- 『JP1 Version 12 JP1/Automatic Job Management System 3 設計ガイド（業務設計編）』（3021-3-D23）

なお，Hitachi Advanced Data Binder のマニュアルを本文中で参照させる場合は，『Hitachi Advanced Data Binder』を『HADB』と表記します。

（例）『HADB　AP 開発ガイド』

また，HA モニタのマニュアルを本文中で参照させる場合は，『高信頼化システム監視機能 HA モニタ Linux(R)(x86)編』を『HA モニタ Linux(R)(x86)編』と表記します。

（例）『HA モニタ Linux(R)(x86)編』

JP1/Base のマニュアルを本文中で参照させる場合は，『JP1 Version 11 JP1/Base 運用ガイド』または『JP1 Version 12 JP1/Base 運用ガイド』を『JP1/Base　運用ガイド』と表記します。

（例）『JP1/Base　運用ガイド』

JP1/AJS3 のマニュアルを本文中で参照させる場合は，『JP1 Version 11 JP1/Automatic Job Management System 3 設計ガイド（業務設計編）』または『JP1 Version 12 JP1/Automatic Job Management System 3 設計ガイド（業務設計編）』を『JP1/AJS3　設計ガイド（業務設計編）』と表記します。

（例）『JP1/AJS3　設計ガイド（業務設計編）』

JP1/Audit のマニュアルを本文中で参照させる場合は，『JP1 Version 11 JP1/Audit Management - Manager 構築・運用ガイド』を『JP1/Audit　構築・運用ガイド』と表記します。

（例）『JP1/Audit　構築・運用ガイド』

■ このマニュアルで使用する製品名・機能名

このマニュアルでは，製品名を次のように表記しています。

表記		製品名
HADB	HADB サーバ	Hitachi Advanced Data Binder
	HADB クライアント	Hitachi Advanced Data Binder Client
Linux	Linux	Linux
	RHEL 6	Red Hat Enterprise Linux Server 6（64-bit x86_64）
	RHEL 7	Red Hat Enterprise Linux Server 7（64-bit x86_64）
	RHEL 8	Red Hat Enterprise Linux Server 8（64-bit x86_64）
HDLM		Hitachi Dynamic Link Manager Software
JP1/AJS3		JP1/Automatic Job Management System 3
JP1/Audit		JP1/Audit Management - Manager

■ このマニュアルで使用する英略語

このマニュアルで使用する英略語を次に示します。

英略語	英字での表記
AP	Application Program
APD	Application Parameter Descriptor
API	Application Programming Interface
ARD	Application Row Descriptor

英略語	英字での表記
BI	Business Intelligence
BLOB	Binary Large Object
BNF	Backus-Naur Form
BOM	Byte Order Mark
CLI	Call Level Interface
CLOB	Character Large Object
CPU	Central Processing Unit
CSV	Character-Separated Values
DB	Database
DBMS	Database Management System
DMMP	Device Mapper Multipath
DNS	Domain Name System
ELF	Executable and Linking Format
ER	Entity Relationship
HBA	Host Bus Adapter
ID	Identification number
IEF	Integrity Enhancement Facility
IP	Internet Protocol
IPD	Implementation Parameter Descriptor
IRD	Implementation Row Descriptor
JAR	Java Archive File
JDBC	Java Database Connectivity
JDK	Java Developer's Kit
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
JTA	Java Transaction API
LOB	Large Object
LRU	Least Recently Used

英略語	英字での表記
LV	Logical Volume
LVM	Logical Volume Manager
LWP	Light Weight Process
MSDN	Microsoft Developer Network
NFS	Network File System
NIC	Network Interface Card
NTP	Network Time Protocol
ODBC	Open Database Connectivity
OS	Operating System
PP	Program Prodsuct
PV	Persistent Volume
PVC	Persistent Volume Claim
RAID	Redundant Array of Independent Disks
RDBMS	Relational Database Management System
SELinux	Security-Enhanced Linux
TLB	Translation Lookaside Buffer
URL	Uniform Resource Locator
VG	Volume Group
WWN	World Wide Name

■ このマニュアルで使用する記号

サーバ定義などのオペランド，およびコマンドの説明で使用している記号を次に示します。

なお，これらの記号は説明のために使用している記号のため，オペランドまたはコマンド中に記述しないでください。

記号	意味	例
〔 〕	この記号で囲まれている項目は省略できます。	`adbsql 〔-V〕` この例の場合，`adbsql` と指定してもよいし，`adbsql -V` と指定してもよいことを意味しています。
{ }	この記号で囲まれている複数の項目のうちから，1 つを選択できます。	`adbcancel {--ALL | -u` *コネクション ID*`}`

記号	意味	例
		この例の場合，--ALL または-u コネクション ID のどちらかを指定できることを意味しています。
…	この記号の直前の項目を繰り返し指定できます。	adbbuff -n *DB エリア名*〔，*DB エリア名*〕… この例の場合，DB エリア名を繰り返し指定できることを意味しています。
{ { } }	この記号で囲まれた複数の項目を 1 つの単位として，繰り返し指定できます。	{ {adbinitdbarea -n *データ用 DB エリア名*} } この例の場合，「adbinitdbarea -n *データ用 DB エリア名*」を繰り返し指定できることを意味しています。
__ （下線）	この記号で示す項目は，省略時の解釈値です。	adb_import_errmsg_lv = {0 | 1} この例の場合，オペランドの指定を省略したとき，0 が仮定されることを意味しています。
～	この記号のあとに，指定値の属性を説明しています。	adb_sys_max_users = *最大同時接続数* ～〈整数〉((1～1,024))《10》 この例の場合，1～1,024 の整数が指定できます。オペランドの指定を省略した場合は，10 が仮定されます。
〈 〉	指定値の種別を説明しています。	
(( ))	指定値の範囲を説明しています。	
《 》	省略値を説明しています。	

■ このマニュアルで使用する構文要素記号

構文要素記号	意味
〈パス名〉	パス名には次に示す文字が使用できます。 • OS が Linux の場合 英字，数字，#，-，/，@，_ • OS が Windows の場合 英字，数字，#，-，/，@，_，¥，： ただし，OS によって使用できる文字が異なります。
〈OS パス名〉	OS パス名には，OS でパス名として使用できるすべての文字が使用できます。使用できる文字の詳細については，OS のマニュアルを参照してください。
〈文字列〉	任意の文字列を指定できます。
〈単位付き整数〉	数字（0～9）の末尾に，MB（メガバイト），GB（ギガバイト），またはTB（テラバイト）のどれかの単位を付けた形式で指定します。数字と単位の間に空白を入れることはできません。 • 指定例 1024MB 512GB 32TB • エラーになる指定例 512 GB

注　すべて半角文字を使用してください。

■ このマニュアルで使用する計算式の記号

このマニュアルで使用する計算式の記号の意味を次に示します。

記号	内容
↑　↑	計算結果の値を小数点以下で切り上げることを意味しています。 （例）↑34÷3↑の計算結果は 12 になります。
↓　↓	計算結果の値を小数点以下で切り捨てることを意味しています。 （例）↓34÷3↓の計算結果は 11 になります。
MAX	計算結果のうち，最も大きい値が有効になることを意味しています。 （例）MAX（3×6，4 ＋ 7）の計算結果は 18 になります。
MIN	計算結果のうち，最も小さい値が有効になることを意味しています。 （例）MIN（3×6，4 ＋ 7）の計算結果は 11 になります。

■ パス名の表記について

- サーバディレクトリ（インストール時）のパスは，$INSTDIR と表記します。
- サーバディレクトリ（運用時）のパスは，$ADBDIR と表記します。
- DB ディレクトリのパスは，$DBDIR と表記します。
- クライアントディレクトリのパスは，%ADBCLTDIR%（HADB クライアントが Windows 版の場合）または$ADBCLTDIR（HADB クライアントが Linux 版の場合）と表記します。
- HADB ODBC ドライバトレースファイルの格納フォルダのパスは，%ADBODBTRCPATH%と表記します。

■ ¥の表記について

本文中で使用されている¥は，Linux 版の場合は半角のバックスラッシュを意味しています。

■ このマニュアルで使用する KB（キロバイト）などの単位表記

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト），1PB（ペタバイト），1EB（エクサバイト）はそれぞれ 1,024 バイト，$1,024^2$ バイト，$1,024^3$ バイト，$1,024^4$ バイト，$1,024^5$ バイト，$1,024^6$ バイトです。

目次

前書き　2
変更内容　9
はじめに　11

第 1 編　解説編

1　概要　47
1.1　情報爆発時代のデータを管理する HADB　48
1.2　HADB の特長　49
1.2.1　大規模データを高速に検索できる　49
1.2.2　ペタバイトやエクサバイト単位の大規模データベースを構築できる　50
1.2.3　スケールアウトによる負荷分散が実現できる（マルチノード機能）　50
1.3　HADB のシステム構成　51
1.3.1　HADB サーバ　51
1.3.2　HADB クライアント　51

2　アーキテクチャ　53
2.1　表の種類　54
2.1.1　実表　54
2.1.2　ビュー表　54
2.2　実表の種類　59
2.2.1　ローストア表とカラムストア表　59
2.2.2　シングルチャンク表とマルチチャンク表　64
2.3　インデクスの種類　66
2.3.1　B-tree インデクス　66
2.3.2　テキストインデクス　66
2.3.3　レンジインデクス　69
2.4　DB エリア（表およびインデクスの格納領域）　75
2.4.1　DB エリアとは　75
2.4.2　DB エリアの種類　76
2.4.3　DB エリアの構造（セグメントとページ）　77
2.4.4　DB エリアのファイル構成　86
2.4.5　DB エリアの自動増分　90
2.5　ユーザとスキーマ　93
2.5.1　ユーザの種類（OS ユーザと HADB ユーザ）　93

2.5.2 スキーマ 94
2.6 ユーザ認証 96
2.7 権限 97
2.7.1 権限の種類 97
2.7.2 ユーザ権限 97
2.7.3 スキーマ操作権限 98
2.7.4 監査権限 99
2.7.5 アクセス権限 99
2.7.6 ビュー表に対するアクセス権限 111
2.7.7 HADB ユーザが参照できるディクショナリ表とシステム表の範囲 113
2.7.8 HADB ユーザに権限を付与する例 113
2.8 データベースのアクセス処理方式 119
2.8.1 グローバルバッファ 119
2.8.2 データベースの更新処理方式 119
2.8.3 データベースの検索処理方式（非順序実行方式） 119
2.9 トランザクション制御 124
2.9.1 トランザクションの開始と終了 124
2.9.2 HADB がサポートしているトランザクション隔離性水準 124
2.9.3 トランザクションの同時実行制御 126
2.9.4 トランザクションの同期点処理 127
2.9.5 トランザクションアクセスモード 127
2.10 排他制御 129
2.10.1 排他制御モード 129
2.10.2 排他資源 131
2.10.3 排他資源が確保される期間 133
2.10.4 確保される排他資源と排他制御モード 134
2.10.5 ディクショナリ表およびシステム表の参照時に排他が取得される実表 142
2.10.6 データベースの参照・更新時に排他が取得されるディクショナリ表とシステム表 143
2.10.7 排他制御の例 145
2.11 データベースの回復処理 154
2.11.1 再開始による回復処理の流れ 154
2.12 クライアントグループ機能 156
2.12.1 クライアントグループ機能の概要 156
2.12.2 クライアントグループ機能で設定できるグループの種類 158
2.12.3 グループごとの接続数・処理リアルスレッド数の設定 158
2.13 クライアント定義の集中管理機能 161
2.13.1 クライアント定義の集中管理機能とは 161
2.13.2 クライアント定義の集中管理機能を使用する場合に作成するファイル 162
2.13.3 クライアント定義の集中管理機能の適用例 163

2.14 バックグラウンドインポート機能 165
2.14.1 バックグラウンドインポートとは 165
2.14.2 データインポート単位のデータ管理（チャンク） 166
2.14.3 チャンクとレンジインデクスの関係 173
2.15 チャンクアーカイブ機能（チャンク内のデータの圧縮） 179
2.15.1 チャンクアーカイブ機能とは 179
2.15.2 チャンクアーカイブ機能を適用するケース 180
2.15.3 アーカイブマルチチャンク表の検索 181
2.15.4 データの伸長（アーカイブ状態の解除） 183
2.16 CSV ファイル中のデータの検索 185
2.17 テキストデータの検索 187
2.17.1 表記ゆれ補正検索 187
2.17.2 正規表現を使用した検索 190
2.17.3 同義語検索 191
2.17.4 ワード検索 193
2.18 監査証跡機能 198
2.18.1 監査証跡機能とは 198
2.18.2 監査人の作成 200
2.18.3 監査証跡の出力契機（監査対象イベント） 200
2.18.4 監査証跡に出力される情報 201
2.18.5 監査証跡の出力先（監査証跡ファイル） 202
2.18.6 監査証跡の参照 205
2.18.7 監査証跡機能の有効化と無効化 207
2.18.8 監査証跡機能で使用するディスクの準備 208
2.18.9 監査証跡の変換（JP1/Audit Management - Manager との連携） 210
2.19 マルチノード機能 212
2.19.1 マルチノード機能とは 212
2.19.2 トランザクションおよびコマンドが実行されるノード 214
2.19.3 ノードの管理 219
2.19.4 マルチノード構成からのノードの切り離し（プライマリノード，セカンダリノード） 221
2.19.5 マルチノード構成へのノードの復帰（プライマリノード，セカンダリノード） 224
2.19.6 マルチノード構成からのワーカーノードの削除と追加 226
2.19.7 マルチノード機能を使用する場合のシステム構成例 227
2.19.8 SQL パラレル実行機能 229
2.20 HADB サーバのメモリ構成 232
2.20.1 HADB サーバが使用するメモリ 232
2.20.2 サーバ定義のオペランドとの関係 235

第 2 編　体験編

3　体験環境構築ナビ　244

3.1　体験環境を構築する流れ　245
3.1.1　構築する HADB サーバのシステム構成　245
3.1.2　体験環境の表に格納するデータ　248
3.2　HADB サーバをインストールする　250
3.2.1　インストール前に準備すること　250
3.2.2　インストールする　254
3.3　カーネルパラメタを設定する　259
3.3.1　カーネルパラメタの設定および確認をする　259
3.4　DB ディレクトリを作成する　264
3.4.1　DB ディレクトリの作成および確認をする　264
3.5　環境変数および定義ファイルを設定する　265
3.5.1　環境変数を設定する　265
3.5.2　定義ファイルを編集する　267
3.6　DB エリアを作成する（データベースの初期設定をする）　271
3.6.1　初期設定オプションファイルを編集する　271
3.6.2　DB エリアを作成する　273
3.7　表の作成・検索用の HADB ユーザを作成する　276
3.7.1　HADB サーバを開始する　276
3.7.2　HADB サーバに接続する　277
3.7.3　HADB ユーザを作成する　278
3.7.4　作成した HADB ユーザに権限を付与する　279
3.8　表を作成する　281
3.8.1　HADB サーバに接続する　281
3.8.2　スキーマを定義する　282
3.8.3　表を作成する　282
3.9　データを格納する　285
3.9.1　データの格納前に準備すること　285
3.9.2　表にデータを格納する　286
3.10　格納したデータを検索する　289
3.10.1　HADB サーバに接続してデータを検索する　289
3.11　HADB サーバをアンインストールする　292
3.11.1　サーバディレクトリを削除する　292
3.11.2　インストールデータを格納するディレクトリを削除する　292
3.11.3　通信情報ファイルを格納するディレクトリを削除する　293
3.11.4　HADB 管理者と HADB 管理グループを削除する　295
3.12　よくある質問と対処方法　297

3.12.1 DB ディレクトリをスーパユーザで作成してしまった 297
3.12.2 表を作成できない 297
3.12.3 HADB サーバの開始時に KFAA91104-Q メッセージが出力された 298
3.12.4 HADB サーバの接続時に KFAA30561-E メッセージが出力された 299

第 3 編 設計編

4 システム設計 301
4.1 システム設計の流れ 302
4.2 リソースの事前見積もり 304
4.2.1 DB ディレクトリの事前見積もり 304
4.2.2 サーバディレクトリの事前見積もり 308
4.2.3 アーカイブディレクトリの事前見積もり 309
4.2.4 アンロードファイルの格納ディレクトリの事前見積もり 309
4.2.5 同義語辞書ファイルの格納ディレクトリの事前見積もり 310
4.2.6 マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの事前見積もり 310
4.3 メモリ所要量の事前見積もり 311
4.3.1 通常運用時のメモリ所要量の事前見積もり 312
4.3.2 adbimport コマンド実行時のメモリ所要量の事前見積もり 313
4.3.3 adbidxrebuild コマンド実行時のメモリ所要量の事前見積もり 314
4.3.4 adbgetcst コマンド実行時のメモリ所要量の事前見積もり 316
4.3.5 adbexport コマンド実行時のメモリ所要量の事前見積もり 317
4.3.6 adbmergechunk コマンド実行時のメモリ所要量の事前見積もり 318
4.3.7 adbarchivechunk コマンド実行時のメモリ所要量の事前見積もり 319
4.3.8 adbunarchivechunk コマンド実行時のメモリ所要量の事前見積もり 320

5 データベースの設計 323
5.1 データベースの設計手順 324
5.2 表の設計 325
5.2.1 表を設計する際の流れ 325
5.2.2 ローストア表とカラムストア表の選択基準 325
5.2.3 シングルチャンク表とマルチチャンク表の選択基準 331
5.2.4 マルチチャンク表を定義する場合の考慮点 332
5.2.5 アーカイブマルチチャンク表を定義する場合の考慮点【ローストア表】 335
5.2.6 表の正規化【ローストア表】 337
5.2.7 行長の固定（FIX 指定）【ローストア表】 340
5.2.8 列の既定値の設定（DEFAULT 句） 341
5.2.9 主キー（一意性制約定義）の指定（PRIMARY KEY）【シングルチャンク表】 342
5.2.10 外部キーの指定（FOREIGN KEY） 343
5.2.11 データページ内の未使用領域の確保（PCTFREE）【ローストア表】 344

5.2.12 可変長データ型の列データの分岐指定（BRANCH）【ローストア表】 345
5.2.13 配列型の列の定義【カラムストア表】 346
5.3 B-tree インデクスの設計 350
5.3.1 B-tree インデクスを定義する場合の注意点（B-tree インデクスの未完状態） 350
5.3.2 B-tree インデクスを定義する列を決めるときの考慮点 351
5.3.3 単一列インデクスと複数列インデクスの使い分け 352
5.3.4 B-tree インデクスのインデクスページ内の未使用領域の確保（PCTFREE） 354
5.3.5 ユニークインデクスを定義する列を決めるときの考慮点 356
5.3.6 ナル値除外指定の設定（EXCLUDE NULL VALUES） 357
5.4 テキストインデクスの設計 358
5.4.1 テキストインデクスを定義する列を決めるときの考慮点 358
5.4.2 テキストインデクスのインデクスページ内の未使用領域の確保（PCTFREE） 360
5.4.3 テキストインデクスを定義する場合の注意点（テキストインデクスの未完状態） 362
5.4.4 テキストインデクスの表記ゆれ補正指定の指定（CORRECTIONRULE） 363
5.4.5 ワード検索用のテキストインデクスの指定（TEXT WORDCONTEXT） 364
5.4.6 ワード検索時の区切り文字の選択（DELIMITER） 365
5.5 レンジインデクスの設計 367
5.5.1 レンジインデクスを定義する際の考慮点 367
5.5.2 レンジインデクス専用のグローバルバッファのページ数の見積もり 370
5.5.3 レンジインデクスの一括先読み 370
5.5.4 レンジインデクスを定義する場合の注意点（レンジインデクスの未完状態） 371
5.6 データ用 DB エリアの設計 373
5.6.1 データ用 DB エリアを設計する際の考慮点 373
5.6.2 データ用 DB エリアにマルチチャンク表を格納する場合の考慮点 375
5.6.3 データ用 DB エリアのページサイズを決める際の考慮点 376
5.6.4 データ用 DB エリアの 1 セグメントの容量を決める際の考慮点 377
5.7 作業表用 DB エリアの設計 381
5.8 データ用 DB エリアの容量見積もり 382
5.8.1 データ用 DB エリアの総ページ数の求め方 382
5.8.2 行の種別ごとの格納ページ数の求め方 408
5.8.3 B-tree インデクスのセグメントごとの格納ページ数の求め方 416
5.8.4 B-tree インデクスのキー長（KEYSZ）の求め方 419
5.8.5 テキストインデクスのセグメントごとの格納ページ数の求め方 421
5.8.6 レンジインデクスを格納するセグメント数の求め方 424
5.9 作業表用 DB エリアの容量見積もり 431
5.9.1 作業表用 DB エリアの総ページ数の求め方 431
5.9.2 作業表を格納するために必要な基本行用ページ数の求め方 432
5.10 マスタディレクトリ用 DB エリアの容量見積もり 435
5.11 ディクショナリ用 DB エリアの容量見積もり 436

5.12 システム表用 DB エリアの容量見積もり 442
5.12.1 システム表の容量見積もり 443

6 リソースの設計 451
6.1 ディスクの設計 452
6.1.1 ディスク設計時の考慮点 452
6.1.2 LVM 設定時の考慮点 454
6.1.3 電源断に対する考慮点 455
6.2 カーネルパラメタの見積もり 456
6.3 HADB サーバのメモリ所要量の見積もり 464
6.3.1 共有メモリの所要量の求め方 466
6.3.2 プロセスメモリの所要量の求め方 471
6.3.3 HADB サーバ開始時のメモリ所要量の求め方 472
6.3.4 通常運用時のメモリ所要量の求め方 499
6.3.5 adbinit コマンド実行時のメモリ所要量の求め方 587
6.3.6 adbimport コマンド実行時のメモリ所要量の求め方 588
6.3.7 adbidxrebuild コマンド実行時のメモリ所要量の求め方 612
6.3.8 adbgetcst コマンド実行時のメモリ所要量の求め方 627
6.3.9 adbdbstatus コマンド実行時のメモリ所要量の求め方 632
6.3.10 adbexport コマンド実行時のメモリ所要量の求め方 634
6.3.11 adbstat コマンド実行時のメモリ所要量の求め方 638
6.3.12 adbmodarea コマンド実行時のメモリ所要量の求め方 638
6.3.13 adbmergechunk コマンド実行時のメモリ所要量の求め方 642
6.3.14 adbchgchunkcomment コマンド実行時のメモリ所要量の求め方 658
6.3.15 adbchgchunkstatus コマンド実行時のメモリ所要量の求め方 658
6.3.16 adbarchivechunk コマンド実行時のメモリ所要量の求め方 660
6.3.17 adbunarchivechunk コマンド実行時のメモリ所要量の求め方 667
6.3.18 adbreorgsystemdata コマンド実行時のメモリ所要量の求め方 686
6.3.19 adbclientdefmang コマンド実行時のメモリ所要量の求め方 701
6.3.20 adbsyndict コマンド実行時のメモリ所要量の求め方 704
6.3.21 adbaudittrail コマンド実行時のメモリ所要量の求め方 705
6.3.22 adbconvertaudittrailfile コマンド実行時のメモリ所要量の求め方 707
6.4 HADB サーバの使用時に単調増加するファイル 708
6.5 サーバメッセージログファイルの容量見積もり 711
6.6 障害情報（コアファイル）の容量見積もり 712
6.7 HADB ダンプファイルの容量見積もり 713
6.8 統計ログファイルの容量見積もり 714
6.9 SQL トレースファイルの容量見積もり 715
6.10 更新行のカラム化機能に必要なファイルの容量見積もり 716

6.11 アクセスパス探索情報ログファイルの容量見積もり 717
6.12 システムログファイルの容量見積もり 718
6.12.1 マスタログファイルの容量の求め方 718
6.12.2 ユーザログファイルの容量の求め方 722
6.12.3 定義系 SQL 実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 723
6.12.4 adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 729
6.12.5 adbidxrebuild コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 739
6.12.6 adbmergechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 743
6.12.7 データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方 748
6.12.8 TRUNCATE TABLE 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 754
6.12.9 PURGE CHUNK 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 757
6.12.10 adbchgchunkstatus コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 760
6.12.11 adbarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 761
6.12.12 adbunarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 766
6.12.13 adbreorgsystemdata コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 767
6.12.14 更新行のカラム化機能のメンテナンス処理に必要なユーザログ容量（変数 max_user_log）の求め方 769
6.12.15 ユーザログファイル数の求め方 770
6.13 アーカイブディレクトリの容量見積もり 775
6.14 同義語辞書ファイルの格納ディレクトリの容量見積もり 778
6.15 監査証跡の出力先ディレクトリの容量見積もり 780
6.16 統一フォーマット用監査証跡の出力先ディレクトリの容量見積もり 781
6.17 マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量見積もり 782
6.18 adbexport コマンドが出力するファイルの容量見積もり 783
6.19 実表の再編成に必要なファイルの容量見積もり 785
6.20 アンロードファイルの容量見積もり 786
6.21 コマンド実行時の作業用一時ファイルの容量見積もり 787
6.21.1 adbimport コマンド実行時の作業用一時ファイルの容量見積もり 787
6.21.2 adbidxrebuild コマンド実行時の作業用一時ファイルの容量見積もり 791
6.21.3 adbmergechunk コマンド実行時の作業用一時ファイルの容量見積もり 795
6.21.4 adbunarchivechunk コマンド実行時の作業用一時ファイルの容量見積もり 798
6.21.5 adbreorgsystemdata コマンド実行時の作業用一時ファイルの容量見積もり 801
6.21.6 adbsyndict コマンド実行時の作業用一時ファイルの容量見積もり 804
6.22 コマンド実行時に増加するデータ量の見積もり 805
6.22.1 adbmergechunk コマンド実行時に増加するデータ量の見積もり 805
6.23 コマンドを同時実行する際の考慮点 806

6.23.1 コマンドの最大同時実行数について 806
6.23.2 コマンドの実行時に使用する処理リアルスレッド数に関する考慮点 807
6.23.3 コマンド同時実行時の排他制御に関する考慮点 809
6.23.4 コマンド同時実行時のメモリ所要量に関する考慮点 810
6.23.5 コマンド同時実行時のシステムログファイルの容量に関する考慮点 810
6.24 クライアントグループ機能を適用する際の考慮点 811
6.24.1 グループの接続数を設定する際の考慮点 811
6.24.2 グループの処理リアルスレッド数を設定する際の考慮点 815
6.24.3 クライアントグループ機能で使用される接続数と処理リアルスレッド数の関係 820
6.25 作業表に関する見積もり 825
6.25.1 グローバル作業表用のグローバルバッファのページ数の見積もり 825
6.25.2 ローカル作業表用バッファのページ数の見積もり 826
6.25.3 ハッシュテーブルを使用した検索時に作成される作業表の数 828

7 サーバ定義の設計 833
7.1 サーバ定義のオペランドの指定形式 834
7.2 サーバ定義のオペランドの内容 838
7.2.1 システム構成に関するオペランド（set 形式） 838
7.2.2 性能に関するオペランド（set 形式） 842
7.2.3 システムログに関するオペランド（set 形式） 859
7.2.4 状態監視に関するオペランド（set 形式） 862
7.2.5 SQL 文に関するオペランド（set 形式） 867
7.2.6 レンジインデクスに関するオペランド（set 形式） 872
7.2.7 統計情報に関するオペランド（set 形式） 873
7.2.8 同義語検索に関するオペランド（set 形式） 874
7.2.9 監査証跡機能に関するオペランド（set 形式） 875
7.2.10 マルチノード機能に関するオペランド（set 形式） 876
7.2.11 グローバルバッファに関するオペランドおよびオプション（コマンド形式） 879
7.2.12 クライアントグループ機能に関するオペランドおよびオプション（コマンド形式） 889
7.3 サーバ定義の文法規則 900
7.3.1 サーバ定義の文法規則の詳細 900

第 4 編 構築編

8 システム構築 905
8.1 システム構築の手順 906
8.2 インストール 908
8.2.1 インストール前に実施する作業 908
8.2.2 インストール手順 916
8.2.3 インストール後に実施する作業 921

8.3 カーネルパラメタの設定 926
8.4 環境変数の設定 927
8.5 サーバ定義の作成および変更方法 931
8.5.1 サーバ定義の作成方法および格納先 931
8.5.2 サーバ定義の変更方法 931
8.6 HADB サーバのバージョンアップ 933
8.6.1 バージョンアップ前に実施すること 933
8.6.2 バージョンアップする 940
8.6.3 バージョンアップ後に実施すること 944
8.6.4 バージョンアップに失敗した場合の対処方法 963
8.6.5 バージョンアップ時の注意事項 969
8.7 HADB サーバのバージョンダウン（旧バージョンに戻す方法） 978
8.7.1 旧バージョンのデータベースのバックアップを使用する場合 979
8.7.2 旧バージョンでデータベースを再構築する場合 982
8.7.3 バージョンアップに失敗したために，旧バージョンに戻す必要がある場合 986
8.8 修正版 HADB サーバとの入れ替え 987
8.8.1 修正版 HADB サーバとの入れ替え前に実施すること 987
8.8.2 修正版 HADB サーバに入れ替える手順 990
8.9 一時的に HADB サーバのバージョンアップとバージョンダウンを行う方法（動作確認テスト用） 994
8.9.1 バージョンアップ前に実施すること（動作確認テスト用） 995
8.9.2 バージョンアップする（動作確認テスト用） 995
8.9.3 バージョンアップ後に実施すること（動作確認テスト用） 996
8.9.4 バージョンダウンする（動作確認テスト用） 996
8.9.5 バージョンダウン後に実施すること（動作確認テスト用） 996
8.10 サーバマシンの OS の時刻変更 997
8.10.1 注意事項（OS の時刻変更） 997
8.10.2 サーバマシンの OS の時刻を進める方法 997
8.10.3 サーバマシンの OS の時刻を戻す方法 998
8.11 サーバマシンの OS のホスト名・IP アドレスの変更 1001
8.12 アンインストール 1003
8.12.1 アンインストール手順 1003
8.12.2 アンインストール後に実施する作業 1004

9 データベースの作成 1006

9.1 データベースの作成手順 1007
9.2 データベースを初期設定する（データ用 DB エリアを作成する） 1008
9.3 HADB サーバを開始する 1010
9.4 実表を定義する HADB ユーザを作成する 1011
9.4.1 HADB ユーザを作成する 1011

9.4.2 認可識別子の指定規則 1012
9.4.3 パスワードの指定規則 1013
9.5 スキーマ，表，およびインデクスを定義する 1014
9.6 実表にデータをインポートする 1016
9.7 コスト情報を収集する 1018

第 5 編 運用編

10 定期運用 1019
10.1 運用項目の一覧 1020
10.2 HADB サーバの開始・終了と稼働モード 1023
10.2.1 HADB サーバの開始方法 1023
10.2.2 HADB サーバの終了方法 1024
10.2.3 HADB サーバの稼働モード 1028
10.2.4 HADB サーバの自動再起動の設定 1030
10.3 データベースのバックアップ 1032
10.3.1 バックアップの取得方法 1032
10.3.2 バックアップからの回復方法 1037
10.3.3 バックアップの運用例（OS コマンドを使用） 1038
10.3.4 バックアップの運用例（ShadowImage を使用） 1045
10.4 メッセージの確認 1049
10.4.1 メッセージの出力先 1049
10.4.2 メッセージログの参照方法（メッセージログの出力先） 1050
10.4.3 メッセージログファイルの運用 1050
10.4.4 メッセージログファイルの縮退状態 1052
10.4.5 syslog へのメッセージの出力抑止 1053
10.4.6 syslog の文字コード変換と信頼性向上（拡張 SYSLOG 機能の適用） 1054
10.5 リソースの使用状況の監視（監視対象メッセージの一覧） 1057
10.6 メモリの使用状況の確認 1065
10.6.1 全メモリの使用状況を確認したい場合 1065
10.6.2 共有メモリの使用状況を確認したい場合 1066
10.6.3 リアルスレッドごとのメモリの使用状況を確認したい場合 1067
10.7 HADB サーバで実行中のスレッドの確認 1068
10.8 トランザクションの処理状況の確認 1069
10.8.1 AP およびコマンドの処理状況を確認する方法 1069
10.8.2 処理リアルスレッドの確保待ちの発生状況を確認する方法 1071
10.8.3 排他資源の確保待ちの発生状況を確認する方法 1071
10.8.4 トランザクションがロールバックしたときの状況を確認する方法 1072
10.9 データベースの状態や使用量の確認 1074

10.9.1 データベースの使用量を確認したい場合 1074
10.9.2 実表の状態や使用量を確認したい場合 1075
10.9.3 B-tree インデクスの状態や使用量を確認したい場合 1075
10.9.4 テキストインデクスの状態や使用量を確認したい場合 1077
10.9.5 レンジインデクスの状態や使用量を確認したい場合 1078
10.9.6 DB エリアファイルの使用量を確認したい場合 1079
10.9.7 アーカイブファイルのサイズを確認したい場合 1080
10.10 統計解析の運用（HADB サーバの稼働情報の確認） 1082
10.10.1 HADB サーバの統計情報の運用方法 1082
10.10.2 コネクションの稼働情報の運用方法 1083
10.10.3 グローバルバッファの統計情報の運用方法 1085
10.10.4 SQL 文の統計情報の運用方法 1088
10.10.5 統計ログファイルの運用 1092
10.10.6 adbstat コマンド実行時に統計ログファイルから出力される情報の容量見積もり 1095
10.11 SQL トレース機能の運用 1096
10.11.1 SQL トレース機能とは 1096
10.11.2 SQL トレース情報に出力される情報 1097
10.11.3 アクセスパスの統計情報の出力例と出力項目 1118
10.11.4 SQL トレース情報の出力例と見方 1143
10.11.5 SQL トレース情報を出力するための準備 1151
10.11.6 SQL トレース情報を出力しているときの運用 1152
10.11.7 SQL トレース情報を使用した SQL 文のエラー原因調査 1155
10.11.8 SQL トレース情報を利用した SQL 文のチューニング 1158
10.11.9 障害によって SQL トレース情報の出力が停止した場合の対処方法 1160
10.11.10 SQL トレース機能の使用をやめる場合 1161
10.11.11 マルチノード機能を使用している場合の留意事項 1161
10.12 システムログファイルの運用 1165
10.12.1 システムログファイルの構成とサーバ定義の指定 1165
10.12.2 システムログファイルの容量の縮小契機 1166

11 非定期運用 1168

11.1 実表の運用 1169
11.1.1 実表を定義する方法 1169
11.1.2 実表に列を追加する方法 1169
11.1.3 実表の列名を変更する方法 1170
11.1.4 実表の表名を変更する方法 1172
11.1.5 実表のデータを検索・更新する方法 1173
11.1.6 実表の全行を削除する方法 1174
11.1.7 シングルチャンク表をマルチチャンク表に変更する方法 1176

11.1.8 実表にデータを格納する方法（データインポート） 1177
11.1.9 実表のデータをファイルに出力する方法（データをエクスポートする方法） 1179
11.1.10 シングルチャンク表の再編成が必要かどうかを確認する方法 1180
11.1.11 シングルチャンク表を再編成する方法 1183
11.1.12 ローストア表をカラムストア表に変更する方法 1185
11.1.13 カラムストア表をローストア表に変更する方法 1188
11.1.14 実表の定義情報を調べる方法（ディクショナリ表の検索） 1190
11.1.15 実表の表定義を変更する方法（別表を定義して動作検証したあとに運用中の実表の表定義を変更する場合） 1192
11.1.16 実表が格納されているデータ用 DB エリアを変更する方法 1195
11.1.17 実表を削除する方法 1198
11.1.18 配列型の列を定義した実表の運用 1199
11.2 ビュー表の運用 1204
11.2.1 ビュー表を定義する方法 1204
11.2.2 ビュー表のデータを検索，更新，追加，削除する方法 1205
11.2.3 ビュー表のデータをファイルに出力する方法（データをエクスポートする方法） 1205
11.2.4 ビュー表を再作成する方法 1205
11.2.5 ビュー表を削除する方法 1206
11.2.6 ビュー表の更新可否を調べる方法 1207
11.2.7 ビュー表が無効化されているかどうかを確認する方法 1207
11.2.8 ビュー表の無効化を解除する方法 1208
11.2.9 無効化されたビュー表を削除する方法 1217
11.2.10 ビュー表の基表を確認する方法 1218
11.2.11 依存するビュー表を確認する方法 1218
11.2.12 ビュー表に対するアクセス権限を確認する方法 1219
11.2.13 ビュー表を使用した運用例 1219
11.3 インデクスの運用 1223
11.3.1 インデクスの定義 1223
11.3.2 B-tree インデクスの再作成 1224
11.3.3 テキストインデクスの再作成 1226
11.3.4 テキストインデクスの確認（指定したインデクスオプションの確認） 1228
11.3.5 レンジインデクスの再作成 1228
11.3.6 レンジインデクスの確認（チャンクのスキップの可否） 1230
11.3.7 レンジインデクスの再定義（チャンクのスキップの設定） 1230
11.3.8 インデクスが格納されているデータ用 DB エリアを変更する方法 1231
11.3.9 インデクスの削除 1232
11.4 マルチチャンク表の運用 1234
11.4.1 マルチチャンク表を定義する方法 1234
11.4.2 マルチチャンク表にデータを格納する方法（バックグラウンドインポート） 1235

11.4.3 アーカイブマルチチャンク表のデータを検索・更新するときの注意事項 1237
11.4.4 マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成） 1238
11.4.5 チャンク単位にデータをエクスポートする方法 1241
11.4.6 チャンク単位にデータを削除する方法 1242
11.4.7 マルチチャンク表に格納されているデータを一括削除する方法 1243
11.4.8 チャンクの状態や作成されているチャンク数を確認する方法 1243
11.4.9 チャンクをマージする方法（チャンク数を減らす方法） 1247
11.4.10 チャンク数の最大値を変更する方法 1256
11.4.11 チャンクのコメントを変更する方法 1257
11.4.12 チャンクの状態を変更する方法 1258
11.4.13 マルチチャンク表の再編成が必要かどうかを確認する方法 1281
11.4.14 マルチチャンク表を再編成する方法（チャンク単位で再編成する場合） 1287
11.4.15 マルチチャンク表を再編成する方法（表全体を再編成する場合） 1295
11.4.16 マルチチャンク表を再編成する方法（サンプルシェルを使用して再編成する場合） 1300
11.4.17 チャンクをアーカイブする方法（アーカイブマルチチャンク表使用時） 1302
11.4.18 チャンクをアンアーカイブする方法（アーカイブマルチチャンク表使用時） 1304
11.4.19 アーカイブ状態のチャンクを確認する方法 1306
11.4.20 レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する方法 1306
11.4.21 アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する方法 1308
11.4.22 マルチチャンク表を削除する方法 1310
11.4.23 バックグラウンドインポートとチャンクを意識した運用例 1（定期的にデータの追加と削除を繰り返す運用） 1310
11.4.24 バックグラウンドインポートとチャンクを意識した運用例 2（待機状態のチャンクを利用した運用） 1327
11.4.25 チャンクのアーカイブを意識した運用例 1340
11.5 コスト情報の収集 1352
11.6 HADB ユーザの管理 1353
11.6.1 HADB ユーザを作成する方法 1353
11.6.2 HADB ユーザのパスワードを変更する方法 1353
11.6.3 HADB ユーザを削除する方法 1354
11.7 ユーザ権限とスキーマ操作権限の管理 1356
11.7.1 HADB ユーザにユーザ権限とスキーマ操作権限を付与する方法 1356
11.7.2 HADB ユーザが持っているユーザ権限とスキーマ操作権限を確認する方法 1356
11.7.3 HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法 1357
11.8 アクセス権限の管理 1359
11.8.1 HADB ユーザにアクセス権限を付与する方法 1359
11.8.2 HADB ユーザが持っているアクセス権限を確認する方法 1360
11.8.3 HADB ユーザに付与したアクセス権限を取り消す方法 1361
11.8.4 HADB ユーザに付与したアクセス権限の付与権だけを取り消す方法 1362

11.8.5 HADB ユーザが持っているアクセス権限を変更する方法 1363
11.9 スキーマの運用 1364
11.9.1 スキーマを定義する方法 1364
11.9.2 スキーマを削除する方法 1364
11.10 データ用 DB エリアの運用 1365
11.10.1 データ用 DB エリアを追加する方法 1365
11.10.2 データ用 DB エリアを削除する方法 1366
11.10.3 データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加） 1366
11.10.4 データ用 DB エリアを再初期化する方法 1367
11.10.5 データ用 DB エリアファイルの格納場所を変更する方法 1368
11.10.6 データ用 DB エリアの空き容量を確保する方法 1372
11.10.7 データ用 DB エリアの使用率を監視する方法 1379
11.11 作業表用 DB エリアの運用 1380
11.11.1 作業表用 DB エリアファイルの格納場所を変更する方法 1380
11.11.2 作業表用 DB エリアファイルの容量を小さくする方法 1380
11.12 バッファの運用 1381
11.12.1 ローカル作業表用バッファを変更する方法 1381
11.13 クライアントグループの設定内容の確認 1383
11.13.1 AP を実行した HADB クライアントが所属しているグループを確認したい場合 1383
11.13.2 リアルスレッドを使用している HADB クライアントが所属しているグループを確認したい場合 1383
11.13.3 各グループに設定した接続数や処理リアルスレッド数を確認したい場合 1383
11.14 クライアント定義の集中管理の運用 1385
11.14.1 クライアント定義の集中管理機能を使用する流れ 1385
11.14.2 クライアント定義の集中管理機能で必要なファイル 1386
11.14.3 新たにクライアント定義の集中管理機能を使用する方法 1390
11.14.4 クライアント定義の集中管理機能の適用内容を変更する方法 1391
11.14.5 クライアント定義の集中管理機能の使用例 1393
11.14.6 クライアント定義の集中管理機能の使用をやめる方法 1396
11.15 CSV ファイル中のデータを検索するときの運用 1397
11.15.1 準備作業 1397
11.15.2 データの検索方法 1397
11.16 同義語検索を行う際の運用 1399
11.16.1 同義語検索を行うための準備をする 1399
11.16.2 同義語検索を行う 1405
11.16.3 同義語辞書に登録されている同義語を確認する 1407
11.16.4 同義語辞書の情報を確認する 1408
11.16.5 同義語を追加する 1410
11.16.6 同義語を削除する 1412
11.16.7 同義語グループを追加する 1413

11.16.8 同義語辞書を登録する 1415
11.16.9 同義語辞書を削除する 1417
11.16.10 同義語辞書を表記ゆれ補正検索対応に変更する 1418
11.16.11 同義語検索のチューニングを行う 1419
11.16.12 同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する 1420
11.16.13 同義語辞書ファイルの格納ディレクトリを変更する 1421
11.16.14 同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合） 1422
11.16.15 同義語一覧定義ファイルの指定規則 1423
11.16.16 辞書情報登録ファイルの指定規則 1425
11.17 システム表の再編成 1429
11.17.1 システム表を再編成する理由 1429
11.17.2 システム表を再編成するタイミング 1430
11.17.3 システム表を再編成する方法 1431
11.17.4 システム表の再編成と排他制御 1432
11.17.5 システム表の状態や使用量を確認する方法 1434
11.18 更新行のカラム化機能の運用（カラムストア表の検索性能の維持） 1437
11.18.1 更新行のカラム化機能とは 1437
11.18.2 準備作業 1439
11.18.3 更新行のカラム化機能の状態（有効または無効）を確認する方法 1440
11.18.4 更新行のカラム化機能を一時的に無効にする場合 1441
11.18.5 更新行のカラム化機能に関するエラーが発生した場合の対処方法 1442
11.18.6 メンテナンス処理が実行されない場合の対処方法 1442
11.18.7 メンテナンス処理の詳細 1443

12 監査証跡機能の運用 1446
12.1 監査証跡機能を使用する際に検討する項目 1447
12.1.1 監査対象定義の検討（監査証跡を出力するイベントの選定） 1447
12.1.2 監査証跡機能で使用するディスクの設計 1447
12.1.3 監査証跡のデータ容量の見積もり 1450
12.1.4 監査証跡ファイルの圧縮率の見積もり 1451
12.1.5 監査証跡機能で使用するディレクトリの容量見積もり 1451
12.1.6 監査人の選定 1453
12.1.7 監査証跡ファイルへの書き込みに失敗した場合の処理方式の検討 1454
12.2 監査証跡機能の環境設定 1456
12.2.1 HADB サーバの正常終了 1456
12.2.2 監査証跡機能で使用するディレクトリの準備 1457
12.2.3 監査証跡機能で使用するサーバ定義の指定 1458
12.2.4 HADB サーバの正常開始（オフラインモード） 1458
12.2.5 監査人の作成 1459

12.2.6 監査証跡機能の有効化 1459
12.2.7 監査対象の定義 1460
12.2.8 HADB サーバの稼働モードの変更（通常モード） 1461
12.3 監査証跡機能の定期運用 1462
12.3.1 監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動） 1462
12.3.2 監査証跡ファイルの移動（監査証跡の長期保存先ディレクトリへの移動） 1464
12.3.3 監査証跡の参照（SELECT 文で監査証跡を参照する場合） 1465
12.3.4 監査証跡の参照（CSV 形式に変換した監査証跡を参照する場合） 1467
12.4 監査証跡機能の非定期運用 1470
12.4.1 監査人の追加，削除，変更（監査権限の付与または取り消し） 1470
12.4.2 現用の監査証跡ファイルの切り替え 1473
12.4.3 監査対象定義の確認 1473
12.4.4 監査対象定義の変更 1474
12.4.5 監査証跡機能の動作状況の確認 1474
12.5 監査証跡機能のトラブルシュート 1476
12.5.1 監査証跡の出力先ディレクトリを作成したディスクが満杯になった場合の対処方法 1476
12.5.2 監査証跡の出力先ディレクトリを作成したディスクで障害が発生した場合の対処方法 1477
12.5.3 ファイルディスクリプタが不足した場合の対処方法 1477
12.6 監査証跡機能の使用をやめる場合 1479
12.7 監査証跡機能の運用例 1482
12.7.1 A 社の情報分析システムの構成 1482
12.7.2 設計 1483
12.7.3 環境設定 1487
12.7.4 定期運用 1491
12.7.5 定期監査の実施 1492
12.7.6 セキュリティインシデント発生時の調査 1496
12.8 監査証跡機能と JP1/Audit Management - Manager との連携 1500
12.8.1 監査証跡機能と JP1/Audit を連携する際の概要 1500
12.8.2 統一フォーマット用監査証跡ファイルの形式と出力項目 1503
12.8.3 監査証跡機能と JP1/Audit を連携する際の環境設定 1510
12.8.4 監査証跡機能と JP1/Audit を連携する際の運用方法 1514
12.8.5 監査の実施 1517
12.9 監査対象イベントと表関数導出表の列構成 1518
12.9.1 監査対象イベントの一覧と出力項目 1518
12.9.2 監査証跡を検索するときの表関数導出表の列構成 1521
12.9.3 監査証跡の出力契機と出力項目 1540
12.10 監査証跡機能使用時の留意事項 1654
12.10.1 1 つのイベントで複数の監査証跡が出力されるケース 1654
12.10.2 コマンドの実行時に出力される監査証跡についての留意事項 1655

13 チューニング 1659
13.1 処理性能の向上に関するチューニング 1660
13.1.1 レンジインデクスの一括先読み 1660
13.1.2 定義したレンジインデクスの見直し 1660
13.1.3 定義したテキストインデクスの見直し 1663
13.1.4 HADB サーバ開始時間の短縮（HugePages の適用） 1664
13.1.5 SQL 文の実行待ち状態の発生防止 1665
13.1.6 SQL 文の処理時間の短縮 1666
13.1.7 ユーザログバッファ面数の拡張 1669
13.1.8 ユーザログファイルの初期化容量の拡張 1670
13.1.9 ユーザログファイルの縮小化容量の見直し 1671
13.1.10 DB エリアの自動増分の発生防止 1673
13.1.11 作業表用 DB エリアのファイル構成の変更 1674
13.2 バッファの見直しによる SQL 文の実行時間の短縮に関するチューニング 1675
13.2.1 チューニングを行う前に確認すること 1675
13.2.2 SQL 文の実行時間を短縮する方法 1676
13.2.3 グローバル作業表を作成する SQL 文の実行時間を短縮する方法 1678
13.2.4 ローカル作業表を作成する SQL 文の実行時間を短縮する方法 1680
13.2.5 テーブルスキャンを実行する SQL 文の実行時間を短縮する方法 1682
13.3 コマンドの実行時間の短縮に関するチューニング 1685
13.3.1 adbimport コマンドの実行時間の短縮 1685
13.3.2 adbidxrebuild コマンドの実行時間の短縮 1685
13.3.3 adbgetcst コマンドの実行時間の短縮 1686
13.3.4 adbexport コマンドの実行時間の短縮 1687
13.3.5 adbmergechunk コマンドの実行時間の短縮 1688
13.3.6 adbarchivechunk コマンドの実行時間の短縮 1689
13.4 メモリ使用量の削減に関するチューニング 1691
13.4.1 HugePages を適用した共有メモリの使用量の削減 1691
13.4.2 グローバルバッファの見直しによるメモリ使用量の削減 1692
13.4.3 ローカル作業表用バッファの見直しによるメモリ使用量の削減 1694
13.5 ハッシュテーブル領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング 1696
13.6 ハッシュグループ化領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング 1699
13.7 ハッシュジョインが適用される INNER JOIN の結合順序の見直しによる SQL 文の実行時間の短縮に関するチューニング 1701
13.8 ハッシュフィルタ領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング 1703
13.9 表定義プールサイズの見直しによる SQL 文の実行時間の短縮に関するチューニング 1706

14 障害運用 1708
14.1 障害発生時の対処の流れ 1709

14.1.1 HADB サーバが異常終了した場合の対処 1709
14.1.2 HADB サーバが異常終了していない場合の対処 1710
14.2 トラブルシュート情報の取得（adbinfoget コマンド） 1712
14.3 トラブルシュート情報の削除（adbinfosweep コマンド） 1715

15 トラブルシュート 1717
15.1 AP に関するトラブル 1718
15.1.1 AP が終了しない場合や AP が異常終了した場合の対処方法 1718
15.1.2 AP が実行できない場合の対処方法（KFAA40005-E メッセージが出力された場合） 1718
15.1.3 AP の HADB サーバへの接続に時間が掛かる場合の対処方法 1718
15.1.4 クライアントマシンで使用中の通信ポート数を確認する方法 1719
15.2 コマンドに関するトラブル 1721
15.2.1 コマンドが応答待ちタイムアウトになる場合の対処方法 1721
15.2.2 コマンドが実行できない場合の対処方法（KFAA40005-E メッセージが出力された場合） 1721
15.2.3 コマンドが終了しない場合の対処方法 1721
15.2.4 adbimport コマンドまたは adbidxrebuild コマンドが再実行できない場合の対処方法 1722
15.2.5 adbunarchivechunk コマンドが再実行できない場合の対処方法 1723
15.2.6 コマンド実行時の作業用一時ファイルを格納するディスクの空き容量が不足した場合の対処方法 1723
15.3 ディスクの空き容量に関するトラブル 1731
15.3.1 DB エリアファイルの容量増加が原因の場合 1731
15.3.2 DB エリアの自動増分の失敗が原因の場合 1732
15.3.3 DB エリアファイル以外のファイルの容量増加が原因の場合 1733
15.4 ディスクに関するトラブル 1735
15.4.1 データベースの回復 1735
15.4.2 サーバディレクトリの回復 1735
15.4.3 クライアントディレクトリの回復 1736
15.5 DB ディレクトリ下のファイルに関するトラブル 1737
15.5.1 DB エリアファイルに障害が発生した場合の対処 1737
15.5.2 システムログファイルでトラブルが発生した場合の対処 1737
15.6 メモリに関するトラブル 1739
15.6.1 HADB サーバ開始時にメモリが不足した場合の対処方法 1739
15.6.2 SQL 文実行時またはコマンド実行時にメモリが不足した場合の対処方法 1742
15.7 DB エリアに関するトラブル 1747
15.7.1 データ用 DB エリアが満杯になった場合 1747
15.7.2 データ用 DB エリアを追加できなくなった場合 1747
15.7.3 データ用 DB エリアを拡張できなくなった場合 1751
15.8 実表に関するトラブル 1753
15.8.1 実表が更新不可状態になった場合 1753
15.8.2 実表に列を追加できない場合 1758

15.9 B-tree インデクスに関するトラブル 1760
15.9.1 B-tree インデクスが未完状態になったときの対処方法 1760
15.9.2 一意性制約に違反した場合の対処方法（KFAA61205-W メッセージが出力された場合） 1761
15.10 テキストインデクスに関するトラブル 1763
15.10.1 テキストインデクスが未完状態になったときの対処方法 1763
15.11 レンジインデクスに関するトラブル 1765
15.11.1 レンジインデクスが未完状態になったときの対処方法 1765
15.12 バックグラウンドインポート機能に関するトラブル 1767
15.12.1 バックグラウンドインポートで作成されたチャンク数を減らす方法 1767
15.12.2 バックグラウンドインポートで作成されるチャンク数の最大値を大きくする方法 1767
15.12.3 バックグラウンドインポートで作成されたチャンク数を変更できないときの対処方法 1768
15.13 アーカイブディレクトリに関するトラブル 1769
15.13.1 アーカイブファイルに障害が発生したときの対処方法 1769
15.13.2 アーカイブディレクトリの空き容量が不足したときの対処方法 1769
15.14 同義語辞書ファイルに関するトラブル 1770
15.14.1 同義語辞書ファイルに障害が発生したときの対処方法 1770
15.14.2 同義語辞書ファイルの格納ディレクトリの空き容量が不足したときの対処方法 1770
15.14.3 同義語辞書ファイルまたは同義語辞書ファイルの格納ディレクトリを誤って削除してしまったときの対処方法 1771
15.15 アンロードファイルに関するトラブル 1773
15.15.1 アンロードファイルを格納するディスクの空き容量が不足したときの対処方法 1773
15.16 メッセージログファイルに関するトラブル 1777
15.16.1 メッセージログファイルの縮退状態を解除する方法 1777
15.17 HADB サーバの再開始に関するトラブル 1779
15.17.1 HADB サーバの再開始に掛かる処理時間が長い場合の対処方法 1779

第 6 編 マルチノード機能編

16 マルチノード機能の運用 1780
16.1 この章を読む際の留意事項 1781
16.2 マルチノード機能を使用したシステム構成例 1782
16.2.1 前提ソフトウェア 1783
16.2.2 サーバ構成 1783
16.2.3 ネットワーク構成 1784
16.2.4 ストレージ構成 1787
16.3 マルチノード機能を使用したシステムの構築 1792
16.3.1 HA モニタのインストール 1792
16.3.2 HADB サーバのインストールおよび環境設定 1792
16.3.3 HADB クライアントのインストールおよび環境設定 1793
16.3.4 HA モニタの環境設定 1793

16.3.5 各ノードでのサーバ定義の作成 1823
16.3.6 クライアント定義の作成 1827
16.3.7 クライアント定義の集中管理機能を使用する場合の環境設定 1828
16.3.8 データベースの作成 1829
16.4 マルチノード構成の HADB サーバの開始と終了 1833
16.4.1 マルチノード構成の HADB サーバの開始方法 1833
16.4.2 マルチノード構成の HADB サーバの終了方法 1835
16.4.3 マルチノード機能使用時の HADB サーバの稼働モード 1838
16.5 AP の運用（マルチノード機能の使用時） 1840
16.5.1 HADB サーバへの接続 1840
16.5.2 マルチノード機能使用時の最大同時接続数 1841
16.5.3 トランザクションを実行するノード 1841
16.5.4 ノード障害が発生した場合 1844
16.6 排他の運用（マルチノード機能の使用時） 1845
16.7 コマンドによるプライマリノードの切り替え 1846
16.8 データベースのバックアップ（マルチノード機能の使用時） 1847
16.8.1 バックアップの取得方法 1847
16.8.2 バックアップからの回復方法 1849
16.8.3 バックアップの運用例（OS コマンドを使用） 1851
16.8.4 バックアップの運用例（ShadowImage を使用） 1857
16.9 状態監視（マルチノード機能の使用時） 1862
16.9.1 全ノードの情報を確認する場合 1862
16.9.2 HADB サーバの状態を確認する場合 1862
16.9.3 AP またはコマンドの実行時間が長い場合 1862
16.9.4 HADB サーバの開始処理または終了処理に掛かる時間が長い場合 1863
16.9.5 各ノードの HADB サーバの状態を確認する場合 1863
16.9.6 メモリの使用状況を確認する場合 1863
16.9.7 SQL メインノードと SQL サブノードを確認する場合 1864
16.10 DB エリアの運用（マルチノード機能の使用時） 1866
16.10.1 グローバルバッファの割り当て 1866
16.10.2 データベースの状態や使用量の確認 1866
16.10.3 データ用 DB エリアの追加・削除・拡張（マルチノード機能の使用時） 1866
16.10.4 DB エリアの格納位置の変更 1870
16.10.5 作業表用 DB エリアの運用（マルチノード機能の使用時） 1871
16.10.6 ディスクを増設した場合の運用 1871
16.11 スキーマ，表およびインデクスの運用（マルチノード機能の使用時） 1873
16.11.1 スキーマの運用（マルチノード機能の使用時） 1873
16.11.2 実表の運用（マルチノード機能の使用時） 1873
16.11.3 ビュー表の運用（マルチノード機能の使用時） 1874

16.11.4 インデクスの運用（マルチノード機能の使用時） 1874
16.11.5 マルチチャンク表の運用（マルチノード機能の使用時） 1876
16.11.6 アーカイブマルチチャンク表の運用（マルチノード機能の使用時） 1877
16.11.7 システム表の再編成（マルチノード機能の使用時） 1879
16.12 システムログの運用（マルチノード機能の使用時） 1880
16.13 統計情報の運用（マルチノード機能の使用時） 1881
16.14 チューニング（マルチノード機能の使用時） 1882
16.15 プライマリノードまたはセカンダリノードでノード障害が発生した場合の運用（ノードの復帰） 1883
16.15.1 プライマリノードでノード障害が発生した場合 1883
16.15.2 セカンダリノードでノード障害が発生した場合 1885
16.15.3 マルチノード構成へのノードの復帰 1886
16.16 ワーカーノードでノード障害が発生した場合の運用（ワーカーノードの追加） 1890
16.16.1 ワーカーノードでノード障害が発生した場合 1890
16.16.2 マルチノード構成から削除されたワーカーノードを再度追加する際の留意事項 1891
16.16.3 マルチノード構成から削除されたワーカーノードを再度追加する手順 1891
16.17 トラブルシュート（マルチノード機能の使用時） 1893
16.17.1 マルチノード構成の HADB サーバの開始または終了に関するトラブル 1893
16.17.2 AP に関するトラブル（マルチノード機能の使用時） 1895
16.17.3 コマンドに関するトラブル（マルチノード機能の使用時） 1896
16.17.4 DB エリアに関するトラブル（マルチノード機能の使用時） 1896
16.17.5 DB ディレクトリのファイルに関するトラブル（マルチノード機能の使用時） 1897
16.17.6 サーバディレクトリ（インストール時）のファイルに関するトラブル（マルチノード機能の使用時） 1897
16.17.7 サーバディレクトリ（運用時）のファイルに関するトラブル（マルチノード機能の使用時） 1897
16.17.8 クライアントディレクトリのファイルに関するトラブル（マルチノード機能の使用時） 1898
16.17.9 OS に関するトラブル（マルチノード機能の使用時） 1898
16.17.10 ハードウェアに関するトラブル（マルチノード機能の使用時） 1898
16.17.11 HA モニタに関するトラブル（マルチノード機能の使用時） 1901
16.17.12 作業表を作成する SQL 文に関するトラブル（マルチノード機能の使用時） 1902
16.17.13 同義語辞書ファイルに関するトラブル（マルチノード機能の使用時） 1902
16.18 セカンダリノードの新規追加，または不要なセカンダリノードの削除 1905
16.18.1 セカンダリノードの新規追加 1905
16.18.2 セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項 1906
16.18.3 セカンダリノードの削除 1910
16.19 ワーカーノードの新規追加，または不要となったワーカーノードの削除 1912
16.19.1 ワーカーノードの新規追加 1912
16.19.2 不要となったワーカーノードの削除 1913
16.20 サーバマシンの OS のホスト名・IP アドレスの変更（マルチノード機能使用時） 1914
16.21 マルチノード機能を使用するシステムへの移行 1916

16.21.1 HADB サーバを正常終了する 1916
16.21.2 サーバディレクトリのバックアップを取得する 1916
16.21.3 DB ディレクトリのバックアップを取得する 1916
16.21.4 HADB サーバをバージョンアップする 1917
16.21.5 HA モニタおよび HADB サーバをインストールする 1917
16.21.6 各ノードの設定をする 1917
16.21.7 DB ディレクトリを準備する 1918
16.21.8 監査証跡機能を使用するための準備を行う 1918
16.21.9 マルチノード構成の HADB サーバを開始する 1919
16.21.10 同義語検索の準備を行う 1919
16.22 HADB サーバのバージョンアップ（マルチノード機能の使用時） 1920
16.23 修正版 HADB サーバとの入れ替え（マルチノード機能の使用時） 1921
16.23.1 マルチノード構成の HADB サーバを正常終了して入れ替えを実施する方法 1921
16.23.2 ノードごとに入れ替えを実施する方法 1921
16.24 SQL トレース機能を使用する際の留意事項（マルチノード機能使用時） 1925
16.25 CSV ファイル中のデータを検索するときの運用（マルチノード機能使用時） 1926
16.26 同義語検索を行う際の運用（マルチノード機能使用時） 1927
16.26.1 同義語検索を行うための準備をする 1927
16.26.2 同義語検索を行う 1935
16.26.3 同義語辞書を登録または削除する 1935
16.26.4 同義語を追加または削除する 1935
16.26.5 同義語辞書ファイルの同期を取る 1936
16.27 監査証跡機能の運用（マルチノード機能使用時） 1938
16.27.1 監査証跡機能の環境設定 1938
16.27.2 監査証跡機能の運用方法 1940
16.27.3 JP1/Audit Management - Manager との連携（マルチノード機能使用時） 1940
16.28 更新行のカラム化機能の運用（マルチノード機能使用時） 1943
16.29 SQL パラレル実行機能の運用 1944
16.29.1 SQL パラレル実行機能を使用するときの準備作業 1944
16.29.2 運用 1946
16.30 マルチノード機能使用時の注意事項 1948

第 7 編 コールドスタンバイ構成編

17 コールドスタンバイ構成の運用 1949
17.1 この章を読む際の留意事項 1950
17.2 コールドスタンバイ構成でのシステム構成例 1951
17.2.1 前提ソフトウェア 1952
17.2.2 サーバ構成 1952

17.2.3 ネットワーク構成 1953
17.2.4 ストレージ構成 1954
17.3 コールドスタンバイ構成でのシステムの構築 1959
17.3.1 HA モニタのインストール 1959
17.3.2 HADB サーバのインストールおよび環境設定 1959
17.3.3 HADB クライアントのインストールおよび環境設定 1960
17.3.4 HA モニタの環境設定 1960
17.3.5 各 HADB サーバでのサーバ定義の作成 1984
17.3.6 クライアント定義の作成 1984
17.3.7 データベースの作成 1985
17.4 コールドスタンバイ構成の開始と終了 1992
17.4.1 コールドスタンバイ構成の開始方法 1992
17.4.2 コールドスタンバイ構成の終了方法 1993
17.5 AP の運用（コールドスタンバイ構成の場合） 1995
17.5.1 HADB サーバへの接続 1995
17.5.2 実行系で障害が発生した場合 1995
17.6 データベースのバックアップ（コールドスタンバイ構成の場合） 1996
17.6.1 バックアップの取得方法 1996
17.6.2 バックアップからの回復方法 1997
17.6.3 バックアップの運用例（OS コマンドを使用） 1998
17.6.4 バックアップの運用例（ShadowImage を使用） 2002
17.7 状態監視（コールドスタンバイ構成の場合） 2006
17.7.1 メモリの使用状況を確認する場合 2006
17.8 DB エリアの運用（コールドスタンバイ構成の場合） 2007
17.8.1 データ用 DB エリアの追加・削除・拡張（コールドスタンバイ構成の場合） 2007
17.8.2 作業表用 DB エリアの運用（コールドスタンバイ構成の場合） 2008
17.9 表およびインデクスの運用（コールドスタンバイ構成の場合） 2009
17.9.1 実表の運用（コールドスタンバイ構成の場合） 2009
17.9.2 インデクスの運用（コールドスタンバイ構成の場合） 2010
17.10 DB ディレクトリの運用（コールドスタンバイ構成の場合） 2013
17.11 統計情報の運用（コールドスタンバイ構成の場合） 2014
17.12 障害が発生した場合の運用 2015
17.12.1 実行系で障害が発生した場合 2015
17.12.2 待機系で障害が発生した場合 2015
17.12.3 実行系と待機系の間で通信障害が発生した場合 2015
17.12.4 コールドスタンバイ構成への系復帰 2016
17.13 コールドスタンバイ構成での計画系切り替え 2017
17.14 トラブルシュート（コールドスタンバイ構成の場合） 2018
17.14.1 コールドスタンバイ構成の開始または終了に関するトラブル 2018

17.14.2 AP に関するトラブル（コールドスタンバイ構成の場合） 2020
17.14.3 コマンドに関するトラブル（コールドスタンバイ構成の場合） 2020
17.14.4 DB ディレクトリのファイルに関するトラブル（コールドスタンバイ構成の場合） 2021
17.15 サーバマシンの OS のホスト名・IP アドレスの変更（コールドスタンバイ構成の場合） 2022
17.16 HADB サーバのバージョンアップ（コールドスタンバイ構成の場合） 2024
17.17 修正版 HADB サーバとの入れ替え（コールドスタンバイ構成の場合） 2025
17.17.1 コールドスタンバイ構成を正常終了して入れ替えを実施する方法 2025
17.17.2 系ごとに入れ替えを実施する方法 2025
17.18 コールドスタンバイ構成で同義語検索を行う場合の運用 2028
17.18.1 同義語辞書ファイルを格納するファイルシステムの準備 2028
17.18.2 servers ファイルの指定の変更 2028
17.18.3 同義語辞書ファイルを格納するファイルシステムの構築 2029
17.18.4 同義語辞書ファイルを格納するファイルシステムのマウント 2029
17.19 コールドスタンバイ構成で監査証跡機能を使用する場合の運用 2031
17.19.1 監査証跡の出力先ディレクトリを作成するファイルシステムの準備 2031
17.19.2 監査証跡の保存先ディレクトリを作成するファイルシステムの準備 2032
17.19.3 servers ファイルの指定の変更 2032
17.19.4 JP1/Audit Management - Manager との連携（コールドスタンバイ構成の場合） 2034
17.20 コールドスタンバイ構成で更新行のカラム化機能を使用する場合の運用 2035

第 8 編 Docker および Kubernetes 編

18 Docker および Kubernetes を使用した HADB の運用 2036
18.1 この章を読む際の留意事項 2037
18.2 Docker および Kubernetes を使用したシステム構成例 2038
18.2.1 前提 OS および前提ソフトウェア 2038
18.2.2 ノード構成 2039
18.2.3 コンテナ構成 2039
18.2.4 ネットワーク構成 2039
18.2.5 ストレージ構成 2039
18.2.6 カーネルパラメタ 2040
18.3 前提ソフトウェアのセットアップ 2041
18.3.1 Docker 環境のインストール 2041
18.3.2 Kubernetes 環境のインストールとノードの構築 2042
18.4 Docker での HADB の環境構築 2043
18.4.1 Dockerfile による Docker イメージの生成 2043
18.4.2 コンテナ内の HADB が使用する DB ディレクトリの初期化 2052
18.5 Kubernetes を使用した HADB の環境構築 2054
18.5.1 Kubernetes の各リソースオブジェクトの生成 2054

18.6 Docker および Kubernetes を使用した HADB の運用 2059
18.6.1 HADB のコマンドの実行方法 2059
18.6.2 adbsql コマンドの実行方法 2060
18.6.3 AP の運用 2062
18.7 Docker および Kubernetes の使用停止 2063
18.7.1 Kubernetes の使用停止 2063
18.7.2 Docker の使用停止 2063

付録 2065

付録 A HADB サーバのディレクトリの構成 2066
付録 A.1 サーバディレクトリ（インストール時）の構成 2066
付録 A.2 サーバディレクトリ（運用時）の構成 2071
付録 A.3 DB ディレクトリの構成 2079
付録 B ディクショナリ表 2081
付録 B.1 ディクショナリ表の概要 2081
付録 B.2 SQL_TABLES の内容 2087
付録 B.3 SQL_COLUMNS の内容 2090
付録 B.4 SQL_DIV_TABLE の内容 2096
付録 B.5 SQL_INDEXES の内容 2096
付録 B.6 SQL_DIV_INDEX の内容 2100
付録 B.7 SQL_DBAREAS の内容 2100
付録 B.8 SQL_SCHEMATA の内容 2101
付録 B.9 SQL_VIEWS の内容 2102
付録 B.10 SQL_VIEW_TABLE_USAGE の内容 2102
付録 B.11 SQL_DEFINE_SOURCE の内容 2103
付録 B.12 SQL_DEFINE_ENVIRONMENT の内容 2104
付録 B.13 SQL_USERS の内容 2105
付録 B.14 SQL_TABLE_CONSTRAINTS の内容 2106
付録 B.15 SQL_INDEX_COLINF の内容 2106
付録 B.16 SQL_KEY_COLUMN_USAGE の内容 2107
付録 B.17 SQL_REFERENTIAL_CONSTRAINTS の内容 2107
付録 B.18 SQL_TABLE_PRIVILEGES の内容 2108
付録 B.19 SQL_AUDITS の内容 2112
付録 B.20 ディクショナリ表の参照時に排他が取得される実表 2112
付録 B.21 ディクショナリ表（実表）の B-tree インデクス 2113
付録 B.22 ディクショナリ表の検索 2115
付録 C システム表 2133
付録 C.1 システム表の概要 2133
付録 C.2 STATUS_TABLES の内容 2136

付録 C.3　STATUS_COLUMNS の内容　2137
付録 C.4　STATUS_INDEXES の内容　2139
付録 C.5　STATUS_CHUNKS の内容　2139
付録 C.6　STATUS_SYNONYM_DICTIONARIES の内容　2141
付録 C.7　システム表の参照時に排他が取得される実表　2142
付録 C.8　システム表（実表）の B-tree インデクス　2142
付録 C.9　システム表の検索　2143
付録 D　HADB の最大値と最小値　2152
付録 D.1　システム構成に関する最大値と最小値　2152
付録 D.2　データベースに関する最大値と最小値　2153
付録 E　HADB サーバで起動するプロセス　2159

用語解説　2160

索引　2181

1 概要

Hitachi Advanced Data Binder（HADB）は大規模なデータを管理し，高速検索を実現するためのデータベース管理システム（DBMS）です。この章では，HADB の概要，特長，およびシステム構成について説明します。

1.1 情報爆発時代のデータを管理する HADB

Hitachi Advanced Data Binder（HADB）は，大規模リレーショナルデータベースの構築と高速検索を実現する，データベース管理システム（DBMS）です。

近年，クラウドコンピューティングの浸透に象徴されるような，社会インフラの変化に伴って，企業や社会が保有するデータ量は爆発的に増加しています。また，大容量ストレージの普及によって，その傾向はより顕著になっています。

社会インフラの変化に伴って，DBMS が管理するデータ量も急激に増加しています。その結果，現行の DBMS では満足な検索性能が得られなくなるケースも考えられるようになり，増加するデータをどのように管理していくかが課題となっています。また一方で，膨大なデータを活用して新たなサービスを立ち上げることができないか，といったことへの関心も高まっています。

このような背景から，次のような DBMS が求められるようになってきています。

- ペタバイトやエクサバイトといった単位の膨大なデータを管理できる。
- 膨大なデータの中から必要な情報を高速に検索できる。

HADB はこれらの要求に応えるために開発された DBMS です。

HADB はペタバイトやエクサバイトといった単位のデータを管理することを想定して設計されています。HADB を使用することによって，膨大なデータを管理するデータベースを構築できます。また，HADB ではハードウェアの性能を有効に利用する検索処理方式（非順序実行方式）を採用しています。これによって大規模データの高速検索を実現しています。

> **メモ**
>
> 1 ペタバイトは，1 テラバイトの約 1,000 倍です。1 エクサバイトは 1 ペタバイトの約 1,000 倍となります。

1.2　HADB の特長

HADB の特長を次に示します。

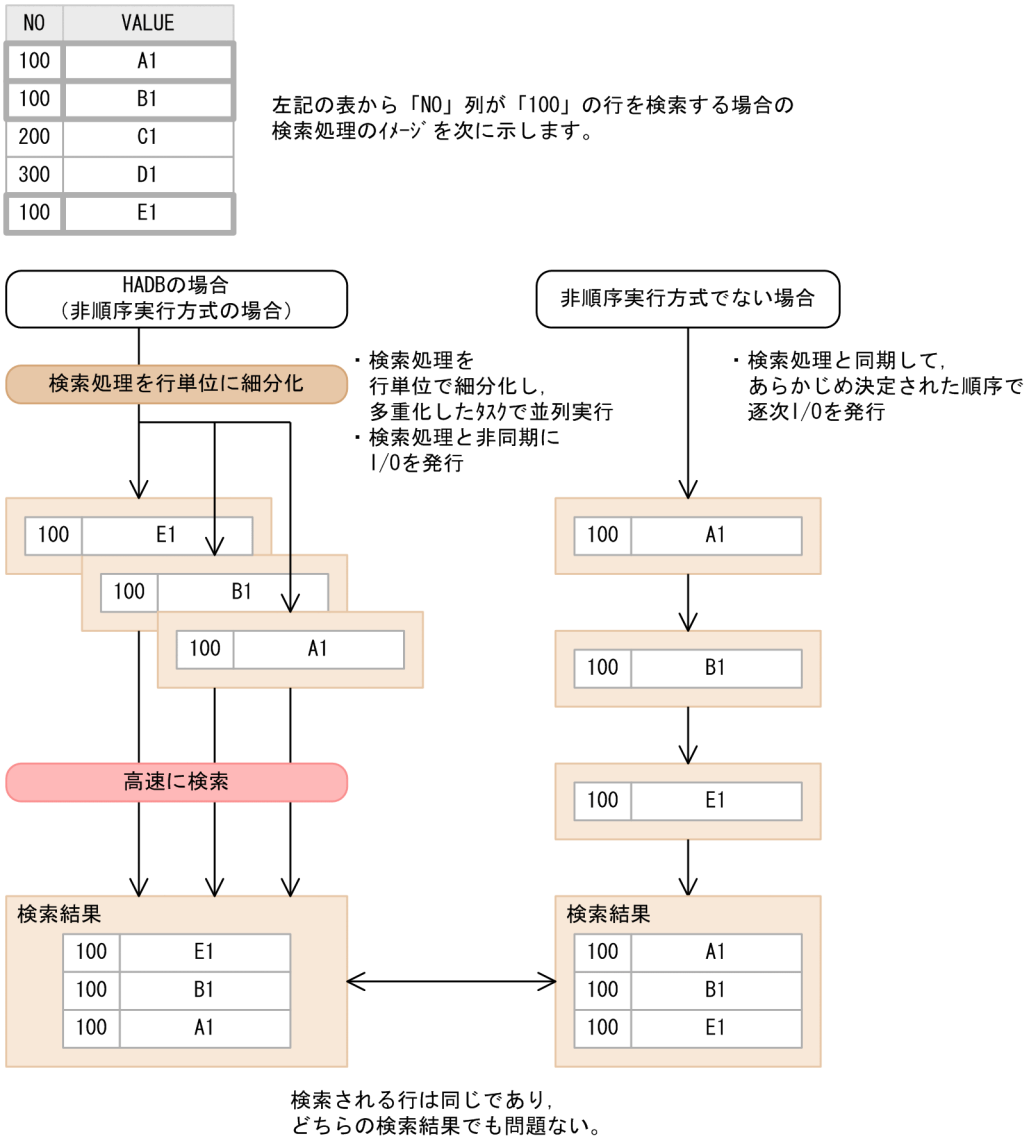
1.2.1　大規模データを高速に検索できる

HADB を使用することで，大規模データに対して，SQL を使用した複雑な検索が高速に実行できます。

HADB では非順序型実行原理に基づく検索処理方式（**非順序実行方式**）を採用しています。非順序実行方式では，検索処理を行単位に細分化し，複数のタスクに割り当てて並列に実行します。非順序実行方式を採用していない DBMS と比較して，検索タスクを超高多重で実行できます。これによって，高速な検索を実現します。

非順序実行方式による検索処理のイメージを次の図に示します。

図 1-1　非順序実行方式による高速検索



非順序実行方式は，「検索処理は行の集合演算であり，ソート処理などを除いて，DBMS が返却する検索結果の順序には規定はない」という，リレーショナルデータベースの実行原理に基づいて考案されています。

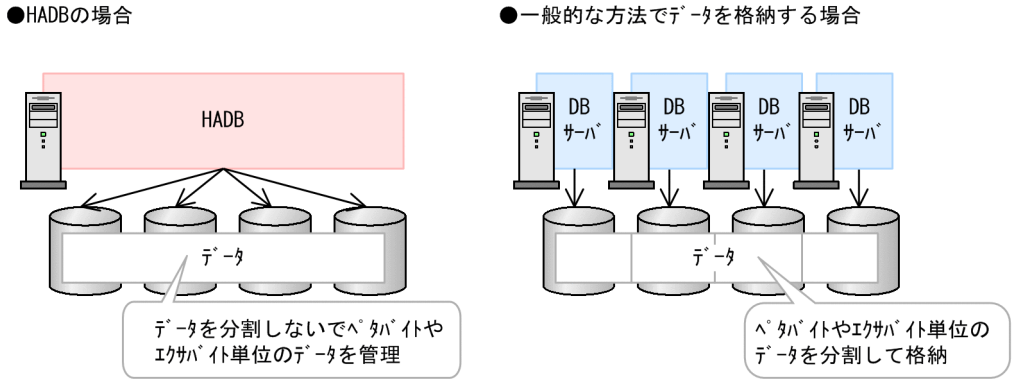
1.2.2　ペタバイトやエクサバイト単位の大規模データベースを構築できる

HADB を使用することで，ペタバイトやエクサバイトといった単位の膨大なデータを管理するデータベースを構築できます。

大規模なデータを管理する場合，データを分割して別々の領域に格納する方法が一般的です。しかし，この方法ではデータ量が増えた場合，データを分割して格納し直す必要があります。また，データを分割格納したことに伴うオーバーヘッドが発生します。

HADB では，複数のディスクに対して 1 つの論理的な領域を作成できます。この領域に表を格納することで，データを分割することなく，1 表でペタバイトやエクサバイトといった単位のデータを管理できます。データ量が増えた場合でも，データを分割して格納し直す必要はありません。大規模データベースの構築イメージを次の図に示します。

図 1-2　ペタバイトやエクサバイト単位の大規模データベースの構築



また，大規模なデータベースを作成するためのデータインポート機能などを提供します。

1.2.3　スケールアウトによる負荷分散が実現できる（マルチノード機能）

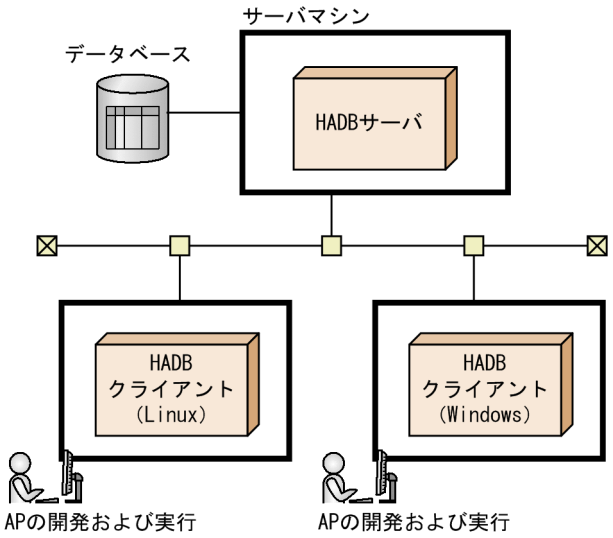
複数の HADB サーバで 1 つのデータベースを共有し，複数の HADB サーバが連携して処理を実行することができます。これによって，スケールアウトによる負荷分散が実現できます。あとから必要に応じてサーバマシンを追加できるため，業務の拡張に合わせたスケールアウトができます。

> **メモ**
>
> 複数の HADB サーバを連携して処理を実行する機能を，**マルチノード機能**といいます。

1.3　HADB のシステム構成

HADB は，クライアント・サーバシステムのネットワーク環境で使用します。HADB のシステム構成を次の図に示します。

図 1-3　HADB のシステム構成例



1.3.1　HADB サーバ

HADB サーバは，HADB のデータベースを管理するサーバ側システムです。データベースに対するトランザクション制御などをします。また，HADB サーバでは，AP の開発や実行もできます。

HADB サーバの適用 OS を次に示します。

- RHEL 6
  6.2 以降のバージョンで動作します。
- RHEL 7
  7.1 以降のバージョンで動作します。
- RHEL 8
  8.1 以降のバージョンで動作します。

なお，HADB は Intel64 アーキテクチャだけで動作します。AMD64 アーキテクチャでは動作しません。

1.3.2　HADB クライアント

HADB サーバに接続するクライアント側システムを **HADB クライアント**といいます。HADB クライアントでは，AP の開発や実行ができます。

HADB クライアントは，Linux および Windows で利用できます。HADB クライアントを利用したい場合は，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

2 アーキテクチャ

この章では，HADB のデータベースの論理構造，データベースのアクセス処理方式などについて説明します。

2.1　表の種類

HADB では，表形式でデータを管理します。

一般的なリレーショナルデータベースと同様に，HADB の表は，行と列から構成されています。表の横方向を**行**，縦方向を**列**といいます。各行の同じ列には同じ形式のデータが格納されます。

1 つの行は 1 つ以上の列から構成されています。行は表を操作するときの単位となります。

表には実表とビュー表があります。各表の詳細を次に示します。

2.1.1　実表

実表とは，データベースに格納している実体がある表のことです。

実表には，表データの格納形式が異なるローストア表とカラムストア表があります。また，表内にチャンクを 1 つだけ作成できるシングルチャンク表と，表内に複数のチャンクを作成できるマルチチャンク表があります。実表を定義する際に，ローストア表にするか，カラムストア表にするかを選択し，さらにシングルチャンク表にするか，マルチチャンク表にするかを選択します。したがって，定義した実表は，次のどれかになります。

- ローストア表でシングルチャンク表
- ローストア表でマルチチャンク表
- カラムストア表でシングルチャンク表
- カラムストア表でマルチチャンク表

ローストア表，カラムストア表，シングルチャンク表，およびマルチチャンク表の特徴については，「2.2 実表の種類」を参照してください。

2.1.2　ビュー表

CREATE VIEW 文に指定した問合せ式の結果を，新たな表として定義した仮想的な表を**ビュー表**といいます。複数の表に対する集合演算の結果をビュー表として定義したり，表中の特定の行や列だけを取り出してビュー表として定義したりすることができます。

何度も指定される条件で検索した結果を，あらかじめビュー表として定義しておくことで，データベースを検索する操作が簡単になります。ビュー表の例を次の図に示します。

図 2-1　ビュー表の例

■STOCK（実表）

PNO（商品番号）	PCODE（商品コード）	PNAME（商品名）	PRICE（単価）	QUANTITY（在庫量）
01010	101	Blouse	3500	62
01011	101	Blouse	3500	85
02021	202	Shirt	3640	67
03530	353	Shirt	4760	18
03531	353	Shirt	4760	26
04121	412	Sweater	8400	8
05910	591	Socks	300	300
05911	591	Socks	300	90

■VSTOCK（ビュー表）

PCODE	PRICE	QUANTITY
202	3640	67
353	4760	18
353	4760	26

[説明]

実表のSTOCK 表から，PNAME（商品名）列が「Shirt」の行である，PCODE（商品コード）列，PRICE（単価）列，およびQUANTITY（在庫量）列で構成されるビュー表として，VSTOCK 表を作成しています。

メモ

ビュー表を定義すると，次のようなセキュリティ上の利点があります。

- 表中の特定の列や行だけを参照可能とするビュー表を定義することで，関係のないデータや，アクセスされては困るデータを隠すことができます。
- 実際にデータが格納されている実表の表名をデータベースの利用者に知らせる必要がなくなるため，実表名を利用した不正アクセスに対する防止効果があります（ビュー表を基表としたビュー表を作成した場合）。

データベースの利用者にデータを参照させる場合，実際にデータが格納されている実表を直接参照させることもできますが，表内の一部のデータ（特定の列など）だけを参照させたい場合は，ビュー表を使用することを推奨します。

(1)　更新可能ビュー表と読み取り専用ビュー表

ビュー表には，次に示す 2 種類があります。ビュー表の定義方法によって，更新可能ビュー表または読み取り専用ビュー表のどちらかになります。

- **更新可能ビュー表**

  実表と同じように，行の挿入，行の更新，および行の削除ができます。ただし，列の追加，およびインデクスの定義はできません。

更新可能ビュー表に対して，行の挿入・更新・削除をした場合，更新可能ビュー表の基表の行も挿入・更新・削除されます。

- **読み取り専用ビュー表**

  行の挿入，行の更新，および行の削除はできません。また，列の追加，およびインデクスの定義もできません。

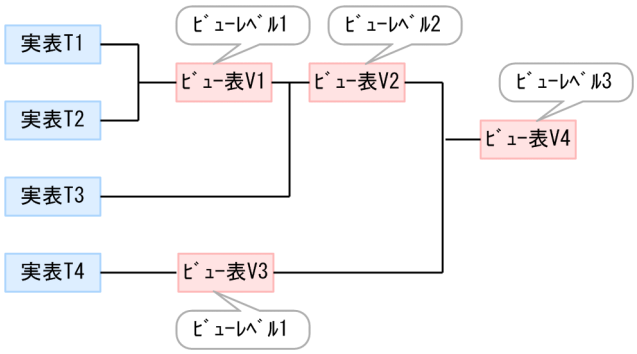
(2) ビュー表の基表について

ビュー表の基になる表を基表といいます。CREATE VIEW 文の問合せ式中に指定した表が基表になります。

■ビューレベルとは

定義したビュー表が，実表から何番目の階層になるのかを示す値のことです。ビューレベルの例を次の図に示します。

図 2-2 ビューレベルの例



[説明]

- ビュー表V1 の基表はすべて実表のため，ビュー表V1 のビューレベルは 1 になります。
- ビュー表V2 の基表は実表とビューレベル 1 のビュー表のため，ビュー表V2 のビューレベルは 2 になります。
- ビュー表V3 の基表は実表のため，ビュー表V3 のビューレベルは 1 になります。
- ビュー表V4 の基表は，ビューレベル 2 のビュー表とビューレベル 1 のビュー表のため，ビュー表V4 のビューレベルは 3 になります。

メモ

- ディクショナリ表およびシステム表は，ビューレベル 1 のビュー表として扱われます。
- 表値構成子によって導出される導出表は，実表と同じ扱いになります。

(3) ビュー表の無効化

実表またはビュー表に対して，次に示す操作を実行した場合，操作対象の表に依存するすべてのビュー表が無効化されます。

1. DROP TABLE 文（削除動作の指定を省略）で実表を削除した場合

   この場合，削除した実表に依存するすべてのビュー表が無効化されます。

2. DROP VIEW 文（削除動作の指定を省略）でビュー表を削除した場合

   この場合，削除したビュー表に依存するすべてのビュー表が無効化されます。

3. ALTER TABLE 文で実表の表名を変更した場合

   この場合，表名を変更した実表に依存するすべてのビュー表が無効化されます。

4. ALTER TABLE 文で実表の列名を変更した場合

   この場合，列名を変更した実表に依存するすべてのビュー表が無効化されます。

5. ALTER TABLE 文で配列型の列の最大要素数を変更した場合

   この場合，最大要素数を変更した実表に依存するすべてのビュー表が無効化されます。

6. ALTER TABLE 文で，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更した場合

   この場合，アーカイブマルチチャンク表に変更した表に依存するすべてのビュー表が無効化されます。

7. ALTER TABLE 文で，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更した場合

   この場合，レギュラーマルチチャンク表に変更した表に依存するすべてのビュー表が無効化されます。

8. ALTER VIEW 文でビュー表を再作成した場合

   この場合，再作成したビュー表に依存するすべてのビュー表が無効化されます。

9. REVOKE 文で表に対するSELECT 権限を取り消した場合※

   この場合，SELECT 権限を取り消した表に依存するすべてのビュー表が無効化されます。

   注※

   - 認可識別子を指定して与えたSELECT 権限と，PUBLIC 指定で与えたSELECT 権限の両方が取り消された場合が該当します。
   - 付与権付きのSELECT 権限が取り消された場合，またはSELECT 権限の付与権だけが取り消された場合，ビュー表が無効化されることがあります。ほかの HADB ユーザにSELECT 権限を付与した際に使用した付与権が取り消された場合，付与したSELECT 権限が取り消されます。SELECT 権限が取り消された表が基表の場合，その基表に依存するビュー表が無効化されます。具体例については，マニュアル『HADB　SQL リファレンス』の『アクセス権限の取り消し』の『例題』を参照してください。

     付与権については，「2.7.5　アクセス権限」の「(3)　アクセス権限の付与」を参照してください。

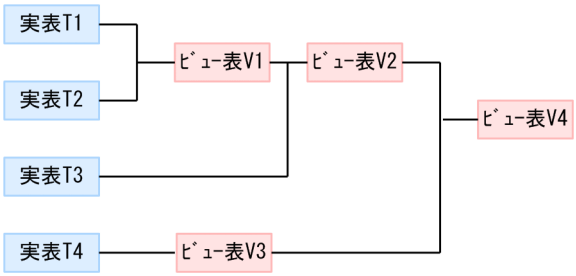
■ビュー表の無効化とは

ビュー表に対するアクセスができなくなる状態のことを，ビュー表の無効化といいます。ビュー表の無効化を解除してビュー表に再びアクセスできるようにするには，ビュー表の無効化要因を対策したあとに，ALTER VIEW 文でビュー表を再作成する必要があります。または，ビュー表の無効化要因を対策したあとに，いったんDROP VIEW 文でビュー表を削除して，CREATE VIEW 文でビュー表を再定義する必要があります。

■依存するビュー表とは

上記の 1.～7.の操作によって，影響を受けるビュー表のことです。依存するビュー表の例を次の図に示します。

図 2-3　依存するビュー表の例



[説明]

- 実表T1 に依存するビュー表は，ビュー表V1，V2，V4 になります。
- 実表T2 に依存するビュー表は，ビュー表V1，V2，V4 になります。
- 実表T3 に依存するビュー表は，ビュー表V2，V4 になります。
- 実表T4 に依存するビュー表は，ビュー表V3，V4 になります。
- ビュー表V1 に依存するビュー表は，ビュー表V2，V4 になります。
- ビュー表V2 に依存するビュー表は，ビュー表V4 になります。
- ビュー表V3 に依存するビュー表は，ビュー表V4 になります。
- ビュー表V4 に依存するビュー表はありません。

2.2 実表の種類

ここでは，実表の種類（ローストア表とカラムストア表，シングルチャンク表とマルチチャンク表）について説明します。

2.2.1 ローストア表とカラムストア表

実表には，ローストア表とカラムストア表があり，それぞれ表データの格納形式が異なります。業務内容や表の利用方法によって，これら 2 つの表を使い分けると，表の検索性能の向上が見込まれます。

(1) ローストア表およびカラムストア表とは

表データの格納形式には，ローストア形式とカラムストア形式の 2 つがあり，表を定義するときに表データの格納形式を指定することができます。表データの格納形式にローストア形式を選択して定義した表を**ローストア表**といい，表データの格納形式にカラムストア形式を選択して定義した表を**カラムストア表**といいます。

ローストア表には，ローストア形式でデータが格納されます。カラムストア表には，カラムストア形式でデータが格納されます。ただし，カラムストア表に対してINSERT 文やUPDATE 文を実行した場合，追加・更新したデータは，ローストア形式でカラムストア表に格納されます。

■カラムストア表と，表データの格納形式の関係

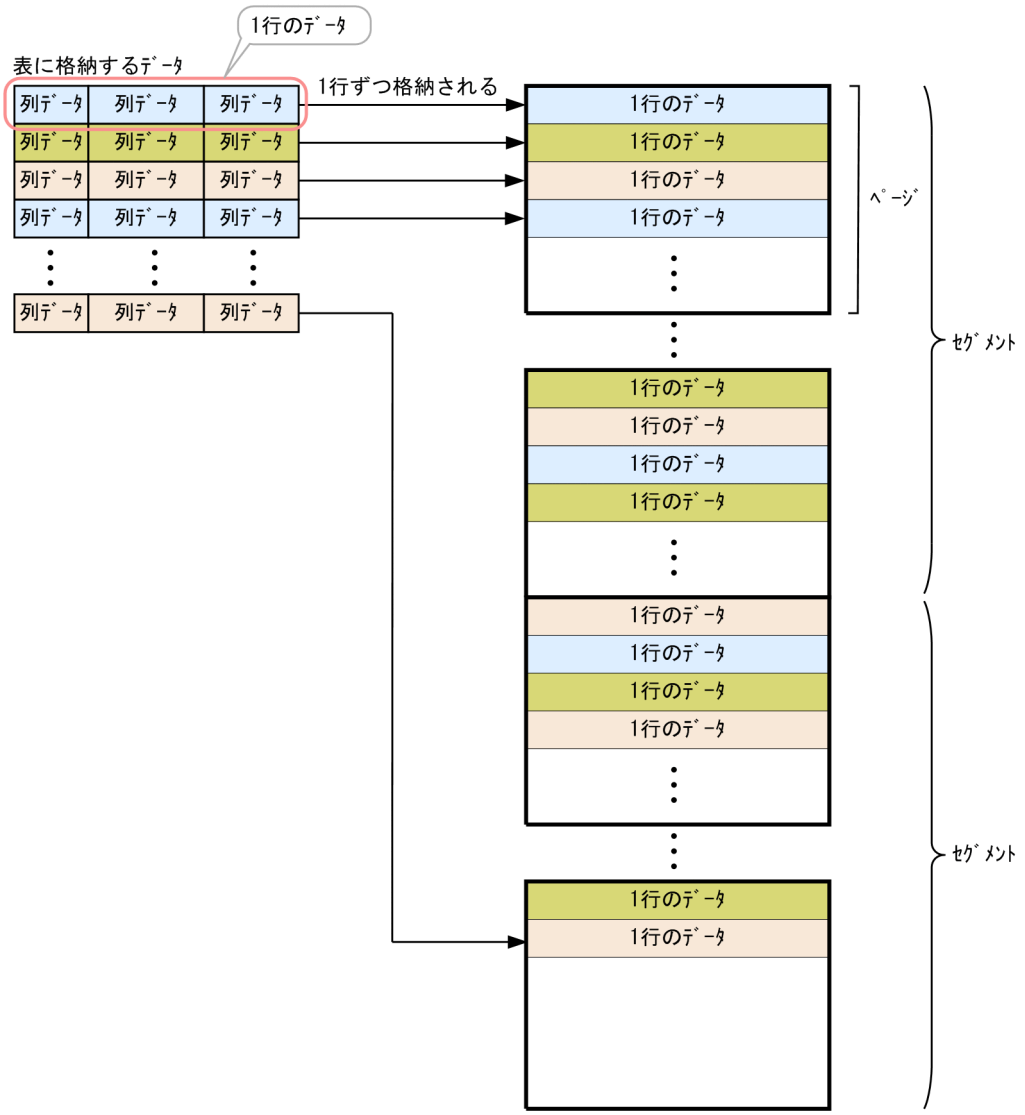
- カラムストア表にadbimport コマンドでインポートしたデータは，カラムストア形式で格納されます。
- カラムストア表にINSERT 文やUPDATE 文で追加・更新したデータは，ローストア形式で格納されます。

> **メモ**
>
> バージョン 04-01 より前のバージョンの HADB サーバで定義した表は，ローストア表になります。

(2) ローストア形式とは

データを行単位でデータベースに格納する形式のことを**ローストア形式**といいます。ローストア形式の場合，1 行のデータが 1 レコードとしてデータベースに格納されます。ローストア形式の場合のデータ格納形式を次の図に示します。

図 2-4　ローストア形式の場合のデータ格納形式



[説明]

- 1 行のデータが 1 レコードとしてデータベースに格納されます。
- 1 ページには，最大 255 行分のデータが格納できます。

> **メモ**
>
> ページ，セグメントについては，「2.4.3　DB エリアの構造（セグメントとページ）」を参照してください。

■ローストア表に適した検索

ローストア表の場合，データが行単位で格納されているため，次のような検索をするときに適しています。

- 行単位でデータにアクセスするような検索をする場合

例えば，次のように，選択式に*を指定したSELECT 文を実行する場合が該当します。

```
SELECT * FROM "T1" WHERE "C1">=DATE'2017-09-06'
```

また，選択式にほぼすべての列を指定するような場合も該当します。

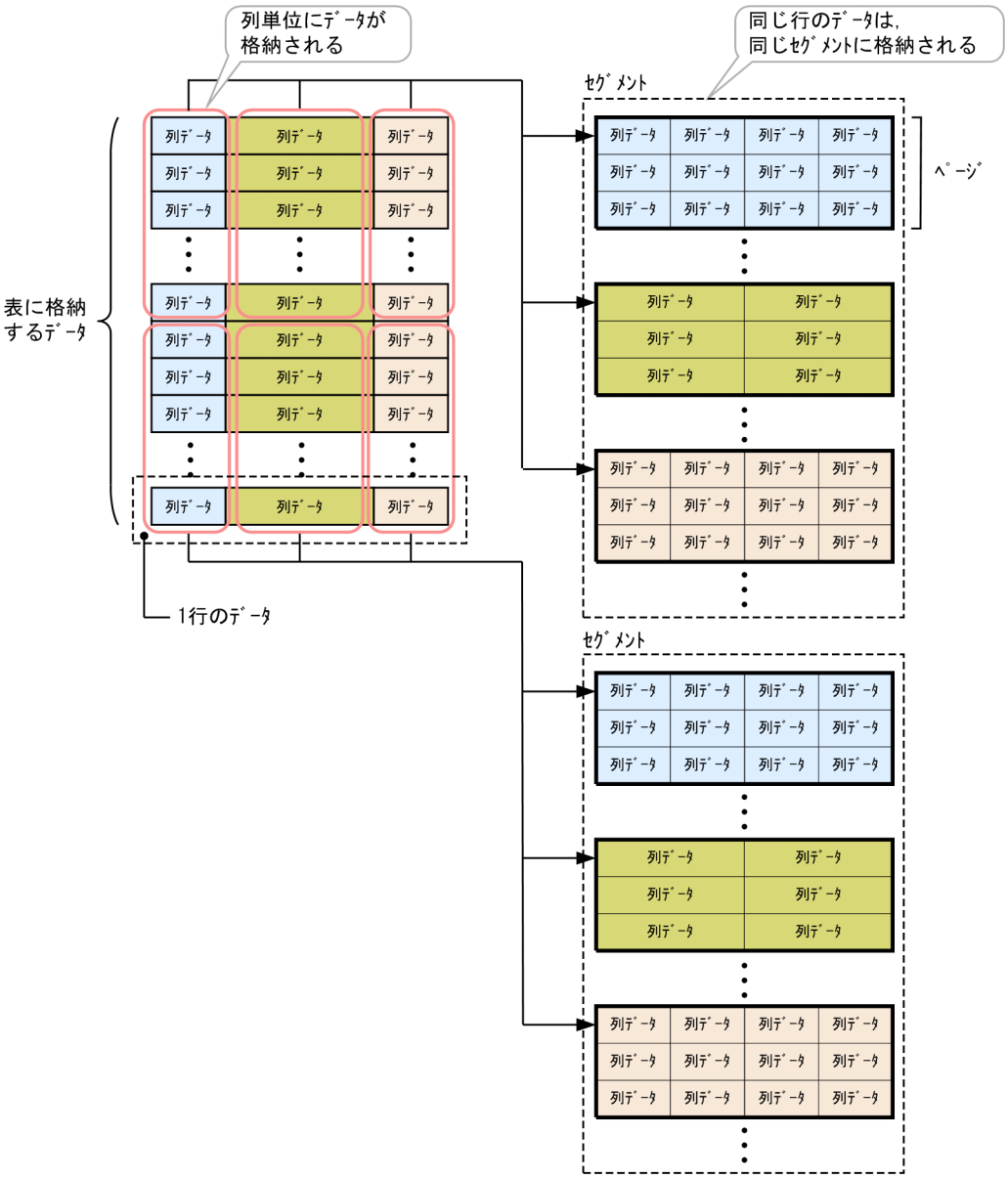
- 検索対象のデータを，B-tree インデクスを使用して絞り込むような検索をする場合

  表の検索が実行された場合，HADB サーバは，検索対象の行が格納されているページだけを読み込みます。そのため，B-tree インデクスを使用して検索対象の行を絞り込むことができる場合は，読み込むページ数を削減することができます。

(3) カラムストア形式とは

データを列単位でデータベースに格納する形式のことを**カラムストア形式**といいます。カラムストア形式の場合，表の各列のデータが列ごとにまとまってデータベースに格納されます。カラムストア形式の場合のデータ格納形式を次の図に示します。

図 2-5　カラムストア形式の場合のデータ格納形式



[説明]

- 各列のデータが，列単位でデータベースに格納されます。
- 同じ行のデータは，同じセグメントに格納されます。
  1 セグメント内には，最大 262,144 行分の列データが格納できます。

メモ

ページ，セグメントについては，「2.4.3　DB エリアの構造（セグメントとページ）」を参照してください。

■カラムストア表に適した検索

カラムストア表の場合，データが列単位でまとまって格納されているため，次のような検索をするときに適しています。

- 検索対象のデータをあまり絞り込まないで，特定の列データ全体を検索する場合
- 特定の範囲内（特定の年や月など）の，特定の列データにアクセスする場合
- 特定の列データに対する値の集計（平均や合計などを求める）をすることが多い場合
- 特定の列データのグループ化をすることが多い場合

表の検索が実行された場合，HADB サーバは，検索対象の列データが格納されているセグメントを最初に探します。そのあとに，セグメント内の，検索対象の列データが格納されているページを読み込みます。そのため，特定の列データの集計をする場合は，集計対象の列データが格納されているページだけにアクセスすればよいため，読み込むページ数を削減することができます。

■列データの圧縮方式

カラムストア表にデータをインポートする際，各列のデータは圧縮されて表に格納されます。列データの圧縮方式には，幾つかの種類があります。列データの圧縮方式の種類については，「5.2.2　ロースト ア表とカラムストア表の選択基準」の「(4)　カラムストア表の列データの圧縮方式」を参照してください。

> **メモ**
>
> カラムストア表にデータをインポートする際，インポートするデータを基に，HADB サーバが自動的に圧縮方式を選択します。また，カラムストア表を定義する際に，列データの圧縮方式を選択することもできます。

■カラムストア形式のデータの削除と，無効情報管理用ページの関係

カラムストア表に格納されているカラムストア形式のデータをDELETE 文で削除すると，対象のデータは無効となります。ディスクからは削除されません。

さらに，HADB サーバは，対象のデータが無効になったことを示す情報を**無効情報管理用ページ**で管理します。カラムストア表に対してDELETE 文を実行すると，無効情報管理用ページが確保される分，データ用 DB エリアの空き容量が減ります。

そのため，データ用 DB エリアの空き容量が少ない場合に，カラムストア形式のデータをDELETE 文で削除すると，空き容量不足でエラーとなるおそれがあります。

■カラムストア表に対して INSERT 文または UPDATE 文を実行する場合

カラムストア表に対してINSERT 文またはUPDATE 文を実行すると，追加または更新されたデータはローストア形式でカラムストア表に格納されます。カラムストア表にローストア形式のデータが格納された場合，カラムストア表の特性が生かされなくなるおそれがあります。例えば，特定の列データにアクセスする際の検索性能が低下したり，データの圧縮率が低下したりします。そのため，カラムストア表に対してINSERT 文またはUPDATE 文を実行することがある場合は，**更新行のカラム化機能**を有効にすることを推奨します。更新行のカラム化機能を有効にすると，HADB が，カラムストア表内のローストア形式のデータをカラムストア形式に自動的に変換します。更新行のカラム化機能については，「11.18 更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

2.2.2　シングルチャンク表とマルチチャンク表

実表には，シングルチャンク表とマルチチャンク表があり，それぞれ使用できる機能が異なります。ここでは，シングルチャンク表とマルチチャンク表の説明と，各表で使用できる機能について説明します。

(1)　シングルチャンク表とは

シングルチャンク表とは，1 表に 1 つのチャンクだけを作成できる実表のことです。シングルチャンク表の場合，バックグラウンドインポート機能を使用することができません。

チャンクおよびバックグラウンドインポート機能については，「2.14　バックグラウンドインポート機能」を参照してください。

(2)　マルチチャンク表とは

マルチチャンク表とは，1 表に複数のチャンクを作成できる実表のことです。マルチチャンク表の場合，バックグラウンドインポート機能を使用できます。

また，マルチチャンク表には，レギュラーマルチチャンク表とアーカイブマルチチャンク表の 2 種類があります。

- **レギュラーマルチチャンク表**
  レギュラーマルチチャンク表の場合，マルチチャンク表の基本機能であるバックグラウンドインポート機能は使用できますが，チャンクアーカイブ機能は使用できません。
- **アーカイブマルチチャンク表**
  アーカイブマルチチャンク表の場合，バックグラウンドインポート機能，およびチャンクアーカイブ機能の両方が使用できます。

チャンクアーカイブ機能については，「2.15　チャンクアーカイブ機能（チャンク内のデータの圧縮）」を参照してください。

重要

シングルチャンク表とマルチチャンク表で，使用できる機能が異なります。関係を次の表に示します。

表 2-1　実表の種類と使用できる機能の関係

実表の種類		使用できる機能	
		バックグラウンドインポート機能	チャンクアーカイブ機能
シングルチャンク表		×	×
マルチチャンク表	レギュラーマルチチャンク表	○	×

実表の種類		使用できる機能	
		バックグラウンドインポート機能	チャンクアーカイブ機能
	アーカイブマルチチャンク表	○	○

（凡例）

○：使用できます。

×：使用できません。

(3) ローストア表およびカラムストア表との関係

表を設計する際は，表をローストア表とカラムストア表のどちらにするかを最初に決めます。そのあとに，表をシングルチャンク表とマルチチャンク表（レギュラーマルチチャンク表またはアーカイブマルチチャンク表）のどちらにするかを決めます。ただし，表をカラムストア表とする場合は，その表をアーカイブマルチチャンク表にすることはできません。

ローストア表およびカラムストア表と，シングルチャンク表およびマルチチャンク表の関係を次の表に示します。

表 2-2 ローストア表およびカラムストア表と，シングルチャンク表およびマルチチャンク表の関係

実表の種類	実表の種類		
	シングルチャンク表	マルチチャンク表	
		レギュラーマルチチャンク表	アーカイブマルチチャンク表
ローストア表	○	○	○
カラムストア表	○	○	×

（凡例）

○：この種類の表を定義できます。

×：この種類の表を定義できません。

2.3　インデクスの種類

インデクスには，B-tree インデクス，テキストインデクス，レンジインデクスの 3 種類があります。それぞれのインデクスについて説明します。

2.3.1　B-tree インデクス

B-tree インデクスは，B-tree 構造でキー値を管理しています。表の検索時に B-tree インデクスを使用することで，探索条件を満たす行に直接アクセスできます。その分，表の検索性能を向上させることができます。

2.3.2　テキストインデクス

テキストインデクスは，SQL 文の検索条件に指定した文字列を含む文字データを検索する場合に使用するインデクスです。テキストインデクスは，テキストインデクスを定義した実表の列に格納されているデータ（文字列）の出現情報を，ページ単位で管理しています。

SQL 文の検索条件に次の指定をして表を検索する場合に，テキストインデクスを使用すると，読み込むページを少なくすることができます。その分，表の検索性能を向上させることができます。

- LIKE 述語
- LIKE_REGEX 述語
- スカラ関数CONTAINS

テキストインデクスは，次の 2 つの管理部で構成されます。

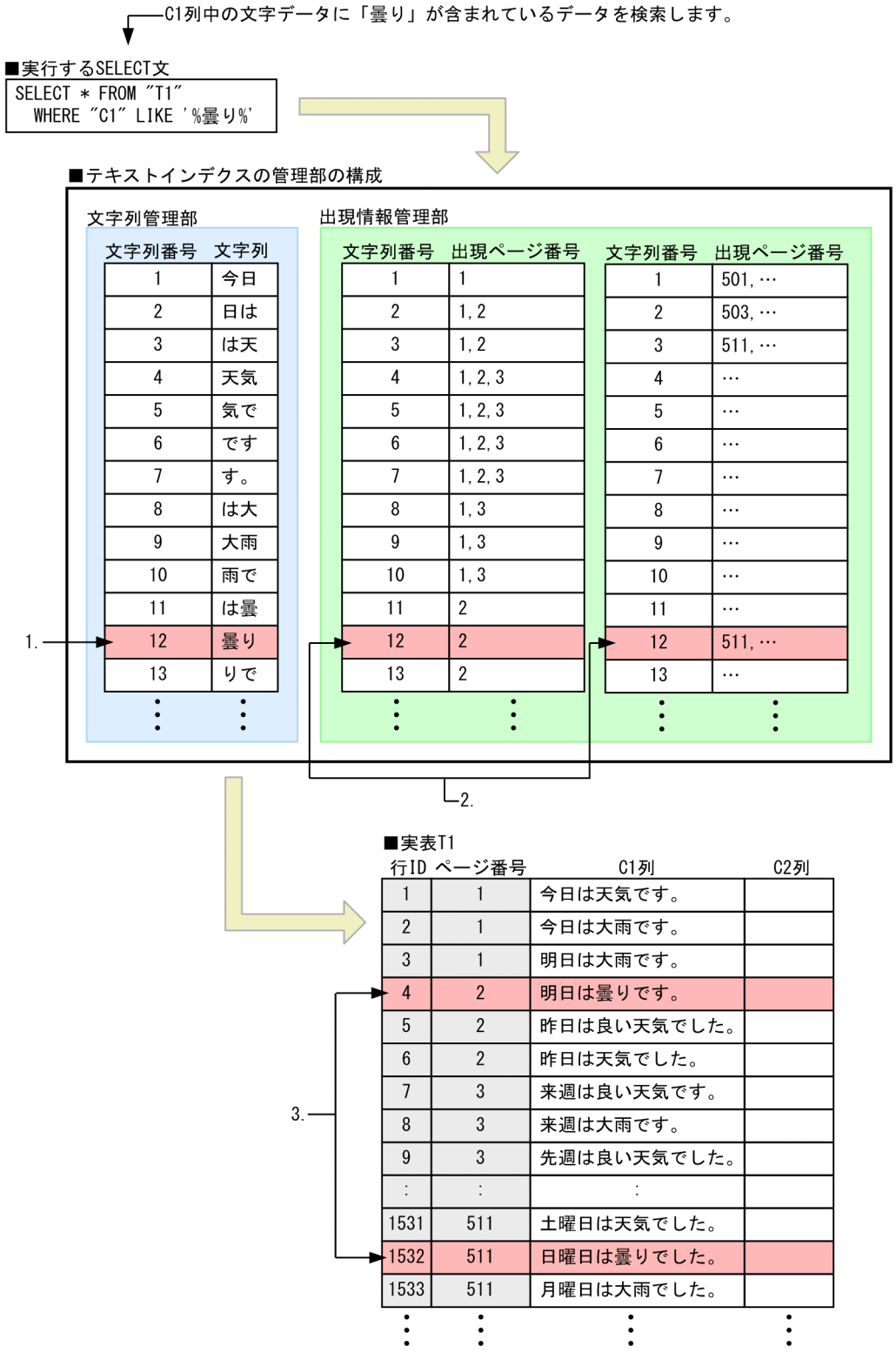
- **文字列管理部**
  テキストインデクス化された文字列を管理します。
- **出現情報管理部**
  テキストインデクス化された文字列の出現情報を管理します。

テキストインデクスの管理部の構成を，次の図に示します。

図 2-6　テキストインデクスの管理部の構成

また，テキストインデクスを使用して実表を検索する場合の処理の流れを，次の図に示します。

図 2-7　テキストインデクスを使用した検索処理の概要



[説明]

次の流れで，テキストインデクスを使用して検索を行います。

1. 文字列管理部から，LIKE 述語中に指定した検索条件の文字列（「曇り」）を検索します。そして，該当する文字列番号（「12」）を取得します。

2. 出現情報管理部から，文字列管理部で検索した文字列番号（「12」）をキーにして，該当する出現ページ番号（「2」と「511」）を検索します。

3. 実表T1 に対して，出現情報管理部の出現ページ番号（「2」と「511」）に対応するデータページを読み込みます。そして，該当する行 ID（「4」と「1532」）を検索して，表データを取得します。このとき，該当する出現ページ番号以外のデータページは読み込みません。

> **メモ**
>
> 検索条件に指定した列に B-tree インデクスおよびテキストインデクスが定義されている場合，表の検索時にはどちらか片方のインデクスだけが使われます。ただし，検索条件の指定内容によっては，両方のインデクスが使われないこともあります。

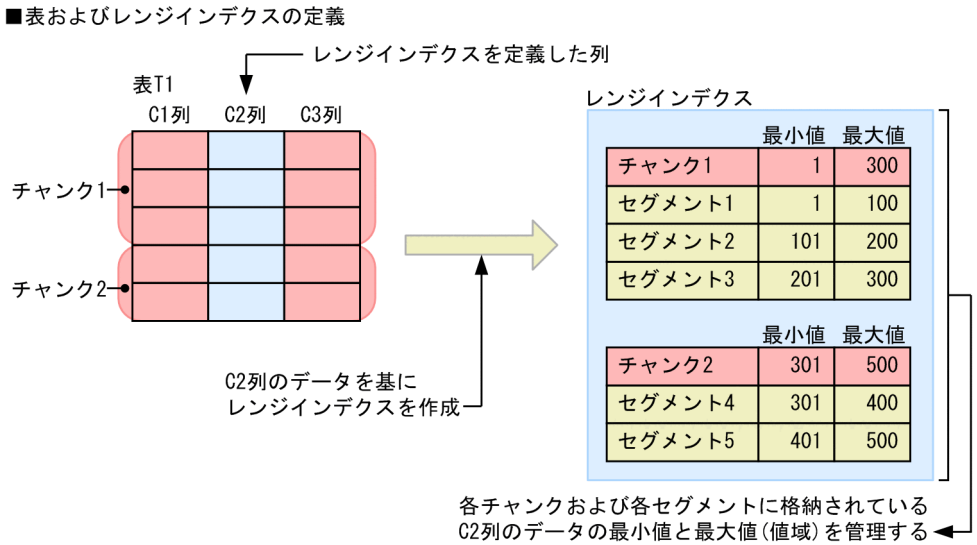
2.3.3 レンジインデクス

レンジインデクスは，レンジインデクスを定義した実表に格納されているデータのうち，次に示す 2 つの値域（列の値の最小値と最大値）を管理しています。

- チャンク内に格納されているデータの値域
- セグメント内に格納されているデータの値域

レンジインデクスが管理する値域の概要について次の図に示します。

図 2-8 レンジインデクスが管理する値域の概要



レンジインデクスを使用すると，探索条件を満たさない値域のデータを格納しているチャンクの検索処理がスキップされます（チャンクのスキップができます）。

また，チャンクのスキップ後に，探索条件を満たす値域のデータを格納しているチャンク内で，探索条件を満たさない値域のデータを格納しているセグメントの検索処理がスキップされます（セグメントのスキップができます）。

不要なチャンクおよびセグメントの検索処理をスキップすることで，その分検索性能が向上します。

- **チャンクのスキップ**

  探索条件を満たさない値域のデータを格納しているチャンクの検索処理をスキップすることです。

  チャンクの概要については，「2.14.2　データインポート単位のデータ管理（チャンク）」を参照してください。チャンクのスキップの詳細については，「2.14.3　チャンクとレンジインデクスの関係」を参照してください。

- **セグメントのスキップ**

  探索条件を満たさない値域のデータを格納しているセグメントの検索処理をスキップすることです。

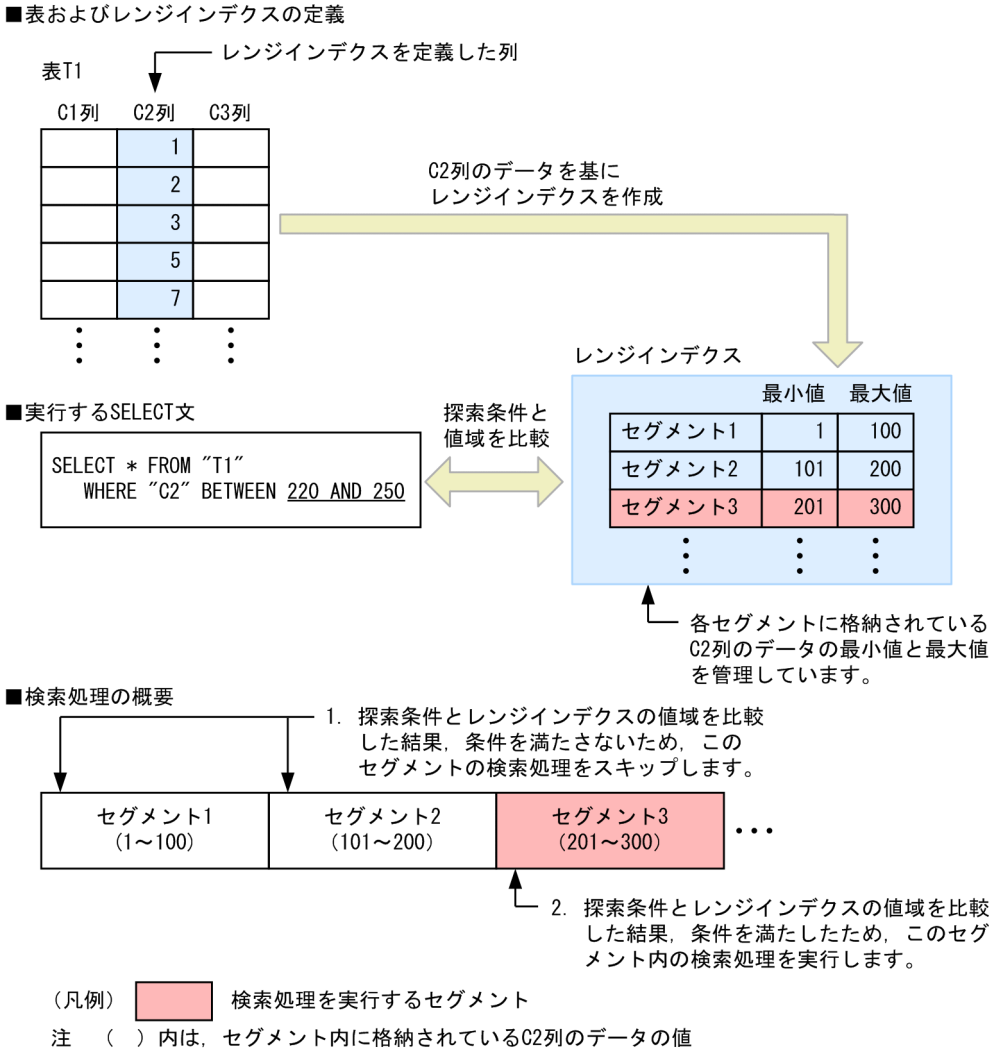
  セグメントの概要については，「2.4.3　DB エリアの構造（セグメントとページ）」の「(1)　セグメント」を参照してください。セグメントのスキップの詳細については，次に示します。

> **メモ**
>
> **■レンジインデクスと非順序実行方式の関係**
>
> 非順序実行方式が適用される SQL 文の場合，検索処理の開始時点（カーソルオープン時点）で，レンジインデクスが管理しているチャンク内に格納されているデータの値域を読み込みます。そして，メモリに保存します。検索処理の実行中には，メモリに保存した情報に従って，チャンクのスキップを行うかどうかを判定します。
>
> なお，次の条件をすべて満たすレンジインデクスが対象になります。
>
> - B-tree インデクスを使用する検索の対象表に定義されているレンジインデクスである
> - チャンクのスキップで使用されるレンジインデクスである

(1)　レンジインデクスの概要

ここでは，セグメントのスキップを例に，レンジインデクスの概要を次の図に示します。

図 2-9　レンジインデクスの概要（セグメントのスキップ）



(2)　レンジインデクスの特徴

レンジインデクスは，主に次に示す 4 つの特徴を持っています。チャンクのスキップができるレンジインデクスの特徴については，「2.14.3　チャンクとレンジインデクスの関係」の「(2)　レンジインデクスの特徴（チャンクのスキップ)」を参照してください。

(a)　B-tree インデクスやテキストインデクスとレンジインデクスを合わせて使用できる

レンジインデクスを使用してチャンクのスキップを行ったあとで，B-tree インデクスやテキストインデクスを使用できます。B-tree インデクスやテキストインデクスとレンジインデクスを使用した検索については，「2.14.3　チャンクとレンジインデクスの関係」の「(2)　レンジインデクスの特徴（チャンクのスキップ)」の「(a)　B-tree インデクスやテキストインデクスとレンジインデクスを合わせて使用できる」を参照してください。

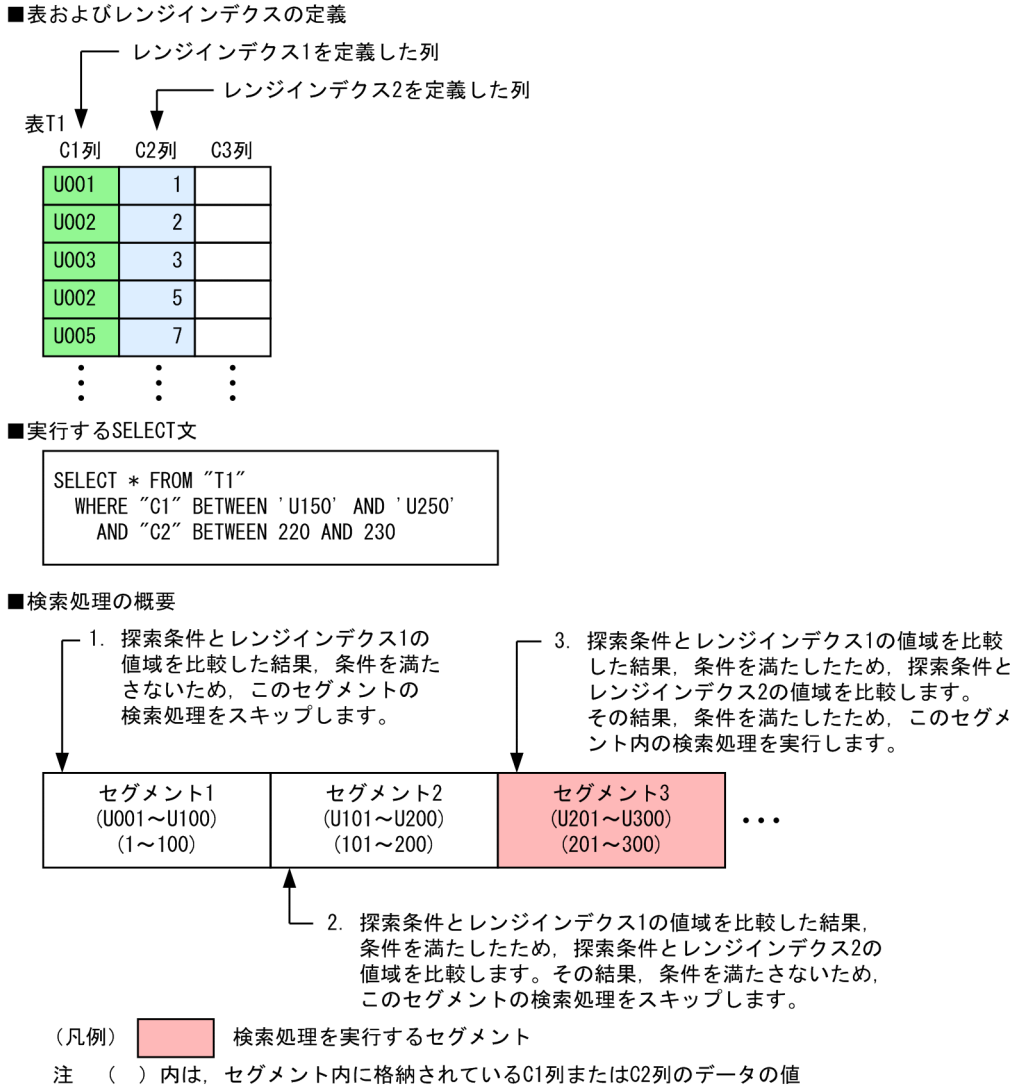
メモ

B-tree インデクスやテキストインデクスと合わせて使用する場合，レンジインデクスを使用した検索（セグメントのスキップ）は行われません。

(b)　複数のレンジインデクスを使用できる

複数のレンジインデクスを使用した検索の例を次の図に示します。

図 2-10　複数のレンジインデクスを使用した検索の例



(c)　B-tree インデクスやテキストインデクスよりデータ容量が少なくなる

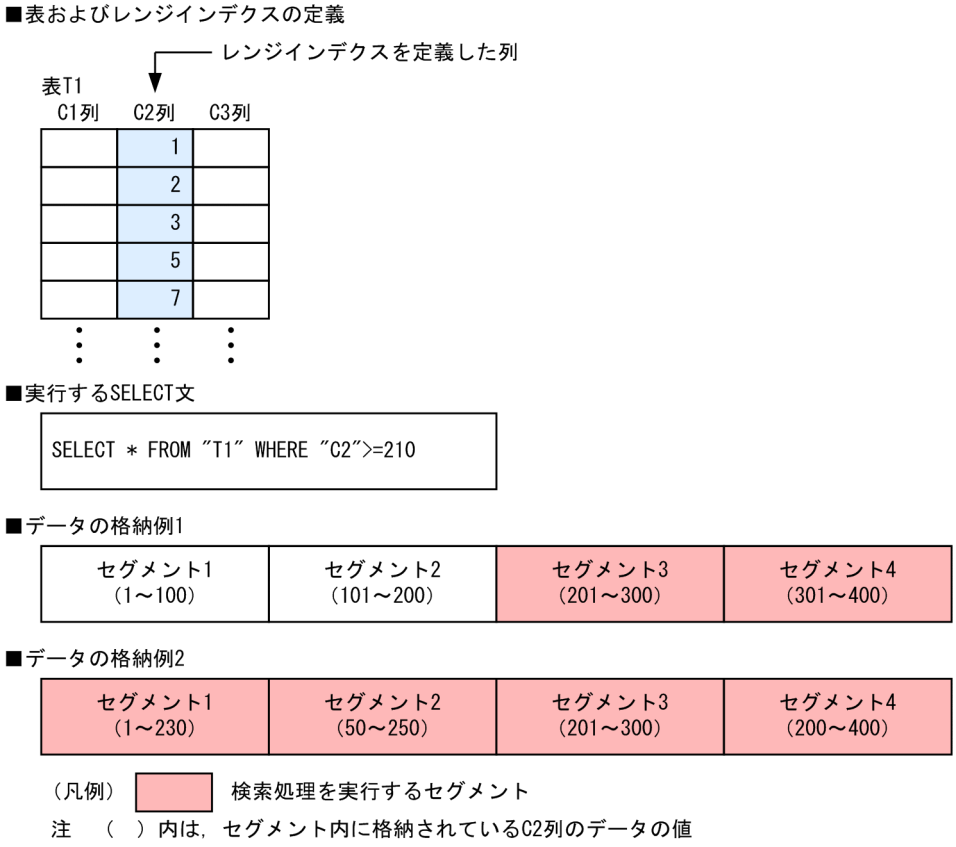
B-tree インデクスの場合は，表の 1 行に対して 1 つのインデクスキーを作成します。また，テキストインデクスの場合は，表の 1 行に対して，そこに含まれる文字列ごとに番号を割り振って管理します。

一方，レンジインデクスの場合は，1 セグメントに対して 1 つの値域を作成します。そのため，レンジインデクスは，B-tree インデクスやテキストインデクスよりデータ容量が少なくなります。

(d)　セグメントに格納されているデータの値域の幅によって検索性能が変わる

レンジインデクスを使用した検索では，セグメント内に格納されているデータの値域の幅によって検索性能が変わります。例を次の図に示します。

図 2-11　セグメント内に格納されているデータの値域の幅によって検索性能が変わる例



[説明]

データの格納例 1 の場合は，検索処理を実行するセグメント数が 2 つになります。一方，データの格納例 2 の場合は，値域の重複があり，各セグメントの値域が格納例 1 に比べて広くなっています。そのため，検索処理を実行するセグメント数が 4 つになります。このようにセグメントに格納されているデータの値域の幅によって，検索性能が変わります。値域を狭くすると，検索性能が向上します。

(3)　レンジインデクスの値域の更新（セグメント）

行の追加，更新を繰り返すと，レンジインデクスの値域が広がり，レンジインデクスの効果が薄れていきます（値域が狭くなることはありません）。また，行の削除を繰り返すと，セグメント内に格納されているデータの値域に比べて，レンジインデクスの値域が広くなり，レンジインデクスの効果が薄れていきます。

レンジインデクスの効果が薄れてきた場合は，対象のレンジインデクスに対してインデクス再作成を実行し，レンジインデクスを作成し直してください。

レンジインデクスの値域が広がる条件を次の表に示します。

表 2-3　レンジインデクスの値域が広がる条件（セグメント）

項番	実行した操作	レンジインデクスの値域が広がる条件
1	行の追加	値域の範囲外の値を追加した場合に，追加行を格納するセグメントの値域が広がります。
2	行の更新	値域の範囲外の値に更新した場合に，更新行を格納するセグメントの値域が広がります。
3	行の削除	値域は広がりません。

2.4　DB エリア（表およびインデクスの格納領域）

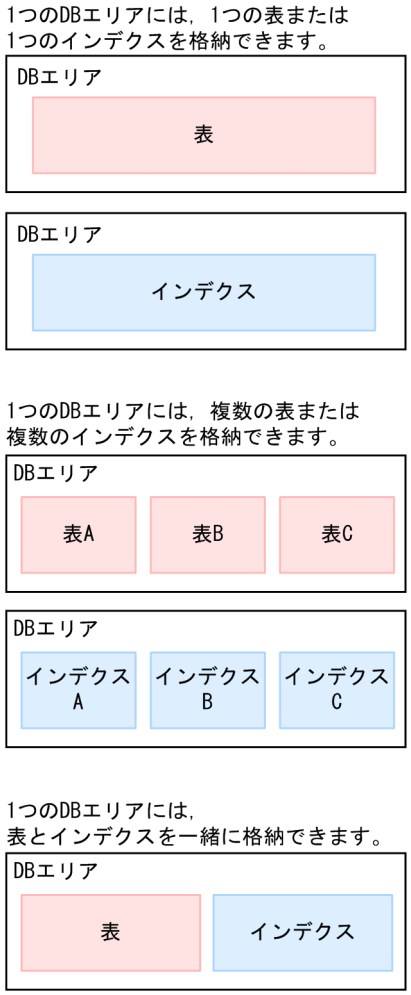
ここでは，表およびインデクスを格納する論理的な領域である DB エリアについて説明します。

2.4.1　DB エリアとは

HADB では，表およびインデクスを **DB エリア**という論理的な領域に格納します。

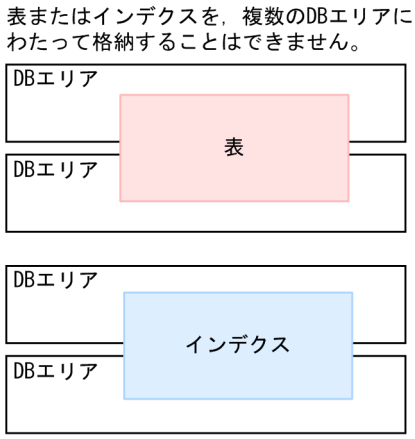
1 つの DB エリアには，複数の表またはインデクスを格納できます。DB エリアに表およびインデクスを格納できるケースを次の図に示します。

図 2-12　DB エリアに表およびインデクスを格納できるケース



ただし，1 つの表またはインデクスを複数の DB エリアにわたって格納することはできません。DB エリアに表またはインデクスを格納できないケースを次の図に示します。

図 2-13　DB エリアに表またはインデクスを格納できないケース



2.4.2　DB エリアの種類

DB エリアは用途によって分類されます。DB エリアの種類を次の表に示します。

表 2-4　DB エリアの種類

項番	DB エリアの種類	説明
1	データ用 DB エリア	表およびインデクスを格納する DB エリアです。
2	作業表用 DB エリア	SQL の実行時に HADB が作成する作業表を格納する DB エリアです。 作業表については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『作業表の種類』を参照してください。
3	マスタディレクトリ用 DB エリア	システムの内部情報を格納する DB エリアです。
4	ディクショナリ用 DB エリア	ディクショナリ表（実表），およびディクショナリ表（実表）の B-tree インデクスを格納する DB エリアです。
5	システム表用 DB エリア	システム表（実表），およびシステム表（実表）の B-tree インデクスを格納する DB エリアです。

注

データ用 DB エリア以外の DB エリアは，HADB によって自動的に作成されます。

■ディクショナリ表とは

ディクショナリ表とは，実表，ビュー表，B-tree インデクス，テキストインデクスおよびレンジインデクスの定義情報や，DB エリアの情報が格納されている表のことです。

ディクショナリ表には，実表とビュー表の 2 種類があります。詳細については，「付録 B.1　ディクショナリ表の概要」を参照してください。

■システム表とは

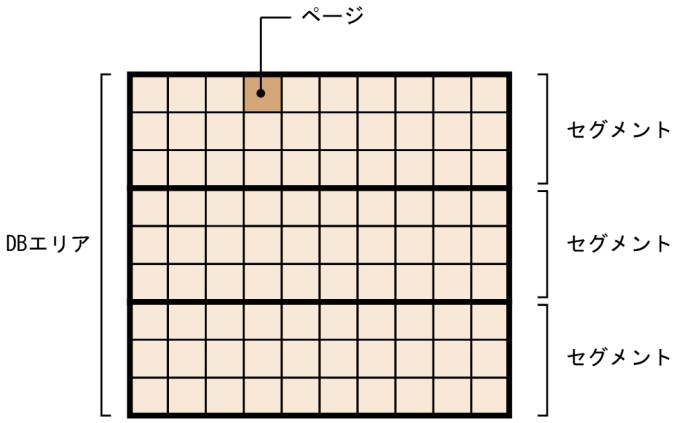
システム表とは，実表，B-tree インデクス，およびテキストインデクスに関するコスト情報や，マルチチャンク表のチャンク情報が格納されている表のことです。

システム表には，実表とビュー表の 2 種類があります。詳細については，「付録 C.1　システム表の概要」を参照してください。

2.4.3　DB エリアの構造（セグメントとページ）

DB エリアは，**セグメント**および**ページ**という単位で構成されます。1 つの DB エリアは，複数のセグメントから構成されます。また，1 つのセグメントは複数のページから構成されます。DB エリアの構造を次の図に示します。

図 2-14　DB エリアの構造

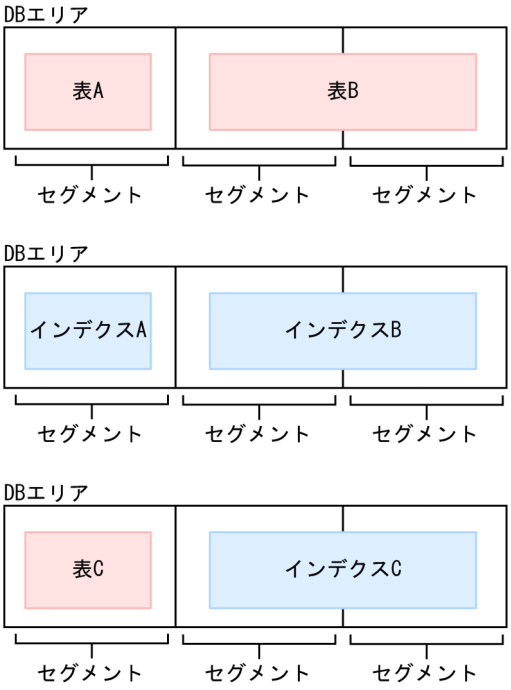


(1)　セグメント

セグメントは，表およびインデクスを格納するときの最小単位です。1 つのセグメントには，1 つの表または 1 つのインデクスが格納されます。セグメントと，表およびインデクスの関係を次の図に示します。

図 2-15　セグメントと，表およびインデクスの関係

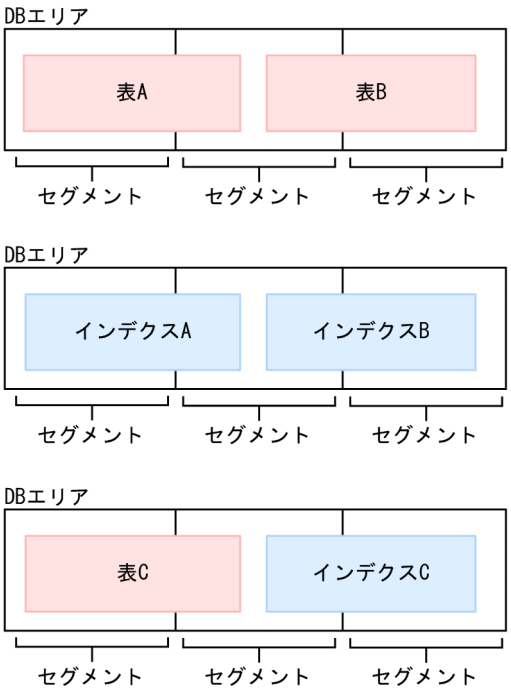
1つのセグメントには，1つの表または
1つのインデクスが格納されます。



なお，1 つのセグメントに複数の表，または複数のインデクスは格納されません。セグメントに，表およびインデクスが格納されない例を次の図に示します。

図 2-16　セグメントに，表およびインデクスが格納されない例

1つのセグメントには，複数の表または
複数のインデクスは格納されません。



(a) セグメントの種類

セグメントは，セグメントに割り当てられるページの種類によって次のように分類されます。セグメントの種類を次の表に示します。ページの種類については，「2.4.3 DB エリアの構造（セグメントとページ）」の「(2) ページ」の「(a) ページの種類」を参照してください。

表 2-5 セグメントの種類

項番	セグメントの種類		説明
1	表（ローストア形式の場合）	基本行用セグメント	基本行を格納するページ（基本行用ページ）を割り当てるセグメントです。
2		分岐行用セグメント	分岐行を格納するページ（分岐行用ページ）を割り当てるセグメントです。
3	表（カラムストア形式の場合）	カラムデータ用セグメント	カラムストア表にデータインポートした際に，カラムストア形式でデータを格納するセグメントです。 また，更新行のカラム化機能によってローストア形式からカラムストア形式に変換されたデータも，カラムデータ用セグメントに格納されます。更新行のカラム化機能については，「11.18 更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。
4		ローデータ用セグメント	カラムストア表にデータインポートした際に，分岐したデータ（分岐行）を格納するセグメントです。 また，次に示すデータも，ローデータ用セグメントに格納されます。 • INSERT 文でカラムストア表に追加したデータ • UPDATE 文で更新したカラムストア表のデータ • UPDATE 文やDELETE 文を実行した場合に，対象のデータが無効となったことを示すデータ
5	B-tree インデクス	上位ページ用のセグメント	B-tree インデクスのルートページ，および中間ページを割り当てるセグメントです。
6		下位ページ用のセグメント	B-tree インデクスのリーフページ，行 ID ディレクトリページおよび行 ID リストページを割り当てるセグメントです。
7	テキストインデクス	文字列管理部用のセグメント	テキストインデクスの文字列管理部ページを割り当てるセグメントです。
8		出現情報管理部用のセグメント	テキストインデクスの出現情報管理部ページを割り当てるセグメントです。
9	レンジインデクス	DB エリアファイル管理部用のセグメント	レンジインデクスの DB エリアファイル管理部ページを割り当てるセグメントです。
10		値域情報部用のセグメント	レンジインデクスのセグメント管理部ページ，および値域情報部ページを割り当てるセグメントです。

(b) セグメントが割り当てられるタイミング，解放されるタイミング

セグメントが割り当てられるタイミング

セグメントは，新規にページを確保したときに割り当てられます。すでに割り当てられているセグメントに，空きページがない場合は，新規にセグメントが確保されて割り当てられます。

ただし，次に示す場合は，すでに割り当てられているセグメントに空きページがあっても，新規にセグメントが確保されて割り当てられます。

- バックグラウンドインポートを実行する場合
- 複数のチャンクをマージする場合
- チャンクの状態変更でカレントチャンクが切り替わり，新規にチャンクが作成される場合
- システム表の再編成を実行する場合
- 更新行のカラム化機能によって，HADB がローストア形式のデータをカラムストア形式のデータに変換する場合

セグメントが解放されるタイミング

割り当てられたセグメントは，次のタイミングで解放されます。

- 実表およびインデクスを削除したとき
- 実表の行データをすべて削除したとき（TRUNCATE TABLE 文を実行したとき）
- 既存データをすべて削除する指定をして，実表にデータをインポートしたとき
- チャンクを削除したとき
- インデクスを再作成したとき
- 複数のチャンクをマージ後に，マージ元チャンクのインデクスのデータを削除したとき
- チャンクをアーカイブ状態にしたとき
- アーカイブマルチチャンク表から，レギュラーマルチチャンク表に変更したとき（ALTER TABLE 文を実行したとき）
- システム表の再編成を実行したとき

(2) ページ

ページは，ディスク I/O 時の最小単位になります。ディスク上のデータを読み込むときや，ディスク上にデータを書き込むときは，ページ単位で実行されます。

(a) ページの種類

ページは，格納するデータの種類によって次のように分類されます。ページの種類を次の表に示します。

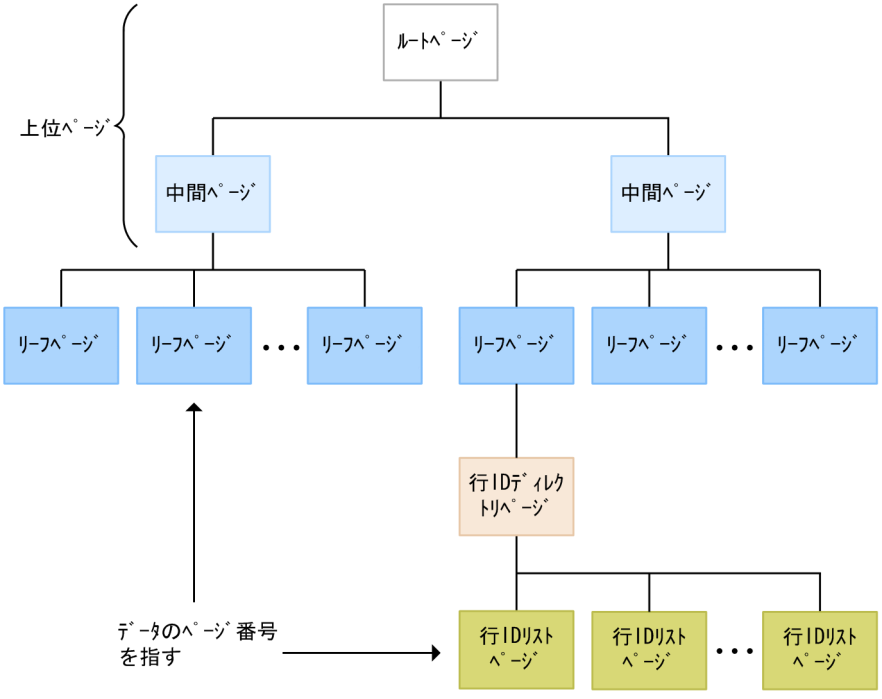
表 2-6　ページの種類

項番	ページの種類			説明
1	データページ（ローストア形式の場合）	基本行用ページ		表データの基本行を格納するページです。
2		分岐行用ページ		表データの分岐行を格納するページです。
3	データページ（カラムストア形式の場合）	エントリページ		カラムストア形式で格納した表データに関する管理情報を格納するページです。このページは，カラムデータ用セグメント内に確保されます。
4		辞書ページ		圧縮方式に辞書圧縮（DICTIONARY）を選択した場合の，辞書データを格納するページです。このページは，カラムデータ用セグメント内に確保されます。
5		列データページ		カラムストア形式で格納した各列のデータ（分岐行のデータを除く）を格納するページです。このページは，カラムデータ用セグメント内に確保されます。
6		基本行用ページ		カラムストア表にINSERT 文で追加したデータを，基本行として格納するページです。 また，カラムストア形式で格納したデータに対してUPDATE 文を実行した場合に，更新データを基本行として格納するページです。 このページは，ローデータ用セグメント内に確保されます。
7		分岐行用ページ		カラムストア形式で格納した行のデータの分岐行を格納するページです。このページは，ローデータ用セグメント内に確保されます。 実長が 128 バイト以上の列データは，分岐行としてこのページに格納されます。
8		無効情報管理用ページ		カラムストア形式で格納したデータに対して，UPDATE 文やDELETE 文を実行した場合に，対象のデータが無効となったことを示す情報を管理するページです。 このページは，ローデータ用セグメント内に確保されます。
9		割り当て管理用ページ		ローデータ用セグメントの先頭に存在し，ローデータ用セグメント内のページの割り当てを管理しています。
10	B-tree インデクスのインデクスページ※1	上位ページ	ルートページ	ルートページとは，B-tree 構造中の最上位のインデクスページのことです。
11			中間ページ	中間の B-tree インデクスページです。
12		リーフページ		最下位の B-tree インデクスページです。
13		行 ID ディレクトリページ		行 ID リストページを管理するページです。

項番	ページの種類		説明
14		行 ID リストページ	同じキー値が 256 個以上重複した場合の，そのキー値専用のリーフページです。
15	テキストインデクスのインデクスページ	文字列管理部ページ	テキストインデクス化された文字列の種類を管理するページです。
16		出現情報管理部ページ	テキストインデクス化された文字列の出現情報をページ単位で管理するページです。
17	レンジインデクスのインデクスページ	DB エリアファイル管理部ページ	レンジインデクスを定義した表の値域が，どの DB エリアファイルの値域かを管理するページです。 また，レンジインデクスを定義した表のチャンクの値域を格納するページです。
18		セグメント管理部ページ	レンジインデクスを定義した表の値域が，どのセグメントの値域かを管理するページです。
19		値域情報部ページ	レンジインデクスを定義した表が格納されているセグメントの値域を格納するページです。
20	作業表用ページ※2		作業表用 DB エリアのページです。作業表用 DB エリアのページサイズは 32 キロバイトを超えることができます。また，分岐行を格納するページはありません。
21	ディレクトリページ※3		DB エリアに関する管理情報を格納するページです。

注※1

B-tree インデクスとインデクスページの関係を次の図に示します。



注※2

作業表用ページは，作業表を作成する SQL を実行したときに確保されます。

注※3

ディレクトリページは，DB エリアを作成するときに一定数確保されます。そのあとは，データページやインデクスページが一定量に達するごとに，新たなページが確保されます。

■基本行および分岐行とは

ローストア形式の場合，1 行のデータは同一ページに格納されますが，定義長が 255 バイトを超える可変長列がある場合，1 行のデータが別ページに分割されて格納されることがあります。別ページに格納された行を**分岐行**といい，分岐先の情報を持つ行を**基本行**といいます。これは，ローストア表のデータの場合も，カラムストア表に更新系 SQL で追加したデータの場合も同じです。

なお，ローストア表の場合，可変長列データが別ページに分割されて格納されるかどうかについては，表定義時（CREATE TABLE 文実行時）に指定できます。

■ディレクトリページ群の存在とディレクトリページの配置

DB エリアファイルの先頭部分には，次に示す情報を管理するディレクトリページ群が存在します。

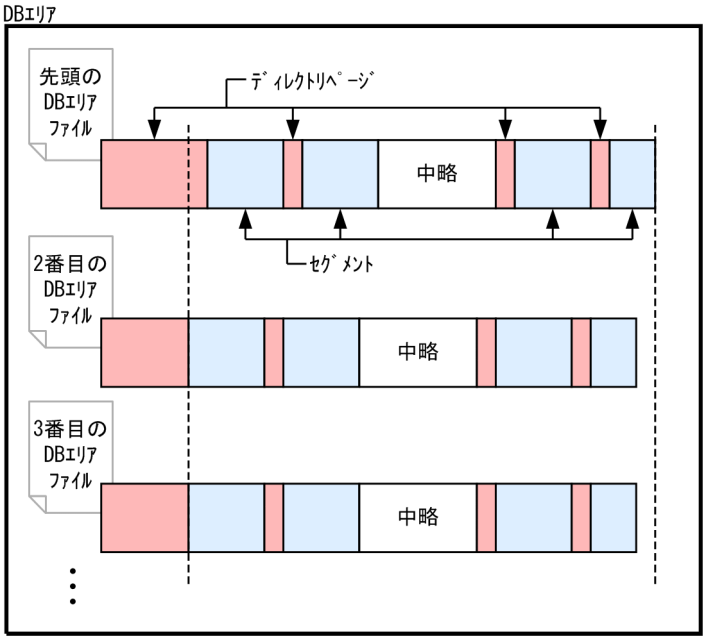
- DB エリアの定義情報
- 定義された表およびインデクスの情報
- チャンクの情報

DB エリアを構成する先頭の DB エリアファイルの場合は，ディレクトリページ群に，DB エリアの情報を管理するディレクトリページも含まれます。そのため，ディレクトリページの配置が，先頭の DB エリアファイルと 2 番目以降の DB エリアファイルでは異なります。

また，表およびインデクスを格納するセグメントは，ディレクトリページ群の後ろに配置されます。また，一定間隔でセグメントの情報を管理するディレクトリページが配置されます。そのため，DB エリアファイルのすべての領域が，表およびインデクスを格納するセグメントに使用されるわけではありません。

DB エリアファイル内のディレクトリページの配置例を，次の図に示します。

図 2-17　DB エリアファイル内のディレクトリページの配置例



(b)　ページが割り当てられるタイミング，解放されるタイミング

ページが割り当てられるタイミング

データページ，およびインデクスページが割り当てられるタイミングを次に示します。

- **ローストア表に行を追加したとき，および行を更新したとき**

  基本行用ページ，および分岐行用ページは，表に行を追加したとき，および行を更新したときに割り当てられます。CREATE TABLE 文を実行してローストア表を定義したときには割り当てられません。行の追加を繰り返して，すでに割り当てられたページが満杯になった場合は，新規にページが確保されて割り当てられます。

  レンジインデクスのインデクスページは，表に行を追加したとき，および行を更新したときに割り当てられます。

- **カラムストア表に行を追加したとき，および行を更新したとき**

  - エントリページ，辞書ページ，および列データページは，adbimport コマンドを実行してカラムストア表に行を追加したときに割り当てられます。

    基本行用ページは，更新系 SQL を実行してカラムストア表に行を追加したとき，または行を更新したときに割り当てられます。

    分岐行用ページは，adbimport コマンドまたは更新系 SQL を実行して，カラムストア表に行を追加したとき，または行を更新したときに割り当てられます。

    エントリページ，辞書ページ，列データページ，基本行用ページ，および分岐行用ページは，CREATE TABLE 文でカラムストア表を定義したときには割り当てられません。行の追加を繰り返し，割り当て済みページが満杯になったときに，新規ページが確保されて割り当てられます。

  - 無効情報管理用ページは，カラムストア形式のデータを削除したとき，またはデータを更新したときに割り当てられます。無効情報管理用ページは，CREATE TABLE 文でカラムストア表を定義

したときには割り当てられません。行の削除や行の更新を繰り返して，すでに割り当てられたページが満杯になったときに，新規にページが確保されて割り当てられます。

なお，更新系 SQL によって追加・更新されたローストア形式のデータを削除したり，更新したりしたときには，無効情報管理用ページは割り当てられません。

- レンジインデクスのインデクスページは，表に行を追加したとき，および行を更新したときに割り当てられます。

- **更新行のカラム化機能によって，HADB がローストア形式のデータをカラムストア形式のデータに変換したとき**

  カラムストア表内のローストア形式のデータをカラムストア形式のデータに変換したときに，新規にページが確保されて割り当てられます。

- **B-tree インデクスを定義したとき**

  CREATE INDEX 文を実行して B-tree インデクスを定義したときに，上位ページと，リーフページが 1 つずつ割り当てられます。

- **B-tree インデクスのインデクスページスプリットが発生したとき**

  B-tree インデクスのインデクスページスプリットが発生するたびに，新規のインデクスページが確保されて，割り当てられます。B-tree インデクスのインデクスページスプリットについては，「5.3.4　B-tree インデクスのインデクスページ内の未使用領域の確保（PCTFREE）」の「(2)　B-tree インデクスのインデクスページスプリット」を参照してください。

- **テキストインデクスを定義したとき**

  CREATE INDEX 文を実行してテキストインデクスを定義したときに，文字列管理部ページと，出現情報管理部ページが 2 つずつ割り当てられます。

- **テキストインデクスのインデクスページスプリットが発生したとき**

  テキストインデクスのインデクスページスプリットが発生するたびに，新規のインデクスページが確保されて，割り当てられます。テキストインデクスのインデクスページスプリットについては，「5.4.2　テキストインデクスのインデクスページ内の未使用領域の確保（PCTFREE）」の「(2)　テキストインデクスのインデクスページスプリット」を参照してください。

- **レンジインデクスを定義したとき**

  CREATE INDEX 文を実行してレンジインデクスを定義したときに，1 セグメント分の DB エリアファイル管理部ページが割り当てられます。

ページが解放されるタイミング

セグメントが解放されたときに，セグメント内のすべてのページが解放されます。

(c)　ページグループ

レンジインデクスの値域情報部用のセグメントからページを割り当てる場合，HADB は複数のページをまとめて確保します。このとき，まとめて割り当てられるページの単位を**ページグループ**といいます。また，ページグループを構成するページ数を**ページグループサイズ**といいます。

ページグループサイズを求める場合は，「5.8.6　レンジインデクスを格納するセグメント数の求め方」の「(1)　変数 SGRI の求め方」の変数*PGGRPSIZE* を参照してください。

レンジインデクスのページグループサイズを確認する場合は，「10.9.5　レンジインデクスの状態や使用量を確認したい場合」を参照してください。

2.4.4　DB エリアのファイル構成

表やインデクスといった DB エリアに格納されるデータは，物理的にはファイルに格納されます。DB エリアを構成するファイルを**DB エリアファイル**といいます。

(1)　DB エリアファイルとして使用できるファイル

HADB では，次に示すファイルを DB エリアファイルとして使用できます。

(a)　レギュラーファイル（ファイルシステム上のファイル）

OS が標準でサポートしているレギュラーファイルを，DB エリアファイルとして使用できます。

レギュラーファイルを更新および参照する際は，ファイルシステムの管理部に対する更新処理および参照処理が行われます。そのため，特にファイルの拡張処理が行われた場合に，ブロックスペシャルファイルよりアクセス性能が低くなります。

(b)　ブロックスペシャルファイル

ブロックスペシャルファイルを，DB エリアファイルとして使用できます。ブロックスペシャルファイルとは，ブロックデバイスの 1 つです。ディスク全体，またはパーティションで区切られたディスク上の領域を 1 ファイルと見なしてアクセスします。

ブロックスペシャルファイルは，ブロック単位にデータを操作します。そのため，ブロックスペシャルファイルは，レギュラーファイルよりも高速にアクセスできます。

ブロックスペシャルファイルは，ユーザがデータベースを作成する前に使用するデータベース領域の大きさを考慮し，割り当てておく必要があります。

なお，ブロックスペシャルファイルには，HADB サーバが読み書きできる権限を付与する必要があります。ブロックスペシャルファイルの設定を変更する場合は，必ずudev ルールを使用してください。udev ルールについては，OS のマニュアルを参照してください。

> **重要**
>
> ブロックスペシャルファイルを write モードでオープンすると，クローズ時にudev にイベントが通知され，udev ルールに設定した状態に戻ります。そのため，chmod コマンド，chown コマンドなどでブロックスペシャルファイルの設定を変更した場合は，データベースの初期設定が

完了したときや HADB サーバが終了したときに，所有者や権限がudev ルールに設定した状態に戻ります。

(2) DB エリアファイルの種類

DB エリアファイルは，DB エリアの種類に合わせて次に示す 5 種類があります。

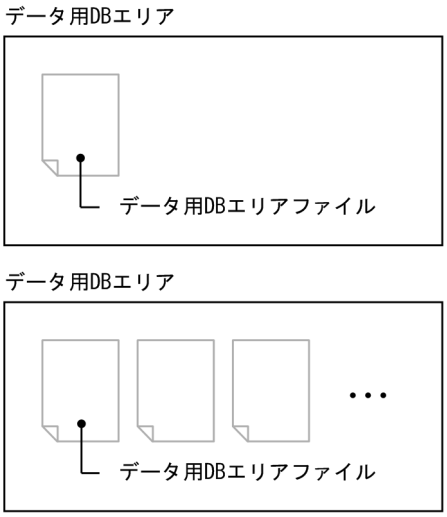
(a) データ用 DB エリアファイル

データ用 DB エリアファイルは，データ用 DB エリアを構成するファイルです。

1 つのデータ用 DB エリアは，1 つ以上のデータ用 DB エリアファイルから構成されます。1 つのデータ用 DB エリアは，最大 1,024 個のデータ用 DB エリアファイルで構成できます。

データ用 DB エリアとデータ用 DB エリアファイルの関係を次の図に示します。

図 2-18 データ用 DB エリアとデータ用 DB エリアファイルの関係



データ用 DB エリアファイルは，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに作成されます。

データ用 DB エリアファイルの格納場所は，データベースの初期設定時に指定します。また，データベースの初期設定後，データ用 DB エリアを追加するときにも指定します。

(b) 作業表用 DB エリアファイル

作業表用 DB エリアファイルは，作業表用 DB エリアを構成するファイルです。

作業表用 DB エリアファイルは，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに，1 つだけ作成されます。

作業表用 DB エリアファイルの格納場所は，データベースの初期設定時に指定します。

作業表用 DB エリアファイルは，ブロックスペシャルファイルに作成することを推奨します。レギュラーファイルに作成すると，作業表を複数作成する SQL 文が実行された場合に，レギュラーファイルを格納しているファイルシステムに対して I/O が集中してしまい，SQL 文の実行時間が長くなるおそれがあります。

(c) マスタディレクトリ用 DB エリアファイル

マスタディレクトリ用 DB エリアファイルは，マスタディレクトリ用 DB エリアを構成するファイルです。

マスタディレクトリ用 DB エリアファイルは，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに，1 つだけ作成されます。

マスタディレクトリ用 DB エリアファイルの格納場所は，データベースの初期設定時に指定します。

(d) ディクショナリ用 DB エリアファイル

ディクショナリ用 DB エリアファイルは，ディクショナリ用 DB エリアを構成するファイルです。

ディクショナリ用 DB エリアファイルは，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに，1 つだけ作成されます。

ディクショナリ用 DB エリアファイルの格納場所は，データベースの初期設定時に指定します。

(e) システム表用 DB エリアファイル

システム表用 DB エリアファイルは，システム表用 DB エリアを構成するファイルです。

システム表用 DB エリアファイルは，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに，1 つだけ作成されます。

システム表用 DB エリアファイルの格納場所は，データベースの初期設定時に指定します。

システム表用 DB エリアファイルは，ブロックスペシャルファイルに作成することを推奨します。ブロックスペシャルファイルに作成すると，システム表用 DB エリアの空き容量が不足してシステム表の再編成が必要になった場合に，HADB サーバから警告メッセージが出力されるようになります。

なお，システム表用 DB エリアファイルをレギュラーファイルに作成した場合は，システム表用 DB エリアの空き容量が不足しても，HADB サーバから警告メッセージは出力されません。システム表用 DB エリアファイルをレギュラーファイルに作成する場合は，運用を開始する前に，システム表の再編成を行うタイミングを検討してください。そして，検討結果を基に，定期的にシステム表を再編成するようにしてください。システム表の再編成については，「11.17　システム表の再編成」を参照してください。

(3) DB ディレクトリ

データ用 DB エリアファイルなど，各 DB エリアファイルが格納されるディレクトリを **DB ディレクトリ** といいます。DB ディレクトリは，データベースの初期設定時に作成されます。

DB ディレクトリには，次のファイルも格納されます。

- データベースの更新履歴情報（システムログファイル）
- HADB サーバが異常終了した場合，再開始するための情報（ステータスファイル）

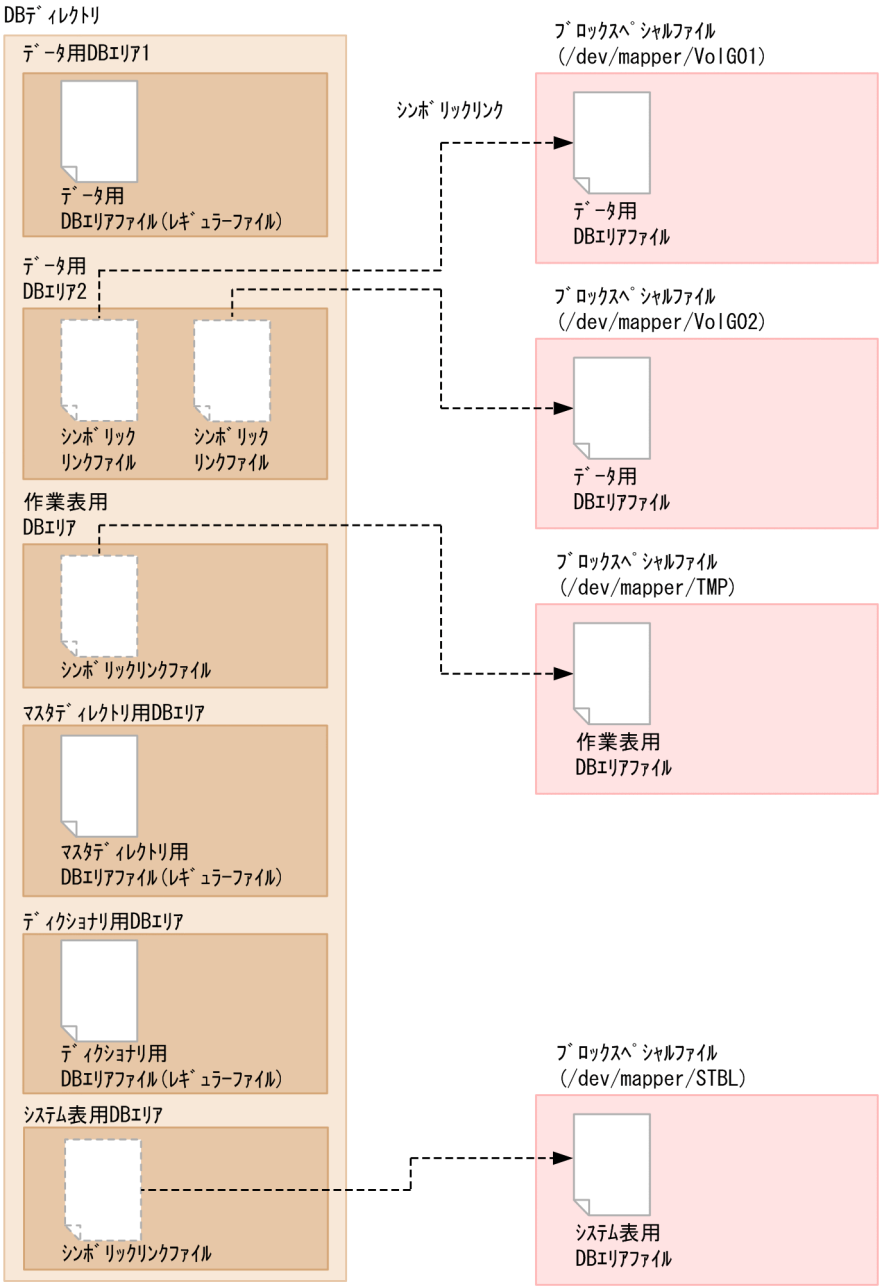
DB ディレクトリの構成については，「付録 A.3　DB ディレクトリの構成」を参照してください。

(4)　DB エリアファイルの構成例

DB エリアファイルを，DB ディレクトリ下以外のブロックスペシャルファイルに作成した場合，DB ディレクトリ下にはファイルの実体へのシンボリックリンクファイルが作成されます。

DB エリアファイルの構成例を次の図に示します。

図 2-19　DB エリアファイルの構成例



(5) ファイルシステムに関する注意事項

- ext3，ext4 または XFS のファイルシステム上に，データベースを構成するファイル（DB エリアファイルおよび DB ディレクトリ内のファイル）を作成してください。
- NFS などのネットワークを経由するファイルシステム上に，データベースを構成するファイル（DB エリアファイルおよび DB ディレクトリ内のファイル）を作成しないでください。

(6) DB エリアファイルとマルチノード機能の関係

マルチノード機能を使用する場合，次の DB エリアファイルには，ブロックスペシャルファイルを割り当てる必要があります。

- データ用 DB エリアファイル
- マスタディレクトリ用 DB エリアファイル
- ディクショナリ用 DB エリアファイル
- システム表用 DB エリアファイル

また，割り当てるブロックスペシャルファイルに対応するディスクは，マルチノード構成の全ノードから参照できる必要があります。

2.4.5 DB エリアの自動増分

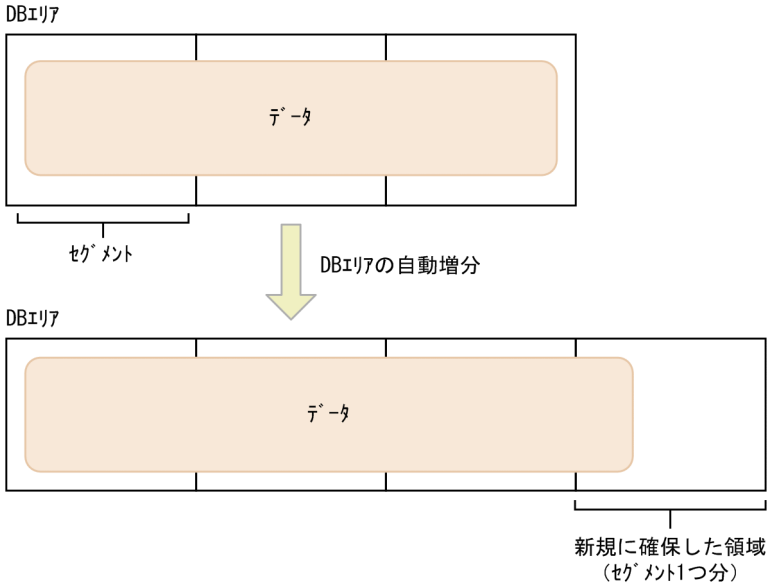
データの追加などによって DB エリアの容量が不足した場合に，HADB が DB エリアの容量を自動的に増やします。これを **DB エリアの自動増分**といいます。

(1) DB エリアの自動増分の単位

DB エリアの自動増分で 1 回に増分する容量は，セグメント 1 つ分です。

DB エリアの自動増分の例を，次の図に示します。

図 2-20　DB エリアの自動増分の例



DB エリアファイルがレギュラーファイルで構成されている場合，自動増分すると DB エリアファイル拡張時の更新処理の性能が低下します。そのため，次のどちらかを実施して，その結果を基に，データベースの初期設定時または DB エリアの追加・拡張時に必要な分だけ初期化しておくことを推奨します。

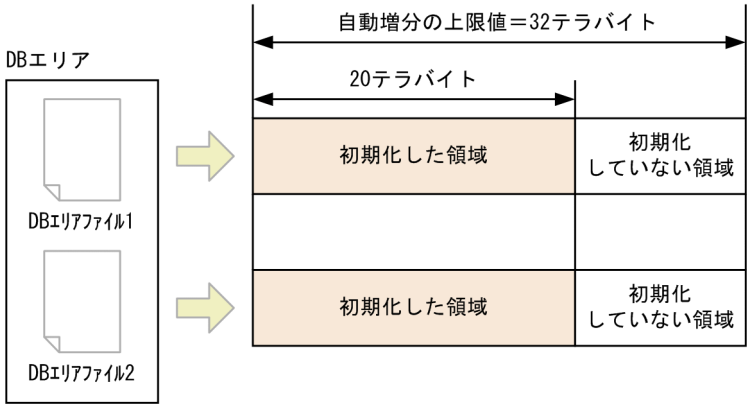
- DB エリアの容量を見積もる
- 小規模のテスト環境に対して`adbdbstatus` コマンドを実行して，DB エリアの容量を確認する

(2)　DB エリアの自動増分の上限値

DB エリアの自動増分では，DB エリアの容量が不足するたびに，自動増分の上限値まで，セグメント単位に増分を繰り返します。

自動増分の上限値は，DB エリアファイルごとに，データベースの初期設定時または DB エリアの追加・拡張時に，初期化した領域を 16 テラバイト単位に切り上げた値になります。自動増分の上限値の例を次の図に示します。

図 2-21　DB エリアの自動増分の上限値の例



［説明］

データベースの初期設定時または DB エリアの追加・拡張時に，2 つの DB エリアファイルを 20 テラバイトずつ初期化した場合，1 つの DB エリアファイルごとに 32 テラバイトが自動増分の上限値になります。そのため，DB エリア全体で 64 テラバイトが上限値になります。

なお，作業表用 DB エリアの場合，作業表用 DB エリアファイルの最大値が自動増分の上限値となります。データベースの初期設定時に初期化した領域を 16 テラバイト単位に切り上げた値ではありません。

2.5　ユーザとスキーマ

ここでは，HADB サーバの運用で使用するユーザの種類について説明します。また，表やインデクスを包括する概念であるスキーマについて説明します。

2.5.1　ユーザの種類（OS ユーザと HADB ユーザ）

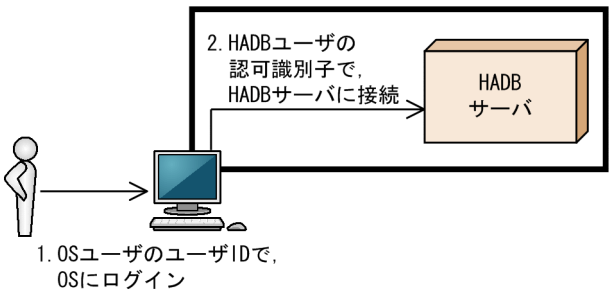
HADB サーバの運用で使用するユーザの種類を次に示します。

- OS ユーザ
- HADB ユーザ

OS ユーザと HADB ユーザは別物であるため，混同しないよう，それぞれ適切に管理する必要があります。

OS ユーザと HADB ユーザの概要を次の図に示します。

図 2-22　OS ユーザと HADB ユーザの概要



(1)　OS ユーザ

OS ユーザとは，OS にログインして，OS の機能を利用できるユーザのことです。OS にログインする場合は，OS ユーザのユーザ ID およびパスワードが必要です。

OS ユーザのユーザ ID およびパスワードでは，HADB サーバには接続できません。HADB サーバに接続する場合は，HADB ユーザのユーザ ID およびパスワードが必要です。

OS ユーザには，**権限**という概念があります。OS に HADB サーバをインストールするためには，管理者権限を持った OS ユーザが必要です。

また，OS には**グループ**という概念があります。OS ユーザが所属するグループによって，実行できる操作が異なります。

HADB サーバの運用で使用する OS ユーザの種類を次に示します。

- **スーパユーザ**
  管理者権限を持った OS ユーザのことです。

スーパユーザは，OS の環境設定，OS ユーザの管理，HADB サーバのインストールなどで使用します。

- **HADB 管理者**

  HADB サーバを管理する専用の OS ユーザのことです。

  HADB 管理者は，HADB サーバをインストールするマシンの OS に対して，スーパユーザが，HADB 管理者として使用する OS ユーザを追加することで作成されます。

  HADB 管理者は，HADB サーバの環境設定，HADB サーバの開始・終了，HADB サーバの管理・運用などで使用します。

  HADB 管理者には，HADB の各種コマンドを実行する権限があります。また，HADB 管理者は，HADB サーバを運用するためのディレクトリ（サーバディレクトリおよび DB ディレクトリ），およびディレクトリ下に作成されるファイルの所有者になります。

- **HADB 管理グループに所属する OS ユーザ**

  HADB 管理者とは別に，HADB サーバを管理する OS ユーザのことです。HADB 管理者と同じ OS グループ（HADB 管理グループ）に所属します。

  HADB サーバをインストールするマシンの OS に対して，スーパユーザが，HADB 管理グループに所属する OS ユーザを追加することで作成されます。

  HADB 管理グループに所属する OS ユーザには，HADB の一部のコマンドを実行する権限があります。また，HADB 管理者が所有するサーバディレクトリおよび DB ディレクトリにアクセスできます。

(2) HADB ユーザ

HADB ユーザとは，HADB サーバに接続して，データのインポート，データの検索・更新，HADB ユーザの管理などを実施できるユーザのことです。

HADB サーバに接続する場合は，HADB ユーザの認可識別子とパスワードを入力する必要があります。認可識別子とは，HADB サーバに接続するときに使用するユーザ ID のことです。

データベースの初期設定時に，最初（1 人目）の HADB ユーザが作成されます。作成した HADB ユーザは，HADB サーバ内で管理されます。また，HADB ユーザを複数作成することもできます。

なお，HADB ユーザが，HADB サーバに接続して各種操作をするためには，権限を持っている必要があります。権限については，「2.7.1　権限の種類」を参照してください。

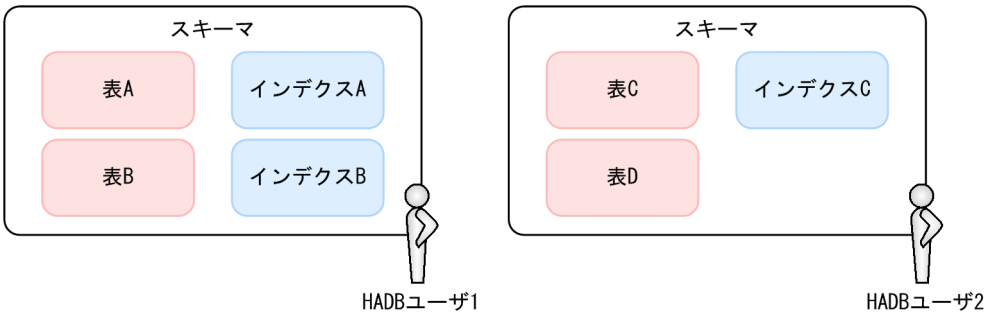
2.5.2　スキーマ

表やインデクスを包括する論理的な概念をスキーマといいます。1HADB ユーザは，1 つのスキーマだけを所有できます。また，スキーマに定義できる要素のことを，スキーマオブジェクトといいます。

HADB では，自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有するスキーマに対してだけ，スキーマオブジェクトを定義できます。ほかの HADB ユーザが所有するスキーマに対して，スキーマオブジェクトを定義できません。

スキーマの概念を次の図に示します。

図 2-23　スキーマの概念



HADB のスキーマオブジェクトを次に示します。

■スキーマオブジェクトの一覧

- 実表
- ビュー表
- B-tree インデクス
- テキストインデクス
- レンジインデクス

HADB では，表やインデクスなどの定義情報を表形式で管理しています。この表を**ディクショナリ表**といいます。また，表やインデクスのコスト情報，チャンク情報なども表形式で管理しています。この表を**システム表**といいます。HADB の初期設定が完了すると，ディクショナリ表およびシステム表を所有するスキーマ（スキーマ名：MASTER）が自動的に作成されます。

スキーマの名前を**スキーマ名**といいます。1HADB ユーザに対して，1 つのスキーマ名が与えられます。スキーマ名を使用することで，HADB のデータベース内の表やインデクスを一意に識別することができます。

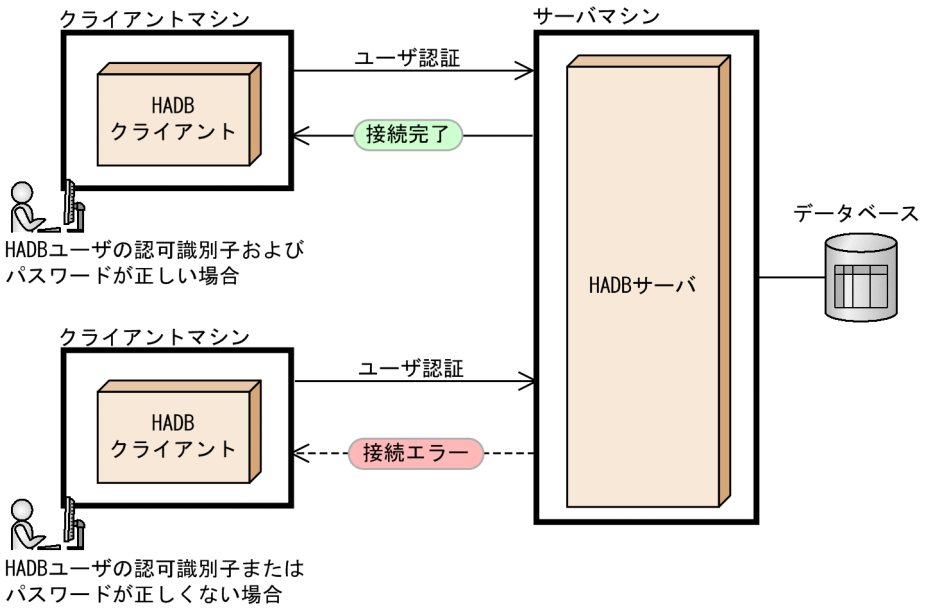
HADB では，HADB サーバにアクセスする HADB ユーザを識別するためのユーザ ID（認可識別子）とスキーマ名が同じになります。

2.6　ユーザ認証

HADB サーバは，HADB ユーザが HADB サーバに接続する際，HADB ユーザの認可識別子およびパスワードを使用して認証を行っています。これを，**ユーザ認証**といいます。

ユーザ認証の概要を次の図に示します。

図 2-24　ユーザ認証の概要



HADB ユーザが正しい認可識別子およびパスワードを入力し，かつCONNECT 権限を持っている場合に，HADB サーバに接続できます。

入力した認可識別子またはパスワードが正しくない場合，またはCONNECT 権限を持っていない場合は，HADB サーバへの接続がエラーとなります。

CONNECT 権限については，「2.7.2　ユーザ権限」の「(2)　CONNECT 権限」で説明します。

2.7　権限

ここでは，HADB サーバの運用で使用する権限について説明します。

2.7.1　権限の種類

HADB ユーザが，HADB サーバを運用するためには，各種の権限が必要になります。

HADB ユーザが，HADB サーバに対して各種操作を実行する際，必要な権限を持っていないと，エラーとなります。そのため，対象の HADB ユーザが実施する運用項目に合わせて，必要な権限だけを付与することで，不正な操作が行われるリスクを減らせます。

HADB ユーザは，複数の権限を持つことができます。権限は，大きく分けて次に示す 4 種類があります。

- **ユーザ権限**
  詳細は「2.7.2　ユーザ権限」を参照してください。
- **スキーマ操作権限**
  詳細は「2.7.3　スキーマ操作権限」を参照してください。
- **監査権限**
  詳細は「2.7.4　監査権限」を参照してください。
- **アクセス権限**
  詳細は「2.7.5　アクセス権限」を参照してください。

> **メモ**
>
> 権限を付与したり，取り消したりするには，定義系 SQL 文のGRANT 文およびREVOKE 文を実行します。

2.7.2　ユーザ権限

ユーザ権限は，HADB ユーザを管理したり，HADB サーバに接続したりするときに必要な権限です。

ユーザ権限には次の権限があります。

- DBA 権限
- CONNECT 権限

(1)　DBA 権限

HADB ユーザの管理およびユーザ権限の付与をする際に必要となるユーザ権限を，**DBA 権限**といいます。

DBA 権限を持つ HADB ユーザは，次に示す操作を実行できます。

- **HADB ユーザの管理**
  - HADB ユーザの作成
  - 作成した HADB ユーザのユーザ情報の変更※
  - 作成した HADB ユーザの削除※
- **ユーザ権限およびスキーマ操作権限の管理**
  - HADB ユーザへのユーザ権限およびスキーマ操作権限の付与
  - HADB ユーザに付与したユーザ権限およびスキーマ操作権限の取り消し※
- **監査権限の付与**
  - HADB ユーザへの監査権限（監査管理権限，監査参照権限）の付与

注※

監査管理権限を持っている HADB ユーザに対しては，次の操作はできません。

- HADB ユーザのユーザ情報の変更
- HADB ユーザの削除
- ユーザ権限およびスキーマ操作権限の取り消し

なお，DBA 権限を持っている HADB ユーザは，ディクショナリ表（SQL_AUDITS を除く），およびシステム表に格納されているすべての情報を参照できます。

(2) CONNECT 権限

HADB ユーザが HADB サーバに接続する際に必要となるユーザ権限を，**CONNECT 権限**といいます。

CONNECT 権限を持っていないと，DBA 権限，スキーマ操作権限，監査権限，またはアクセス権限を持っていても HADB サーバに接続できません。

2.7.3 スキーマ操作権限

スキーマ操作権限は，HADB ユーザのスキーマを管理するときに必要な権限です。

スキーマ操作権限には次の権限があります。

- スキーマ定義権限

(1) スキーマ定義権限

HADB ユーザがスキーマを定義する際に必要となるスキーマ操作権限を，**スキーマ定義権限**といいます。

スキーマ，表およびインデクスを定義・削除する場合は，スキーマ定義権限が必要になります。

2.7.4　監査権限

監査権限とは，次に示す 2 つの権限の総称です。監査証跡機能を使用する場合に，監査権限が必要になります。監査証跡機能については，「2.18　監査証跡機能」を参照してください。

- 監査管理権限
  監査管理権限とは，監査証跡機能の運用を行う際に必要となる権限です。監査管理権限を持っていると，次の操作ができます。
  - adbaudittrail コマンドによる監査証跡機能の操作
  - CREATE AUDIT 文による監査対象の定義
  - DROP AUDIT 文による監査対象定義の削除
  - REVOKE 文による監査権限の取り消し
  - 監査対象定義の情報の参照（SQL_AUDITS の検索）

> **重要**
>
> 同一の HADB ユーザが，監査管理権限とDBA 権限の両方を持つことはできません。

- 監査参照権限
  監査参照権限とは，監査証跡を参照する際に必要となる権限です。監査参照権限を持っていると，次の操作ができます。
  - ADB_AUDITREAD 関数を使用した監査証跡の参照
  - adbconvertaudittrailfile コマンドによる監査証跡ファイルの変換

> **メモ**
>
> ADB_AUDITREAD 関数とは，出力された監査証跡を HADB サーバが検索できる表形式のデータ集合に変換する関数です。SELECT 文にADB_AUDITREAD 関数を指定することで，監査証跡を参照できます。

2.7.5　アクセス権限

スキーマオブジェクトに対するアクセス制御を行う権限（アクセス権限）について説明します。

(1)　アクセス権限とは

アクセス権限とは，次に示すスキーマオブジェクトにアクセスする際に必要となる権限のことです。

- 実表
- ビュー表

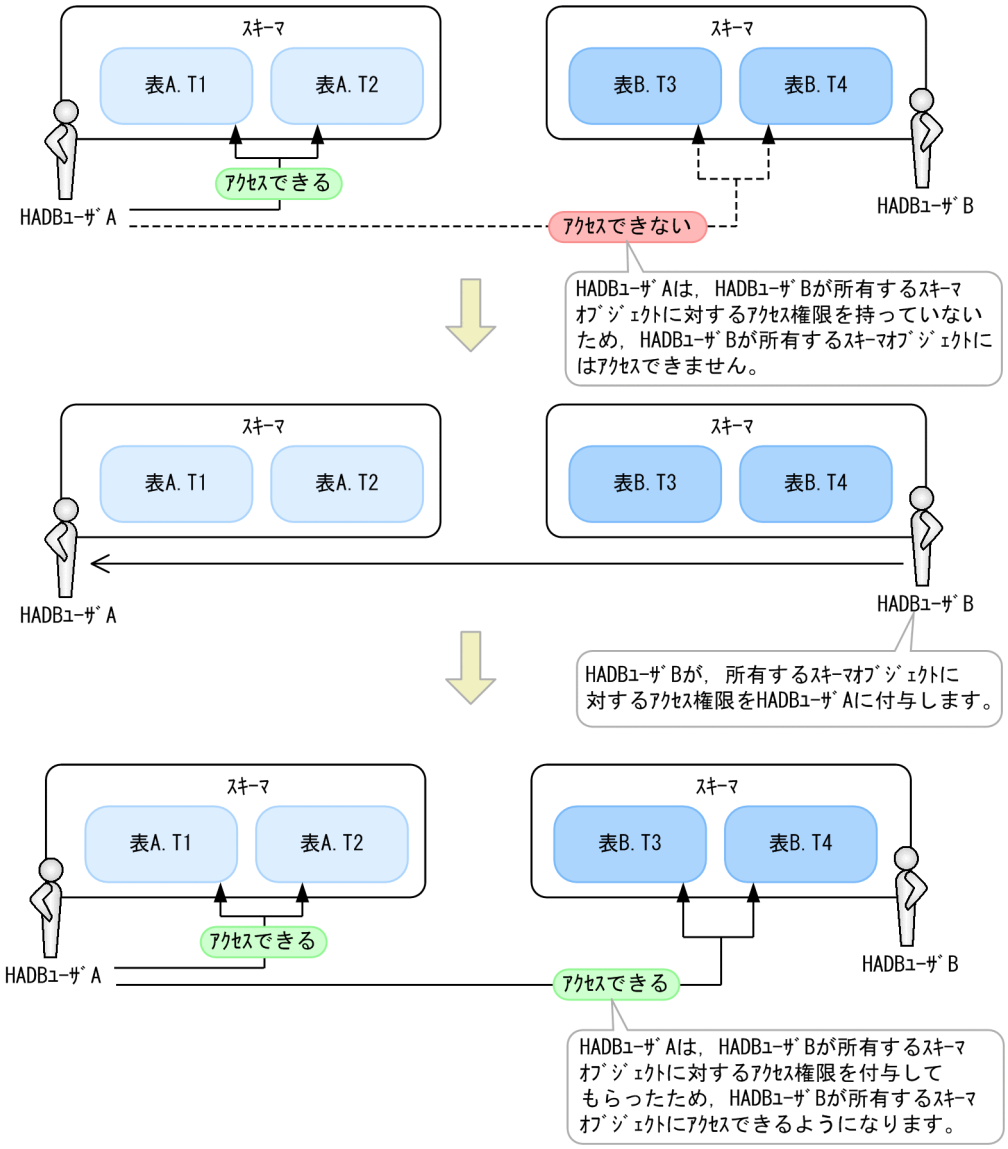
スキーマオブジェクトのデータを SQL 文で検索したり，スキーマオブジェクトをコマンドで操作したりするには，スキーマオブジェクトに対するアクセス権限が必要です。

自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有しているスキーマオブジェクトに対しては，自動的にアクセス権限を持ちます。そのため，自分が所有しているスキーマオブジェクトにはアクセスできます。

ほかの HADB ユーザが所有しているスキーマオブジェクトに対しては，アクセス権限を通常は持っていないため，アクセスできません。ほかの HADB ユーザが所有しているスキーマオブジェクトにアクセスするには，そのスキーマオブジェクトに対するアクセス権限を持っている HADB ユーザから，アクセス権限を付与してもらう必要があります。

ほかの HADB ユーザが所有しているスキーマオブジェクトにアクセスする場合の例を次の図に示します。

図 2-25　ほかの HADB ユーザが所有しているスキーマオブジェクトにアクセスする場合の例



[説明]

HADB ユーザ A は，表A.T1 と表A.T2 を所有しています。HADB ユーザ B は，表B.T3 と表B.T4 を所有しています。

HADB ユーザ A は，自分が所有している表A.T1 と表A.T2 に対するアクセス権限を持っているため，表A.T1 と表A.T2 にアクセスできますが，HADB ユーザ B が所有している表B.T3 と表B.T4 に対するアクセス権限を持っていません。そのため，表B.T3 と表B.T4 にはアクセスできません。HADB ユーザ A が，表B.T3 と表B.T4 にアクセスできるようにするには，表B.T3 と表B.T4 に対するアクセス権限を HADB ユーザ B に付与してもらう必要があります。

メモ

HADB ユーザは，アクセス権限を持っているスキーマオブジェクトに対してだけアクセスできます。

(2) アクセス権限の種類

アクセス権限には，次の表に示す種類があります。持っているアクセス権限の種類によって，スキーマオブジェクトに対して実行できる操作が変わります。

表 2-7 アクセス権限の一覧

項番	アクセス権限の種類	説明	対象のスキーマオブジェクト
1	SELECT 権限	SELECT 権限とは，HADB ユーザが，対象のスキーマオブジェクトのデータを参照するために必要となる権限です。 SELECT 文およびCREATE VIEW 文を実行する場合，そのスキーマオブジェクトに対するSELECT 権限が必要になります。	• 実表 • ビュー表
2	INSERT 権限	INSERT 権限とは，HADB ユーザが，対象のスキーマオブジェクトにデータを挿入するために必要となる権限です。 INSERT 文を実行する場合，そのスキーマオブジェクトに対するINSERT 権限が必要になります。	• 実表 • ビュー表
3	UPDATE 権限	UPDATE 権限とは，HADB ユーザが，対象のスキーマオブジェクトのデータを変更するために必要となる権限です。 UPDATE 文を実行する場合，そのスキーマオブジェクトに対するUPDATE 権限が必要になります。	• 実表 • ビュー表
4	DELETE 権限	DELETE 権限とは，HADB ユーザが，対象のスキーマオブジェクトのデータを削除するために必要となる権限です。 DELETE 文を実行する場合，そのスキーマオブジェクトに対するDELETE 権限が必要になります。	• 実表 • ビュー表
5	TRUNCATE 権限	TRUNCATE 権限とは，HADB ユーザが，対象のスキーマオブジェクトのデータを一括で削除するために必要となる権限です。 TRUNCATE TABLE 文およびPURGE CHUNK 文を実行する場合，そのスキーマオブジェクトに対するTRUNCATE 権限が必要になります。	実表
6	REFERENCES 権限	REFERENCES 権限とは，HADB ユーザが，CREATE TABLE 文で参照制約（外部キー）を定義する際に必要となる権限です。 外部キーを定義する場合，被参照表（主キーを定義している表）に対するREFERENCES 権限が必要になります。	実表
7	IMPORT TABLE 権限	IMPORT TABLE 権限とは，HADB ユーザが，対象のスキーマオブジェクトにデータをインポートするために必要となる権限です。	実表

項番	アクセス権限の種類	説明	対象のスキーマオブジェクト
		adbimport コマンドを実行する場合，そのスキーマオブジェクトに対するIMPORT TABLE 権限が必要になります。	
8	REBUILD INDEX 権限	REBUILD INDEX 権限とは，HADB ユーザが，対象のスキーマオブジェクトのインデクスを再作成するために必要となる権限です。 adbidxrebuild コマンドを実行する場合，そのスキーマオブジェクトに対するREBUILD INDEX 権限が必要になります。	実表
9	GET COSTINFO 権限	GET COSTINFO 権限とは，HADB ユーザが，対象のスキーマオブジェクトとそのスキーマオブジェクトのインデクスからコスト情報を収集したり，収集したコスト情報を削除したりするために必要となる権限です。 adbgetcst コマンドを実行する場合，そのスキーマオブジェクトに対するGET COSTINFO 権限が必要になります。	実表
10	EXPORT TABLE 権限	EXPORT TABLE 権限とは，HADB ユーザが，対象のスキーマオブジェクトのデータをエクスポートするために必要となる権限です。 adbexport コマンドを実行する場合，そのスキーマオブジェクトに対するEXPORT TABLE 権限が必要になります。	• 実表 • ビュー表
11	MERGE CHUNK 権限	MERGE CHUNK 権限とは，HADB ユーザが，対象のスキーマオブジェクトに対して作成された複数のチャンクを，1 つのチャンクにマージするために必要となる権限です。 adbmergechunk コマンドを実行する場合，そのスキーマオブジェクトに対するMERGE CHUNK 権限が必要になります。	実表
12	CHANGE CHUNK COMMENT 権限	CHANGE CHUNK COMMENT 権限とは，HADB ユーザが，対象のスキーマオブジェクトに定義されたチャンクのコメントを設定・変更・削除するために必要となる権限です。 adbchgchunkcomment コマンドを実行する場合，そのスキーマオブジェクトに対するCHANGE CHUNK COMMENT 権限が必要になります。	実表
13	CHANGE CHUNK STATUS 権限	CHANGE CHUNK STATUS 権限とは，HADB ユーザが，対象のスキーマオブジェクトに定義されたチャンクの状態を変更するために必要となる権限です。 adbchgchunkstatus コマンドを実行する場合，そのスキーマオブジェクトに対するCHANGE CHUNK STATUS 権限が必要になります。	実表

項番	アクセス権限の種類	説明	対象のスキーマオブジェクト
14	ARCHIVE CHUNK 権限	ARCHIVE CHUNK 権限とは，HADB ユーザが，チャンクをアーカイブ状態にする際に必要となる権限です。 adbarchivechunk コマンドを実行する場合，対象のスキーマオブジェクトに対するARCHIVE CHUNK 権限が必要です。	実表
15	UNARCHIVE CHUNK 権限	UNARCHIVE CHUNK 権限とは，HADB ユーザが，チャンクのアーカイブ状態を解除する際に必要となる権限です。 adbunarchivechunk コマンドを実行する場合，対象のスキーマオブジェクトに対するUNARCHIVE CHUNK 権限が必要です。	実表

メモ

- 実表の所有者（実表を定義した HADB ユーザ）は，所有している実表に対して，上記の表に示すすべての種類のアクセス権限を持っています。
- ビュー表の所有者（ビュー表を定義した HADB ユーザ）は，所有しているビュー表に対して，上記の表に示すすべての種類のアクセス権限を持っているわけではありません。どの種類のアクセス権限を持っているかは，ビュー表の所有者が基表に対して持っているアクセス権限によって決まります。詳細については，「2.7.6　ビュー表に対するアクセス権限」を参照してください。

(3)　アクセス権限の付与

スキーマオブジェクトに対するアクセス権限をほかの HADB ユーザに付与するには，そのスキーマオブジェクトに対するアクセス権限の付与権が必要になります。アクセス権限の付与権とは，スキーマオブジェクトに対するアクセス権限をほかの HADB ユーザに付与できる権限のことです。

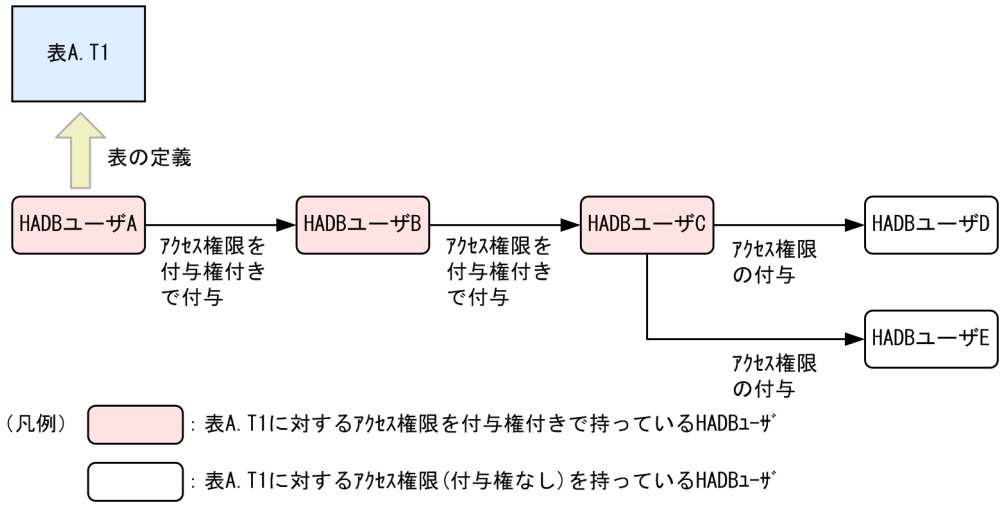
実表の所有者は，所有している実表に対するアクセス権限を付与権付きで持っています。ビュー表の所有者は，所有しているビュー表に対するアクセス権限を持っていますが，付与権を持っているかどうかは，基表に対して付与権付きのアクセス権限を持っているかどうかによって決まります。詳細については，「2.7.6　ビュー表に対するアクセス権限」を参照してください。

重要

アクセス権限を付与する際，アクセス権限だけを付与するか，アクセス権限と一緒にその付与権も付与するかを選択できます。

スキーマオブジェクトに対するアクセス権限の付与例を次の図に示します。

図 2-26　スキーマオブジェクトに対するアクセス権限の付与例



[説明]

- HADB ユーザ A

  スキーマオブジェクト（表A.T1）の所有者です。表A.T1 に対するアクセス権限を付与権付きで持っています。

- HADB ユーザ B

  表A.T1 に対するアクセス権限を付与権付きで，HADB ユーザ A から付与された HADB ユーザです。表A.T1 にアクセスでき，かつ表A.T1 に対するアクセス権限をほかの HADB ユーザに付与できます。

- HADB ユーザ C

  表A.T1 に対するアクセス権限を付与権付きで，HADB ユーザ B から付与された HADB ユーザです。表A.T1 にアクセスでき，かつ表A.T1 に対するアクセス権限をほかの HADB ユーザに付与できます。

- HADB ユーザ D

  表A.T1 に対するアクセス権限（付与権なし）を，HADB ユーザ C から付与された HADB ユーザです。表A.T1 にアクセスできます。HADB ユーザ D は，表A.T1 に対するアクセス権限を付与権付きで持っていないため，表A.T1 に対するアクセス権限をほかの HADB ユーザに付与できません。

- HADB ユーザ E

  表A.T1 に対するアクセス権限（付与権なし）を，HADB ユーザ C から付与された HADB ユーザです。表A.T1 にアクセスできます。HADB ユーザ E は，表A.T1 に対するアクセス権限を付与権付きで持っていないため，表A.T1 に対するアクセス権限をほかの HADB ユーザに付与できません。

ほかの HADB ユーザにアクセス権限を付与するには，GRANT 文を実行します。アクセス権限を付与権付きで付与する場合は，WITH GRANT OPTION オプションを指定してGRANT 文を実行します。

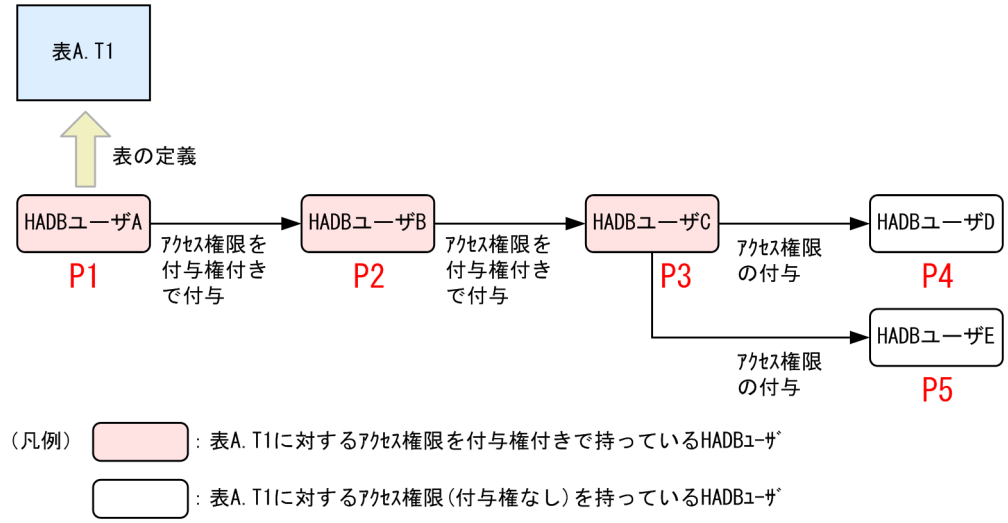
重要

上記の例の場合，HADB ユーザ B～HADB ユーザ E が持っている，表A.T1 に対するアクセス権限は依存権限となります。

■依存権限とは

ほかの HADB ユーザから付与されたアクセス権限のことを依存権限といいます。

（例）



- HADB ユーザ A が持っている表A.T1 に対するアクセス権限をP1 とします。
- HADB ユーザ B が持っている表A.T1 に対するアクセス権限をP2 とします。
- HADB ユーザ C が持っている表A.T1 に対するアクセス権限をP3 とします。
- HADB ユーザ D が持っている表A.T1 に対するアクセス権限をP4 とします。
- HADB ユーザ E が持っている表A.T1 に対するアクセス権限をP5 とします。

この場合，アクセス権限P1 の依存権限は，アクセス権限P2～P5 になります。

アクセス権限P2 の依存権限は，アクセス権限P3～P5 になります。

アクセス権限P3 の依存権限は，アクセス権限P4，P5 になります。

アクセス権限P4 とP5 には，依存権限はありません。

依存権限は，自分に対してアクセス権限を付与した HADB ユーザ（その延長上の HADB ユーザも含む）が持っているアクセス権限が取り消された場合に影響を受けます。例えば，アクセス権限P2 が取り消された場合，依存権限であるアクセス権限P3～P5 も取り消されます。アクセス権限P3 が取り消された場合，依存権限であるアクセス権限P4，P5 も取り消されます。

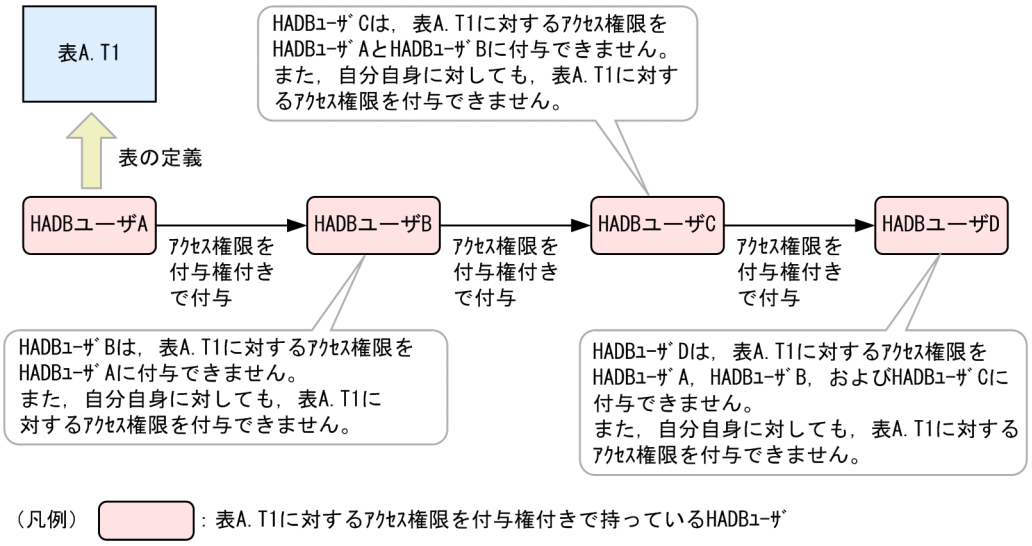
■アクセス権限を付与できない HADB ユーザ

アクセス権限の付与権を持っている場合でも，次に示す HADB ユーザにはアクセス権限を付与できません。

- 付与権付きのアクセス権限を自分に付与した HADB ユーザ

- 上記の HADB ユーザに，対象のアクセス権限を付与権付きで付与している延長上の HADB ユーザ
- 自分自身（付与されたアクセス権限を自分に対して付与することはできません）

（例）



(4) アクセス権限の取り消し

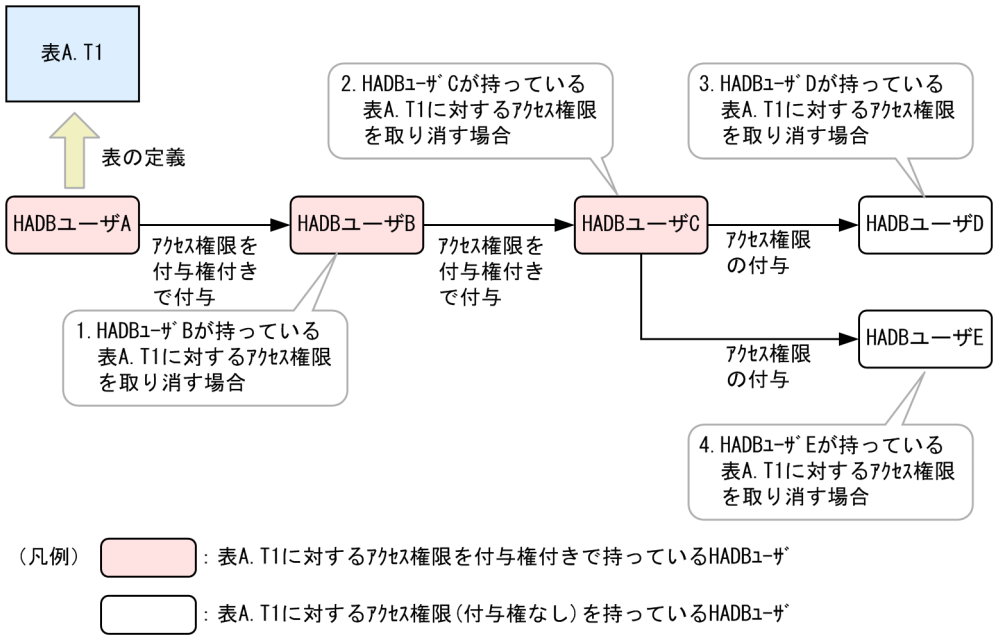
自分が付与したアクセス権限を取り消すことができます。ほかの HADB ユーザが付与したアクセス権限を取り消すことはできません。

重要

- 取り消したアクセス権限に依存権限がある場合，依存権限も一緒に取り消されます。
- 付与権付きのアクセス権限を取り消すと，アクセス権限と一緒に付与権も取り消されます。

スキーマオブジェクトに対するアクセス権限の取り消し例を次の図に示します。

図 2-27　スキーマオブジェクトに対するアクセス権限の取り消し例



[説明]

1. HADB ユーザ B が持っている表A.T1 に対するアクセス権限を取り消す場合

   このアクセス権限を取り消せるのは，HADB ユーザ A だけです。

   HADB ユーザ B が持っている表A.T1 に対するアクセス権限を取り消すと，その依存権限である HADB ユーザ C，HADB ユーザ D，および HADB ユーザ E が持っている，表A.T1 に対するアクセス権限も取り消されます。

2. HADB ユーザ C が持っている表A.T1 に対するアクセス権限を取り消す場合

   このアクセス権限を取り消せるのは，HADB ユーザ B だけです。

   HADB ユーザ C が持っている表A.T1 に対するアクセス権限を取り消すと，その依存権限である HADB ユーザ D と HADB ユーザ E が持っている，表A.T1 に対するアクセス権限も取り消されます。

3. HADB ユーザ D が持っている表A.T1 に対するアクセス権限を取り消す場合

   このアクセス権限を取り消せるのは，HADB ユーザ C だけです。HADB ユーザ D が持っている表A.T1 に対するアクセス権限には依存権限はありません。

4. HADB ユーザ E が持っている表A.T1 に対するアクセス権限を取り消す場合

   このアクセス権限を取り消せるのは，HADB ユーザ C だけです。HADB ユーザ E が持っている表A.T1 に対するアクセス権限には依存権限はありません。

付与したアクセス権限を取り消すには，REVOKE 文を実行します。

メモ

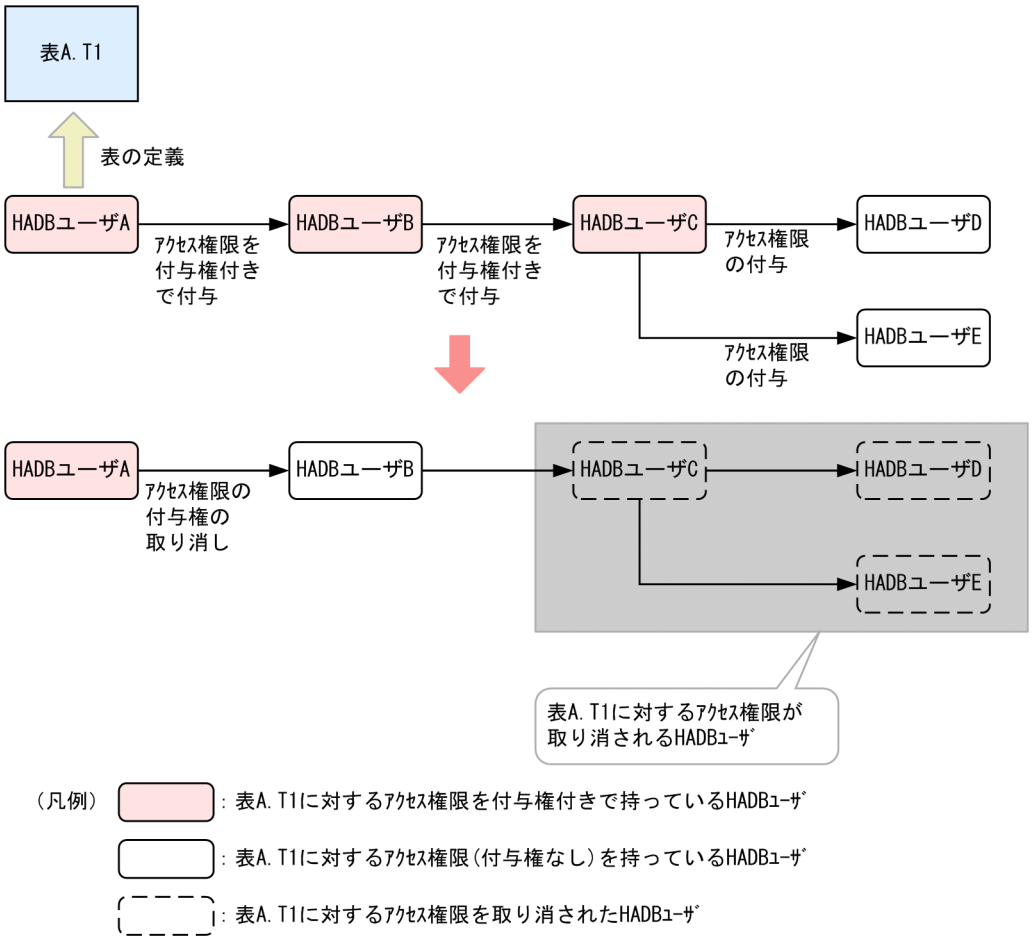
- スキーマオブジェクトの所有者（上記の例の HADB ユーザ A）が持っているスキーマオブジェクト（上記の例の表A.T1）に対するアクセス権限は取り消すことはできません。

- 上記の例の場合，HADB ユーザ A は，自分が付与したアクセス権限ではないため，HADB ユーザ C が持っている表A.T1 に対するアクセス権限を取り消すことはできません。ただし，HADB ユーザ A は，HADB ユーザ B に付与した表A.T1 に対するアクセス権限，またはその付与権を取り消すことによって，HADB ユーザ C が持っている表A.T1 に対するアクセス権限も一緒に取り消すことができます。HADB ユーザ C が持っている表A.T1 に対するアクセス権限は，HADB ユーザ B が持っている表A.T1 に対するアクセス権限の依存権限であるため，一緒に取り消すことができます。

■アクセス権限の付与権だけを取り消す場合

アクセス権限の付与権だけを取り消すことができます。付与権だけを取り消した場合，その付与権を使用して付与したアクセス権限とその依存権限が取り消されます。

（例）



［説明］

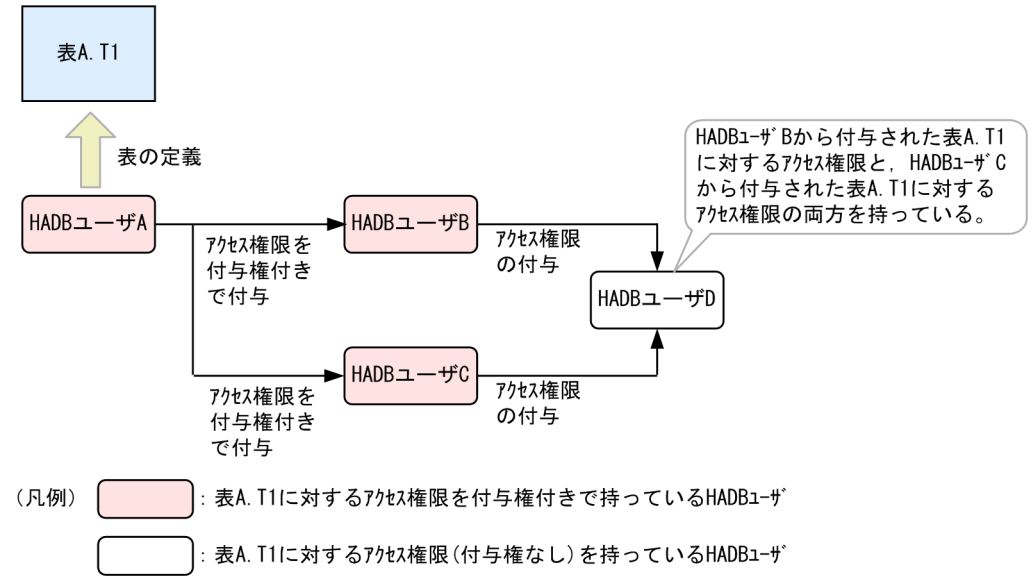
HADB ユーザ B が持っている表A.T1 に対するアクセス権限の付与権を，HADB ユーザ A が取り消した場合，HADB ユーザ C が持っている表A.T1 に対するアクセス権限が取り消されます。

また，HADB ユーザ D と HADB ユーザ E が持っている，表A.T1 に対するアクセス権限も取り消されます（HADB ユーザ C が持っている表A.T1 に対するアクセス権限の依存権限であるため）。

■同一のスキーマオブジェクトに対するアクセス権限を複数の HADB ユーザから付与されている場合

次に示す例のように，HADB ユーザ B と HADB ユーザ C が，表A.T1 に対するアクセス権限を HADB ユーザ D に付与した場合，HADB ユーザ D は，HADB ユーザ B から付与された表A.T1 に対するアクセス権限と，HADB ユーザ C から付与された表A.T1 に対するアクセス権限をそれぞれ持ちます。

（例）



例えば，HADB ユーザ D が持っている，HADB ユーザ B が付与した表A.T1 に対するアクセス権限を取り消しても，HADB ユーザ C が付与した表A.T1 に対するアクセス権限は取り消されません（自分が付与したアクセス権限しか取り消せないため）。HADB ユーザ D は，表A.T1 に対するアクセス権限をまだ持っているため，表A.T1 に対してアクセスできます。

(5) 複数のスキーマオブジェクトにアクセスする場合に必要となるアクセス権限

HADB ユーザが SQL 文を実行した場合に，SQL 文の処理過程で複数のスキーマオブジェクトにアクセスするときは，それらすべてのスキーマオブジェクトに対するアクセス権限が必要となります。

対象のスキーマオブジェクトが実表の場合を例に，次で説明します。

複数の実表にアクセスする SQL 文の実行例

```
SELECT * FROM "ADBUSER02"."T1", "ADBUSER02"."T2", "ADBUSER02"."T3"
```

[説明]

HADB ユーザ（ADBUSER01）が，ほかの HADB ユーザ（ADBUSER02）が所有する実表T1，T2 およびT3 を検索する場合は，検索対象すべての実表に対してアクセス権限（SELECT 権限）が必要になります。もし，HADB ユーザ（ADBUSER01）が実表T3 のアクセス権限（SELECT 権限）を持っていない場合は，上記のSELECT 文は実行できません。

これは，SELECT 権限だけでなく，INSERT 権限，UPDATE 権限，DELETE 権限など，ほかのアクセス権限でも同様です。

2.7.6　ビュー表に対するアクセス権限

ビュー表に対するアクセス権限について説明します。

■定義したビュー表に対するアクセス権限

ビュー表を定義したHADBユーザが基表に対して持っているアクセス権限によって，定義したビュー表に対するアクセス権限が決まります。例えば，SELECT権限とINSERT権限を持っている表を基表としたビュー表を定義した場合，ビュー表を定義したHADBユーザは，そのビュー表に対するSELECT権限とINSERT権限を持ちます。

なお，基表が複数ある場合は，すべての基表に対して持っている共通のアクセス権限が，ビュー表に対するアクセス権限となります。

(例)

HADBユーザAは，次の表に対して次のアクセス権限を持っているものとします。

- 表X.T1：SELECT権限，INSERT権限，DELETE権限
- 表X.T2：SELECT権限，INSERT権限，UPDATE権限
- 表X.T3：SELECT権限，UPDATE権限，DELETE権限

表X.T1と表X.T2を基表としたビュー表A.V1をHADBユーザAが定義した場合，HADBユーザAは，ビュー表A.V1に対するSELECT権限とINSERT権限を持ちます。

表X.T1，表X.T2，および表X.T3を基表としたビュー表A.V2をHADBユーザAが定義した場合，HADBユーザAは，ビュー表A.V2に対するSELECT権限だけを持ちます。

> **メモ**
>
> 定義したビュー表に対して，例えばUPDATE権限を持ちたい場合は，すべての基表に対してUPDATE権限を持っている必要があります。

定義したビュー表に対するアクセス権限の決定規則の詳細については，マニュアル『HADB　SQLリファレンス』の『CREATE VIEW文の指定形式および規則』の『規則』を参照してください。

■ビュー表に対するアクセス権限の付与

ビュー表に対するアクセス権限をほかのHADBユーザに付与するには，基表に対するアクセス権限を付与権付きで持っている必要があります。

(例)

HADBユーザAは，次の表に対して次のアクセス権限を持っているものとします。

- 表X.T1：SELECT権限（付与権付き），INSERT権限（付与権付き），UPDATE権限
- 表X.T2：SELECT権限（付与権付き），INSERT権限，DELETE権限（付与権付き）

表X.T1を基表としたビュー表A.V1をHADBユーザAが定義した場合，HADBユーザAは，ビュー表A.V1に対するSELECT権限（付与権付き），INSERT権限（付与権付き），およびUPDATE権限を持ちます。SELECT権限とINSERT権限については付与権があるため，HADBユーザAは，ビュー表A.V1に対するSELECT権限とINSERT権限をほかのHADBユーザに付与できます。

表X.T1 と表X.T2 を基表としたビュー表A.V2 を HADB ユーザ A が定義した場合，HADB ユーザ A は，ビュー表A.V2 に対するSELECT 権限（付与権付き）とINSERT 権限を持ちます。SELECT 権限については付与権があるため，HADB ユーザ A は，ビュー表A.V2 に対するSELECT 権限をほかの HADB ユーザに付与できます。

■ビュー表に対するアクセス権限の伝搬

基表に対するアクセス権限が新たに付与された場合，その基表に依存するビュー表に対するアクセス権限も新たに付与されます。これを**アクセス権限の伝搬**といいます。

（例）

HADB ユーザ A が，表X.T1 を基表としたビュー表A.V1 と，ビュー表A.V1 を基表としたビュー表A.V2 を定義しているとします。この場合に，表X.T1 に対するINSERT 権限が HADB ユーザ A に新たに付与されると，ビュー表A.V1，A.V2 に対するINSERT 権限も HADB ユーザ A に新たに付与されます。

また，基表に対するアクセス権限が取り消された場合も，アクセス権限の伝搬が発生します。

（例）

HADB ユーザ A が，表X.T1 を基表としたビュー表A.V1 と，ビュー表A.V1 を基表としたビュー表A.V2 を定義しているとします。この場合に，HADB ユーザ A が持っている表X.T1 に対するINSERT 権限が取り消されると，HADB ユーザ A が持っているビュー表A.V1，A.V2 に対するINSERT 権限も取り消されます。

アクセス権限の伝搬規則の詳細については，マニュアル『HADB　SQL リファレンス』の『アクセス権限の付与』の『規則』を参照してください。

■表に対するSELECT 権限の取り消しによるビュー表への影響

ビュー表の基表に対するSELECT 権限が取り消された場合，その基表に依存するすべてのビュー表が無効化されます。

（例）

HADB ユーザ A が，表X.T1 を基表としたビュー表A.V1 と，ビュー表A.V1 を基表としたビュー表A.V2 を定義しているとします。この場合に，HADB ユーザ A が持っている表X.T1 に対するSELECT 権限が取り消されると，表X.T1 に依存するビュー表A.V1，A.V2 が無効化されます。

また，付与権付きのSELECT 権限が取り消された場合，またはSELECT 権限の付与権が取り消された場合，その付与権を使用して付与されたSELECT 権限も取り消されます。それに伴いビュー表が無効化されることがあります。

（例）

HADB ユーザ A が持っている表X.T1 に対するSELECT 権限の付与権が取り消されたとします。HADB ユーザ A が，表X.T1 に対するSELECT 権限を HADB ユーザ B に付与していた場合，HADB ユーザ B に付与した表X.T1 に対するSELECT 権限が取り消されます。HADB ユーザ B が，表X.T1 を基表としたビュー表B.V1 を定義している場合，ビュー表B.V1 が無効化されます。

2.7.7　HADB ユーザが参照できるディクショナリ表とシステム表の範囲

HADB ユーザは，CONNECT 権限を持っていれば，ディクショナリ表およびシステム表中に格納されているスキーマオブジェクトに関する情報などを参照することができます（更新などはできません）。ただし，HADB ユーザが持っている権限によって，ディクショナリ表およびシステム表中の参照できる情報の範囲が異なります。HADB ユーザが参照できる主な情報を，権限ごとに次に示します。

DBA 権限を持っている HADB ユーザ

ディクショナリ表（SQL_AUDITS を除く）およびシステム表に格納されているすべての情報を参照できます。

DBA 権限を持っていない HADB ユーザ

自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が定義した定義情報を参照できます。また，ほかの HADB ユーザが定義した定義情報であっても，次の情報は参照できます。

- 表，インデクスなど，自分がアクセス権限を持っているスキーマオブジェクトに関する情報や，そのスキーマに関する情報
- PUBLIC 指定でアクセス権限を許可されたスキーマオブジェクトの情報や，そのスキーマに関する情報

監査管理権限を持っている HADB ユーザ

監査証跡機能で監査の対象とする情報（ディクショナリ表のSQL_AUDITS に格納されている監査対象定義の情報）を参照できます。また，アクセス権限を持っていなくても，ほかの HADB ユーザが定義したスキーマオブジェクトに関する情報の一部や，そのスキーマに関する情報を参照できます。

HADB ユーザが参照できるディクショナリ表およびシステム表の情報の範囲の詳細については，次に示す個所を参照してください。

- 「付録 B.1　ディクショナリ表の概要」の「(3)　HADB ユーザが参照できるディクショナリ表の範囲」
- 「付録 C.1　システム表の概要」の「(3)　HADB ユーザが参照できるシステム表の範囲」

2.7.8　HADB ユーザに権限を付与する例

ここでは，HADB ユーザに権限を付与する流れを，例を使って説明します。

(1)　HADB ユーザの役割と運用例

HADB ユーザに権限を付与する前提となる運用例を次の図に示します。この例では，HADB ユーザを 4 種類作成します。

図 2-28　各 HADB ユーザの役割と運用例

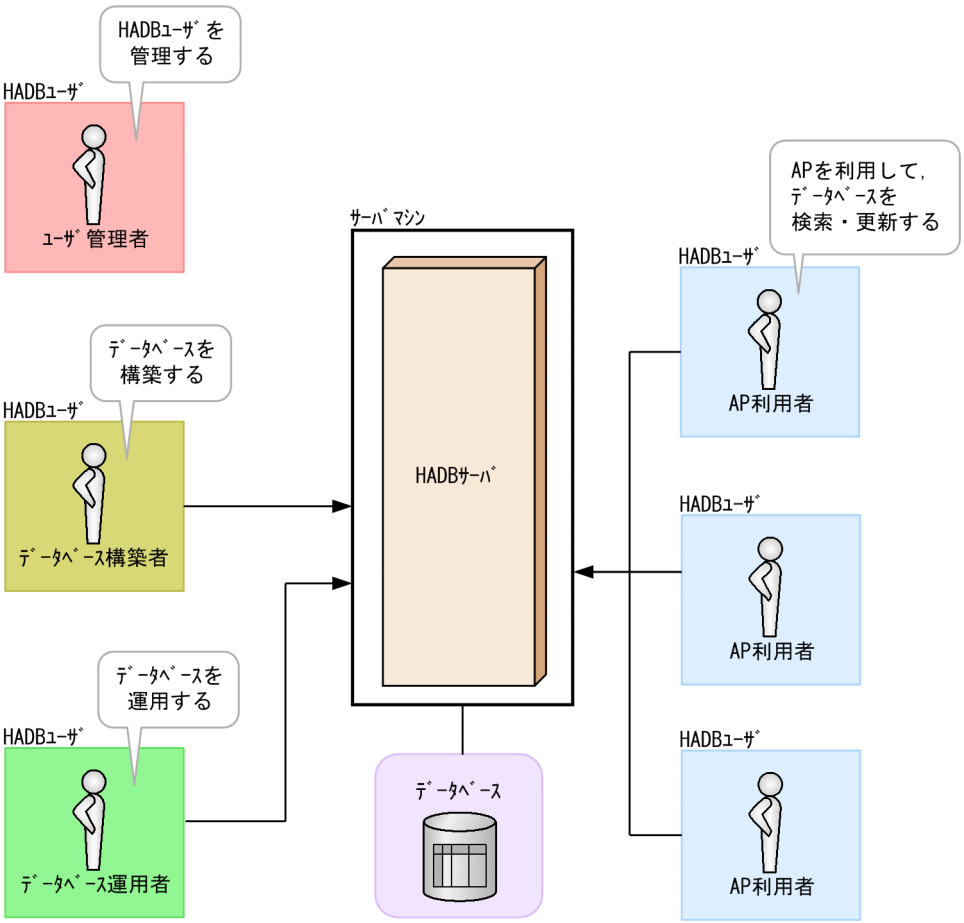


表 2-8　作成する HADB ユーザの役割分担

項番	HADB ユーザの役割	説明
1	ユーザ管理者	HADB ユーザの管理を担当する HADB ユーザです。ユーザ管理者は，次に示す役割の HADB ユーザを作成します。 • データベース構築者 • データベース運用者 • AP 利用者 また，作成した HADB ユーザに，CONNECT 権限やスキーマ定義権限を付与します。 この例では，データベースの初期設定時に作成される最初（1 人目）の HADB ユーザを，ユーザ管理者とします。
2	データベース構築者	データベースの構築を担当する HADB ユーザです。データベース構築者は，次に示す作業を行います。 • スキーマ，表およびインデクスの定義 • データのインポート • データベース運用者と AP 利用者に，作成した表へのアクセス権限の付与
3	データベース運用者	データベースの運用を担当する HADB ユーザです。データベース構築者が作成した表に対して，次に示す作業を行います。 • データのインポート • インデクスの再作成

項番	HADB ユーザの役割	説明
		• コスト情報の収集 • SELECT 文によるデータの検索
4	AP 利用者	AP を利用して，データベースを検索・更新する HADB ユーザです。データベース構築者が作成した表に対して，次に示す作業を行います。 • SELECT 文によるデータの検索 • INSERT 文，UPDATE 文およびDELETE 文によるデータの追加・更新・削除

メモ

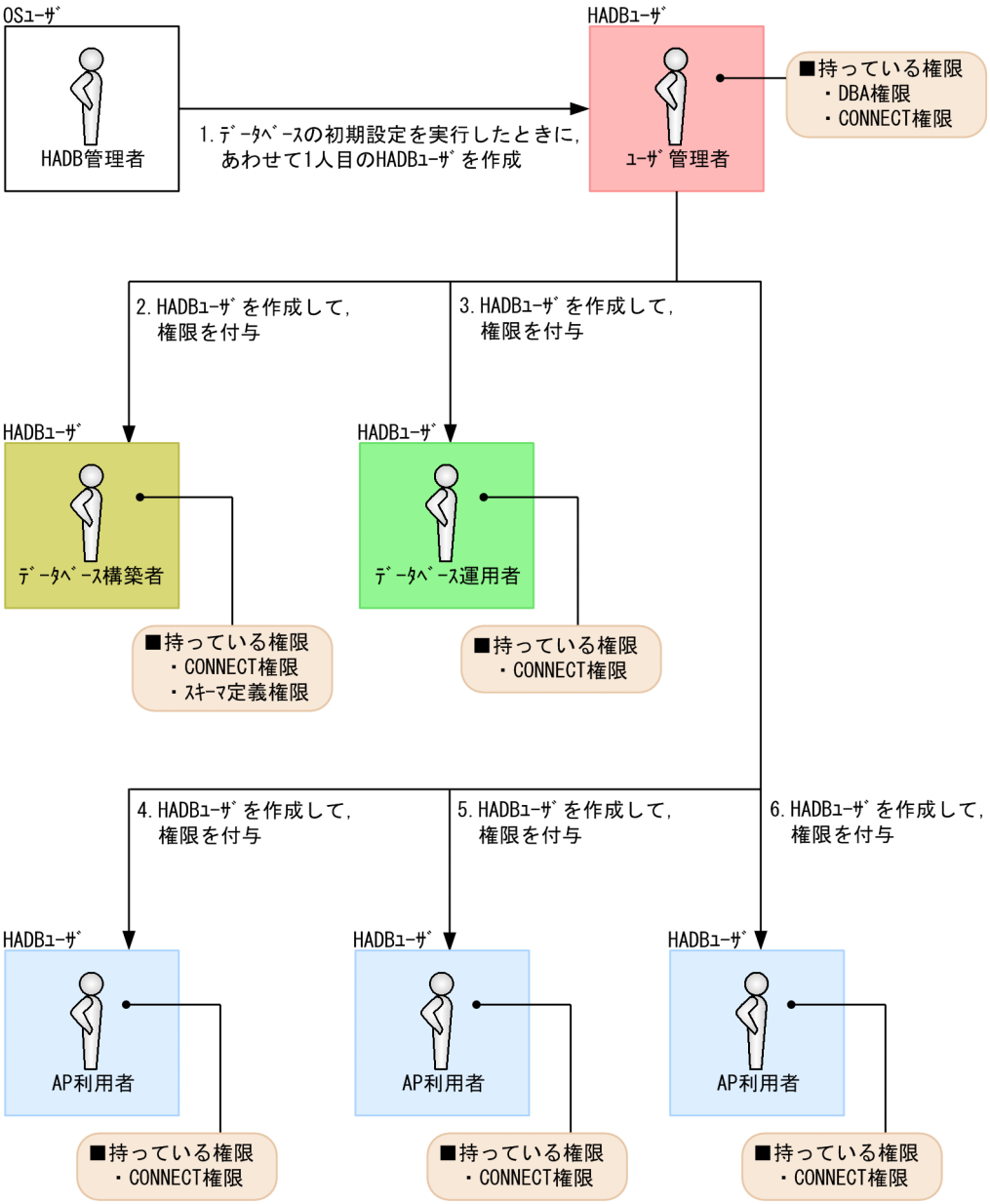
ここで説明する例には，監査証跡機能を使用する際に必要となる監査人は登場しません。監査証跡機能，および監査人の役割については，「2.18　監査証跡機能」を参照してください。

(2)　HADB ユーザに権限を付与する流れ

「(1)　HADB ユーザの役割と運用例」で示した HADB ユーザの役割を基に，HADB ユーザに権限を付与する流れについて説明します。

まず，ユーザ管理者が，各 HADB ユーザを作成して権限を付与する流れについて，次の図に示します。

図 2-29　権限の付与の流れ（その 1）



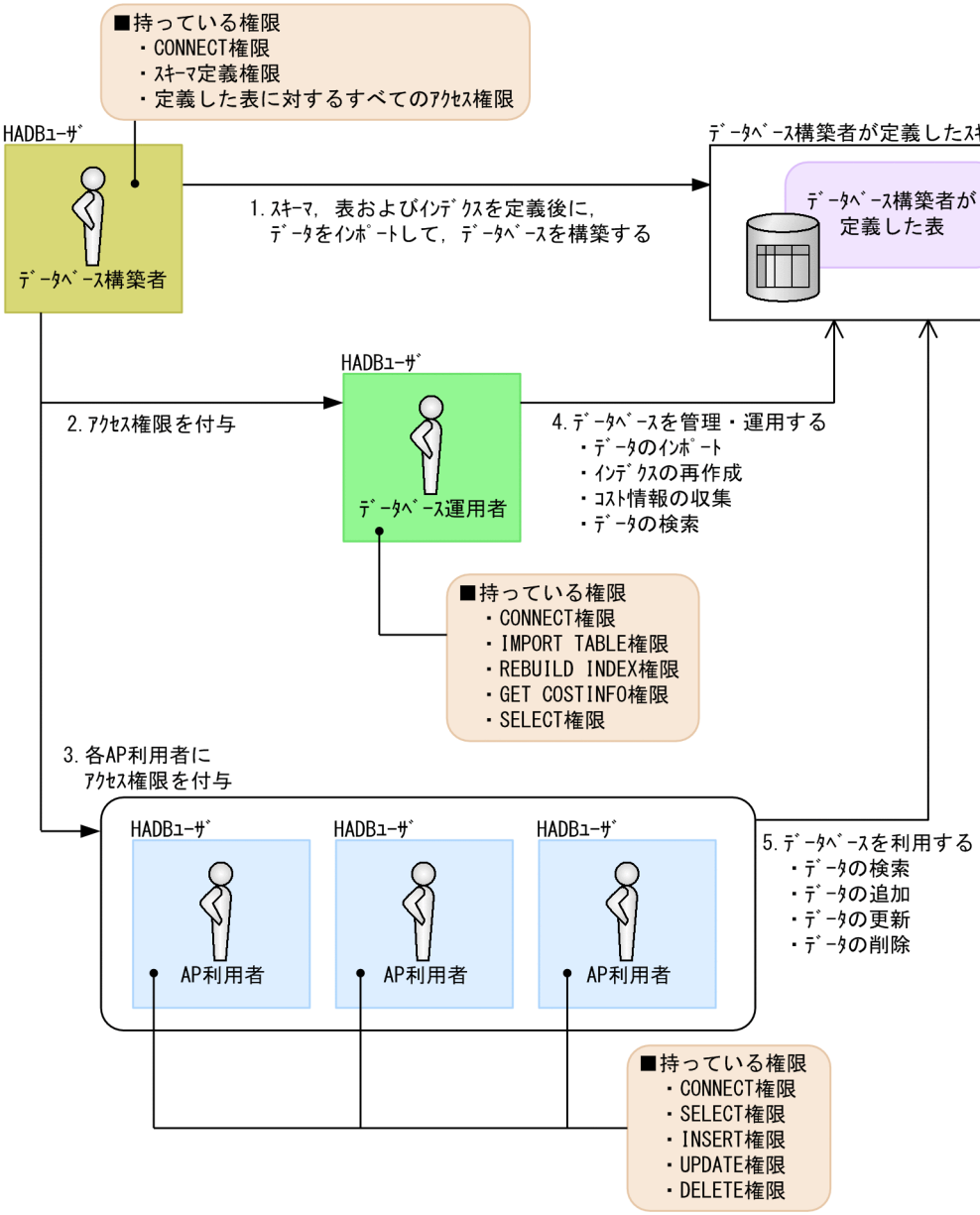
[説明]

OS ユーザである HADB 管理者が，データベースの初期設定を実行します。その際，最初（1 人目）の HADB ユーザとして，ユーザ管理者が作成されます。

そのあとで，ユーザ管理者が，各 HADB ユーザを作成して，CONNECT 権限やスキーマ定義権限を付与します。

次に，データベース構築者が定義した表へのアクセス権限を，データベース運用者と AP 利用者に付与する流れについて，次の図に示します。

図 2-30　権限の付与の流れ（その 2）



［説明］

- データベース構築者が，表を定義します。そのあとで，データベース運用者と AP 利用者に対して，表へのアクセス権限をそれぞれ付与します。データベース運用者と AP 利用者は，データベース構築者が定義した表に対してアクセスできるようになり，持っているアクセス権限に従って作業を行うことができます。
- データベース運用者が複数人いる場合，『2. アクセス権限を付与』で，データベース運用者の代表者に付与権付きのアクセス権限を付与すれば，データベース運用者の代表者が，ほかのデータベース運用者にアクセス権限を付与できるようになります。
- 『3. 各 AP 利用者にアクセス権限を付与』で，AP 利用者の代表者に付与権付きのアクセス権限を付与すれば，AP 利用者の代表者が，ほかの AP 利用者にアクセス権限を付与できるようになります。

メモ

- HADB ユーザに，権限を付与したり，付与した権限を取り消したりする場合は，定義系 SQL のGRANT 文およびREVOKE 文を実行します。
- データベースの初期設定時に，最初（1 人目）の HADB ユーザが作成されます。最初（1 人目）の HADB ユーザには，DBA 権限およびCONNECT 権限が自動的に付与されます。

2.8　データベースのアクセス処理方式

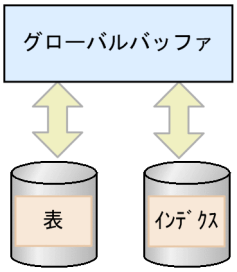
ここでは，データベースのアクセス処理方式として，データベースの入出力処理に使用するグローバルバッファ，データベースの更新処理方式，およびデータベースの検索処理方式（非順序実行方式）について説明します。

2.8.1　グローバルバッファ

HADB は，データベースの入出力処理にグローバルバッファを使用します。

グローバルバッファとは，ディスク上の DB エリアに格納されているデータを入出力するための領域のことです。グローバルバッファは共有メモリ上に確保されます。グローバルバッファの概念を次の図に示します。

図 2-31　グローバルバッファの概念



表およびインデクスを格納するデータ用 DB エリアには，必ずグローバルバッファを割り当てます。1 つのグローバルバッファには，1 つ以上の DB エリアを割り当てることができます。

2.8.2　データベースの更新処理方式

HADB では，データベースの更新処理方式として追記型の更新処理方式を採用しています。

追記型の更新処理方式では，行を削除した場合，削除対象の行は無効になりますが，ディスクからは削除されません。また，行を更新した場合，更新対象の行は無効になり，更新後の行が新たな行として追加されます。更新前の無効になった行は，ディスクからは削除されません。

ただし，操作系 SQL の`TRUNCATE TABLE` 文を実行して行を削除した場合は，セグメントが解放され，削除対象の行はディスクから削除されます。

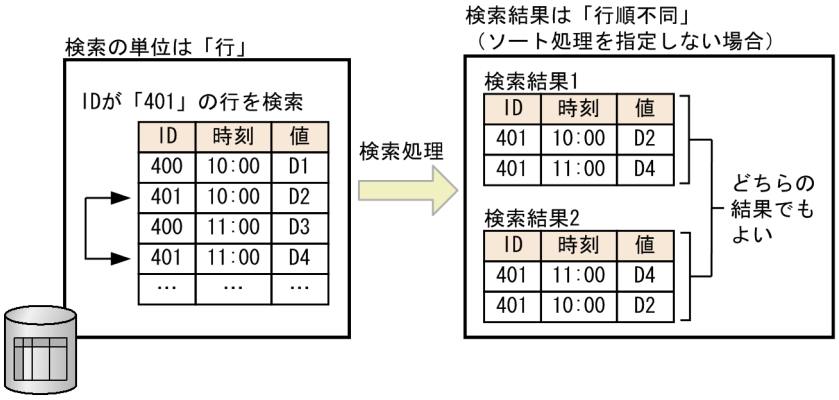
2.8.3　データベースの検索処理方式（非順序実行方式）

HADB には，データベースの検索処理方式として非順序実行方式が実装されています。非順序実行方式は，大規模なデータベースを高速に検索するために考案された処理方式です。

(1) 非順序実行方式の考え方

リレーショナルデータベースでは，検索処理は行の集合演算であり，ソート処理などを除いて，DBMS が返却する検索結果の順序に規定はありません。非順序実行方式は，この点に着目して考案されています。非順序実行方式の考え方を次の図に示します。

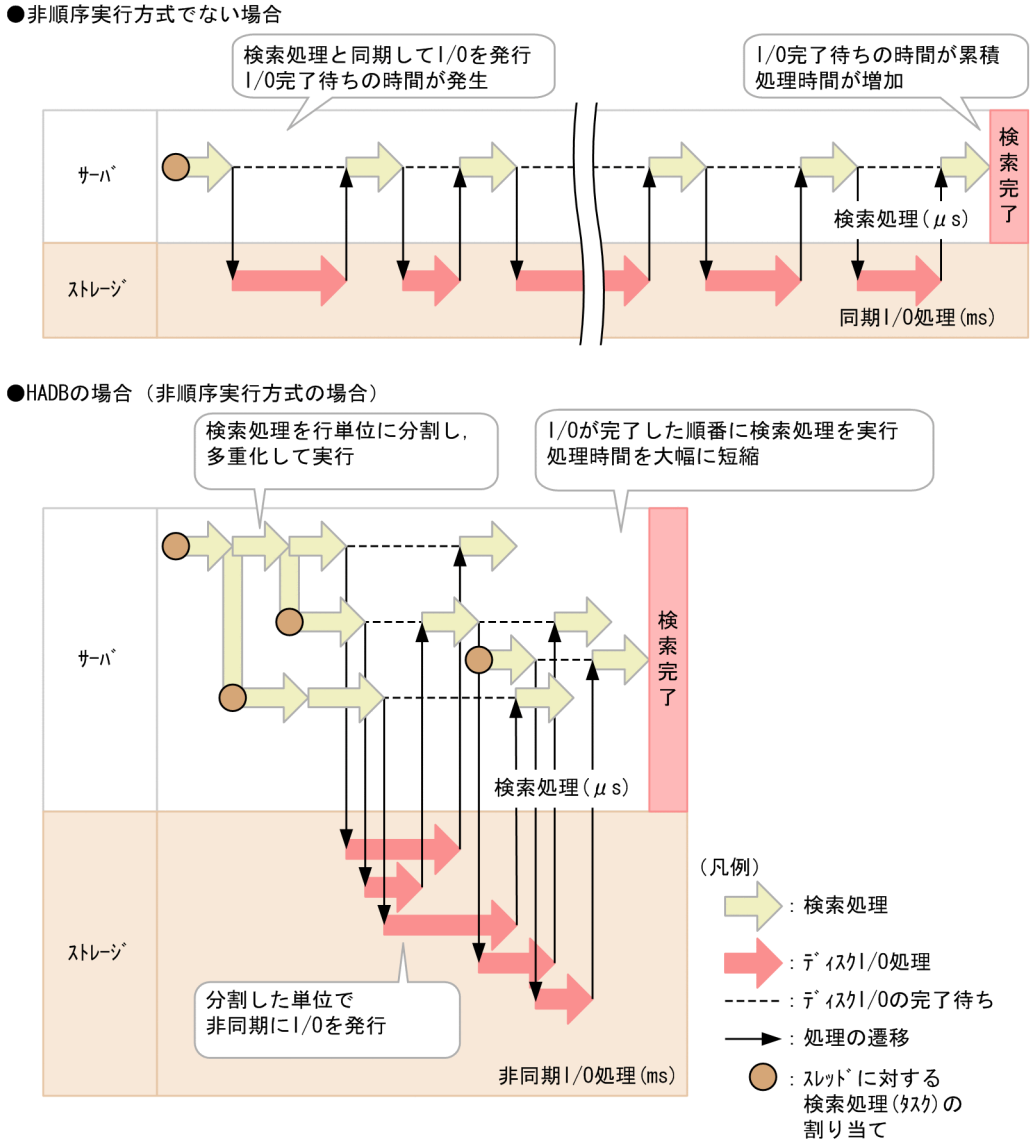
図 2-32　非順序実行方式の考え方



(2) 非順序実行方式による検索処理

非順序実行方式では，ハードウェア資源を有効に活用することで，大規模なデータベースに対しても高速な検索ができます。非順序実行方式による検索処理を，非順序実行方式でない場合と比較して，次の図に示します。

図 2-33　非順序実行方式による検索処理（非順序実行方式でない場合との比較）



非順序実行方式でない場合，1 つのスレッドに 1 つの検索処理だけが割り当てられます。また，発行した I/O が完了してから新たな I/O が発行されます。そのため，ストレージの I/O 性能の上限を下回る回数でしか I/O を発行できません。結果として，データベースのデータ量が大規模になると，I/O 完了待ちの時間が長くなってしまい，検索処理に時間が掛かります。

非順序実行方式では，検索処理が行単位に分割され，各スレッドに割り当てられます。割り当てられた検索処理はスレッドごとに独立して実行されます。また，各スレッドの処理は I/O の進捗に応じて非順序に実行されます。つまり，各スレッドは I/O を発行したあと，その I/O の完了を待たないで，別のスレッドに処理を切り替えながら並列に動作します（非同期 I/O 処理）。これによって，高速な検索を実現します。

(3)　非順序実行方式で使用するスレッド

HADB では，依存関係のない処理を次に示す複数のスレッドに割り当て，I/O および処理を並列化しています。非順序実行方式で使用するスレッドを次に示します。

- リアルスレッド

  OS（カーネルとライブラリ）によって管理されるスレッドです。CPU のコア数が 2 以上の場合，複数の処理を並列に実行できます。

- 擬似スレッド

  HADB によって管理されるスレッドです。1 つのリアルスレッド内で複数の擬似スレッドを生成できますが，同時に 2 つ以上の擬似スレッドが動作することはできません。HADB では，複数の擬似スレッドを使うことによって処理や I/O を並列に実行します。

(4) 非順序実行方式が適用されない場合

次のどれかの場合は，非順序実行方式は適用されないで，SQL が実行されます。

- 検索（SELECT 文）以外の SQL を実行する場合
- SQL 文実行時に使用するリアルスレッド数（adb_sql_exe_max_rthd_num オペランド）に0 を指定した場合
- 擬似スレッドが生成できない場合（1 リアルスレッドで生成できる擬似スレッド数（adb_sys_uthd_num オペランド）に0 を指定した場合）

(5) 非順序実行方式と待ち状態の関係

SQL 文の実行時に使用する処理リアルスレッド数が確保できない場合に発生する待ち状態と，非順序実行方式の関係について説明します。

(a) SQL 文を実行する場合

SQL 文の実行時に使用する処理リアルスレッド数を確保できない場合，その SQL 文は待ち状態になります。待ち状態になった SQL 文は，必要な処理リアルスレッド数が確保されたあとに，実行されます。

なお，先発の SQL 文またはコマンドが待ち状態になった場合，後発の SQL 文はすべて待ち状態になります（後発の SQL 文に必要な処理リアルスレッド数が確保できる場合でも，待ち状態になります）。この場合，先発の SQL 文またはコマンドの待ち状態が解放されたあとに，後発の SQL 文の待ち状態が解放されます。

■クライアントグループ機能を適用しているとき

グループに所属している HADB クライアントが実行する SQL 文が待ち状態になった場合，同じグループ内のほかの HADB クライアントが実行する SQL 文は，すべて待ち状態になります。この場合，先発の SQL 文の待ち状態が解放されたあとに，後発の SQL 文の待ち状態が解放されます。

なお，あるグループに所属している HADB クライアントが実行する SQL 文が待ち状態になった場合でも，ほかのグループに所属している HADB クライアントが実行する SQL 文は影響を受けません。

SQL 文の実行時に使用する処理リアルスレッドについては，「7.2.2 性能に関するオペランド（set 形式）」の adb_sql_exe_max_rthd_num オペランドを参照してください。

メモ

対象の SQL 文が待ち状態かどうかを確認する方法については，「10.8.2　処理リアルスレッドの確保待ちの発生状況を確認する方法」を参照してください。

(b)　コマンドを実行する場合

adbimport コマンドなど，一部のコマンドの実行時には，処理リアルスレッドが使用されます。ただし，使用する処理リアルスレッド数を確保できない場合，そのコマンドは待ち状態になります。該当するコマンドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドの『対象のコマンド』を参照してください。

なお，先発のコマンドまたは SQL 文が待ち状態になった場合，後発のコマンドはすべて待ち状態になります（後発のコマンドに必要な処理リアルスレッド数が確保できる場合でも，待ち状態になります）。この場合，先発のコマンドまたは SQL 文の待ち状態が解放されたあとに，後発のコマンドの待ち状態が解放されます。

■クライアントグループ機能を適用しているとき

グループに所属しているコマンドが待ち状態になった場合，同じグループ内のほかのコマンドを実行しても，すべて待ち状態になります。この場合，先発のコマンドの待ち状態が解放されたあとに，後発のコマンドの待ち状態が解放されます。

コマンド実行時に使用する処理リアルスレッドについては，マニュアル『HADB　コマンドリファレンス』の各コマンドを参照してください。

メモ

対象のコマンドが待ち状態かどうかを確認する方法については，「10.8.2　処理リアルスレッドの確保待ちの発生状況を確認する方法」を参照してください。

2.9　トランザクション制御

ここでは，HADB のトランザクション制御について説明します。

2.9.1　トランザクションの開始と終了

SQL 文の前処理を実行したとき，または文ハンドルを確保したときに，HADB のトランザクションが開始します。また，同期点（コミットまたはロールバック）を設定したとき，または HADB サーバから切り離したときに，HADB のトランザクションが終了します。

2.9.2　HADB がサポートしているトランザクション隔離性水準

HADB では，次に示すトランザクション隔離性水準をサポートしています。

- READ COMMITTED
- REPEATABLE READ

(1)　READ COMMITTED

SQL 文を実行した時点（検索の場合はカーソルオープン時点）でコミット済みのデータが SQL 文の処理対象になります。そのため，トランザクション内で同じSELECT 文を複数回実行した場合，ほかのトランザクションによるデータ更新によって，SELECT 文の実行結果が異なることがあります。

(2)　REPEATABLE READ

REPEATABLE READ のトランザクション内で同じ探索条件となる SQL 文を繰り返し実行した場合，その SQL 文の処理対象となるデータは，他トランザクションに関係なく常に同じものとなります。

なお，SQL 文の処理対象となるデータは，次に示す表に従って変わります。

- 「表 2-9　データが格納されているチャンクと SQL 文の処理対象となるデータの関係」
- 「表 2-10　チャンク内のデータを更新したトランザクションと SQL 文の処理対象となるデータの関係」

表 2-9　データが格納されているチャンクと SQL 文の処理対象となるデータの関係

項番	データが格納されているチャンク	SQL 文の処理対象となるデータ
1	• 実表の定義時に作成されたチャンク • TRUNCATE TABLE 文実行時に作成されたチャンク • -d オプションを指定したadbimport コマンド実行時に作成されたチャンク • 自トランザクションの実行前に，バックグラウンドインポート機能を使用したadbimport コマンド実行時に作成されたチャンク • adbmergechunk コマンド実行時に作成されたチャンク（マージ先チャンク）	「表 2-10　チャンク内のデータを更新したトランザクションと SQL 文の処理対象となるデータの関係」を参照してください。

項番	データが格納されているチャンク	SQL 文の処理対象となるデータ
	• adbchgchunkstatus コマンドで状態を変更されたチャンク • 自トランザクションの実行中に，バックグラウンドインポート機能を使用したadbimport コマンド実行時に作成されたチャンクで，アーカイブ状態のチャンク	
2	自トランザクションの実行中に，バックグラウンドインポート機能を使用したadbimport コマンド実行時に作成されたチャンクで，アーカイブ状態ではないチャンク	なし※

注※

自トランザクションの実行中に，次に示す 2 つのコマンドが実行された場合，SQL 文の処理対象となるデータは，「表 2-9　データが格納されているチャンクと SQL 文の処理対象となるデータの関係」および「表 2-10　チャンク内のデータを更新したトランザクションと SQL 文の処理対象となるデータの関係」に従って変わります。

- バックグラウンドインポート機能を使用したadbimport コマンド
- adbimport コマンド実行時に作成されたチャンクを含む，複数のチャンクをマージするadbmergechunk コマンド

例を次の図に示します。

図 2-34　コマンドの実行に伴う自トランザクションの処理対象となるデータの変化

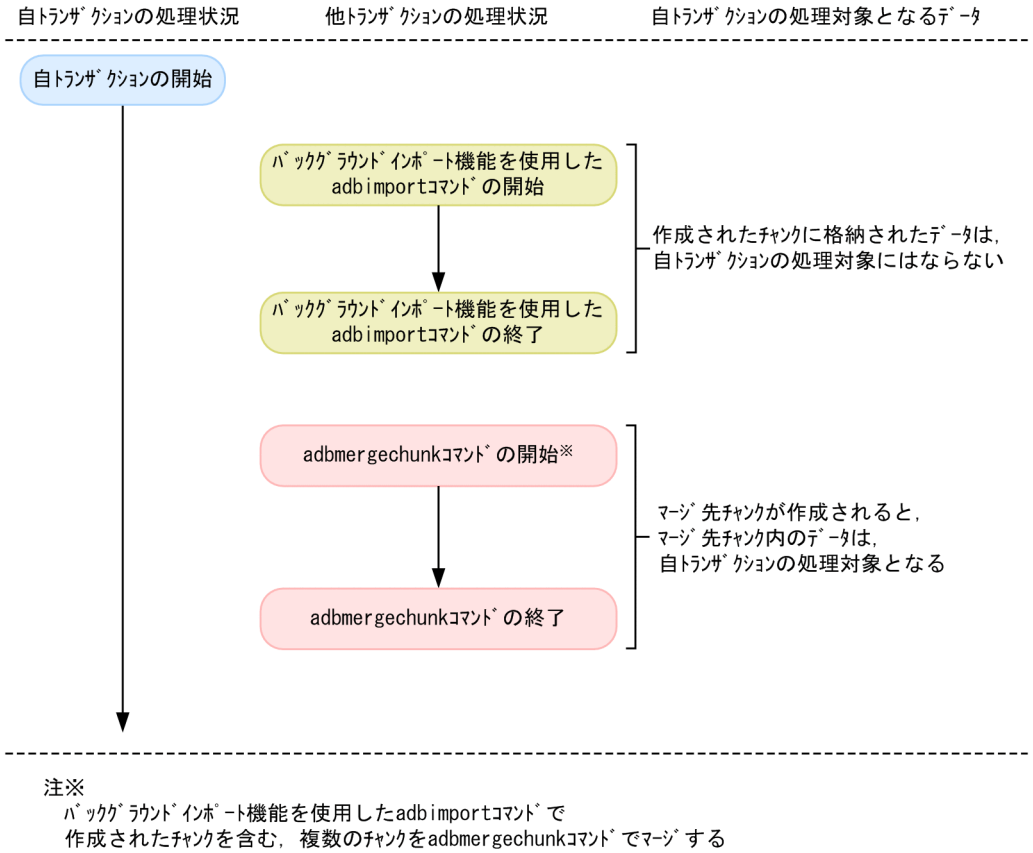


表 2-10　チャンク内のデータを更新したトランザクションと SQL 文の処理対象となるデータの関係

項番	チャンク内のデータを更新したトランザクション		SQL 文の処理対象となるデータ
1	自トランザクション		すべてのデータ
2	他トランザクション	SQL 文による更新 • INSERT 文 • UPDATE 文 • DELETE 文	自トランザクション開始時にコミット済みのデータ
3		adbimport コマンドによるデータの追加	adbimport コマンドの処理が正常終了したときのデータ
4		PURGE CHUNK 文による更新	PURGE CHUNK 文が正常終了したときのデータ
5		TRUNCATE TABLE 文による更新	TRUNCATE TABLE 文が正常終了したときのデータ
6		adbchgchunkstatus コマンドによるチャンクの状態変更	adbchgchunkstatus コマンドが正常終了したときのデータ

重要

チャンク ID を指定した検索（adbsql コマンドのサブコマンド#GETDATA など）を実行した場合は，トランザクション隔離性水準がREPEATABLE READ にならないことがあります。詳細については，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」の「(4) adbmergechunk コマンド実行中に検索対象となるチャンク」を参照してください。

2.9.3　トランザクションの同時実行制御

複数のトランザクションが同時に同じ行にアクセスする際の処理方式について説明します。

(1)　トランザクション A とトランザクション B が同じ行を参照した場合

トランザクション A とトランザクション B は，お互いに影響はありません。

(2)　トランザクション A が更新した行をトランザクション B が参照した場合

トランザクション B が参照する行の内容は，トランザクション A の状態とトランザクション B のトランザクション隔離性水準によって決まります。トランザクション B が参照する行の内容を次の表に示します。

表 2-11　トランザクション B が参照する行の内容

項番	トランザクション A の状態	トランザクション B のトランザクション隔離性水準	
		READ COMMITTED	REPEATABLE READ
1	実行中	トランザクション A が更新する前の行の内容	トランザクション A が更新する前の行の内容
2	コミット済み	トランザクション A が更新したあとの行の内容	

(3)　トランザクション A が更新した行をトランザクション B が更新しようとした場合

実行中のトランザクション A が更新した行を，トランザクション B が更新しようとした場合，トランザクション B をロールバックします。

2.9.4　トランザクションの同期点処理

トランザクションがコミットされると，そのトランザクションによる更新内容が確定されます。

トランザクションがロールバックされると，そのトランザクションによる更新内容が無効になり，データベースはトランザクション開始前の状態に戻ります。

2.9.5　トランザクションアクセスモード

トランザクションアクセスモードは，トランザクションで実行する SQL 文を制御する際に使用します。HADB では，次に示すトランザクションアクセスモードをサポートしています。

- 読み書き可能モード
- 読み取り専用モード

> **メモ**
>
> トランザクションアクセスモードは，クライアント定義adb_clt_trn_access_mode オペランド，およびコネクション属性で指定できます。クライアント定義およびコネクション属性については，マニュアル『HADB　AP 開発ガイド』を参照してください。

(1)　読み書き可能モード

トランザクションアクセスモードが読み書き可能モードの場合に，トランザクションを開始すると，そのトランザクションは**読み書き可能トランザクション**（read/write transaction）となります。

読み書き可能トランザクションでは，すべての SQL 文が実行できます。

(2) 読み取り専用モード

トランザクションアクセスモードが読み取り専用モードの場合に，トランザクションを開始すると，そのトランザクションは**読み取り専用トランザクション**（read-only transaction）となります。

読み取り専用トランザクションでは，定義系 SQL 文および更新系 SQL 文は実行できません。

読み取り専用トランザクションで実行できる SQL 文を次に示します。

■読み取り専用トランザクションで実行できる SQL 文

- SELECT 文

2.10　排他制御

ここでは，HADB の排他制御の仕組みについて説明します。

2.10.1　排他制御モード

排他制御モードの種類や同時実行の関係などについて説明します。

(1)　排他制御モードの種類

HADB サーバの排他制御モードには，複数の種類があります。

- 占有モード（EXCLUSIVE）（EX）
  占有モードは，排他資源を確保したトランザクションが，排他資源を占有するモードです。占有モードで排他資源が確保されている場合，ほかのトランザクションは，排他資源に対して参照処理と更新処理を実行できません。
- 保護更新モード（PROTECTED UPDATE）（PU）
  保護更新モードは，排他資源を確保したトランザクションが更新処理を行うモードです。保護更新モードで排他資源が確保されている場合，ほかのトランザクションは，排他資源に対して参照処理だけを実行できます。
- 共有更新モード（SHARED UPDATE）（SU）
  共有更新モードは，排他資源を確保したトランザクションが更新処理を行うモードです。共有更新モードで排他資源が確保されている場合，ほかのトランザクションは，排他資源に対して参照処理と更新処理を実行できます。
- 共有参照モード（SHARED RETRIEVE）（SR）
  共有参照モードは，排他資源を確保したトランザクションが参照処理を行うモードです。共有参照モードで排他資源が確保されている場合，ほかのトランザクションは，排他資源に対して参照処理と更新処理を実行できます。

(2)　排他制御モードの同時実行の関係性（排他資源の確保の競合が発生したときの動作）

複数のトランザクションが同一の排他資源を確保しようとした場合，排他資源の確保の競合が発生します。競合が発生した場合，先発のトランザクションの排他制御モードと，後発のトランザクションの排他制御モードの組み合わせによって，トランザクションの同時実行可否が決まります。

同一の排他資源に対する，トランザクションと排他制御モードの同時実行の関係性について，次の表に示します。

表 2-12　トランザクションと排他制御モードの同時実行の関係性

先発のトランザクションの排他制御モード	後発のトランザクションの排他制御モード			
	SR	SU	PU	EX
共有参照モード（SR）	○	○	○	×
共有更新モード（SU）	○	○	×	×
保護更新モード（PU）	○	×	×	×
占有モード（EX）	×	×	×	×

（凡例）

○：先発のトランザクションと後発のトランザクションを同時に実行できます。

×：後発のトランザクションは実行できません。

上記の表で×の組み合わせの場合，後発のトランザクションは排他資源を確保できません。そのため，後発のトランザクションの動作は，次のどちらかになります。

- 後発のトランザクションがエラーになり，ロールバックする
- 後発のトランザクションが，排他資源を確保できるまで待つ※

詳細については，「2.10.4　確保される排他資源と排他制御モード」を参照してください。

注※

排他資源を確保できるまで待つ場合に，同一トランザクション内で確保中の排他資源に対して，次に示す事象が発生したときはエラーになります。

- 確保中の排他資源に対する排他制御モードの遷移が発生したが，ほかのトランザクションと排他の競合が発生したため，排他制御モードの遷移ができないとき

なお，adbarchivechunk コマンドによるチャンクのアーカイブ処理の延長で，次に示す場合はエラーになりません。排他制御モードの遷移が完了するまで処理を待ちます。

- 前処理表に対する排他を取得している際に，共有参照モード（SR）から占有モード（EX）に排他制御モードを遷移する場合
- 処理対象表に対する排他を取得している際に，保護更新モード（PU）から占有モード（EX）に排他制御モードを遷移する場合

(3)　排他制御モードの遷移

1 つのトランザクション内で同じ排他資源に対して，異なる種類の排他制御モードで続けて排他を取得した場合，次に示す表のとおりに排他制御モードが遷移します。

表 2-13　排他制御モードの遷移

最初に確保した排他制御モード	次に確保した排他制御モード			
	SR	SU	PU	EX
共有参照モード（SR）	－	SU に遷移する	PU に遷移する	EX に遷移する※1，※2
共有更新モード（SU）	－	－	PU に遷移する	EX に遷移する※2
保護更新モード（PU）	－	－	－	EX に遷移する※2
占有モード（EX）	SR に遷移する※3	－	PU に遷移する※4	－

（凡例）

－：遷移しません。

注※1

ALTER VIEW 文の処理の延長で，次に示す排他資源を確保する場合に，占有モード（EX）に遷移します。それ以外の場合は，排他資源を確保するときにエラーになります。

- 前処理表

注※2

adbarchivechunk コマンドの処理の延長で，次に示す排他資源を確保する場合に，占有モード（EX）に遷移します。それ以外の場合は，排他資源を確保するときにエラーになります。

- 前処理表
- DB エリア（処理対象表およびインデクスの格納処理）
- 処理対象表

注※3

adbarchivechunk コマンドの処理の延長で，次に示す排他資源を確保する場合に，共有参照モード（SR）に遷移します。それ以外の場合は，排他制御モードは遷移しません。

- 前処理表
- DB エリア（処理対象表およびインデクスの格納処理）

注※4

adbarchivechunk コマンドの処理の延長で，次に示す排他資源を確保する場合に，保護更新モード（PU）に遷移します。それ以外の場合は，排他制御モードは遷移しません。

- 処理対象表

2.10.2　排他資源

排他資源の一覧を次の表に示します。

各排他資源の排他制御モードの詳細については，「表 2-15　確保される排他資源と排他制御モード」を参照してください。

表 2-14　排他資源の一覧

項番	排他資源	説明
1	シングルコネクション	AP が HADB サーバに接続したとき，または次に示すコマンドを実行したときに確保される排他資源です。 • adbimport コマンド • adbidxrebuild コマンド • adbgetcst コマンド • adbdbstatus コマンド • adbmodarea コマンド • adbexport コマンド • adbmergechunk コマンド • adbchgchunkcomment コマンド • adbchgchunkstatus コマンド • adbarchivechunk コマンド • adbunarchivechunk コマンド • adbreorgsystemdata コマンド • adbclientdefmang コマンド • adbsyndict コマンド • adbaudittrail コマンド • adbconvertaudittrailfile コマンド • adbmodbuff コマンド adbmodarea コマンドと，AP およびそのほかのコマンドが同時実行されないように排他制御しています。
2	データベースアクセス	次に示すときに確保される排他資源です。 1. 更新行のカラム化機能のメンテナンス処理の実行時 2. AP のトランザクションの実行時 3. HADB サーバに接続するコマンドの実行時 この排他資源によって，1.のメンテナンス処理と 2.の AP のトランザクション処理が同時実行されないように排他制御しています。また，1.のメンテナンス処理と 3.のコマンド処理が同時実行されないように排他制御しています。 トランザクションおよびコマンドの処理性能に影響を及ぼさないように，トランザクションまたはコマンドが実行されているときはメンテナンス処理が実行されないように制御しています。また，メンテナンス処理の実行中にトランザクションまたはコマンドが開始された場合は，メンテナンス処理を中断します。 なお，更新行のカラム化機能を無効にする場合は，この排他資源を意識する必要はありません。 HADB サーバに接続するコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧と共通規則』の『コマンドの一覧』を参照してください。

項番	排他資源		説明
3	ディクショナリ		定義系 SQL の実行時に確保される排他資源です。
4	前処理表		次に示すときに，処理対象表ごとに確保される排他資源です。 • 操作系 SQL の前処理時 • 定義系 SQL の実行時 • コマンドの実行時
5	DB エリア	データ用 DB エリア	データ用 DB エリア内のデータ更新時に確保される排他資源です。同一データ用 DB エリア内のデータを AP とコマンドによって同時に更新されないように排他制御しています。
6		ディクショナリ用 DB エリア	次に示すときに確保される排他資源です。 • 定義系 SQL の実行時 • adbdbstatus コマンドの実行時 ただし，adbdbstatus コマンドを実行しても次のどれかに該当する場合，この排他資源は確保されません。 • DB エリアのサマリ情報を出力する場合 • --shared-lock オプションを指定して情報を出力する場合 • 再編成要否の情報を出力する場合
7		システム表用 DB エリア	システム表用 DB エリア内のデータへのアクセス時に確保される排他資源です。
8	表	処理対象表	処理対象表へのアクセス時に確保される排他資源です。
9		ディクショナリ表	ディクショナリ表（実表）へのアクセス時に確保される排他資源です。
10		システム表（コスト情報）	コスト情報が格納されているシステム表（実表）へのアクセス時に確保される排他資源です。対象のシステム表（実表）を次に示します。 • STATUS_TABLES • STATUS_COLUMNS • STATUS_INDEXES
11		システム表（チャンク情報）	チャンク情報が格納されているシステム表（実表）へのアクセス時に確保される排他資源です。対象のシステム表（実表）は，STATUS_CHUNKS です。
12		システム表（同義語辞書情報）	同義語辞書情報が格納されているシステム表（実表）へのアクセス時に確保される排他資源です。対象のシステム表（実表）は，STATUS_SYNONYM_DICTIONARIES です。

2.10.3 排他資源が確保される期間

トランザクションが確保した排他資源は，トランザクションがコミットまたはロールバックすると解放されます。

なお，次に示すコマンドを実行した場合は，各コマンドの開始時に排他資源が確保され，各コマンドの終了時に排他資源が解放されます。

- adbimport コマンド
- adbidxrebuild コマンド
- adbgetcst コマンド
- adbdbstatus コマンド
- adbmodarea コマンド
- adbexport コマンド
- adbmergechunk コマンド
- adbchgchunkcomment コマンド
- adbchgchunkstatus コマンド
- adbarchivechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド
- adbclientdefmang コマンド
- adbsyndict コマンド
- adbaudittrail コマンド
- adbconvertaudittrailfile コマンド
- adbmodbuff コマンド

2.10.4　確保される排他資源と排他制御モード

確保される排他資源と排他制御モードを次の表に示します。

各排他資源の詳細については，「表 2-14　排他資源の一覧」を参照してください。また，各処理の排他制御の例については，「2.10.7　排他制御の例」を参照してください。

表 2-15　確保される排他資源と排他制御モード

処理内容		排他資源										
		シングルコネクション※16	ディクショナリ	前処理表	DB エリア			表				
					データ用 DB エリア	ディクショナリ用 DB エリア	システム表用 DB エリア	処理対象表	ディクショナリ表	システム表（コスト情報）	システム表（チャンク情報）	システム表（同義語辞書情報）
定義系 SQL の実行	次の SQL 文以外の定義系 SQL を実行する場合 • CREATE VIEW 文 • ALTER VIEW 文 • CREATE AUDIT 文 • DROP AUDIT 文	－	EX（WAIT）	EX	EX	SR	SR	EX	EX	SR	SR※2	－
	CREATE VIEW 文を実行する場合	－	EX（WAIT）	EX※1	－	SR	SR	－	SR（WAIT）と EX※14	SR	－	SR
	ALTER VIEW 文を実行する場合	－	EX（WAIT）	SR（WAIT）と EX※1, ※18	－	SR	SR	－	SR（WAIT）と EX※14	SR	－	SR
	CREATE AUDIT 文とDROP AUDIT 文を実行する場合	－	EX（WAIT）	EX	EX	SR	SR	－	EX	SR	－	－
検索系 SQL の実行		－	－	SR（WAIT）※3	－	－	－	SR（WAIT	SR（WAIT）	SR	SR※7	SR

処理内容		排他資源										
		シングルコネクション※16	ディクショナリ	前処理表	DB エリア			表				
					データ用 DB エリア	ディクショナリ用 DB エリア	システム表用 DB エリア	処理対象表	ディクショナリ表	システム表（コスト情報）	システム表（チャンク情報）	システム表（同義語辞書情報）
								）※4，※5，※6				
INSERT 文，UPDATE 文，および DELETE 文の実行	更新対象表の場合	－	－	SR（WAIT）※3	SU※4，※5，※6	－	－	SU	SR（WAIT）	SR	SR※7	SR
	更新対象表以外の場合	－	－	SR（WAIT）※3	－	－	－	SR（WAIT）※4，※5，※6	SR（WAIT）	SR	SR※7	SR
TRUNCATE TABLE 文の実行		－	－	SR（WAIT）	EX	－	SR※17	EX	SR（WAIT）	SR	SR※17	－
PURGE CHUNK 文の実行	更新対象表の場合	－	－	SR（WAIT）	EX	－	SR	EX	SR（WAIT）	SR	SR※15	SR
	更新対象表以外の場合※8	－	－	SR（WAIT）※3	－	－	－	SR	SR（WAIT）	SR	SR※15	SR
adbimport コマンドの実行	バックグラウンドインポート機能を適用する場合	SR	－	SR（WAIT）	PU	－	SR	SR（WAIT）	SR（WAIT）	SR	SR	－
	バックグラウンドインポート機能を適用しない場合	SR	－	SR（WAIT）	EX	－	SR※9	EX（WAIT）	SR（WAIT）	SR	SR※9	－
adbidxrebuild コマンドの実行	バックグラウンドインポートの中断後に実行	SR	－	SR（WAIT）	EX	－	SR	SR（WAIT）	SR（WAIT）	SR	SR	－

処理内容		排他資源										
		シングルコネクション※16	ディクショナリ	前処理表	DBエリア			表				
					データ用DBエリア	ディクショナリ用DBエリア	システム表用DBエリア	処理対象表	ディクショナリ表	システム表（コスト情報）	システム表（チャンク情報）	システム表（同義語辞書情報）
	した場合※10											
	バックグラウンドインポートの中断後以外に実行した場合	SR	－	SR（WAIT）	EX	－	－	EX（WAIT）	SR（WAIT）	SR	－	－
adbgetcst コマンドの実行		SR	－	SR（WAIT）	EX	－	SR	SR（WAIT）	SR（WAIT）	SR	－	－
adbdbstatus コマンドの実行	DBエリアのサマリ情報を出力する場合	SR	－	－	－	－	－	－	SR（WAIT）	－	－	－
	--shared-lock オプションを指定して情報を出力する場合，または再編成要否の情報を出力する場合	SR	－	SR※19	－	－	－	SR※19	SR（WAIT）	－	－	－
	上記以外の場合	SR	－	SR※19	EX	EX	EX	－	SR（WAIT）	SR	SR	－
adbmodarea コマンドの実行		EX	－	－	－	－	－	－	SR（WAIT）	－	－	－
adbexport コマンドの実行※11		SR	－	SR（WAIT）※3	－	－	－	SR（WAIT）	SR（WAIT）	SR	SR※7	SR

処理内容		排他資源										
		シングルコネクション※16	ディクショナリ	前処理表	DB エリア			表				
					データ用 DB エリア	ディクショナリ用 DB エリア	システム表用 DB エリア	処理対象表	ディクショナリ表	システム表（コスト情報）	システム表（チャンク情報）	システム表（同義語辞書情報）
adbmergechunk コマンドの実行		SR	−	SR（WAIT）	EX	−	SR	SU（WAIT）	SR（WAIT）	SR	SR	−
adbchgchunkcomment コマンドの実行		SR	−	SR（WAIT）	EX	−	SR	SR（WAIT）	SR（WAIT）	SR	SR	−
adbchgchunkstatus コマンドの実行		SR	−	EX（WAIT）	EX	−	SR	EX（WAIT）	SR（WAIT）	SR	SR	−
adbarchivechunk コマンドの実行		SR	−	SR（WAIT）と EX（WAIT）※12, ※13	SR と EX ※12	−	SR	PU と EX（WAIT）※12, ※13	SR（WAIT）	SR	SR	−
adbunarchivechunk コマンドの実行		SR	−	SR（WAIT）	EX	−	SR	EX（WAIT）	SR（WAIT）	SR	SR	−
adbreorgsystemdata コマンドの実行		SR	−	SR（WAIT）	−	−	EX	PU	SR（WAIT）	SR	−	−
adbclientdefmang コマンドの実行		SR	−	−	−	−	−	−	SR（WAIT）	−	−	−
adbsyndict コマンドの実行	同義語辞書の登録，更新，または削除の場合	SR	−	−	−	−	SR	−	−	−	−	SR
	同義語辞書ファイルの同期，同義語一覧の出力，または不要なファ	SR	−	−	−	−	−	−	−	−	−	SR

処理内容	排他資源										
	シングルコネクション※16	ディクショナリ	前処理表	DB エリア			表				
				データ用 DB エリア	ディクショナリ用 DB エリア	システム表用 DB エリア	処理対象表	ディクショナリ表	システム表（コスト情報）	システム表（チャンク情報）	システム表（同義語辞書情報）
イルの削除の場合											
adbaudittrail コマンドの実行	SR	－	－	－	－	－	－	SR（WAIT）	－	－	－
adbconvertaudittrailfile コマンドの実行	SR	－	－	－	－	－	－	SR（WAIT）	－	－	－
adbmodbuff コマンドの実行	SR	－	－	－	－	－	－	－	－	－	－

（凡例）

－：排他資源を確保しません。

EX：占有モード（EX）で排他資源を確保します。

PU：保護更新モード（PU）で排他資源を確保します。

SU：共有更新モード（SU）で排他資源を確保します。

SR：共有参照モード（SR）で排他資源を確保します。

WAIT：排他資源を確保できるまで待ち状態になります。

注

- 複数のトランザクションで同一の排他資源を確保しようとした場合の動作については，「2.10.1　排他制御モード」の「(2)　排他制御モードの同時実行の関係性（排他資源の確保の競合が発生したときの動作）」を参照してください。
- 排他資源「データベースアクセス」については，上記の表に記載していません。排他資源「データベースアクセス」の目的および機能については，「2.10.2　排他資源」を参照してください。

注※1

ビュー表の基表に対しては共有参照モード（SR）で排他を取得します。その際，排他資源を確保できない場合は，排他資源を確保できるまで待ち状態（WAIT）になります。

注※2

CREATE TABLE 文を実行してマルチチャンク表を作成した場合，および，DROP TABLE 文を実行してマルチチャンク表を削除した場合に，排他資源を確保します。

また，DROP SCHEMA 文などの定義系 SQL を実行して，処理の延長でマルチチャンク表が削除された場合にも，排他資源を確保します。

注※3

処理対象表がアーカイブマルチチャンク表の場合，処理対象表およびシステム表（STATUS_CHUNKS）に対する排他が確保されます。ただし，処理対象表に対してINSERT 文を実行する場合は該当しません。

注※4

同一トランザクション内で，保持機能にHOLD_CURSORS_OVER_COMMIT を指定して作成したResultSet オブジェクトがある場合，コミットしても排他資源は解放されません。そのResultSet オブジェクトをクローズしたあとに，コミットすると解放されます。なお，ロールバックした場合は排他資源は解放されます。

注※5

SQL 文の実行時にエラーが発生した場合，そのトランザクションが確保した排他資源はトランザクションが終了するまで解放されません。

注※6

複数の排他資源を確保するトランザクションが，排他資源の競合によってエラーとなった場合，エラー発生前までに確保していた排他資源はそのトランザクションが終了するまで解放されません。

注※7

処理対象表がアーカイブマルチチャンク表のときだけ，排他資源を確保します。

注※8

PURGE CHUNK 文の探索条件に副問合せを指定して実行した場合に，副問合せに含まれる表を指します。

注※9

マルチチャンク表に対して，-d オプションまたは-m オプションを指定してadbimport コマンドを実行したときだけ，排他資源を確保します。

注※10

バックグラウンドインポート機能を適用した（-b オプションを指定した）adbimport コマンドの中断後に，--create-temp-file オプションを指定してadbidxrebuild コマンドを実行した場合を指します。

注※11

adbexport コマンドを実行すると，処理の延長で検索系 SQL が実行されます。そのため，検索系 SQL と同様の排他資源も確保されます。

注※12

adbarchivechunk コマンドによる処理の延長で，排他制御モードの遷移が発生します。排他制御モードの遷移後，占有モード（EX）で排他資源が確保されます。

注※13

adbarchivechunk コマンドの実行時に，排他制御モードの遷移が発生した場合は，遷移によって排他資源に対する確保待ちが発生することがあります。確保待ちが発生した場合，ほかのコネクションによる排他資源の確保が優先されることがあります。

注※14

前処理時は，共有参照モード（SR）で排他資源を確保します。その際，排他資源を確保できない場合は，排他資源を確保できるまで待ち状態（WAIT）になります。

また，実行時は，占有モード（EX）で排他資源を確保します。

注※15

更新対象表以外の表にアーカイブマルチチャンク表が含まれる場合は，前処理時に排他資源を確保します。それ以外の場合は，実行時に排他資源を確保します。

注※16

対象のコマンドを実行した場合に，排他資源を確保します。また，AP が HADB サーバに接続した場合にも，排他資源を確保します。AP が HADB サーバに接続した場合は，共有参照モード（SR）で排他資源を確保します。

注※17

処理対象表がマルチチャンク表のときだけ，排他資源を確保します。

注※18

ALTER VIEW 文を実行すると，排他制御モードの遷移が発生します。排他制御モードの遷移後，占有モード（EX）で排他資源が確保されます。

注※19

adbdbstatus コマンドの対象がインデクスである場合は，インデクスが定義された表に対して排他を取得します。

adbdbstatus コマンドの対象が DB エリアである場合は，対象の DB エリアに格納された表，および対象の DB エリアに格納されたインデクスが定義されている表に対して，排他を取得します。

上記の表の見方を，検索系 SQL を実行した場合を例にして説明します。

検索系 SQL を実行した場合

アーカイブマルチチャンク表に対して検索系 SQL を実行すると，次に示す排他資源を共有参照モード（SR）で確保します。

- 前処理表
- 処理対象表
- ディクショナリ表
- システム表（コスト情報）
- システム表（チャンク情報）
- システム表（同義語辞書情報）

上記の排他資源を，共有参照モード（SR），共有更新モード（SU），および保護更新モード（PU）で確保する処理は実行できます。例えば，adbgetcst コマンドは実行できます。

ただし，上記の排他資源を，占有モード（EX）で確保する処理は実行できません。例えば，定義系 SQL，TRUNCATE TABLE 文などは実行できません。

2.10.5　ディクショナリ表およびシステム表の参照時に排他が取得される実表

SQL 文を実行してディクショナリ表およびシステム表を参照すると，ディクショナリ表（実表）およびシステム表（実表）の排他が取得されます。ディクショナリ表およびシステム表の参照時に排他が取得される実表について，次の表に示します。

表 2-16　ディクショナリ表およびシステム表の参照時に排他が取得される実表

参照するディクショナリ表およびシステム表	排他が取得されるディクショナリ表（実表）およびシステム表（実表）																						
	SQL_TABLES	SQL_COLUMNS	SQL_INDEXES	SQL_DIV_TABLE	SQL_DIV_INDEX	SQL_DBAREAS	SQL_SCHEMATA	SQL_VIEWS	SQL_VIEW_TABLE_USAGE	SQL_VIEW_OBJECT	SQL_DEFINE_SOURCE	SQL_DEFINE_ENVIRONMENT	SQL_USERS	SQL_TABLE_CONSTRAINTS	SQL_INDEX_COLINF	SQL_KEY_COLUMN_USAGE	SQL_REFERENTIAL_CONSTRAINTS	SQL_TABLE_PRIVILEGES	SQL_AUDITS	STATUS_TABLES	STATUS_COLUMNS	STATUS_INDEXES	STATUS_CHUNKS	STATUS_SYNONYM_DICTIONARIES
SQL_TABLES	◎	△	△	△	△	－	－	◎	－	△	－	－	○	－	－	－	－	◎	－	△	△	△	－	－
SQL_COLUMNS	◎	◎	△	△	△	－	－	◎	－	△	－	－	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_INDEXES	△	△	◎	△	△	－	－	△	－	△	－	－	○	－	－	－	－	◎	－	△	△	△	－	－
SQL_DIV_TABLE	△	△	△	◎	△	－	－	△	－	△	－	－	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_DIV_INDEX	△	△	△	△	◎	－	－	△	－	△	－	－	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_DBAREAS	△	△	△	△	△	○	－	△	－	△	－	－	－	－	－	－	－	△	－	△	△	△	－	－
SQL_SCHEMATA	△	△	△	△	△	－	○	△	－	△	－	－	○	－	－	－	－	◎	－	△	△	△	－	－
SQL_VIEWS	△	△	△	△	△	－	－	◎	－	△	－	－	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_VIEW_TABLE_USAGE	△	△	△	△	△	－	－	◎	○	△	－	－	○	－	－	－	－	◎	－	△	△	△	－	－
SQL_DEFINE_SOURCE	△	△	△	△	△	－	－	◎	－	△	○	－	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_DEFINE_ENVIRONMENT	△	△	△	△	△	－	－	◎	－	△	－	○	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_USERS	△	△	△	△	△	－	－	△	－	△	－	－	○	－	－	－	－	△	－	△	△	△	－	－
SQL_TABLE_CONSTRAINTS	△	△	△	△	△	－	－	△	－	△	－	－	－	○	－	－	－	◎	－	△	△	△	－	－
SQL_INDEX_COLINF	△	△	△	△	△	－	－	△	－	△	－	－	－	－	○	－	－	◎	－	△	△	△	－	－
SQL_KEY_COLUMN_USAGE	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	○	－	◎	－	△	△	△	－	－
SQL_REFERENTIAL_CONSTRAINTS	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	○	◎	－	△	△	△	－	－
SQL_TABLE_PRIVILEGES	◎	△	△	△	△	－	－	◎	－	△	－	－	－	－	－	－	－	◎	－	△	△	△	－	－
SQL_AUDITS	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	－	△	○	△	△	△	－	－
STATUS_TABLES	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	－	◎	－	◎	△	△	－	－
STATUS_COLUMNS	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	－	◎	－	◎	◎	△	－	－
STATUS_INDEXES	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	－	◎	－	△	△	◎	－	－
STATUS_CHUNKS	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	－	◎	－	△	△	△	○	－
STATUS_SYNONYM_DICTIONARIES	△	△	△	△	△	－	－	△	－	△	－	－	－	－	－	－	－	△	－	△	△	△	－	○

（凡例）

◎：SQL 文の前処理時，および SQL 文の実行時に共有参照モード（SR）で排他を取得します。なお，SQL 文の実行時に，対象の排他資源がすでに占有モード（EX）で確保されている場合は，確保できるまで待ち状態になります。

○：SQL 文の実行時に共有参照モード（SR）で排他を取得します。なお，対象の排他資源がすでに占有モード（EX）で確保されている場合は，確保できるまで待ち状態になります。

△：SQL 文の前処理時に共有参照モード（SR）で排他を取得します。

－：排他を取得しません。

2.10.6 データベースの参照・更新時に排他が取得されるディクショナリ表とシステム表

SQL 文やコマンドを実行してデータベースを参照したり更新したりすると，ディクショナリ表（実表）とシステム表（実表）に対する排他が取得されます。データベースを参照したり更新したりした場合に，排他が取得されるディクショナリ表とシステム表について，次の 2 つの表に示します。

表 2-17　データベースの参照・更新時に排他が取得されるディクショナリ表とシステム表（その 1）

データベースの参照・更新処理（実行するSQL文とコマンド）			排他が取得されるディクショナリ表（実表）およびシステム表（実表）																						
			SQL_TABLES	SQL_COLUMNS	SQL_INDEXES	SQL_DIV_TABLE	SQL_DIV_INDEX	SQL_DBAREAS	SQL_SCHEMATA	SQL_VIEWS	SQL_VIEW_TABLE_USAGE	SQL_VIEW_OBJECT	SQL_DEFINE_SOURCE	SQL_DEFINE_ENVIRONMENT	SQL_USERS	SQL_TABLE_CONSTRAINTS	SQL_INDEX_COLINF	SQL_KEY_COLUMN_USAGE	SQL_REFERENTIAL_CONSTRAINTS	SQL_TABLE_PRIVILEGES	SQL_AUDITS	STATUS_TABLES	STATUS_COLUMNS	STATUS_INDEXES	STATUS_CHUNKS	STATUS_SYNONYM_DICTIONARIES
接続		CONNECT													SRW											
		DISCONNECT																								
定義系SQL	HADBユーザ	CREATE USER文													EX							SR	SR	SR	SR	
		ALTER USER文													EX							SR	SR	SR	SR	
		DROP USER文	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX		SR	SR	SR	SR	
	スキーマ	CREATE SCHEMA文							EX						EX							SR	SR	SR	SR	
		DROP SCHEMA文	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX		EX	EX	EX	EX	EX		SR	SR	SR	SR	
	実表	CREATE TABLE文	EX	EX	EX	EX	EX	EX	EX							EX	EX	EX		EX		SR	SR	SR	SR	
		ALTER TABLE文（列の追加）	EX	EX		EX	EX															SR	SR	SR	SR	
		ALTER TABLE文（チャンク数の最大値の変更）	EX			EX	EX															SR	SR	SR	SR	
		ALTER TABLE文（レギュラーマルチチャンク表からアーカイブマルチチャンク表への変更）	EX	EX	EX	EX	EX	EX	EX	EX	EX					EX	EX	EX		EX		SR	SR	SR	SR	
		ALTER TABLE文（アーカイブマルチチャンク表からレギュラーマルチチャンク表への変更）	EX	EX	EX	EX	EX	EX		EX	EX	EX	EX	EX		EX	EX	EX	EX	EX		SR	SR	SR	SR	
		ALTER TABLE文（列名の変更）	EX	EX						EX	EX						EX	EX				SR	SR	SR	SR	
		ALTER TABLE文（表名の変更）	EX	EX	EX	EX	EX			EX	EX					EX	EX	EX	EX	EX	EX	SR	SR	SR	SR	
		DROP TABLE文	EX	EX	EX	EX	EX	EX		EX	EX	EX	EX	EX		EX	EX	EX	EX	EX		SR	SR	SR	SR	
	ビュー表	CREATE VIEW文（前処理時）	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			●					SRW		SR	SR	SR	SR	SR
		CREATE VIEW文（実行時）	EX	EX					EX	EX	EX	EX	EX	EX						EX		SR	SR	SR	SR	
		ALTER VIEW文（前処理時）	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW	SRW	SRW	●					SRW		SR	SR	SR	SR	SR
		ALTER VIEW文（実行時）	EX	EX				EX	EX	EX	EX	EX		EX						EX		SR	SR	SR	SR	
		DROP VIEW文	EX	EX					EX	EX	EX	EX	EX	EX						EX		SR	SR	SR	SR	
	インデクス	CREATE INDEX文	EX	EX	EX		EX	EX									EX					SR	SR	SR	SR	
		DROP INDEX文	EX	EX	EX		EX	EX									EX					SR	SR	SR	SR	
	監査対象定義	CREATE AUDIT文													EX						EX	SR	SR	SR	SR	
		DROP AUDIT文													EX						EX	SR	SR	SR	SR	
	権限	GRANT文（ユーザ権限，スキーマ操作権限，または監査権限の付与）													EX							SR	SR	SR	SR	
		GRANT文（アクセス権限の付与）	EX												EX					EX		SR	SR	SR	SR	
		REVOKE文（ユーザ権限，スキーマ操作権限，または監査権限の取り消し）	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX		SR	SR	SR	SR	
		REVOKE文（アクセス権限の取り消し）	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX		SR	SR	SR	SR	
操作系SQL		SELECT文	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			●					SRW		SR	SR	SR	SR	SR
		INSERT文	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			●					SRW		SR	SR	SR	SR	SR
		UPDATE文	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			●					SRW		SR	SR	SR	SR	SR
		DELETE文	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			●					SRW		SR	SR	SR	SR	SR
		TRUNCATE TABLE文	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW								SRW		SR	SR	SR	SR	
		PURGE CHUNK文	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			●					SRW		SR	SR	SR	SR	SR

（凡例）

EX

占有モード（EX）で排他を取得します。

SRW

共有参照モード（SR）で排他を取得します。なお，ほかのトランザクションによって対象の排他資源がすでに確保されている場合は，確保できるまで待ち状態になります。

SR

共有参照モード（SR）で排他を取得します。

●

SQL 文中に次のどれかの指定がある場合に限り，共有参照モード（SR）で排他を取得します。なお，ほかのトランザクションによって対象の排他資源がすでに確保されている場合は，確保できるまで待ち状態になります。

1. ADB_AUDITREAD 関数を指定した表関数導出表
2. ディクショナリ表のSQL_AUDITS 表
3. ADB_AUDITREAD 関数を指定した表関数導出表をCREATE VIEW 文中に指定して定義されたビュー表
4. ディクショナリ表のSQL_AUDITS 表をCREATE VIEW 文中に指定して定義されたビュー表
5. 3.または 4.のビュー表に依存するビュー表

表 2-18　データベースの参照・更新時に排他が取得されるディクショナリ表とシステム表（その 2）

データベースの参照・更新処理（実行するSQL文とコマンド）		排他が取得されるディクショナリ表（実表）およびシステム表（実表）																							
		SQL_TABLES	SQL_COLUMNS	SQL_INDEXES	SQL_DIV_TABLE	SQL_DIV_INDEX	SQL_DBAREAS	SQL_SCHEMATA	SQL_VIEWS	SQL_VIEW_TABLE_USAGE	SQL_VIEW_OBJECT	SQL_DEFINE_SOURCE	SQL_DEFINE_ENVIRONMENT	SQL_USERS	SQL_TABLE_CONSTRAINTS	SQL_INDEX_COLINF	SQL_KEY_COLUMN_USAGE	SQL_REFERENTIAL_CONSTRAINTS	SQL_TABLE_PRIVILEGES	SQL_AUDITS	STATUS_TABLES	STATUS_COLUMNS	STATUS_INDEXES	STATUS_CHUNKS	STATUS_SYNONYM_DICTIONARIES
コマンド	adbimportコマンド（バックグラウンドインポート機能を適用しない場合）	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbimportコマンド（バックグラウンドインポート機能を適用する場合）	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbexportコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	SR
	adbgetcstコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR		
	adbidxrebuildコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbmergechunkコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbchgchunkcommentコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbchgchunkstatusコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbdbstatusコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbmodareaコマンド																								
	adbarchivechunkコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbunarchivechunkコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW			SRW					SRW		SR	SR	SR	SR	
	adbreorgsystemdataコマンド	SRW	SRW	SRW	SRW	SRW	SRW		SRW		SRW								SRW		SR/SUW	SR/SUW	SR/SUW	-/SUW	-/SUW
	adbclientdefmangコマンド													SRW											
	adbsyndictコマンド																								SR
	adbauditrailコマンド（--startオプションを指定する場合）													SRW						SRW					
	adbauditrailコマンド（--startオプション以外のオプションを指定する場合）													SRW											
	adbconvertauditrailfileコマンド													SRW											

（凡例）

SRW

共有参照モード（SR）で排他を取得します。なお，ほかのトランザクションによって対象の排他資源がすでに確保されている場合は，確保できるまで待ち状態になります。

SR

共有参照モード（SR）で排他を取得します。

SR/SUW

SQL 文やコマンドの操作対象表ではない場合は，共有参照モード（SR）で排他を取得します。

SQL 文やコマンドの操作対象表である場合は，共有更新モード（SU）で排他を取得します。なお，ほかのトランザクションによって対象の排他資源がすでに確保されている場合は，確保できるまで待ち状態になります。

－/SUW

SQL 文やコマンドの操作対象表ではない場合は，排他を取得しません。

SQL 文やコマンドの操作対象表である場合は，共有更新モード（SU）で排他を取得します。なお，ほかのトランザクションによって対象の排他資源がすでに確保されている場合は，確保できるまで待ち状態になります。

2.10.7　排他制御の例

ここでは，HADB の排他制御の具体例を説明します。

排他制御の仕組みを理解するために，「2.10.2　排他資源」，および「2.10.4　確保される排他資源と排他制御モード」とあわせてお読みください。

(1)　CREATE TABLE 文実行時の排他制御（実表の定義）

CREATE TABLE 文を実行すると，次に示す排他資源が確保されます。

- ディクショナリ（占有モード（EX））
- 前処理表（占有モード（EX））
- DB エリア
  - データ用 DB エリア（占有モード（EX））
  - ディクショナリ用 DB エリア（共有参照モード（SR））
  - システム表用 DB エリア（共有参照モード（SR））
- 表
  - 処理対象表（占有モード（EX））
  - ディクショナリ表（占有モード（EX））
  - システム表（コスト情報）（共有参照モード（SR））
  - システム表（チャンク情報）（共有参照モード（SR））

ディクショナリの排他が占有モード（EX）で取得されるため，CREATE TABLE 文の実行時に後発で実行した定義系 SQL は，CREATE TABLE 文が完了するまで待ち状態になります。また，ディクショナリ表の排他が占有モード（EX）で取得されるため，CREATE TABLE 文の実行時に後発で実行した検索系 SQL や更新系 SQL も，CREATE TABLE 文が完了するまで待ち状態になります。

CREATE TABLE 文の実行時に後発で実行した検索系 SQL（SELECT 文）が待ち状態になる例を次に示します。

例 1：定義中の実表に対して SELECT 文を実行する場合

```
CREATE TABLE T1(”ID” INT, ”NAME” VARCHAR(100)) in ADBUTBL01   ...1
SELECT * FROM T1                                              ...2
```

2 のSELECT 文は，1 のCREATE TABLE 文が完了するまで待ち状態になります。

例 2：定義中の実表とは別の実表に対して SELECT 文を実行する場合

```
CREATE TABLE T1(”ID” INT, ”NAME” VARCHAR(100)) in ADBUTBL01   ...1
SELECT * FROM T2                                              ...2
```

注　実表T2 は実表T1 とは別のデータ用 DB エリアに定義されているものとします。

2 のSELECT 文は，1 のCREATE TABLE 文が完了するまで待ち状態になります。

また，検索系 SQL 以外にも，CREATE TABLE 文の実行時に後発で実行すると，CREATE TABLE 文が完了するまで待ち状態になる操作があります。検索系 SQL 以外の，後発で実行すると待ち状態になる主な操作を次に示します。

- INSERT 文
- UPDATE 文
- DELETE 文
- TRUNCATE TABLE 文
- PURGE CHUNK 文
- adbimport コマンド
- adbdbstatus コマンド
- adbexport コマンド
- adbmergechunk コマンド

(2)　adbimport コマンド実行時の排他制御（データインポート）

adbimport コマンド実行時の排他制御は，バックグラウンドインポート機能を適用するかどうかで異なります。

(a)　バックグラウンドインポート機能を適用する場合の排他制御

バックグラウンドインポート機能を適用したadbimport コマンドを実行すると，データ用 DB エリアの排他が保護更新モード（PU）で取得されます。このため，インポート対象表に対する後発の更新操作は，排他資源を確保するときにエラーになります。

バックグラウンドインポート機能を適用したadbimport コマンド実行時に，後発で実行するとエラーになる主な操作※を次に示します。

- 定義系 SQL

- INSERT 文
- UPDATE 文
- DELETE 文
- TRUNCATE TABLE 文
- PURGE CHUNK 文
- adbdbstatus コマンド
  ただし，次のどれかの場合はadbdbstatus コマンドを実行できます。
  - DB エリアのサマリ情報を出力する場合
  - --shared-lock オプションを指定して情報を出力する場合
  - 再編成要否の情報を出力する場合
- adbmergechunk コマンド

注※

次のどれかの場合が該当します。

- SQL 文やコマンドの操作対象表が，インポート対象表と同一の場合
- SQL 文やコマンドの操作対象表が，インポート対象表と同じデータ用 DB エリアに格納されている場合
- SQL 文やコマンドの操作対象表が，インポート対象表に定義されているインデクスと同じデータ用 DB エリアに格納されている場合
- SQL 文やコマンドの操作対象表に定義されているインデクスが，インポート対象表と同じデータ用 DB エリアに格納されている場合

(b) バックグラウンドインポート機能を適用しない場合の排他制御

バックグラウンドインポート機能を適用しないadbimport コマンドを実行すると，データ用 DB エリアおよび処理対象表の排他が占有モード（EX）で取得されます。このため，インポート対象表に対する操作は，排他資源を確保するときにエラーになるか，排他資源を確保できるまで待ち状態になります。

■後発で実行するとエラーになる操作

バックグラウンドインポート機能を適用しないadbimport コマンド実行時に，後発で実行するとエラーになる主な操作※を次に示します。

- 定義系 SQL
- INSERT 文
- UPDATE 文
- DELETE 文
- TRUNCATE TABLE 文

- PURGE CHUNK 文
- adbdbstatus コマンド

  ただし，次のどれかの場合はadbdbstatus コマンドを実行できます。
  - DB エリアのサマリ情報を出力する場合
  - --shared-lock オプションを指定して情報を出力する場合
  - 再編成要否の情報を出力する場合
- adbmergechunk コマンド

注※

次のどれかの場合が該当します。
- SQL 文やコマンドの操作対象表が，インポート対象表と同一の場合
- SQL 文やコマンドの操作対象表が，インポート対象表と同じデータ用 DB エリアに格納されている場合
- SQL 文やコマンドの操作対象表が，インポート対象表に定義されているインデクスと同じデータ用 DB エリアに格納されている場合
- SQL 文やコマンドの操作対象表に定義されているインデクスが，インポート対象表と同じデータ用 DB エリアに格納されている場合

■後発で実行すると待ち状態になる操作

バックグラウンドインポート機能を適用しないadbimport コマンド実行時に，インポート対象表に対して後発で実行すると待ち状態になる主な操作を次に示します。

- 検索系 SQL
- adbexport コマンド

> **メモ**
>
> インポート対象表以外の表に対する上記の操作は，待ち状態にならないで実行できます（データ用 DB エリアの排他は影響しません）。

(3) adbexport コマンド実行時の排他制御（データエクスポート）

adbexport コマンドを実行すると，次に示す排他資源が共有参照モード（SR）で確保されます。

- シングルコネクション
- 前処理表
- 表
  - 処理対象表

- ディクショナリ表
- システム表（コスト情報）
- システム表（チャンク情報）
- システム表（同義語辞書情報）

このため，上記の排他資源を占有モード（EX）で確保する操作以外は，adbexport コマンド実行時，同時に実行できます。上記の排他資源を占有モード（EX）で確保する操作は，排他資源を確保するときにエラーになるか，排他資源を確保できるまで待ち状態になります。

■エクスポート対象表に対して後発で実行するとエラーになる操作

adbexport コマンド実行時に，エクスポート対象表に対して後発で実行するとエラーになる主な操作を次に示します。

- 定義系 SQL
- TRUNCATE TABLE 文
- PURGE CHUNK 文

> **メモ**
>
> エクスポート対象表以外の表に対する上記の操作は，エラーにならないで実行できます（データ用 DB エリアの排他は影響しません）。

■エクスポート対象表に対して後発で実行すると待ち状態になる操作

adbexport コマンド実行時に，エクスポート対象表に対して後発で実行すると待ち状態になる主な操作を次に示します。

- バックグラウンドインポート機能を適用しないadbimport コマンド
- adbmergechunk コマンド（--purge-chunk オプションの指定を省略，または--purge-chunk オプションにWAIT を指定した場合）
- adbreorgsystemdata コマンド（エクスポート対象がシステム表の場合）

> **メモ**
>
> エクスポート対象表以外の表に対する上記の操作は，待ち状態にならないで実行できます（データ用 DB エリアの排他は影響しません）。

(4) adbmergechunk コマンド実行時の排他制御（チャンクのマージ）

adbmergechunk コマンドを実行すると，次に示す排他資源が確保されます。

- シングルコネクション（共有参照モード（SR））

- 前処理表（共有参照モード（SR））
- DB エリア
  - データ用 DB エリア（占有モード（EX））
  - システム表用 DB エリア（共有参照モード（SR））
- 表
  - 処理対象表（共有更新モード（SU））
  - ディクショナリ表（共有参照モード（SR））
  - システム表（コスト情報）（共有参照モード（SR））
  - システム表（チャンク情報）（共有参照モード（SR））

このため，次に示す後発の操作は，排他資源を確保するときにエラーになります。

- マージチャンク対象表と同じ DB エリアに格納されている表に対して，排他を取得する後発の操作
- マージチャンク対象表に対して，保護更新モード（PU），または占有モード（EX）で排他を取得する後発の操作
- シングルコネクション，前処理表，ディクショナリ表，またはシステム表に対して，占有モード（EX）で排他を取得する後発の操作

adbmergechunk コマンド実行時に，後発で実行するとエラーになる主な操作※を次に示します。

- 定義系 SQL 文
- INSERT 文
- UPDATE 文
- DELETE 文
- TRUNCATE TABLE 文
- PURGE CHUNK 文
- adbimport コマンド
- adbdbstatus コマンド
  ただし，次のどれかの場合はadbdbstatus コマンドを実行できます。
  - DB エリアのサマリ情報を出力する場合
  - --shared-lock オプションを指定して情報を出力する場合
  - 再編成要否の情報を出力する場合

注※

次のどれかの場合が該当します。

- SQL 文やコマンドの操作対象表が，マージチャンク対象表と同一の場合

- SQL 文やコマンドの操作対象表が，マージチャンク対象表と同じデータ用 DB エリアに格納されている場合
- SQL 文やコマンドの操作対象表が，マージチャンク対象表に定義されているインデクスと同じデータ用 DB エリアに格納されている場合
- SQL 文やコマンドの操作対象表に定義されているインデクスが，マージチャンク対象表と同じデータ用 DB エリアに格納されている場合

(5) adbdbstatus コマンド実行時の排他制御（データベースの状態解析）

adbdbstatus コマンド実行時の排他制御は，出力する情報によって異なります。

(a) DB エリアのサマリ情報を出力する場合の排他制御

adbdbstatus コマンドで DB エリアのサマリ情報を出力する場合，シングルコネクションおよびディクショナリ表の排他が共有参照モード（SR）で取得されます。このため，ディクショナリ表に対して占有モード（EX）で排他を取得する後発の操作は，排他資源を確保するときにエラーになります。

adbdbstatus コマンド実行時に，後発で実行するとエラーになる主な操作を次に示します。

- 定義系 SQL

(b) --shared-lock オプションを指定する場合と再編成要否の情報を出力する場合の排他制御

adbdbstatus コマンドに--shared-lock オプションを指定して情報を出力する場合，またはadbdbstatus コマンドで再編成要否の情報を出力する場合，次に示す排他資源が共有参照モード（SR）で確保されます。

- シングルコネクション
- 前処理表
- 表
  - 処理対象表
  - ディクショナリ表

このため，上記の排他資源に対して占有モード（EX）で排他を取得する後発の操作は，排他資源を確保するときにエラーになります。

adbdbstatus コマンド実行時に，後発で実行するとエラーになる主な操作を次に示します。

- 定義系 SQL
- TRUNCATE TABLE 文
- PURGE CHUNK 文
- バックグラウンドインポート機能を適用しないadbimport コマンド

(c)　その他の情報を出力する場合の排他制御

「(a)　DB エリアのサマリ情報を出力する場合の排他制御」または「(b)　--shared-lock オプションを指定する場合と再編成要否の情報を出力する場合の排他制御」以外の場合，次に示す排他資源が確保されます。

- シングルコネクション（共有参照モード（SR））
- 前処理表（共有参照モード（SR））
- DB エリア
  - データ用 DB エリア（占有モード（EX））
  - ディクショナリ用 DB エリア（占有モード（EX））
  - システム表用 DB エリア（占有モード（EX））
- 表
  - ディクショナリ表（共有参照モード（SR））

このため，データ用 DB エリア，ディクショナリ用 DB エリア，またはシステム表用 DB エリアに対して排他を取得する後発の操作は，排他資源を確保するときにエラーになります。また，シングルコネクション，前処理表，またはディクショナリ表に対して占有モード（EX）で排他を取得する後発の操作もエラーになります。

adbdbstatus コマンド実行時に，後発で実行するとエラーになる主な操作を次に示します。

- 定義系 SQL
- INSERT 文
- UPDATE 文
- DELETE 文
- TRUNCATE TABLE 文
- PURGE CHUNK 文
- adbimport コマンド
- adbmergechunk コマンド

(6)　adbmodarea コマンド実行時の排他制御（DB エリアの追加・変更）

adbmodarea コマンドを実行すると，シングルコネクションの排他が占有モード（EX）で取得されます。このため，adbmodarea コマンド実行時，シングルコネクションの排他を取得する操作（HADB サーバに接続する操作）を実行すると，すべてエラーになります。

adbmodarea コマンド実行時に，後発で実行するとエラーになる主な操作を次に示します。

- 定義系 SQL※
- 検索系 SQL※

- INSERT 文※
- UPDATE 文※
- DELETE 文※
- TRUNCATE TABLE 文※
- PURGE CHUNK 文※
- adbimport コマンド
- adbdbstatus コマンド
- adbexport コマンド
- adbmergechunk コマンド

注※

AP が HADB サーバに接続する際にエラーになるため，SQL 文を実行できません。

(7) adbls コマンド実行時の排他制御（HADB サーバの状態表示）

adbls コマンドは排他資源を確保しません。このため，排他資源を確保するほかの操作と同時に実行できます。

2.11　データベースの回復処理

ここでは，障害が発生して HADB サーバが異常終了した場合のデータベースの回復処理について説明します。

2.11.1　再開始による回復処理の流れ

HADB サーバが異常終了した場合，HADB を再度開始すると，HADB サーバはステータス情報を使用して（自動的に）再開始します。再開始すると，システムログを使用してデータベースの回復処理を行い，HADB サーバが異常終了する直前の状態に回復させます。

(1)　ステータス情報

HADB サーバが異常終了した場合に備えて，HADB サーバは（自動的に）再開始するために必要な情報を出力します。これを**ステータス情報**といいます。

ステータス情報は，HADB サーバの状態が変化した時点で出力され，ステータスファイルに格納されます。ステータス情報として出力される情報を次に示します。

- HADB サーバの終了モード（正常終了，異常終了など）
- HADB サーバの稼働状況（開始処理中，稼働中，終了処理中など）

ステータスファイルが作成される場所は，DB ディレクトリ下（$DBDIR/ADBSYS/ADBSTS）になります。

(2)　システムログ

データベースを更新すると，データベースの更新履歴情報が出力されます。この情報を**システムログ**といいます。また，システムログが出力されるファイルを**システムログファイル**といいます。

システムログは，HADB が異常終了したときの再開始，およびロールバックが発生したときの回復処理に使用されます。

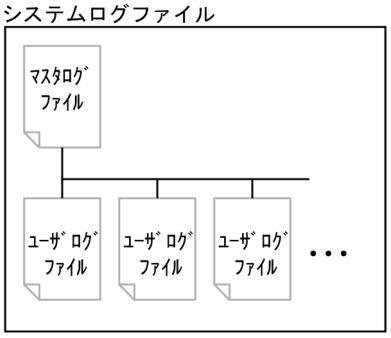
(a)　システムログファイルの作成先

システムログファイルは，DB ディレクトリ下（$DBDIR/ADBSYS/ADBSLG）に作成されます。

(b)　システムログファイルの構成

システムログファイルは，マスタログファイルおよびユーザログファイルの 2 種類のファイルで構成されます。システムログファイルの構成を次の図に示します。

図 2-35　システムログファイルの構成



- マスタログファイル

  マスタログファイルには，トランザクションの開始および終了など，システム制御に関する履歴情報（イベントログ）が格納されます。また，ユーザログファイル群を管理します。

  マスタログファイルは，HADB サーバを開始したときに自動的に 1 つ作成されます。ファイル名称は MSTLOG になります。

- ユーザログファイル

  ユーザログファイルには，データベースに対する操作履歴（**ユーザログ**）が格納されます。

  ユーザログファイルは，データベースを更新するトランザクションおよびリアルスレッドごとに割り当てられます。

  ユーザログファイルは，HADB サーバを開始したときに，ユーザの指定に従って作成されます。ファイル名称はUSRLOG で始まる名称になります。

  ユーザログファイルに関する指定を変更した場合，HADB サーバを開始したときにシステムログファイルは再作成されます。

2.12　クライアントグループ機能

ここでは，クライアントグループ機能について説明します。

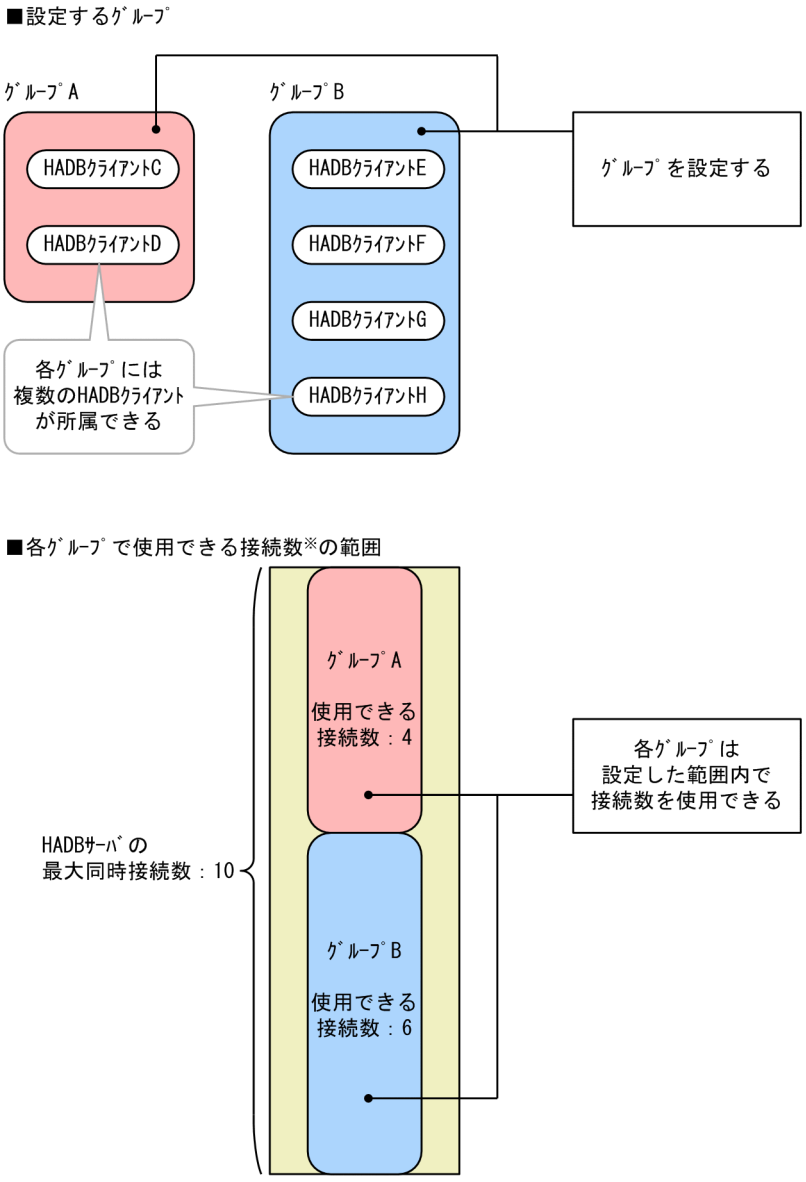
2.12.1　クライアントグループ機能の概要

HADB では，複数の HADB クライアントを，1 つのグループにまとめて管理する**クライアントグループ機能**をサポートしています。

クライアントグループ機能を適用すると，1 つのグループが使用できる接続数や処理リアルスレッド数を設定できます。グループに所属する HADB クライアントは，グループに設定した接続数や処理リアルスレッド数の範囲内で，処理を実行するようになります。

クライアントグループ機能の概要について，接続数の管理を例に説明します。

図 2-36　クライアントグループ機能の適用例（接続数を管理する場合）



注※
処理ﾘｱﾙｽﾚｯﾄﾞ数についても，接続数の場合と同様に管理します。

クライアントグループ機能を適用して，複数のグループに対して，必要な接続数や処理リアルスレッド数を適切に設定すると，次のような運用を実現できます。

- 特定の HADB クライアントやコマンドが，HADB サーバに集中して接続したり，処理リアルスレッドを使用し過ぎたりすることを防ぐ（ほかの HADB クライアントやコマンドが行う処理に影響を与えないようにする）。
- 定期的に実行する処理がある HADB クライアントやコマンドのために，実行に必要な接続数や処理リアルスレッド数を常に確保しておく（ほかの HADB クライアントやコマンドが行う処理の影響を受けないようにする）。

2.12.2　クライアントグループ機能で設定できるグループの種類

クライアントグループ機能で設定できるグループの種類を次に示します。

- **クライアントグループ**

  HADB クライアントが所属できるグループです。1 つのクライアントグループに，複数の HADB クライアントが所属できます。また，クライアントグループは複数設定できます。

- **コマンドグループ**

  HADB サーバに接続するコマンドが所属できるグループです。例えば，データインポートを実行するコマンドや，インデクスの再作成を実行するコマンドなどが所属できます。HADB サーバに接続するコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧』を参照してください。コマンドグループは 1 つだけ設定できます。

> **メモ**
>
> コマンドグループには，HADB クライアントも所属することができます。詳細については，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」の「adbcltgrp（コマンドグループを設定する場合）」を参照してください。

2.12.3　グループごとの接続数・処理リアルスレッド数の設定

クライアントグループ機能を適用すると，HADB クライアントおよびコマンドの接続数や処理リアルスレッド数について，使用できる上限値をグループごとに設定できます。また，最低限使用できる数もグループごとに設定できます。

(1)　グループごとの接続数の設定

クライアントグループ機能で設定できる接続数の種別を，次に示します。

- **最大同時接続数**

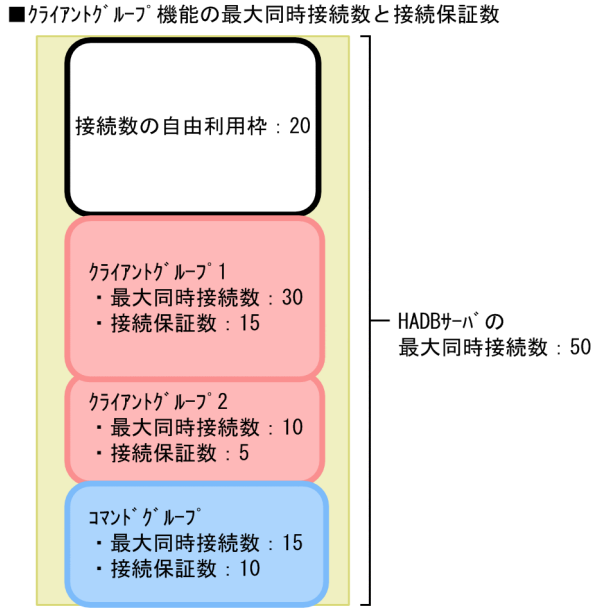
  対象のグループが使用できる接続数の上限値を設定できます。設定した最大同時接続数の範囲内で，対象のグループに所属している HADB クライアントやコマンドは同時接続します。

- **接続保証数**

  対象のグループが最低限使用できる接続数を設定できます。対象のグループは，設定した接続保証数までは，常に同時接続できます。自グループ以外の HADB クライアントやコマンドの接続状況に影響されません。

クライアントグループ機能の最大同時接続数と接続保証数の概要を，次の図に示します。

図 2-37　クライアントグループ機能の最大同時接続数と接続保証数の概要



[説明]

対象のグループは，設定した接続保証数までは，常に接続できます。例えば，クライアントグループ 1 は，接続保証数（15）までは常に接続できます。

HADB サーバの最大同時接続数（50）から，各グループで設定した接続保証数の合計（30）を除いた数が，接続数の自由利用枠（20）になります。

接続数の自由利用枠の範囲内で，すべての HADB クライアントやコマンドは接続できます。グループの所属有無は関係ありません。

なお，グループに所属している場合，接続数の自由利用枠（20）が未使用なら，設定した最大同時接続数までは接続できます。例えば，クライアントグループ 1 は最大同時接続数が 30 のため，接続保証数（15）とは別に，追加で 15 まで接続できます。

また，グループに所属していない場合は，接続数の自由利用枠（20）が未使用なら，接続数の自由利用枠までは接続できます。

メモ

最大同時接続数の残り枠が少なくなったときに，警告メッセージKFAA40020-W を出力する設定がグループごとにできます。

(2)　グループごとの処理リアルスレッド数の設定

クライアントグループ機能で設定できる処理リアルスレッド数の種別を，次に示します。

- **使用できる処理リアルスレッドの最大数**

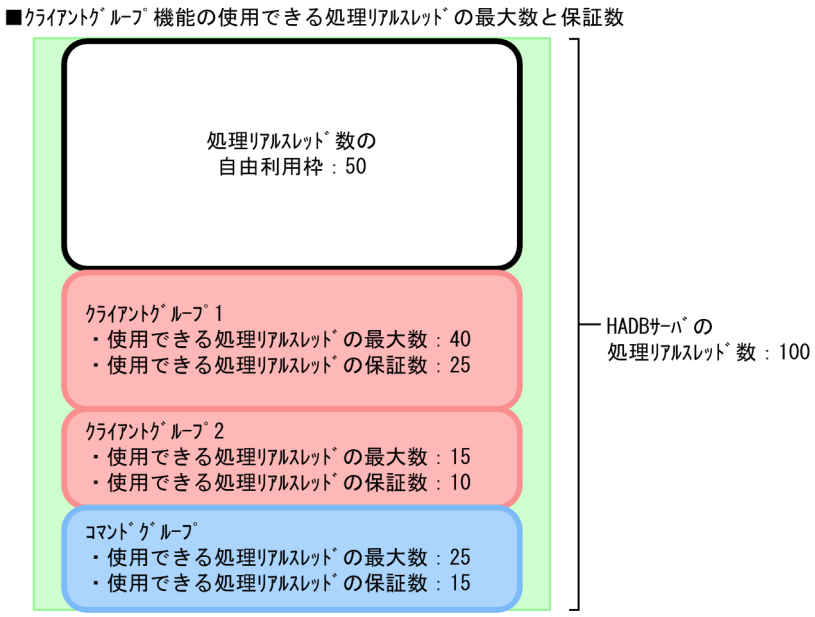
  対象のグループが使用できる処理リアルスレッド数の上限値を設定できます。設定した処理リアルスレッド数の範囲内で，対象のグループに所属している HADB クライアントやコマンドは処理を行います。

- 使用できる処理リアルスレッドの保証数

  対象のグループが最低限使用できる処理リアルスレッド数を設定できます。対象のグループは，設定した使用できる処理リアルスレッドの保証数までは，常に処理リアルスレッドを使用できます。自グループ以外の HADB クライアントやコマンドの使用状況に影響されません。

クライアントグループ機能の使用できる処理リアルスレッドの最大数と保証数の概要を，次の図に示します。

図 2-38　クライアントグループ機能の使用できる処理リアルスレッドの最大数と保証数の概要



[説明]

対象のグループは，設定した使用できる処理リアルスレッドの保証数までは，常に処理リアルスレッドを使用できます。例えば，クライアントグループ 1 は，使用できる処理リアルスレッドの保証数（25）までは常に使用できます。

HADB サーバの処理リアルスレッド数（100）から，各グループで設定した使用できる処理リアルスレッドの保証数の合計（50）を除いた数が，処理リアルスレッド数の自由利用枠（50）になります。

処理リアルスレッド数の自由利用枠の範囲内で，すべての HADB クライアントやコマンドは処理リアルスレッドを使用できます。グループの所属有無は関係ありません。

なお，グループに所属している場合，処理リアルスレッド数の自由利用枠（50）が未使用なら，設定した使用できる処理リアルスレッドの最大数までは使用できます。例えば，クライアントグループ 1 は使用できる処理リアルスレッドの最大数が 40 のため，使用できる処理リアルスレッドの保証数（25）とは別に，追加で 15 の処理リアルスレッドを使用できます。

また，グループに所属していない場合は，処理リアルスレッド数の自由利用枠（50）が未使用なら，処理リアルスレッド数の自由利用枠までは使用できます。

2.13　クライアント定義の集中管理機能

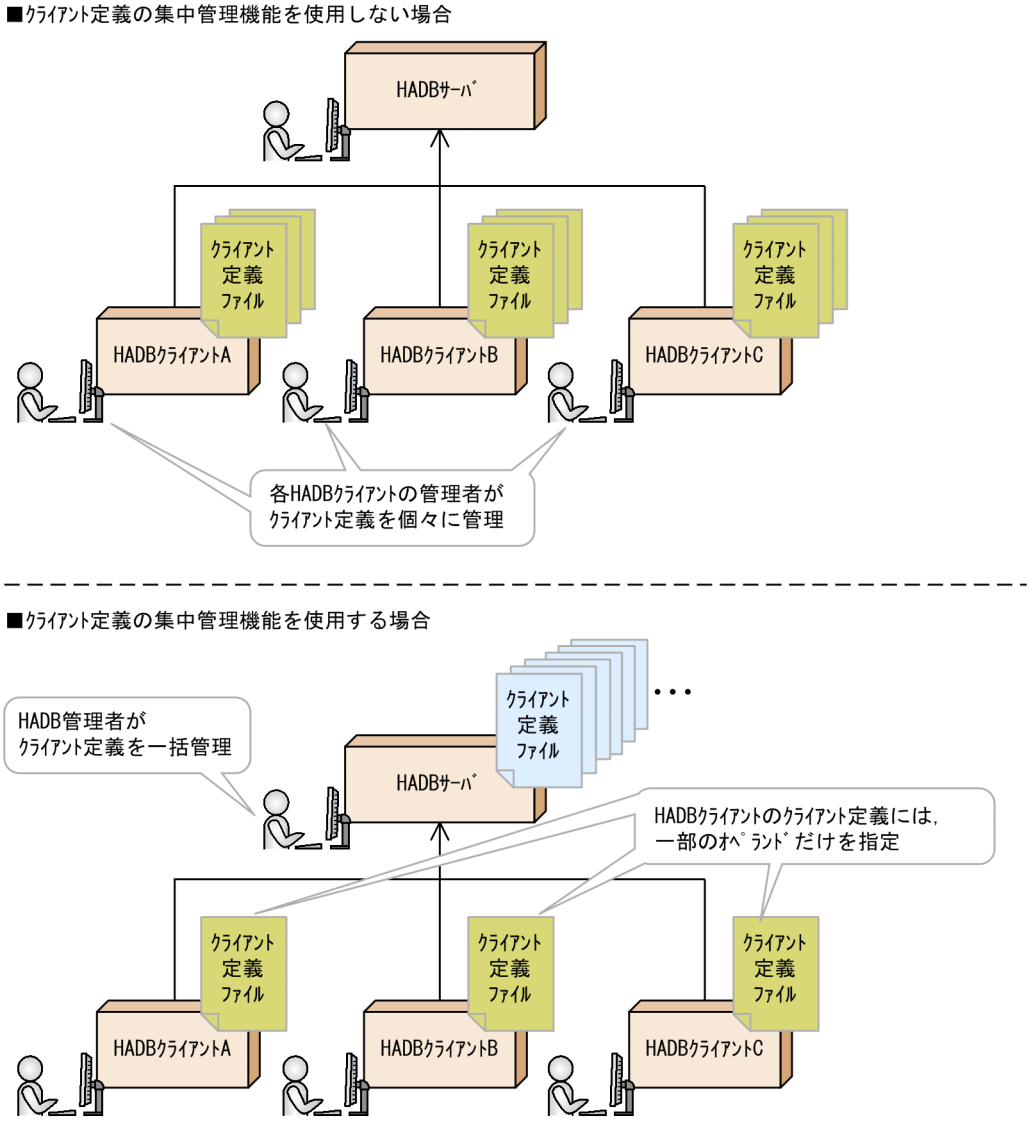
ここでは，クライアント定義の集中管理機能について説明します。

2.13.1　クライアント定義の集中管理機能とは

各 HADB クライアントで個々に管理しているクライアント定義を，HADB サーバ上で一括管理できます。この機能をクライアント定義の集中管理機能といいます。この機能を使用すると，HADB 管理者がクライアント定義を HADB サーバ上で一括管理できるため，各 HADB クライアントの管理者がクライアント定義を個々に管理する必要がなくなります。

クライアント定義の集中管理機能を使用する場合と使用しない場合の比較を次の図に示します。

図 2-39　クライアント定義の集中管理機能を使用する場合と使用しない場合の比較



［説明］

クライアント定義の集中管理機能を使用すると，HADB 管理者が，全 HADB クライアントのクライアント定義を HADB サーバ上で追加，修正できます。

ただし，次に示すオペランドについては，HADB クライアントのクライアント定義で指定する必要があります。

- adb_clt_rpc_srv_host
- adb_clt_rpc_srv_port
- adb_clt_rpc_con_wait_time
- adb_clt_rpc_sql_wait_time

> **メモ**
>
> 上記のオペランドには，HADB サーバへの接続時に必要となる情報を指定しています。そのため，HADB クライアントのクライアント定義に指定する必要があります。

■クライアント定義の集中管理機能を使用するメリット

- 各クライアント定義のオペランドの指定値を，HADB 管理者が管理できます。例えば，adb_sql_exe_max_rthd_num オペランド（SQL 文を処理するリアルスレッド数の最大値を指定するオペランド）などの性能に関するオペランドの指定値を，HADB 管理者がチューニング結果を基に変更できます。
- クライアントグループ機能を新規に使用する場合，該当する各 HADB クライアントのクライアント定義にadb_clt_group_name オペランド（クライアントグループ名を指定するオペランド）を追加する必要があります。この場合，各 HADB クライアントの管理者が，adb_clt_group_name オペランドを追加する必要があります。クライアント定義の集中管理機能を使用すると，HADB サーバ上のクライアント定義にadb_clt_group_name オペランドを追加すればよいため，各 HADB クライアントのクライアント定義にadb_clt_group_name オペランドを追加する必要がありません。

このようにクライアント定義の集中管理機能を使用すると，クライアント定義をより簡単に，より速く変更できます。

2.13.2　クライアント定義の集中管理機能を使用する場合に作成するファイル

クライアント定義の集中管理機能を使用する場合，HADB サーバ上に次に示すファイルを作成します。

- クライアント管理定義ファイル
- クライアント定義ファイル

(1)　クライアント管理定義ファイルに指定する情報

クライアント管理定義ファイルには，HADB サーバに接続する認可識別子と，その認可識別子で接続した AP（adbsql コマンドを含む）に適用するクライアント定義ファイルを指定します。指定例を次に示します。

指定例

```
adbclientmang -f client01.def -i ADBUSER01,ADBUSER02             ...1
adbclientmang -f client02.def -i ADBUSER03                       ...2
adbclientmang -f client03.def -i ADBUSER04,ADBUSER05,ADBUSER06   ...3
      :               :              :
```

[説明]

1. 認可識別子ADBUSER01 またはADBUSER02 で接続した AP には，クライアント定義ファイルclient01.def のクライアント定義を適用します。
2. 認可識別子ADBUSER03 で接続した AP には，クライアント定義ファイルclient02.def のクライアント定義を適用します。
3. 認可識別子ADBUSER04，ADBUSER05，またはADBUSER06 で接続した AP には，クライアント定義ファイルclient03.def のクライアント定義を適用します。

クライアント管理定義ファイルは，HADB サーバの$ADBDIR/conf ディレクトリ下に格納します。

(2) クライアント定義ファイルに指定する情報

クライアント定義ファイルには，クライアント定義を指定します。クライアント定義ファイルは，HADB サーバの$ADBDIR/conf ディレクトリ下に格納します。

なお，次に示すオペランドは，HADB サーバのクライアント定義に指定しても無視されるため，指定する必要はありません。次に示すオペランドは，HADB クライアントのクライアント定義に指定します。

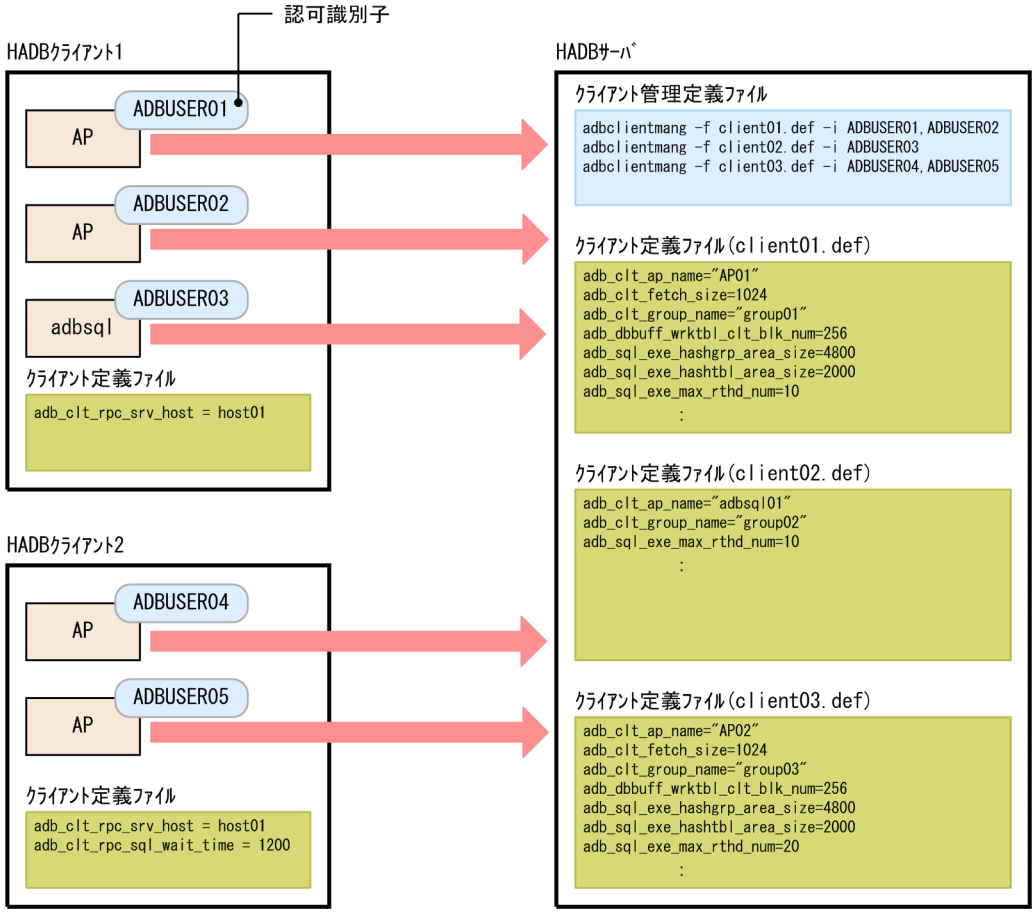
- adb_clt_rpc_srv_host
- adb_clt_rpc_srv_port
- adb_clt_rpc_con_wait_time
- adb_clt_rpc_sql_wait_time

上記以外のオペランドを省略した場合，そのオペランドの省略値が仮定されます。

2.13.3 クライアント定義の集中管理機能の適用例

クライアント定義の集中管理機能の適用例を次の図に示します。

図 2-40　クライアント定義の集中管理機能の適用例



[説明]

HADB サーバに接続した AP またはadbsql コマンドは，指定された認可識別子によって適用されるクライアント定義が決まります。上記の例では，認可識別子ADBUSER01 の場合は，client01.def に指定されているクライアント定義が適用されます。

■留意事項

- クライアント管理定義ファイルに指定されていない認可識別子を使用して HADB サーバに接続した場合（上記の例ではADBUSER06 など），その AP またはadbsql コマンドについては，HADB クライアントで指定されたクライアント定義が適用されます。
- クライアント定義の集中管理機能は，次のうちのどちらかを実行する場合にだけ適用されます(adbsql コマンド以外のコマンドを実行する場合は，機能は適用されません)。

  ・AP

  ・adbsql コマンド

2.14　バックグラウンドインポート機能

ここでは，バックグラウンドインポート機能について説明します。

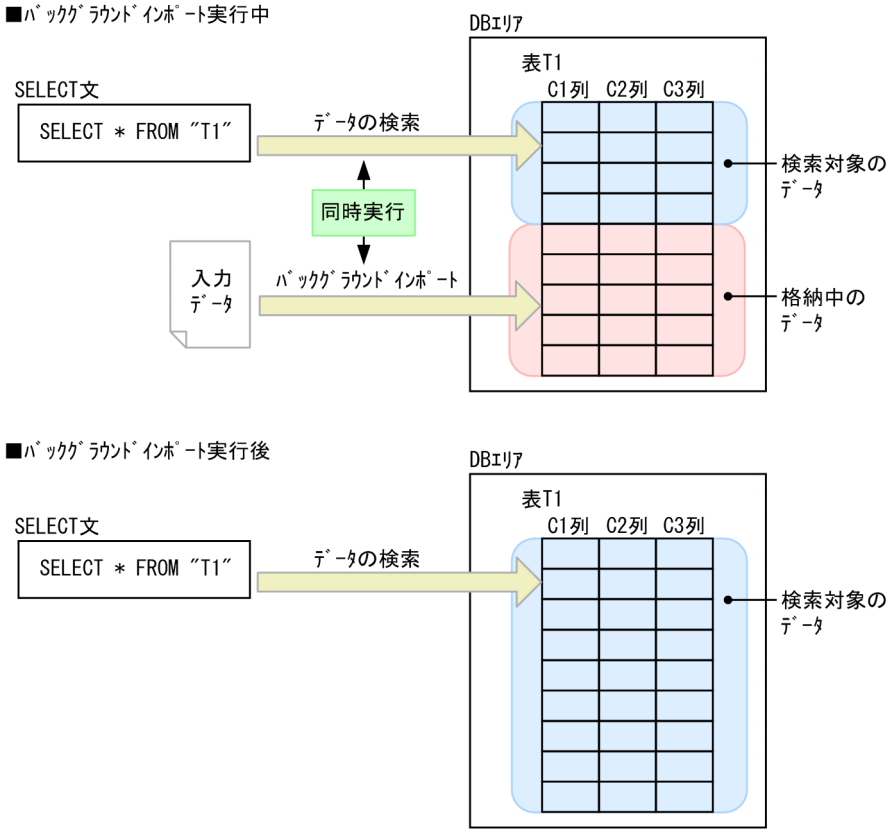
2.14.1　バックグラウンドインポートとは

HADB では，データ検索とデータインポートの同時実行を実現するために，**バックグラウンドインポート機能**をサポートしています。

バックグラウンドインポート機能を適用してデータインポートを実行することで，同一表に対してデータの検索処理とデータの格納処理を同時に実行できるようになります。

バックグラウンドインポートの概要について，次の図に示します。

図 2-41　バックグラウンドインポートの概要



[説明]

表T1 に格納されたデータの検索処理と，表T1 に対するバックグラウンドインポートによるデータの格納処理を，同時に実行できます。

なお，バックグラウンドインポート中のデータは，データの格納処理が完了したあとで検索できるようになります。

メモ

データの検索と，大量データの格納を同時実行したい場合に，バックグラウンドインポート機能を使用することを推奨します。データの検索と，少量データの格納を同時実行したい場合は，データの格納効率を低下させないために，バックグラウンドインポート機能ではなく，SQL 文（INSERT 文）を使用することを検討してください。詳細については，「5.2.4　マルチチャンク表を定義する場合の考慮点」を参照してください。

2.14.2　データインポート単位のデータ管理（チャンク）

HADB は，バックグラウンドインポートを実現するために，1 回のバックグラウンドインポートで実表に格納したデータを，1 つの固まりとして管理します。この固まりを，**チャンク**といいます。

(1)　チャンクの概要

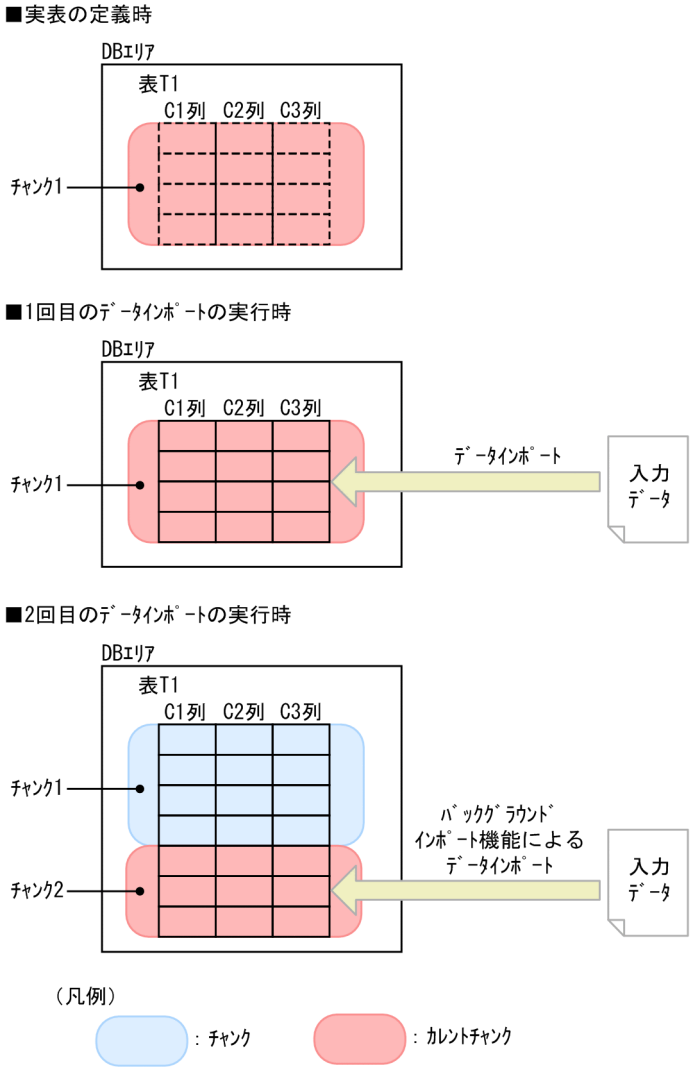
HADB は，複数のチャンクを管理することができ，各チャンクには実表内で一意となる番号が付与されます。この番号を，**チャンク ID** といいます。

バックグラウンドインポートを実行するたびに，チャンクが新規に作成されます。作成されたチャンクのうち，データが追加されるチャンクを**カレントチャンク**といいます。カレントチャンクは実表内に 1 つだけ存在します。

バックグラウンドインポートの実行などで，新たなチャンクが作成されると，カレントチャンクは変更されます（データが追加されるチャンクが変わります）。カレントチャンクが変更されることを，**カレントチャンクの切り替え**といいます。

バックグラウンドインポートとカレントチャンクの関係を次の図に示します。

図 2-42　バックグラウンドインポートとカレントチャンクの関係



[説明]

実表として表T1 を新規に定義すると，カレントチャンクとして空のチャンク（チャンク 1）が作成されます。

表T1 に対して 1 回目のデータインポートを実行すると，データが格納されます。格納されたデータの固まりは，チャンク 1 として管理されます。

表T1 に対して，2 回目のデータインポートをバックグラウンドインポートで実行すると，新たなチャンク（チャンク 2）が作成されて，データが格納されます。格納されたデータの固まりは，チャンク 2 として管理されます。このとき，カレントチャンクは，チャンク 1 からチャンク 2 に変更されます。

メモ

■カレントチャンクの切り替えが発生するタイミング

次に示す場合に，カレントチャンクが変更されます。

・バックグラウンドインポートを実行した場合

・カレントチャンクを含む複数のチャンクをマージした場合

また，次に示す場合に，カレントチャンクが変更されることがあります。

・チャンクの状態を変更した場合

■行の挿入や更新と，チャンクの関係

バックグラウンドインポートでデータを格納した実表に対して，INSERT 文で行を挿入すると，挿入した行はカレントチャンクに格納されます。一方，UPDATE 文で行を更新すると，更新後の行は更新対象の行があるチャンクに格納されます。

■インデクスが定義されている場合のチャンクの作成とチャンク ID

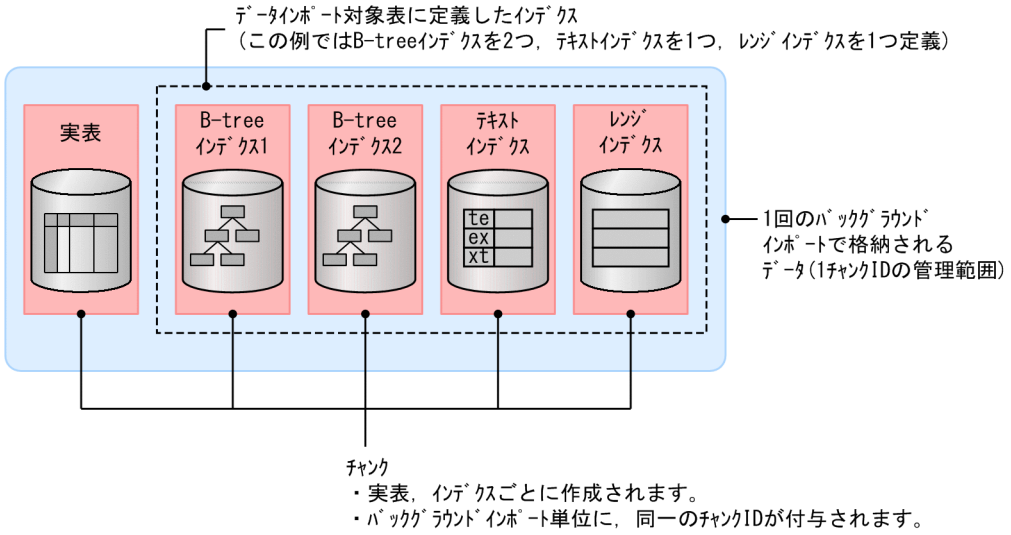
データインポート対象の実表にインデクスが定義されている場合，バックグラウンドインポート時に作成されるインデクスのデータも，チャンク単位に管理されます。

例えば，データインポート対象の実表にインデクス（B-tree インデクス，テキストインデクスおよびレンジインデクス）が定義されている場合，実表のデータだけでなく，バックグラウンドインポート時に作成されるインデクスのデータ（B-tree インデクスのデータ，テキストインデクスのデータおよびレンジインデクスのデータ）についても，それぞれチャンクが作成されます。

HADB では，バックグラウンドインポートした単位でチャンクを管理します。そのため，上記のチャンクに同一のチャンク ID が付与されます。

インデクスが定義されている場合のチャンクとチャンク ID の概要を次の図に示します。

図 2-43　インデクスが定義されている場合のチャンクの作成とチャンク ID の概要



■作成されるチャンクの数とインデクス数の関係

データインポート対象の実表にインデクスを定義している場合，1 回のバックグラウンドインポートで作成されるチャンクの数は，インポート対象の実表，および実表に定義されたインデクスの数の合計になります。

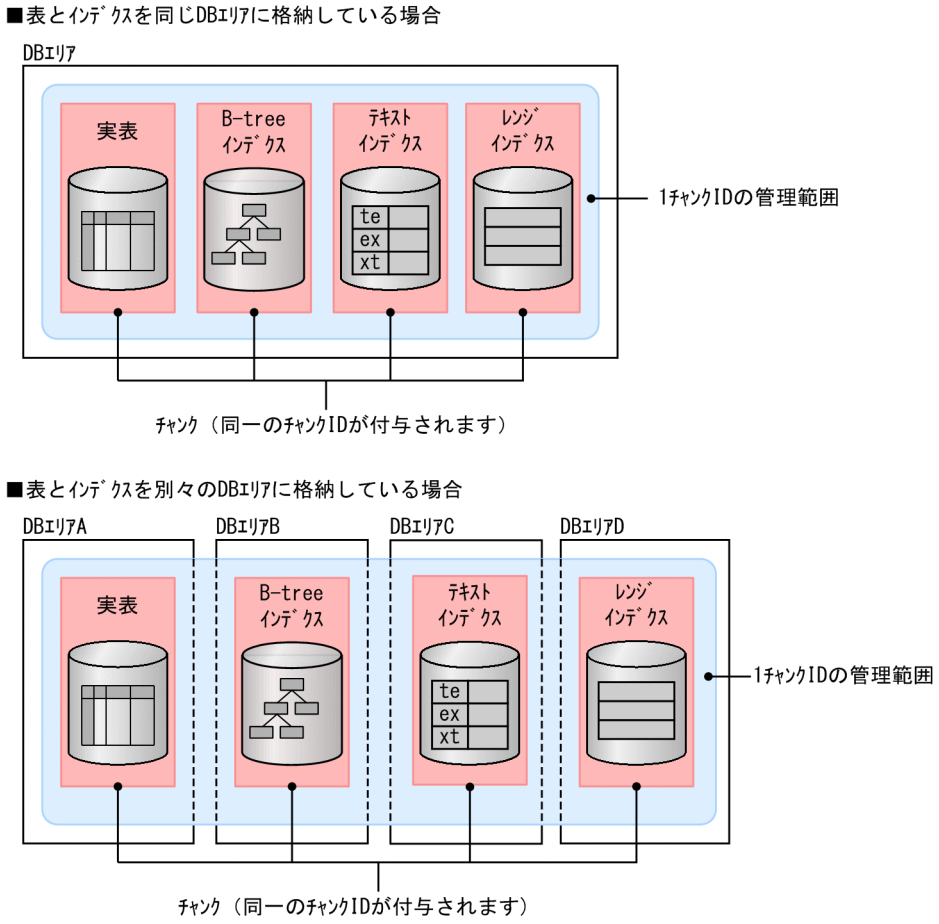
例えば，実表に B-tree インデクスを 2 つ，テキストインデクスを 1 つ，およびレンジインデクスを 1 つ定義している場合，1 回のバックグラウンドインポートで作成されるチャンク数は 5 個になります。

HADB では，1 つの DB エリア，および 1 つの実表で管理できるチャンク数には上限があります。1DB エリア中のチャンク数，および 1 つの実表に作成できるチャンク数の上限については，「付録 D.1　システム構成に関する最大値と最小値」を参照してください。

■チャンクと，DB エリアの関係

表とインデクスを同じ DB エリアに格納しても，別々の DB エリアに格納しても，バックグラウンドインポートは実行できます。どちらの場合でも，実表，およびインデクスごとにチャンクが作成されて，1 つのチャンク ID が付与されます。チャンクと DB エリアの関係を，次の図に示します。

図 2-44　チャンクと DB エリアの関係



(2)　チャンクのマージ

バックグラウンドインポートを実行して作成された複数のチャンクは，1 つの新規のチャンクにマージできます。チャンクをマージすると，使用しているチャンク数を減らすことができます。

なお，マージ対象のチャンクに対して検索を実行しているユーザが存在しても，チャンクをマージできます。

重要

■チャンクのマージと，B-tree インデクスやテキストインデクスを使用した検索性能の関係

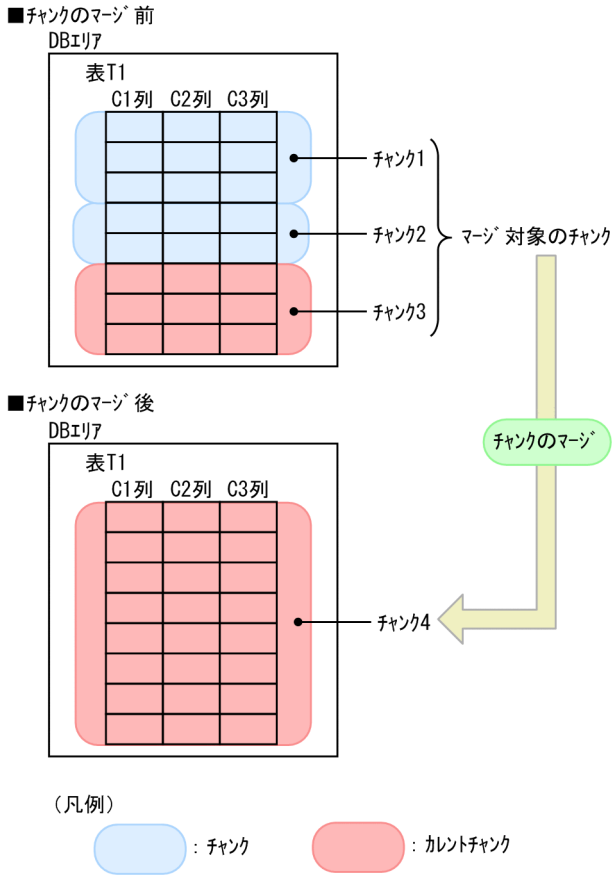
バックグラウンドインポートを繰り返すことによってチャンク数が増加すると，B-tree インデクスやテキストインデクスを使用した検索時に性能が低下するおそれがあります。チャンクをマージしてチャンク数を減らすと，検索性能の低下を回避できます。

■チャンクのマージと，作成できるチャンク数の関係

HADB では，1 つの DB エリア，および 1 つの実表で管理できるチャンク数に上限があります。チャンクをマージしてチャンク数を減らすと，この上限に達することを回避できます。1DB エリア中のチャンク数，および 1 つの実表に作成できるチャンク数の上限については，「付録 D.1　システム構成に関する最大値と最小値」を参照してください。

複数のチャンクをマージする例を次の図に示します。

図 2-45　複数のチャンクをマージする例



[説明]

カレントチャンクを含む 3 つのチャンク（チャンク 1 からチャンク 3 まで）をマージします。マージすると，3 つのチャンクは，1 つの新規のチャンク（チャンク 4）として管理されます。このとき，カレントチャンクは，チャンク 3 からチャンク 4 に変更されます。

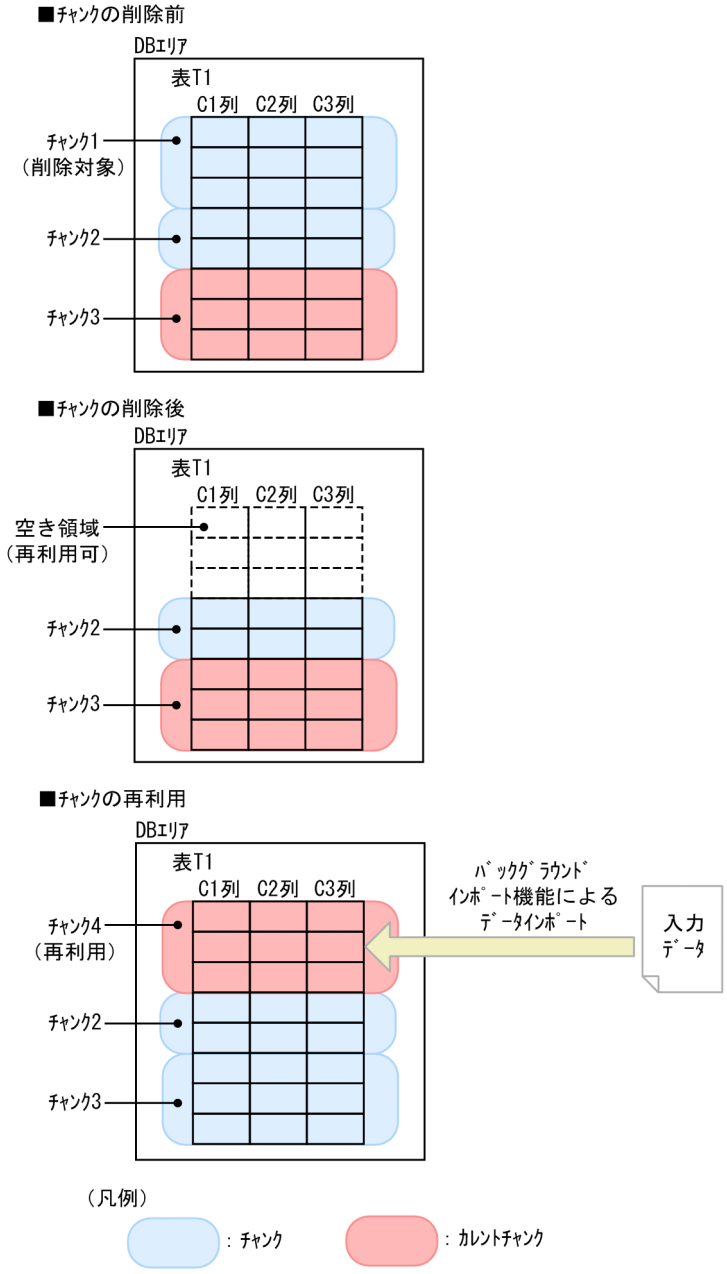
> **メモ**
>
> マージ対象のチャンクに，カレントチャンクを含まない場合は，カレントチャンクは変更されません。

(3) チャンクの削除

実表に格納されたデータは，チャンク単位で削除できます。チャンクを削除すると，行の削除とは異なり，データの格納領域を再利用できます。ただし，カレントチャンクは削除できません。

チャンクの削除と格納領域の再利用の例について次の図に示します。

図 2-46 チャンクの削除と格納領域の再利用の例



［説明］

チャンク 1 を削除した場合，チャンク 1 が使用していた領域を再利用できます。チャンク 1 の削除後に，表T1 に対してバックグラウンドインポートを実行すると，チャンク 1 が使用していた領域に対して，新たなチャンク（チャンク 4）が作成されてデータが格納されます。

(4) チャンクの状態変更

チャンクには，幾つかの状態があります。

バックグラウンドインポートを実行するときに，作成されるチャンクの状態を指定できます。チャンクの状態はバックグラウンドインポート後に変更することもできます。

HADB サーバが管理するチャンクの状態を次に示します。

表 2-19 チャンクの状態の一覧

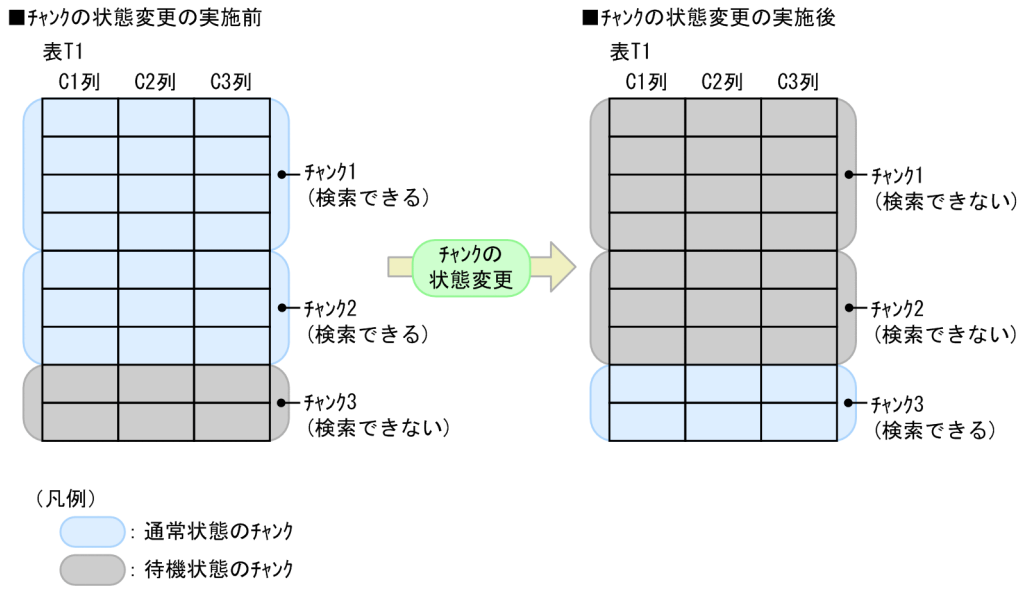
項番	チャンクの状態	説明
1	通常状態	通常状態のチャンクとは，チャンクに含まれるデータが操作系 SQL の対象となるチャンクのことです。そのため，通常状態のチャンクに含まれるデータは検索できます。 通常状態のチャンクは，バックグラウンドインポートを実行すると作成されます。 また，通常状態のチャンクは，待機状態のチャンクに変更できます。
2	待機状態	待機状態のチャンクとは，チャンクに含まれるデータが操作系 SQL の対象外となるチャンクのことです。そのため，待機状態のチャンクに含まれるデータは検索できません。ただし，PURGE CHUNK 文とTRUNCATE TABLE 文によるチャンクの削除はできます。 待機状態のチャンクは，バックグラウンドインポートを実行するときに，作成されるチャンクの状態を指定することで作成されます。 また，待機状態のチャンクは，通常状態のチャンクに変更できます。 なお，待機状態のチャンクは，カレントチャンクにはなりません。 待機状態のチャンクを利用すると，次のような運用ができます。 • バックグラウンドインポートで待機状態のチャンクにデータを格納しておき，任意のタイミングで通常状態のチャンクに変更して，検索できるようにする • 通常状態のチャンクを，任意のタイミングで待機状態のチャンクに変更することで，通常状態のチャンクに格納していたデータを検索できないようにする
3	削除仕掛中	削除仕掛中のチャンクとは，チャンクに含まれるデータが操作系 SQL の対象外となるチャンクのことです。そのため，削除仕掛中のチャンクに含まれるデータは検索できません。ただし，PURGE CHUNK 文とTRUNCATE TABLE 文によるチャンクの削除はできます。 削除仕掛中のチャンクは，チャンクのマージ処理が中断した場合に，削除されないで残ってしまった不要なチャンク（マージ元チャンク）のことです。 また，削除仕掛中のチャンクは，状態を変更できません。 待機状態のチャンクとは異なり，削除仕掛中のチャンクは不要であるため，手動で削除する必要があります。

注

チャンクの状態によって，チャンクに含まれるデータに対して，実行できる SQL 文とコマンドが異なります。詳細については，「11.4.12　チャンクの状態を変更する方法」の「(2)　チャンクの状態と実行できる SQL 文・コマンドの関係」を参照してください。

バックグラウンドインポートで作成された複数のチャンクの状態を変更する例を，次の図に示します。

図 2-47　チャンクの状態を変更する例



[説明]

通常状態のチャンク（チャンク 1 と 2）を，待機状態のチャンクに変更します。待機状態のチャンクに変更すると，チャンク 1 と 2 に格納されているデータは検索できなくなります。

また，待機状態のチャンク（チャンク 3）を，通常状態のチャンクに変更します。通常状態のチャンクに変更すると，チャンク 3 に格納されているデータを検索できるようになります。

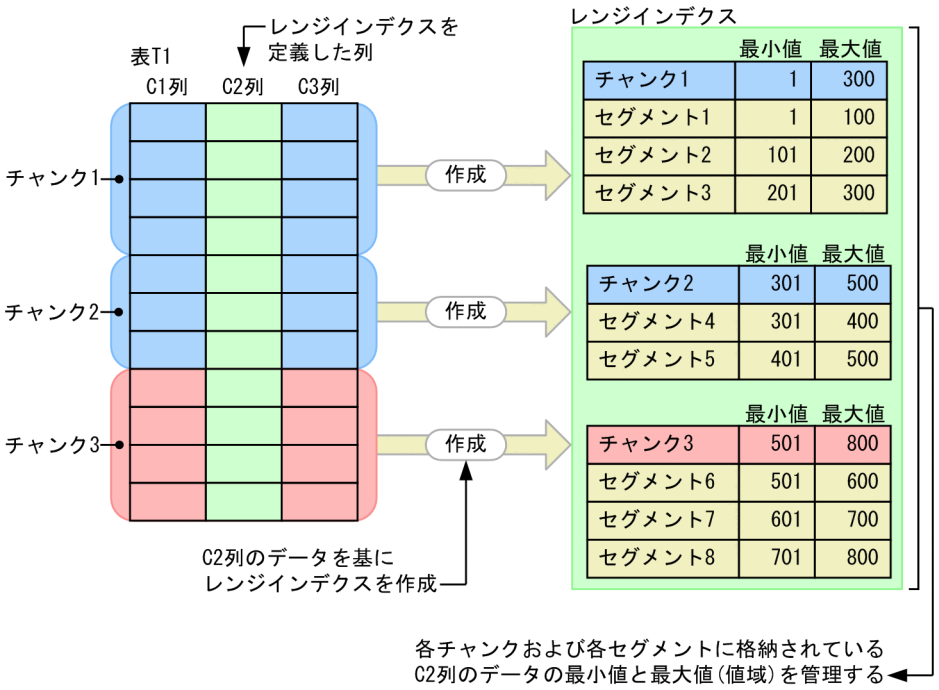
2.14.3　チャンクとレンジインデクスの関係

ここでは，チャンクとレンジインデクスの関係について説明します。レンジインデクスの詳細については，「2.3.3　レンジインデクス」を参照してください。

レンジインデクスは，実表のセグメント内に格納されているデータの値域（列の値の最小値と最大値）だけでなく，チャンク内に格納されているデータの値域も管理しています。そのため，レンジインデクスを使用すると，探索条件を満たさない値域のデータを格納しているチャンクおよびセグメントの検索処理をスキップでき，その分検索性能が向上します。

レンジインデクスと，チャンクおよびセグメントの概要を次の図に示します。

図 2-48　レンジインデクスと，チャンクおよびセグメントの概要



[説明]

表T1 のC2 列にレンジインデクスを定義します。表T1 に対して，バックグラウンドインポートを実行すると，表にデータが格納されます。また，C2 列のデータを基にレンジインデクスが作成されます。それらのデータの固まりは，1 つのチャンクとして管理されます。

このとき，レンジインデクスは，セグメント内の表データの値域を管理するだけでなく，チャンク内の表データの値域も管理します。

(1)　レンジインデクスを使用した検索処理（チャンクのスキップ）

レンジインデクスを使用して表を検索した場合，探索条件を満たさない値域のデータを格納しているチャンクおよびセグメントの検索処理を，次に示すように段階的にスキップできます。

■レンジインデクスを使用した検索処理

1. チャンクのスキップ
2. セグメントのスキップ

> **重要**
>
> バージョン 02-02 より前のバージョンで定義されたレンジインデクスの場合は，チャンクのスキップはできません。

それぞれの検索処理について，次に示します。

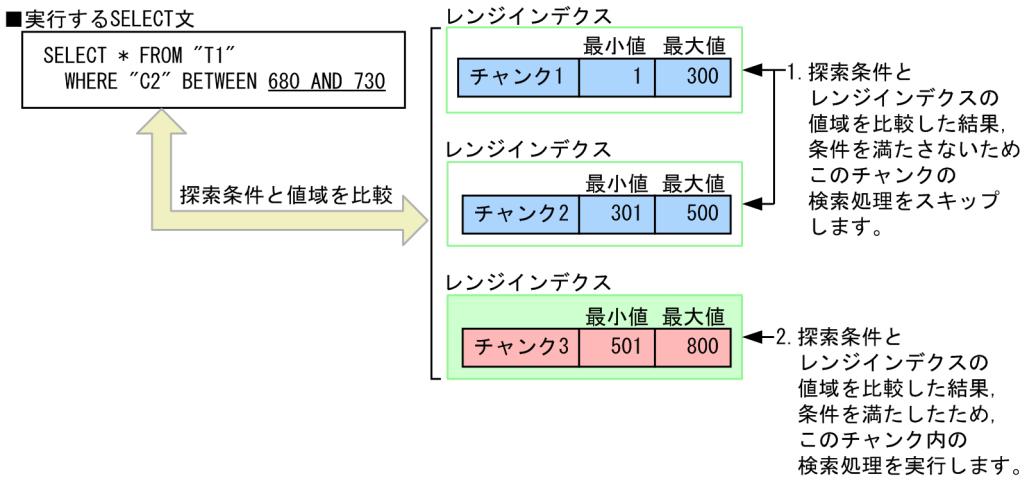
1. チャンクのスキップ

レンジインデクスを使用した表の検索では，まず，チャンク単位で検索処理をスキップできます。HADBは，探索条件を満たす値域のデータを格納しているチャンクかどうかを判定し，探索条件を満たさない値域のデータを格納しているチャンクであれば，そのチャンクを検索しません。

このとき，表のデータ，B-tree インデクスのデータおよびテキストインデクスのデータにはアクセスしないため，検索性能が向上します。

チャンクのスキップの概要を，次の図に示します。

図 2-49　チャンクのスキップの概要

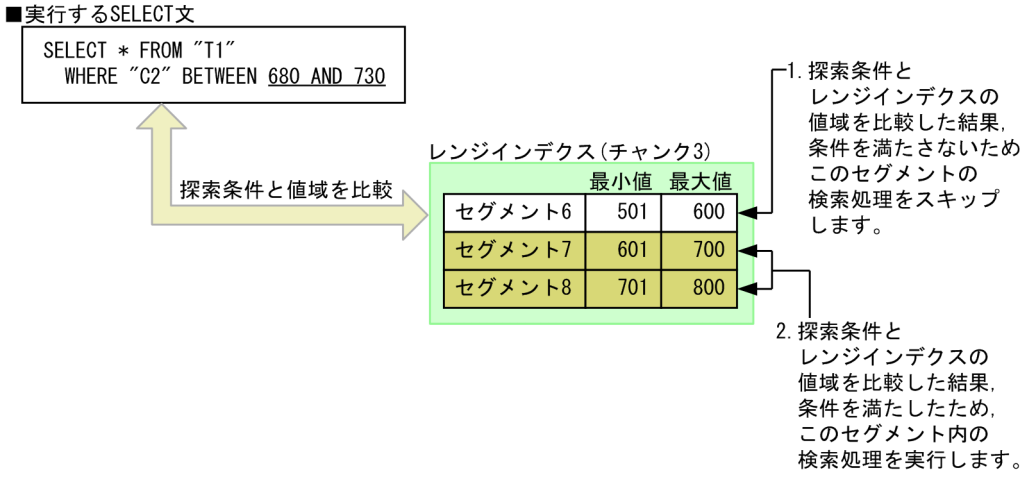


2. セグメントのスキップ

探索条件とレンジインデクスの値域を比較した結果，チャンク３が条件を満たしている場合，次はチャンク３内の表のセグメントに対して検索処理を実行します。その際，「2.3.3　レンジインデクス」と同様に，探索条件を満たさない値域のデータを格納しているセグメントの検索処理をスキップできます。

セグメントのスキップの概要を，次の図に示します。

図 2-50　セグメントのスキップの概要



(2)　レンジインデクスの特徴（チャンクのスキップ）

チャンクのスキップができるレンジインデクスと，「2.3.3　レンジインデクス」の「(2)　レンジインデクスの特徴」で説明している４つの特徴との違いについて説明します。

(a)　B-tree インデクスやテキストインデクスとレンジインデクスを合わせて使用できる

チャンクのスキップができるレンジインデクスは，B-tree インデクスやテキストインデクスと合わせて使用できます。

B-tree インデクスやテキストインデクスと合わせて使用したときの検索処理を，次に示します。

■チャンクのスキップができるレンジインデクスと B-tree インデクスを使用した検索処理

1. レンジインデクスを使用した検索（チャンクのスキップ）

   まず，レンジインデクスを使用した検索処理が行われます。探索条件を満たす値域のデータを格納しているチャンクかどうかを判定し，探索条件を満たさない値域のデータを格納しているチャンクであれば，そのチャンクの検索をスキップします。チャンクのスキップによって，探索条件を満たすチャンクを絞り込みます。

2. B-tree インデクスを使用した検索

   探索条件を満たすチャンクを絞り込んだら，次に対象のチャンク内の B-tree インデクスを使用した検索処理が行われます。

■チャンクのスキップができるレンジインデクスとテキストインデクスを使用した検索処理

1. レンジインデクスを使用した検索（チャンクのスキップ）

   まず，レンジインデクスを使用した検索処理が行われます。探索条件を満たす値域のデータを格納しているチャンクかどうかを判定し，探索条件を満たさない値域のデータを格納しているチャンクであれば，そのチャンクの検索をスキップします。チャンクのスキップによって，探索条件を満たすチャンクを絞り込みます。

2. テキストインデクスを使用した検索

   探索条件を満たすチャンクを絞り込んだら，次に対象のチャンク内のテキストインデクスを使用した検索処理が行われます。

> **メモ**
>
> B-tree インデクスやテキストインデクスと合わせて使用する場合，レンジインデクスを使用した検索（セグメントのスキップ）は行われません。

(b)　複数のレンジインデクスを使用できる

違いはありません。「複数のレンジインデクスを使用できる」の詳細については，「2.3.3　レンジインデクス」の「(2)　レンジインデクスの特徴」の「(b)　複数のレンジインデクスを使用できる」を参照してください。

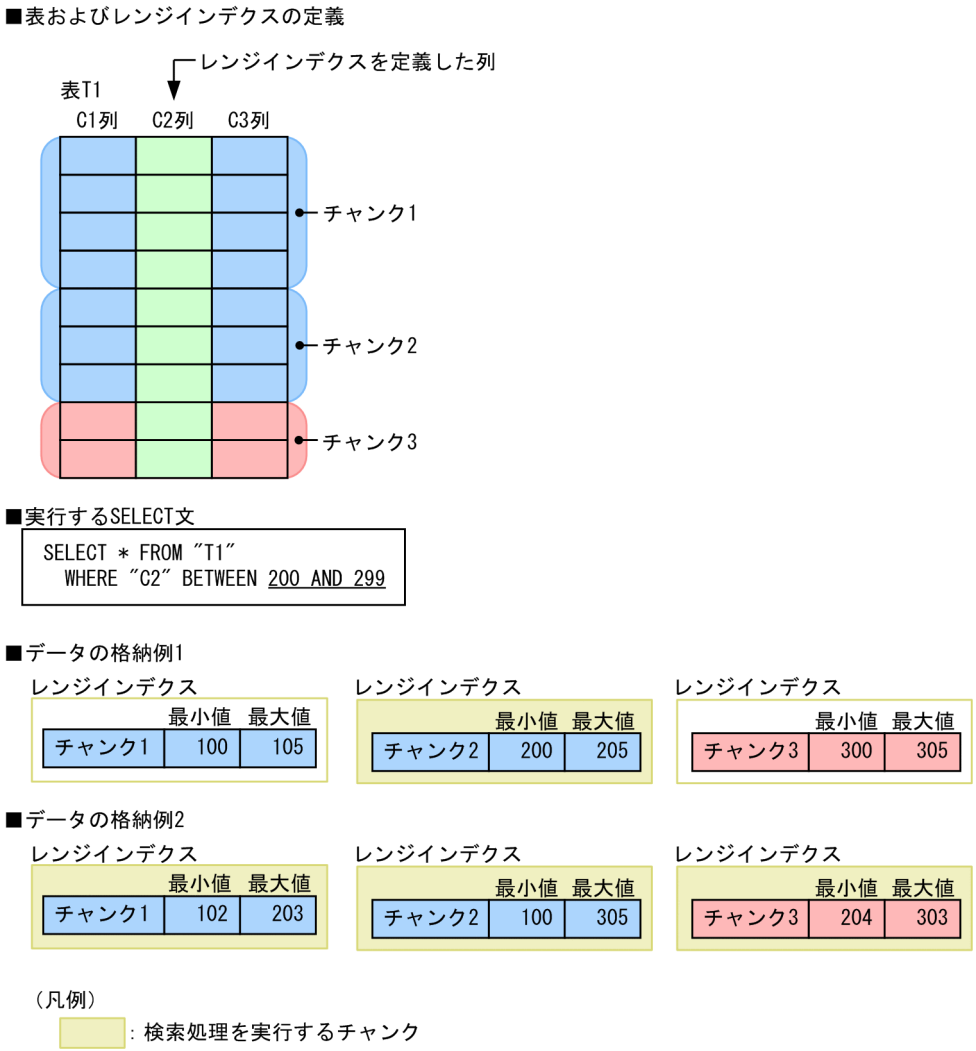
(c)　B-tree インデクスやテキストインデクスよりデータ容量が少なくなる

違いはありません。「B-tree インデクスやテキストインデクスよりデータ容量が少なくなる」の詳細については，「2.3.3　レンジインデクス」の「(2)　レンジインデクスの特徴」の「(c)　B-tree インデクスやテキストインデクスよりデータ容量が少なくなる」を参照してください。

(d)　チャンクに格納されているデータの値域の幅によって検索性能が変わる

レンジインデクスを使用した検索では，チャンク内に格納されているデータの値域の幅によって検索性能が変わります。

図 2-51　チャンク内に格納されているデータの値域の幅によって検索性能が変わる例



[説明]

データの格納例 1 の場合は，検索処理を実行するチャンク数は 1 つになります。一方で，データの格納例 2 の場合は，値域の重複があり，各チャンクの値域が格納例 1 に比べて広くなっています。そのため，検索処理を実行するチャンク数が 3 つになります。このように同じ探索条件でも，チャンクに格納されているデータの値域の幅によって，検索性能が変わります。値域を狭くすることで，検索性能が向上します。

(3)　レンジインデクスの値域の更新（チャンクとセグメント）

行の追加，更新を繰り返すと，レンジインデクスが管理するチャンクの値域およびセグメントの値域が広がり，レンジインデクスの効果が薄れていきます（値域が狭くなることはありません）。また，行の削除を

繰り返すと，チャンク内およびセグメント内に格納されているデータの値域に比べて，レンジインデクスの値域が広くなり，レンジインデクスの効果が薄れていきます。

レンジインデクスの効果が薄れてきた場合は，対象のレンジインデクスに対してインデクス再作成を実行し，レンジインデクスを作成し直してください。

レンジインデクスの値域が広がる条件を次の表に示します。

表 2-20　レンジインデクスの値域が広がる条件（チャンクとセグメント）

項番	実行した操作	レンジインデクスの値域が広がる条件
1	行の追加	値域の範囲外の値を追加した場合に，追加行を格納するチャンクおよびセグメントの値域が広がります。
2	行の更新	値域の範囲外の値に更新した場合に，更新行を格納するチャンクおよびセグメントの値域が広がります。
3	行の削除	値域は広がりません。

2.15　チャンクアーカイブ機能（チャンク内のデータの圧縮）

ここでは，チャンク内のデータを圧縮することができるチャンクアーカイブ機能について説明します。

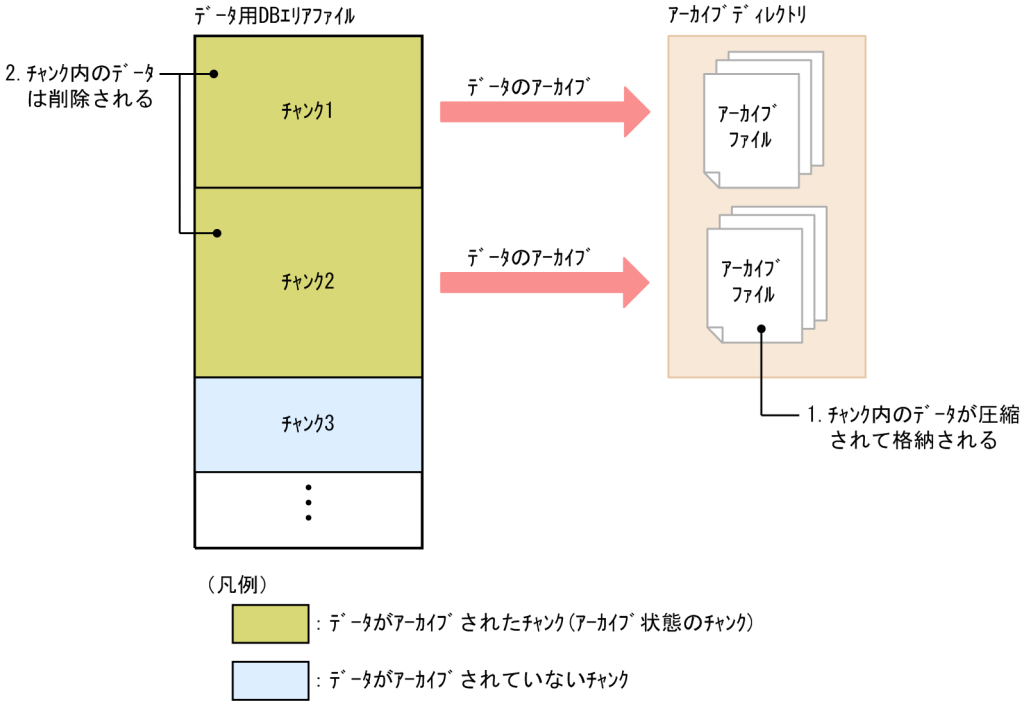
チャンクアーカイブ機能は，アーカイブマルチチャンク表に対してだけ使用できます。アーカイブマルチチャンク表については，「2.2.2　シングルチャンク表とマルチチャンク表」の「(2)　マルチチャンク表とは」を参照してください。

2.15.1　チャンクアーカイブ機能とは

センサデータや売上データのように，毎日大量に増加するデータを長期間保管する場合，ストレージの容量不足が発生したり，ストレージの追加コストが想定よりも増えたりすることがあります。このような問題がある場合，DB エリア内のデータを圧縮してファイルに出力（アーカイブ）することで，データベースの容量を削減することができます。データのアーカイブはチャンク単位で実行します。この機能をチャンクアーカイブ機能といいます。

チャンクアーカイブ機能を使用したデータのアーカイブを次の図に示します。

図 2-52　チャンクアーカイブ機能を使用したデータのアーカイブ



[説明]

1. チャンク内のデータが圧縮されてファイルに出力（アーカイブ）されます。このファイルをアーカイブファイルといいます。1 つのチャンクをアーカイブすると，1 つ以上のアーカイブファイルが作成されます。

   アーカイブファイルを格納するディレクトリをアーカイブディレクトリといいます。

アーカイブされるのは表データだけです。インデクスデータはアーカイブされないで削除されます。

2. アーカイブ対象のチャンク内のデータは削除されます。

データがアーカイブされたチャンクは，アーカイブ状態のチャンクになります。

この一連の処理をチャンクのアーカイブといいます。チャンクをアーカイブするには，adbarchivechunk コマンドを実行します。

> **メモ**
>
> チャンクアーカイブ機能を使用すると，データベースの容量を削減できますが，アーカイブされているデータを検索する場合，次に示す原因によって，アーカイブされていないデータを検索するときに比べて検索時間が長くなります。
>
> - データの伸長に時間が掛かる
> - インデクスを使用した検索ができない

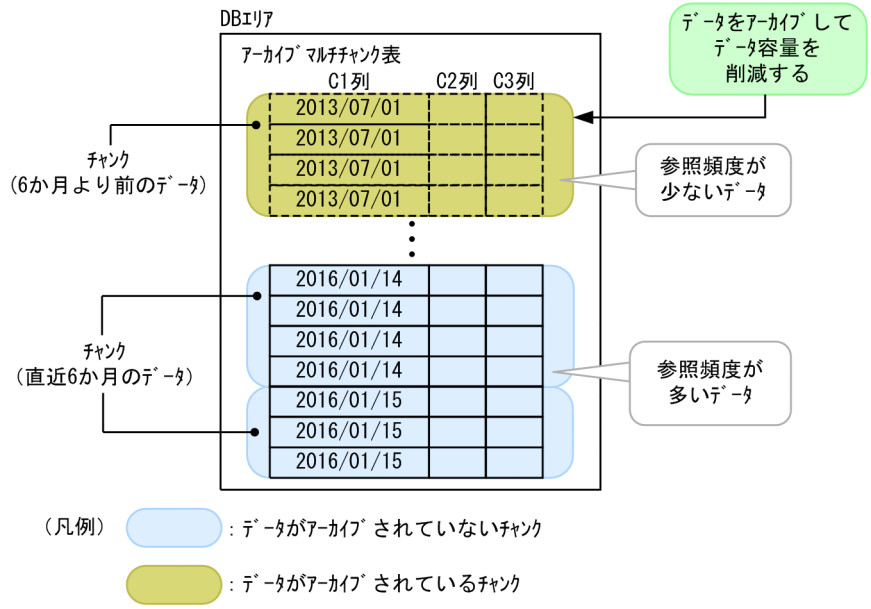
2.15.2 チャンクアーカイブ機能を適用するケース

チャンクアーカイブ機能は，次に示す条件をすべて満たす場合に適用することを推奨します。

- センサデータや売上データのような時系列データを毎日大量にインポートする
- 上記の時系列データを長期間保管する必要がある
- 分析業務の際，最近のデータは頻繁に参照するが，古いデータは参照する頻度が非常に少ない

チャンクアーカイブ機能の適用例を次の図に示します。

図 2-53 チャンクアーカイブ機能の適用例



[説明]

この例では，直近 6 か月のデータは参照頻度が多いため，データをアーカイブしていません。6 か月より前のデータは参照頻度が非常に少ないため，データをアーカイブしてデータ容量を削減しています。

アーカイブされているデータを検索する場合，アーカイブされていないデータを検索するときに比べて検索時間が長くなります。

重要

- データ中に日時を示すデータがあることが，チャンクアーカイブ機能を使用する際の前提条件です。
- アーカイブされているデータは，更新または削除できません。更新または削除するには，データをいったん伸長して DB エリアに戻す必要があります。そのため，アーカイブされているデータを更新または削除するような機会がある場合は，チャンクアーカイブ機能を使用しないことを推奨します。

2.15.3 アーカイブマルチチャンク表の検索

アーカイブマルチチャンク表を定義したときに，アーカイブレンジ列に自動的にレンジインデクスが定義されます。また，ロケーション表が自動的に作成されます。

ここでは，アーカイブレンジ列およびロケーション表について説明します。

(1) アーカイブレンジ列とは

HADB サーバは，アーカイブマルチチャンク表に格納されたデータの範囲（最大値と最小値）を，チャンク単位で管理しています。チャンク単位で管理しているデータの範囲を使用すると，検索範囲を絞り込むことができます。そのため，長期間管理する膨大なデータの中から必要なデータだけを検索できるようにしています。

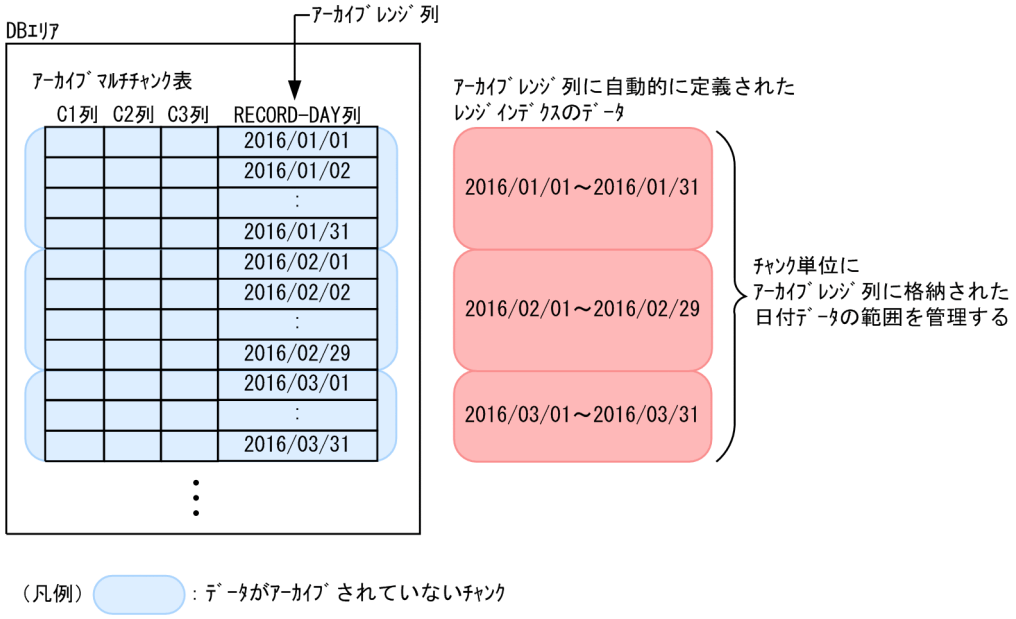
データが格納された列で，かつ検索範囲を絞り込むために使用する列を，アーカイブレンジ列といいます。

アーカイブマルチチャンク表には，アーカイブレンジ列として使用する列が必要です。日時データの列，数データの列などを，アーカイブレンジ列として使用できます。

アーカイブレンジ列には，HADB サーバが自動的にレンジインデクスを定義します。そして，アーカイブ状態ではないチャンクごとに，アーカイブレンジ列に格納されたデータの範囲（最大値と最小値）を管理します。

アーカイブレンジ列の概要を次の図に示します。

図 2-54　アーカイブレンジ列の概要（日付データの列をアーカイブレンジ列として使用する場合）



(2)　ロケーション表とは

アーカイブマルチチャンク表を定義したときに，ロケーション表とロケーション表のインデクスが HADB サーバによって自動的に定義されます。ロケーション表は，アーカイブ状態のチャンクに関する情報を管理するシステム専用の表です。ロケーション表が管理する情報を次に示します。

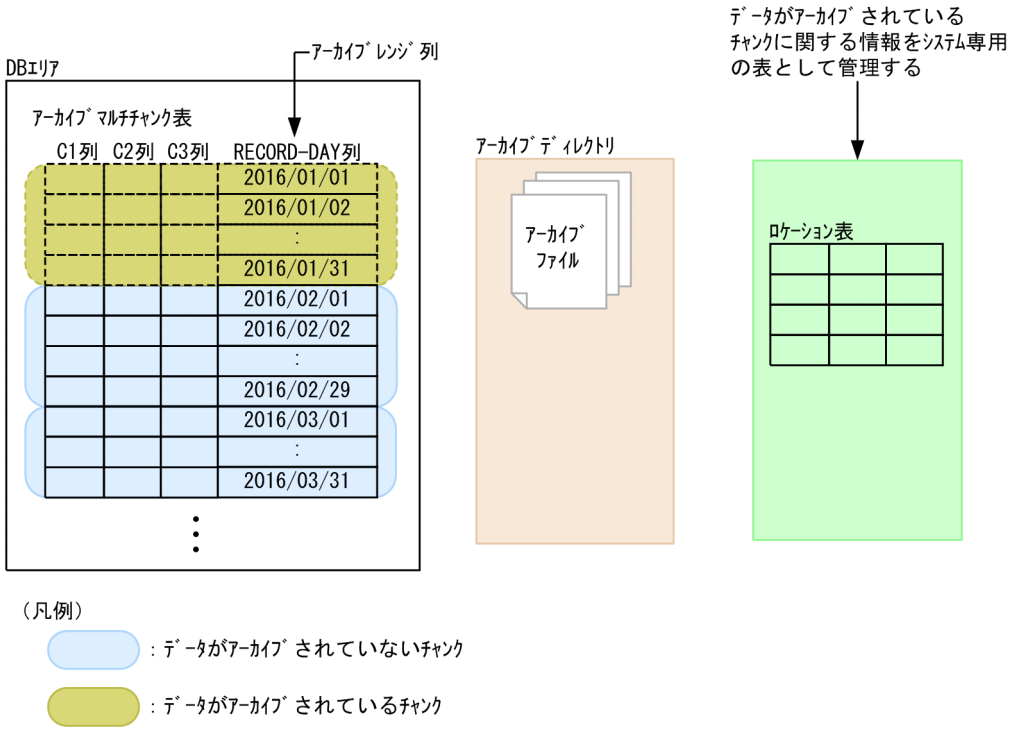
ロケーション表が管理する情報

- チャンク ID
- アーカイブファイルのファイルパス
- アーカイブファイル単位のアーカイブレンジ列の値の範囲（最大値と最小値）

アーカイブ状態のチャンクには，インデクスデータは含まれません。そのため，アーカイブ状態のチャンク内のデータを検索する場合，インデクスを使った検索はできません。その代わりに，アーカイブレンジ列に格納されたデータをロケーション表が管理することで，検索範囲の絞り込みを実現しています。

ロケーション表の概要を次の図に示します。

図 2-55　ロケーション表の概要



メモ

- ロケーション表は，HADB サーバが使用するための表で，ユーザが使用することはありません。
- 1 つのアーカイブマルチチャンク表に対して，1 つのロケーション表が作成されます。
- ロケーション表とロケーション表に定義されたインデクスは，アーカイブマルチチャンク表が格納されているデータ用 DB エリア内に格納されます。

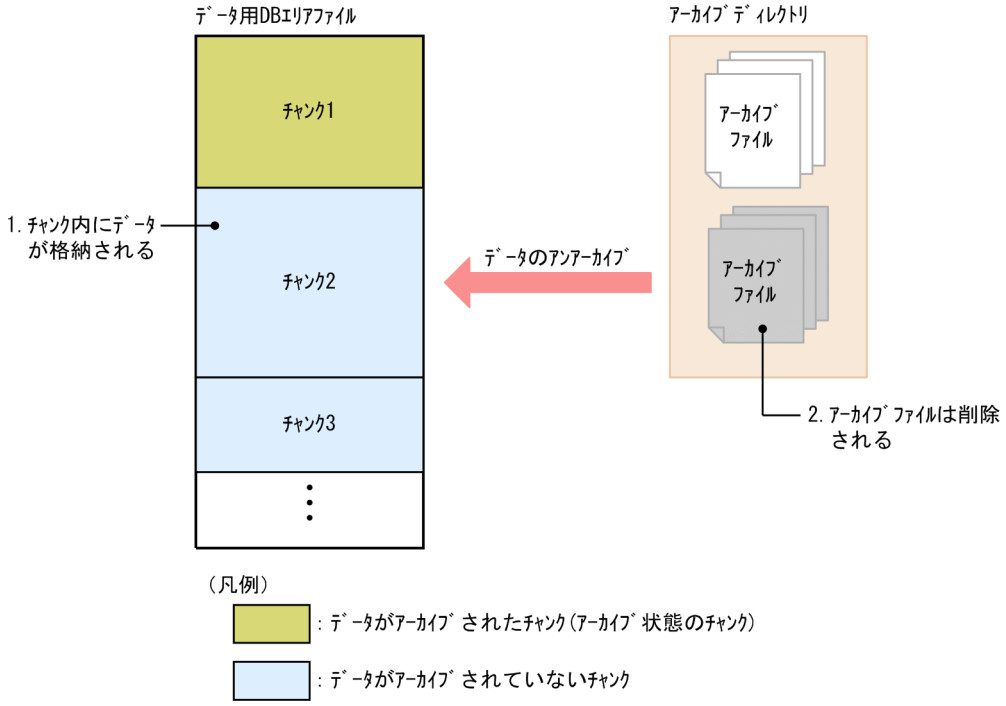
アーカイブマルチチャンク表を検索する場合，HADB サーバはアーカイブレンジ列に格納されたデータを基に，検索対象のアーカイブファイルを絞り込みます。詳細については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『アーカイブマルチチャンク表を検索する際の考慮点』を参照してください。

2.15.4　データの伸長（アーカイブ状態の解除）

アーカイブされているデータは，更新または削除することができません。そのため，アーカイブされているデータを更新または削除する必要が生じた場合は，アーカイブされているデータを伸長（アンアーカイブ）する必要があります。また，アーカイブされているデータの参照頻度が増えてきた場合も，アーカイブされているデータのアンアーカイブを検討してください。

チャンクアーカイブ機能を使用したデータのアンアーカイブを次の図に示します。

図 2-56　チャンクアーカイブ機能を使用したデータのアンアーカイブ



［説明］

上記の図では，チャンク 2 のデータをアンアーカイブしています。

1. アーカイブファイルに格納されているデータがチャンク 2 に格納されます。このとき，インデクスも作成されます。
2. チャンク内にデータが格納され，インデクスも作成されると，アーカイブファイルは削除されます。

この一連の処理をチャンクのアンアーカイブといいます。チャンクをアンアーカイブするには，adbunarchivechunk コマンドを実行します。

メモ

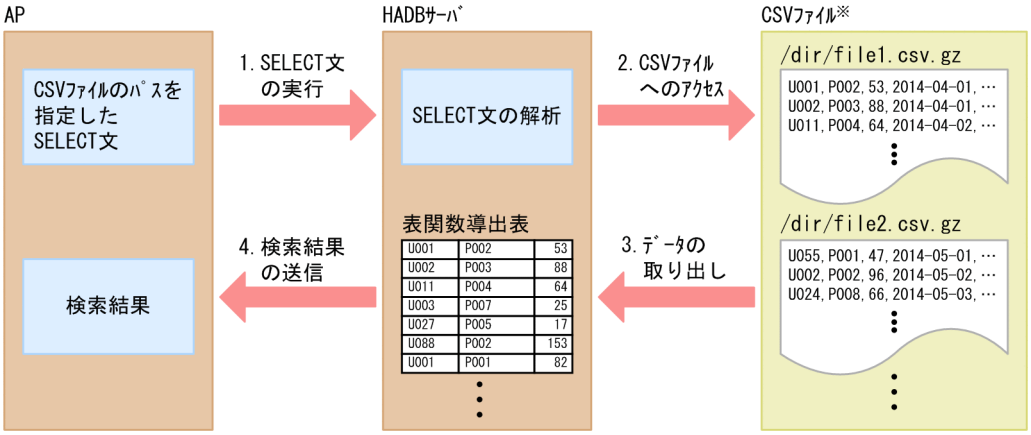
データの格納処理とインデクスの作成処理は，adbimport コマンドのデータインポート処理で実行されています。

2.16　CSV ファイル中のデータの検索

HADB では，SELECT 文を使って CSV ファイル（CSV 形式のファイル）中のデータを検索することができます。例えば，前年度の販売管理データを CSV ファイルに保管している場合，その販売管理データを表にインポートしなくても検索できます。

検索の際は，検索対象の CSV ファイルのパスを指定したSELECT 文を実行します。また，SELECT 文に探索条件を指定することで，表にインポートしたデータの検索と同様に，条件に合致したデータを検索できます。CSV ファイル中のデータ検索の概要を次の図に示します。

図 2-57　CSV ファイル中のデータ検索の概要



注※

圧縮されていない CSV ファイル，または GZIP 形式で圧縮されている CSV ファイル中のデータを検索できます。

[説明]

1. CSV ファイルのパスを指定したSELECT 文を実行します。

   CSV ファイルのパスは，HADB が提供している関数（CSV ファイル中のデータを取り出す ADB_CSVREAD 関数）中に指定します。

2. HADB サーバは，関数中に指定されたパスの CSV ファイルにアクセスします。

3. HADB サーバは，CSV ファイル中のデータを取り出し，表の形式で一時的に保存します。この一時的に保存された表形式のデータ集合を**表関数導出表**といいます。

4. HADB サーバは，SELECT 文に指定された探索条件に従って，表関数導出表を検索し，AP に検索結果を返します。

■ADB_CSVREAD 関数を指定したSELECT 文の例

GZIP 形式で圧縮した次の CSV ファイル中のデータを検索します。

- /dir/file1.csv.gz
- /dir/file2.csv.gz
- /dir/file3.csv.gz

```
SELECT * FROM TABLE(ADB_CSVREAD(
           MULTISET['/dir/file1.csv.gz','/dir/file2.csv.gz','/dir/file3.csv.gz'],
          'COMPRESSION_FORMAT=GZIP;
           FIELD_NUM=1,2;
           ENCLOSING_CHAR=";
           DELIMITER_CHAR=,;'))
           AS "T1" ("C1" INTEGER, "C2" VARCHAR(32))
```

[説明]

下線部分がADB_CSVREAD 関数の指定です。

ADB_CSVREAD 関数には，CSV ファイルの絶対パス名や，CSV ファイル中の取り出し対象のデータなどを指定します。

ADB_CSVREAD 関数については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『システム定義関数』の『ADB_CSVREAD 関数』を参照してください。

重要

- CSV ファイルは，次のどれかの条件を満たす必要があります。
  - ・圧縮されていない CSV ファイル
  - ・OS のgzip コマンドで GZIP 形式に圧縮された CSV ファイル
  - ・adbexport コマンドでエクスポートした，GZIP 形式に圧縮された CSV ファイル
- CSV ファイル中のデータを更新することはできません。
- 表にインポートされているデータの検索と比べると，CSV ファイル中のデータの検索は，CSV ファイルの解析処理が必要となる分，時間が掛かります。

2.17　テキストデータの検索

ここでは，テキストデータを便利に検索できる機能について説明します。

2.17.1　表記ゆれ補正検索

ここでは，テキストデータの表記ゆれ補正検索について説明します。

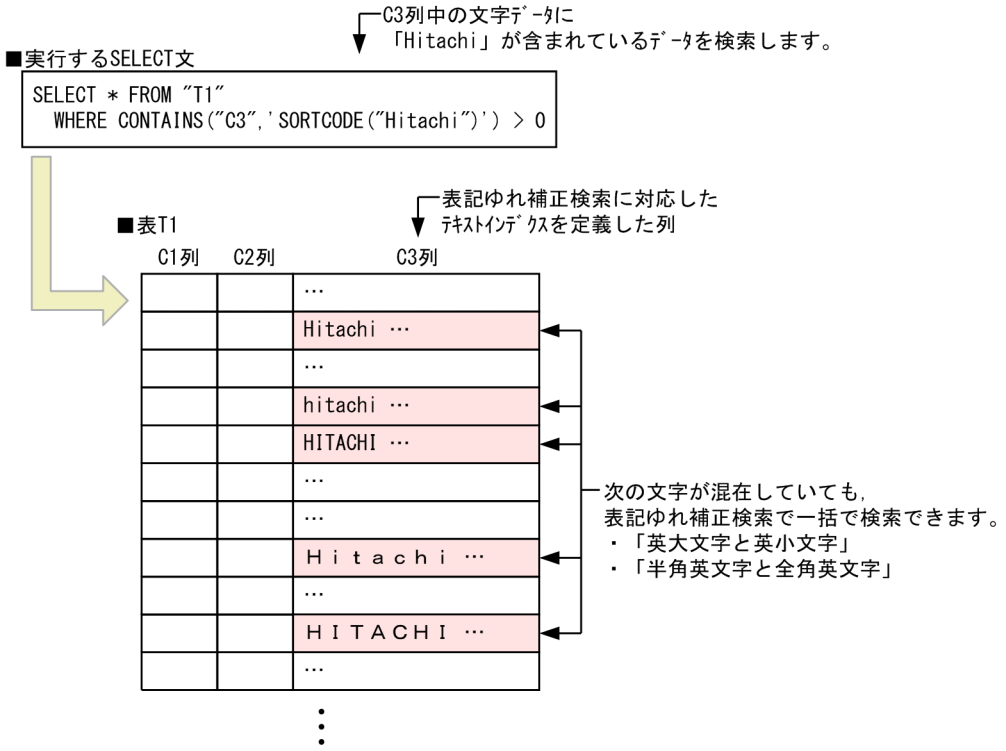
(1)　表記ゆれ補正検索とは

表記ゆれ補正検索とは，テキストデータ中の文字列に「大文字と小文字」，「半角文字と全角文字」，「ひらがなとカタカナ」などの表記ゆれがある場合に，それらの文字列を一括して検索する機能のことです。例えば，「Hitachi」で検索した場合，「Hitachi」だけでなく，「HITACHI」や「ＨＩＴＡＣＨＩ」などもまとめて検索できます。そのため，表記ゆれ補正検索を使用すると，複数の探索条件をOR でつないだ SQL 文を作成しなくても，表記がゆれているテキストデータを一括して検索できます。

コールセンタの履歴情報や製品の利用申請書のように，作成者が複数存在するテキストデータの場合，表記のゆれが発生しやすいと考えられます。そのようなテキストデータから，社名，製品名，担当者の名前など，ユニークな文字列をキーにして漏れなく検索する場合に，表記ゆれ補正検索を使用します。

表記ゆれ補正検索の例を次の図に示します。

図 2-58　表記ゆれ補正検索の例



［説明］

検索条件に「Hitachi」を指定した場合，「大文字と小文字」や「半角文字と全角文字」が混在していても，「Hitachi」が含まれている文字列を一括して検索できます。

表記ゆれ補正検索を行う場合は，スカラ関数CONTAINS を使用します。スカラ関数CONTAINS については，マニュアル『HADB　SQL リファレンス』の『スカラ関数』の『文字列関数（文字列情報の取得)』の『CONTAINS』を参照してください。

なお，表記ゆれ補正検索を行う場合は，表記ゆれ補正検索に対応しているテキストインデクスを定義すると，読み込むページを少なくすることができます。その分，表の検索性能を向上させることができます。

重要

表記ゆれ補正検索は，HADB サーバで使用する文字コードが，Unicode の場合（環境変数ADBLANG の指定値がUTF8 の場合）に使用できます。Shift-JIS の場合（環境変数ADBLANG の指定値がSJIS の場合）は使用できません。

(2)　表記ゆれ補正検索の規則

表記ゆれ補正検索の規則を次の表に示します。

表 2-21　表記ゆれ補正検索の規則

項番	文字の種類	規則	表記ゆれ補正検索の例
1	英字	次に示す文字の違いは，表記ゆれ補正検索の対象になります。 • 英大文字 • 英小文字 • 半角英字 • 全角英字	検索条件に「max」を指定した場合，次に示す文字列が，表記ゆれ補正検索の対象になります。 ・「max」 ・「MAX」 ・「ＭＡＸ」 ・「ｍａｘ」
2	数字	次に示す文字の違いは，表記ゆれ補正検索の対象になります。 • 半角数字 • 全角数字	
3	• ひらがな • カタカナ	**■ひらがなとカタカナ** 次に示す文字の違いは，表記ゆれ補正検索の対象になります。 • ひらがな • カタカナ **■全角カタカナと半角カタカナ** 次に示す文字の違いは，表記ゆれ補正検索の対象になります。 • 全角カタカナ • 半角カタカナ	**■例 1** 「りんご」を条件に検索した場合， 次に示す文字列が，表記ゆれ補正検索の対象になります。 ・「りんご」 ・「リンゴ」 ・「ﾘﾝｺﾞ」 ・「りんごﾞ」 ・「りんご゛」 **■例 2**

項番	文字の種類	規則	表記ゆれ補正検索の例
		■濁点と半濁点 次に示す文字の後ろに，全角の濁点または半角の濁点がある場合，対象の文字は濁点が付いた文字と見なされます。 • ひらがな • 全角カタカナ • 半角カタカナ また，全角の半濁点，および半角の半濁点の場合も同様です。 **■拗音と促音** 拗音および促音の場合，それぞれの大文字と同一であると見なされます。 **■長音記号** 長音記号の有無は，表記ゆれ補正検索の対象外になるため，別の文字列として扱われます。	「バット」を条件に検索した場合，次に示す文字列が，表記ゆれ補正検索の対象になります。 ・「バット」 ・「ﾊﾞｯﾄ」 ・「ばっと」 ・「は゛つと」 **■例 3** 検索条件に「プリンター」を指定した場合，「プリンタ」は検索対象外になります。「プリンタ」と「プリンター」は別の文字列となります。
4	ダイアクリティカルマーク	次に示す文字の違いは，表記ゆれ補正検索の対象になります。 • ダイアクリティカルマーク（ウムラウトなど）が付いた文字 • ダイアクリティカルマーク（ウムラウトなど）が付いていない文字	次に示す文字列は，同じ文字列と見なされます。 ・「A」 ・「ä」 ・「Ä」 ・「â」 そのため，検索条件に「MAX」を指定した場合，次に示す文字列が，表記ゆれ補正検索の対象になります。 ・「MAX」 ・「mäx」 ・「MÄX」 ・「mâx」
5	特定の文字列を 1 文字で表記した文字	特定の文字列を 1 文字で表記している場合は，対象の 1 文字を文字列に展開します。そして，展開後の文字列と同一と見なされます。	次に示す文字列は，同じ文字列と見なされます。 **■例1** 「㌘」と「グラム」 **■例2** 「㎏」と「kg」 例1の「㌘」を条件に検索した場合，次に示す文字列が，表記ゆれ補正検索の対象になります。 ・「㌘」 ・「グラム」 ・「ぐらむ」 ・「ｸﾞﾗﾑ」

「ひらがな」,「全角カタカナ」および「半角カタカナ」で，濁点や半濁点の有無が異なる文字の場合，別の文字と見なされます。そのため，表記ゆれ補正検索の対象外です。例えば，次に示す文字は，濁点や半濁点の有無が同じであるため，それぞれ同じ文字と見なされます。

- 「は」と「ハ」
- 「ば」と「バ」

- 「ぱ」と「パ」

一方，「は」と「ば」と「ぱ」は，濁点や半濁点の有無が異なるため，それぞれ別の文字と見なされます。例えば，検索条件に「バイク」を指定した場合，「バイク」と「ばいく」は検索できます。しかし，「ハイク」，「はいく」，「パイク」および「ぱいく」は，表記ゆれ補正検索の対象外であるため，検索できません。

表記ゆれ補正検索では，ISO/IEC14651:2011 で規定されている文字の並べ替えや比較を行うためのコード（ソートコード）を使用しています。ソートコードが一致している文字を同じ文字と見なしています。

メモ

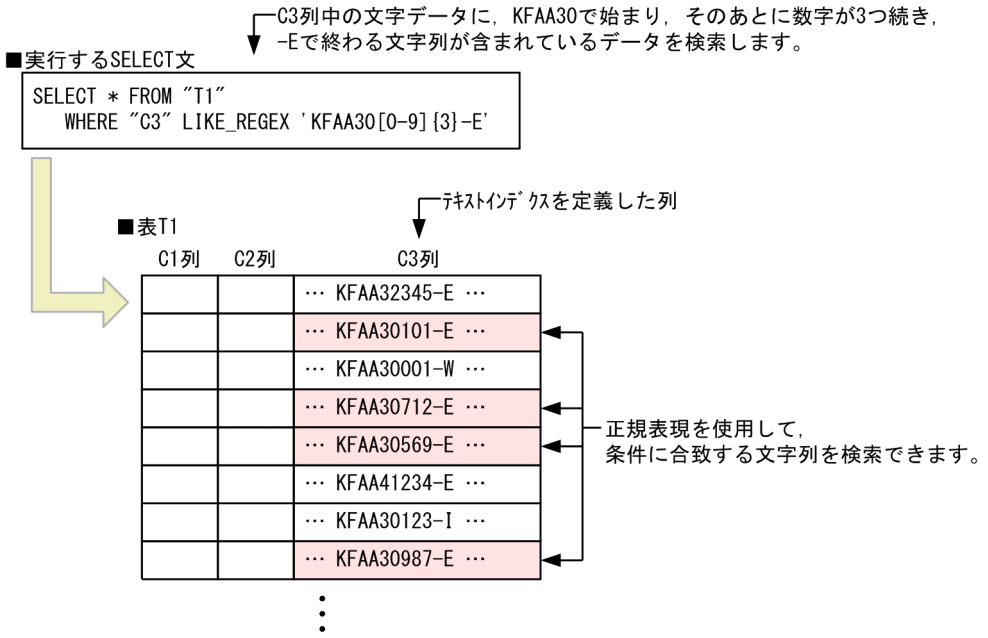
- ソートコードでは存在しないと判断される文字（「記号」など）の場合，バイトコードを使用して文字の並べ替えや比較を行います。
- スカラ関数CONTAINS の「*表記ゆれ補正検索指定*」に「SORTCODE(*単純文字列指定*)」を指定した場合に，ソートコードを使用します。

2.17.2 正規表現を使用した検索

正規表現を使用してテキストデータを検索することができます。正規表現を使用すると，複雑な検索条件を 1 つの検索条件で表すことができます。例えば，文字列ABC のあとに，5 つの数字がある文字列などを検索することができます。

正規表現を使用した検索の例を次の図に示します。

図 2-59 正規表現を使用した検索の例



正規表現を使用した検索を行う場合は，LIKE_REGEX 述語を使用します。LIKE_REGEX 述語については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『述語』の『LIKE_REGEX 述語』を参照してください。

なお，正規表現を使用してテキストデータを検索する際，テキストインデクスを使用すると，読み込むページを少なくすることができます。その分，表の検索性能を向上させることができます。

メモ

POSIX 準拠の正規表現（POSIX1003.2 準拠の拡張正規表現）を使用できます。ただし，次に示す点が異なります。

- 文字列挙に，照合要素（[. .]），および文字等価クラス（[= =]）は使用できません。
- 後方参照表現（¥*n*）（*n* は1 から9 までの整数）は使用できません。
- 特殊文字をリテラル文字として扱う場合，常にエスケープ文字の指定が必要です。また，文字列挙に「] 」をリテラル文字として指定する場合，および「^」を先頭以外でリテラル文字として指定する場合もエスケープ文字の指定が必要です。
- Perl 拡張の演算子の一部が使用できます。使用できる表現については，マニュアル『HADB SQL リファレンス』の『構成要素』の『述語』の『LIKE_REGEX 述語』にある文字クラスを参照してください。

重要

- 正規表現を使用する場合，複数の文字と一致する表現，または複数回の繰り返しの表現などの指定ができます。そのため，指定した正規表現の内容によっては，列のデータを複数回評価することがあります。したがって，列のデータ長が長い場合は，正規表現を使用した検索時に，メモリを大量に使用するおそれがあります。また，検索に時間が掛かるおそれがあります。
- 繰り返し因子に大きな数を指定した場合，または繰り返し因子をネストして連続で指定した場合は，メモリを大量に使用するおそれがあります。また，検索に時間が掛かるおそれがあります。

2.17.3　同義語検索

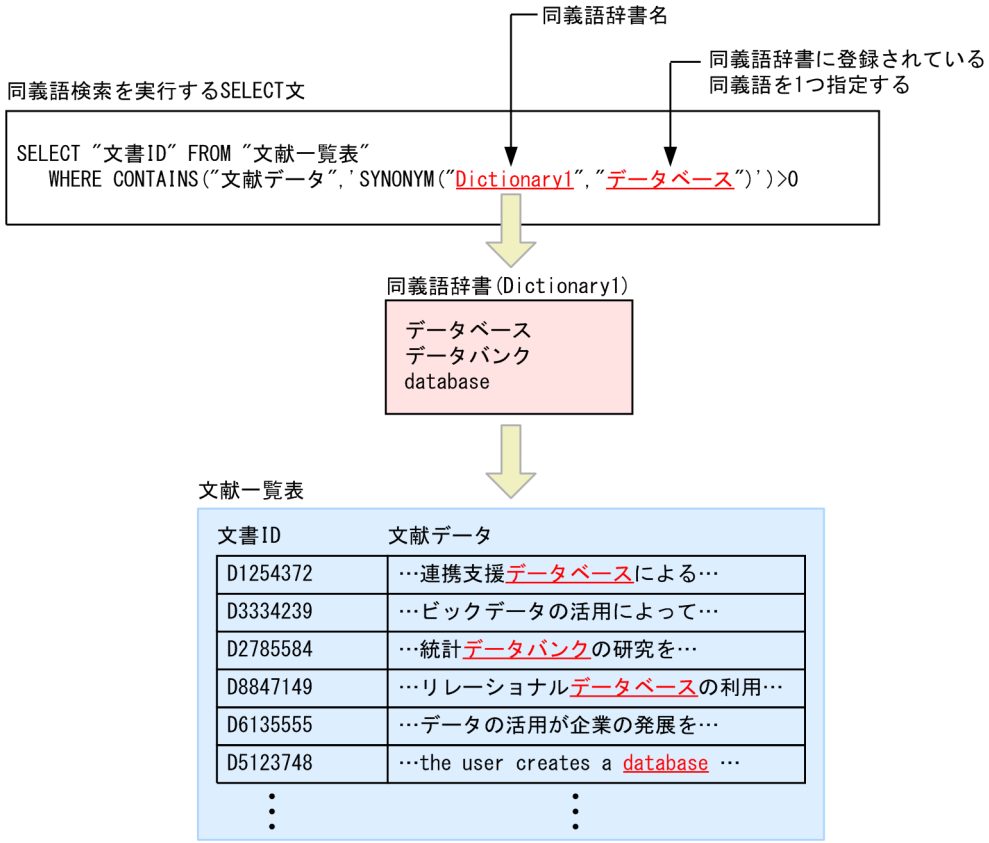
ここでは，テキストデータ（文書データ）の同義語検索について説明します。

(1)　同義語検索とは

文献や論文などの文書データを検索する際，検索対象に指定した語句と，その語句の同義語をまとめて検索することができます。この機能を同義語検索といいます。同義語検索をする場合は，検索対象とする同

義語の一覧を辞書に登録しておく必要があります。この辞書を同義語辞書といいます。同義語検索の概要を次の図に示します。

図 2-60　同義語検索の概要



[説明]

- 同義語検索を行う前に，同義語辞書を作成しておく必要があります。この例では，同義語辞書名が Dictionary1 という同義語辞書を作成しています。その同義語辞書に「データベース，データバンク，database」という同義語を登録しています。
- 同義語検索を行う場合，SELECT 文でスカラ関数CONTAINS を指定します。スカラ関数CONTAINS 中に同義語辞書名と，同義語辞書に登録されている同義語の 1 つを指定します。上記の例では，同義語の 1 つとして「データベース」を指定しています。
- 上記のSELECT 文を実行すると，同義語検索が行われて「データベース，データバンク，database」のどれかが含まれる文書データが検索されます。

■英語文章への利用方法

語尾の変化を考慮した英単語を同義語辞書に登録するという利用方法も考えられます。例えば，同義語辞書に「electromagnet，electromagnets，electromagnetic」を登録します。同義語検索の実行時に「electromagnet」を検索対象文字列に指定すると，「electromagnet，electromagnets，electromagnetic」のすべての語句が検索対象になります。

(例)

```
SELECT "TITLE" FROM "REPORTS"
    WHERE CONTAINS("DOCUMENTS",'SYNONYM("Dictionary2","electromagnet")')>0
```

メモ

同義語検索の環境設定および使い方については，「11.16　同義語検索を行う際の運用」を参照してください。

(2)　表記ゆれ補正検索との組み合わせ

同義語検索と表記ゆれ補正検索の両方を合わせて使用できます。スカラ関数CONTAINS の指定で，同義語検索と表記ゆれ補正検索の両方を使用することを選択できます。

（例）

```
SELECT "文書ID" FROM "文献一覧表"
    WHERE CONTAINS("文献データ",'SYNONYM("Dictionary1",SORTCODE("データベース"))')>0
```

上記のSELECT 文を実行すると，同義語検索が行われて「データベース，データバンク，database」のどれかが含まれる文書データが検索されます。さらに，表記ゆれ補正検索も適用されるため，「ﾃﾞｰﾀﾍﾞｰｽ，ﾃﾞｰﾀﾊﾞﾝｸ，DATABASE」などの語句が含まれている文書データも検索されます。

(3)　テキストインデクスとの関係

同義語検索を行う場合，テキストインデクスを使用すると検索性能の向上が見込まれるため，検索対象の文書データを格納している列にテキストインデクスを定義することを推奨します。

なお，同義語検索と表記ゆれ補正検索の両方を合わせて使用する場合にテキストインデクスを使用するときは，表記ゆれ補正検索に対応しているテキストインデクスを定義する必要があります。ただし，表記ゆれ補正検索に対応しているテキストインデクスは，対応していないテキストインデクスよりもデータ容量が増加します。

2.17.4　ワード検索

ここでは，テキストデータ（文書データ）のワード検索について説明します。

(1)　ワード検索とは

ワード検索とは，英語で書かれたテキストデータ（文書データ）から，検索対象に指定した英単語を高速に検索する機能です。ワード検索では，次の 2 つの方法で英単語を検索できます。

- 完全一致検索

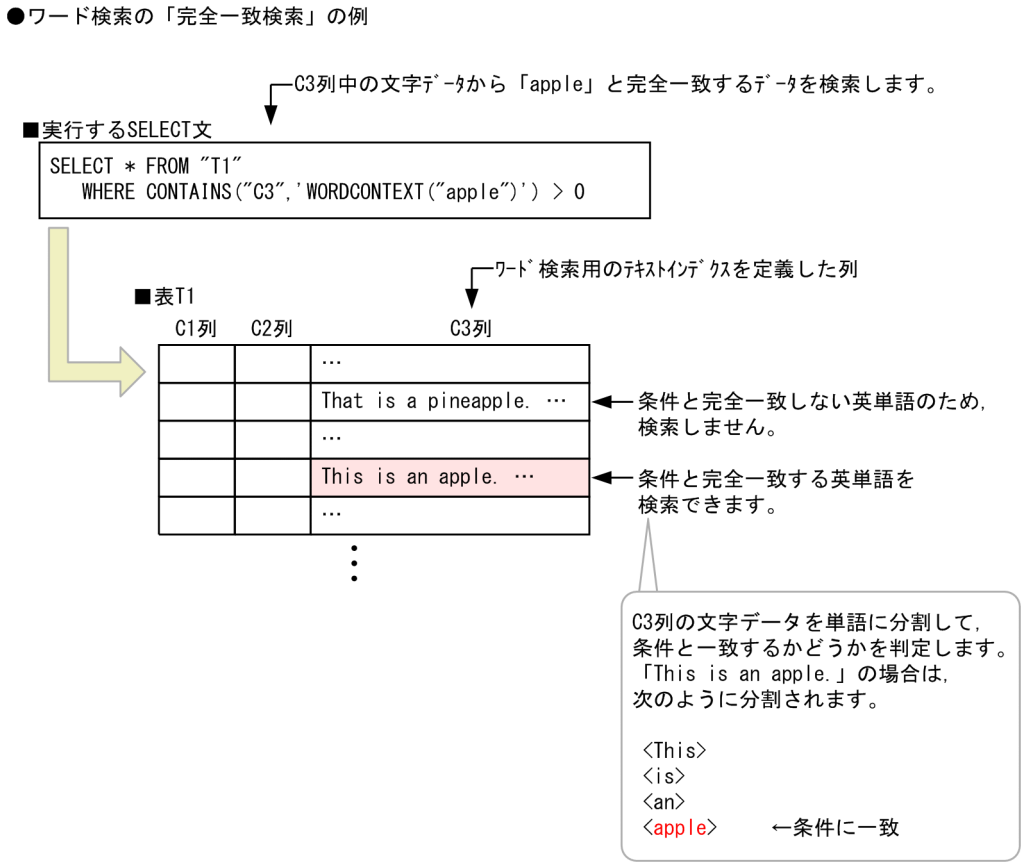
指定した英単語と完全に一致する英単語を検索する方法です。例えば，「apple」という英単語を検索したい場合は，「apple」と指定します。

- 前方一致検索

  検索したい英単語の先頭から数文字を指定して検索する方法です。例えば，「apple」という英単語を検索したい場合は，「app」のように指定します。

英語で書かれたテキストデータ（文書データ）を，完全一致検索や前方一致検索で高速に検索したい場合に，ワード検索を使用します。ワード検索の概要を次の図に示します。

図 2-61　ワード検索の概要



ワード検索を行う場合は，スカラ関数CONTAINS を使用します。スカラ関数CONTAINS については，マニュアル『HADB　SQL リファレンス』の『スカラ関数』の『文字列関数（文字列情報の取得)』の『CONTAINS』を参照してください。

また，ワード検索を行う場合は，ワード検索用のテキストインデクスを定義すると，読み込むページ数を少なくすることができます。その分，表の検索性能を向上させることができます。ワード検索用のテキストインデクスについては，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義)』を参照してください。

ワード検索の対象となるテキストデータ（文書データ）は，次の条件を満たす必要があります。

- テキストデータは，半角英数字の単語で構成されている。

- 単語と単語の間は，半角空白，タブ，改行，またはピリオドなどの記号で区切られている（区切り文字が存在している）。

区切り文字そのものは，ワード検索では検索されません。上記の条件を満たしている場合は，英語で書かれた文章以外でもワード検索を使用できます。区切り文字については，「(2)　ワード検索と区切り文字の関係」を参照してください。

メモ

英単語中に全角文字や全角空白が含まれている場合でも，ワード検索が実行されます。例えば，「Hitach ｉ 」（「 ｉ 」が全角文字）という英単語が存在する場合，「Hitach ｉ 」でワード検索が実行されます。

重要

ワード検索では，中間一致検索と後方一致検索は実行できません。例えば，「apple」を指定してワード検索を行っても，「pineapple」は検索できません。

中間一致検索や後方一致検索を実行したい場合は，LIKE 述語による中間一致検索を行ってください。または，スカラ関数CONTAINS を使用した中間一致検索を行ってください。

(2)　ワード検索と区切り文字の関係

ワード検索では，英語の文章中に含まれる区切り文字で区切られた単語を検索します。ワード検索で区切り文字として扱われる文字を次に示します。

- 半角空白（0x20）
- タブ（0x09）
- 改行（0x0A）
- 復帰（0x0D）
- ピリオド（0x2E）
- 疑問符（0x3F）
- 感嘆符（0x21）

区切り文字そのものは，単語には含まれません。そのため，区切り文字が 1 文字の場合でも，2 文字以上連続して存在した場合でも，単語には影響がありません。

■区切り文字が連続で存在した場合

次の文章の場合，どちらも「This」，「is」，「an」，「apple」がそれぞれ単語として扱われます。

- This△is◇an△apple.
- This△is◇an△△◇apple.

（凡例）

△：半角空白

◇：タブ

なお，ワード検索用のテキストインデクスを定義する際，区切り文字として扱う文字を指定します。詳細については，「5.4.6　ワード検索時の区切り文字の選択（DELIMITER）」を参照してください。

また，ワード検索の完全一致検索の場合，複数の単語（熟語）を指定して検索することができます。このときも，区切り文字そのものは単語に含まれません。そのため，区切り文字が 1 文字の場合でも，2 文字以上連続して存在した場合でも，検索する単語には影響がありません。

■ワード検索の完全一致検索による複数単語（熟語）の検索

ワード検索で「this△△△is」を指定して検索した場合は，次のような複数の単語（熟語）を検索できます。なお，「is△this」のように記載順序が異なったり，「this△apple△is」のように単語と単語の間に別の文字が含まれていたりした場合は，検索されません。

- this△is△an△apple
- this△◇is◇an◇apple

（凡例）

△：半角空白

◇：タブ

> **メモ**
>
> 検索対象に区切り文字だけが指定された場合は，ワード検索の対象となるテキストデータ（文書データ）の内容に関係なく，全件検索が行われます。

(3)　ワード検索と表記ゆれ補正検索の組み合わせ

ワード検索は，表記ゆれ補正検索と組み合わせて使用できます。表記ゆれ補正検索については，「2.17.1 表記ゆれ補正検索」を参照してください。

スカラ関数CONTAINS の指定で，ワード検索と表記ゆれ補正検索の両方を使用することを選択できます。ワード検索の完全一致検索と前方一致検索どちらの場合でも，表記ゆれ補正検索と組み合わせて使用できます。

ワード検索と表記ゆれ補正検索を組み合わせて使用すると，英単語に，次のような表記ゆれが存在しても一括して検索できます。

- 英大文字と英小文字
- ダイアクリティカルマーク（ウムラウトなど）が付いた文字と付いていない文字

(4)　ワード検索と同義語検索の組み合わせ

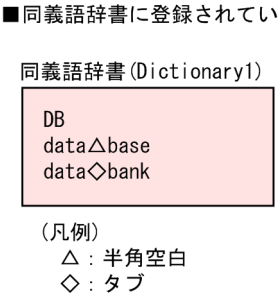
ワード検索の完全一致検索は，同義語検索と組み合わせて使用できます。同義語検索については，「2.17.3 同義語検索」を参照してください。

スカラ関数CONTAINS の指定で，ワード検索と同義語検索の両方を使用することを選択できます。そのため，検索対象とする単語とその単語の同義語を，ワード検索でまとめて検索できます。

> **重要**
>
> ワード検索の前方一致検索は，同義語検索と組み合わせて使用できません。

ワード検索の場合，単語そのものを検索するため，単語と単語の間の区切り文字の種類に関係なく，検索対象の単語と一致すれば検索されます。しかし，同義語検索の場合，区切り文字も含めた単語を同義語辞書に登録していると，区切り文字も含めて同じ単語のときだけ検索されます。例を次の図に示します。

図 2-62　ワード検索と同義語検索を組み合わせた場合の検索例



[説明]

- ワード検索と同義語検索を組み合わせて，「data◇base」で検索した場合は，同義語辞書に登録されている同義語「data△base」と一致しません。そのため，同義語辞書に登録されている同義語は検索されません。
- ワード検索と同義語検索を組み合わせて，「data△base」で検索した場合は，同義語辞書に登録されている同義語「data△base」と一致します。同義語辞書に登録されている同義語が検索されるため，同義語の「data◇bank」も検索されます。また，このときワード検索も行われるため，区切り文字が異なる「data△bank」も検索されます。

(5)　ワード検索・表記ゆれ補正検索・同義語検索の組み合わせ

ワード検索の完全一致検索は，表記ゆれ補正検索および同義語検索の両方と組み合わせて使用することもできます。

2.18　監査証跡機能

ここでは，データベースの利用状況を監査する際に必要となる監査証跡を出力する監査証跡機能について説明します。

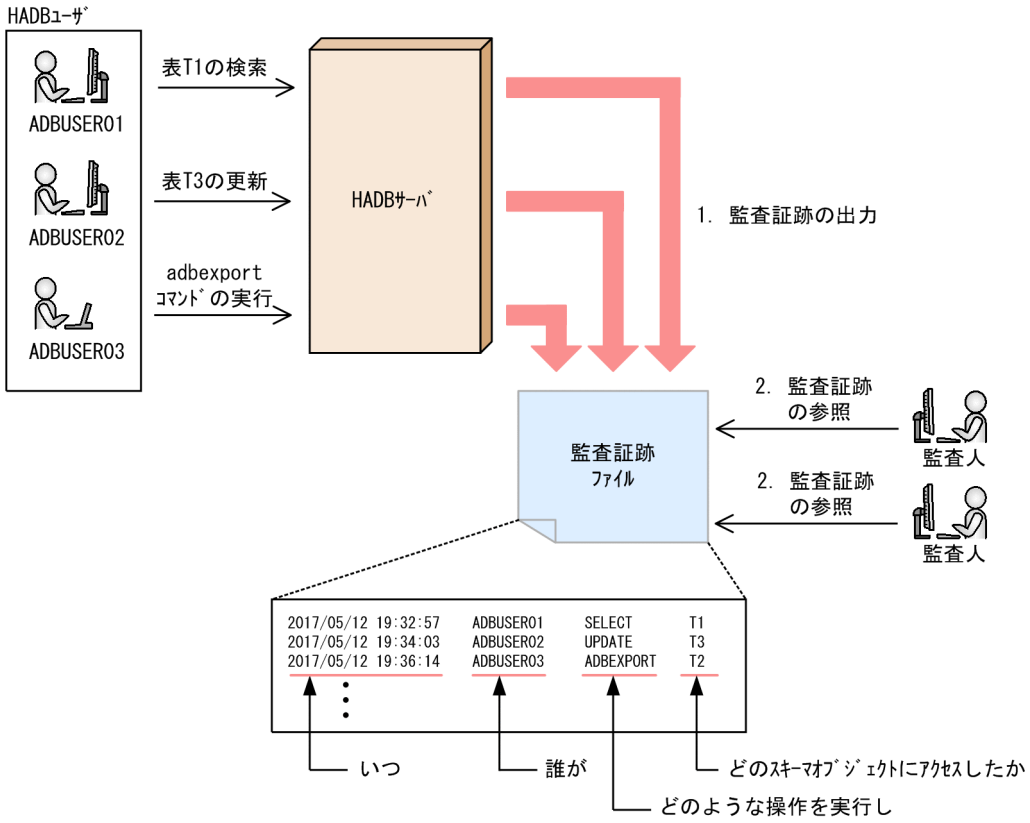
2.18.1　監査証跡機能とは

HADB ユーザがデータベースにアクセスした際のアクセス記録や，HADB ユーザが実行したコマンドの操作記録などをファイル（監査証跡ファイル）に出力します。この機能を**監査証跡機能**といいます。また，出力されるアクセス記録や操作記録などを**監査証跡**といいます。例えば，HADB ユーザが表にアクセスした場合，アクセスした時刻，アクセスした HADB ユーザの認可識別子，操作内容，操作対象のスキーマオブジェクトなどが，操作記録（監査証跡）として出力されます。出力された監査証跡を参照すると，「いつ」，「誰が」，「どのような操作を実行し」，「どのスキーマオブジェクトにアクセスしたか」，などを調査できます。出力された監査証跡の利用方法の例を次に示します。

- データベースの利用状況を監査する際に，監査証跡を使用してデータベース利用者の操作内容などを調査します。
- セキュリティインシデントが発生した際に，監査証跡を使用して原因の調査や，漏えいしたおそれがある情報を調査します。

監査証跡機能の概要を次の図に示します。

図 2-63　監査証跡機能の概要



[説明]

1. HADB ユーザが表の検索や更新，コマンドを実行した際の操作記録が，監査証跡として監査証跡ファイルに出力されます。
2. データベースの利用状況を監査する際，監査の実行者（監査人）が監査証跡を参照して調査を行います。また，セキュリティインシデントの発生時の調査資料の一つとして，監査証跡を利用します。

データベースシステムを使用している場合，データベースの不正利用が行われていないか，セキュリティポリシーに従ってデータベースが利用されているかなどを，定期的に監査する必要があります。この定期監査の際，監査の結果を証明するために監査証跡を使用します。

メモ

監査証跡機能は，セキュリティを直接強化する機能ではありません。データベースの利用者が，セキュリティポリシーに従ってデータベースを適切に利用しているかどうかを調査する際に使用する機能です。したがって，監査証跡機能を使用することによって，データベースの利用者の不正操作を必ず防止できるわけではありません。しかし，監査証跡機能を使用して監査を実施していることをデータベースの利用者に知らせることによって，悪意を持った利用者による不正操作を心理的に抑止できると考えられます。それによって，総合的にセキュリティ効果を高める狙いがあります。

2.18.2　監査人の作成

監査証跡機能を使用する場合は，監査人を作成する必要があります。監査人とは，監査権限（監査管理権限または監査参照権限）を持っている HADB ユーザのことです。監査人は次のことを実施します。

- 監査証跡機能の運用

  監査証跡を出力する操作（監査イベント）の選定，監査証跡の出力開始，HADB ユーザが持っている監査権限の取り消しなど，監査証跡機能の運用を行います。監査証跡機能の運用を行うには，**監査管理権限**が必要になります。

- 監査の実行

  監査証跡を参照して，データベースの利用状況の監査や，統一フォーマット用監査証跡ファイルの出力を行います。

  監査証跡を参照するには，**監査参照権限**が必要になります。

DBA 権限を持っている HADB ユーザが，GRANT 文を実行して HADB ユーザに監査権限を付与することで監査人を作成します。監査管理権限だけを持っている監査人，監査参照権限だけを持っている監査人，または監査管理権限と監査参照権限の両方を持っている監査人を作成できます。また，これらの監査人を複数人作成することもできます。監査人の役割や作業内容を考慮して，必要な権限を付与してください。

2.18.3　監査証跡の出力契機（監査対象イベント）

監査証跡の出力対象となる操作を**監査対象イベント**といいます。監査対象イベントには，監査証跡機能が有効な場合は監査証跡が必ず出力される**必須監査イベント**と，監査証跡を出力するかどうかを選択できる**選択監査イベント**があります。

例えば，次に示す操作などが監査対象イベントになります。これらの監査対象イベントの終了時に監査証跡が出力されます。

- 必須監査イベントの例
  - adbstart コマンドによる HADB サーバの開始
  - adbmodarea コマンドによる DB エリアの追加または変更
  - GRANT 文による監査権限の付与
  - adbaudittrail コマンドによる監査証跡機能の無効化
  - ADB_AUDITREAD 関数を使用した監査証跡の参照
- 選択監査イベントの例
  - HADB サーバへの接続
  - CREATE TABLE 文による表の定義
  - SELECT 文による表の検索
  - UPDATE 文による行の更新

- adbimport コマンドによるデータのインポート

監査対象イベントとなる操作の詳細については，「12.9.1　監査対象イベントの一覧と出力項目」を参照してください。

重要

監査対象イベントの対象となるスキーマオブジェクトが複数ある場合，スキーマオブジェクトの数だけ監査証跡が出力されます。

(例)

```
SELECT * FROM "T1","T2"
```

[説明]

上記のSELECT 文を実行した場合，表T1 に関する監査証跡と，表T2 に関する監査証跡がそれぞれ出力されます。

上記のように，監査対象イベントの対象となるスキーマオブジェクトごとに監査証跡は出力されます。そのため，必須監査イベントと選択監査イベントの両方で監査証跡を出力すると，監査証跡の出力量が非常に多くなる可能性があります。また，監査証跡の出力量が多くなると，監査の作業効率にも影響を及ぼすおそれがあります。したがって，監査の目的を明確にして，選択監査イベントでも監査証跡を出力するかどうかを選択します。選択監査イベントでも監査証跡を出力する場合は，CREATE AUDIT 文で監査対象を定義します。

メモ

- 監査対象イベントの対象となるスキーマオブジェクトが 1 つの場合，または監査対象イベントの対象となるスキーマオブジェクトがない場合は，監査証跡が 1 つだけ出力されます。
- コマンドの入力誤りによるエラー発生時は，監査証跡は出力されません。

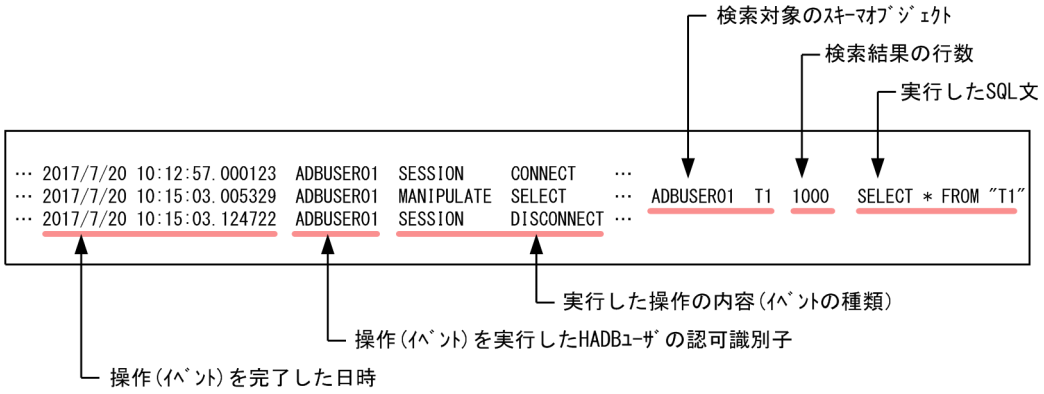
2.18.4　監査証跡に出力される情報

監査証跡には，「いつ」，「誰が」，「どのような操作を実行したか」などの情報が記録されます。監査証跡に出力される情報の例を次に示します。

(例 1)

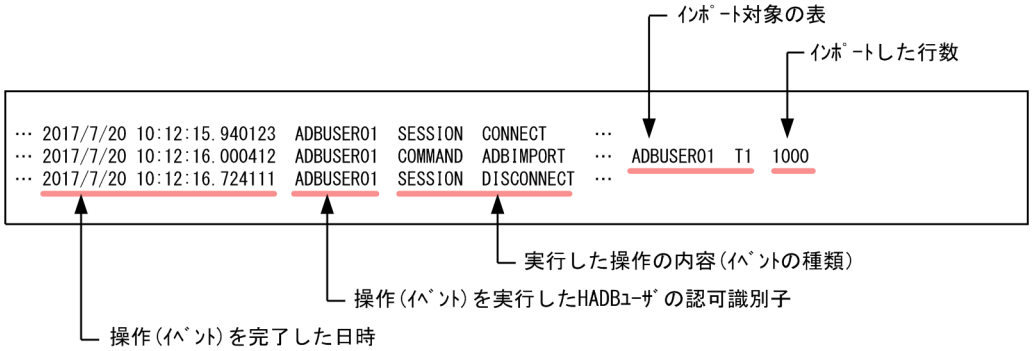
```
SELECT * FROM "T1" WHERE "C1"=100
```

上記の SQL 文が実行された場合，監査証跡には次のような情報が出力されます。



（例 2）

adbimport コマンドが実行された場合，監査証跡には次のような情報が出力されます。



ここでは監査証跡に出力される情報の一部だけを紹介しています。監査証跡に出力される情報の詳細については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」および「12.9.3　監査証跡の出力契機と出力項目」を参照してください。

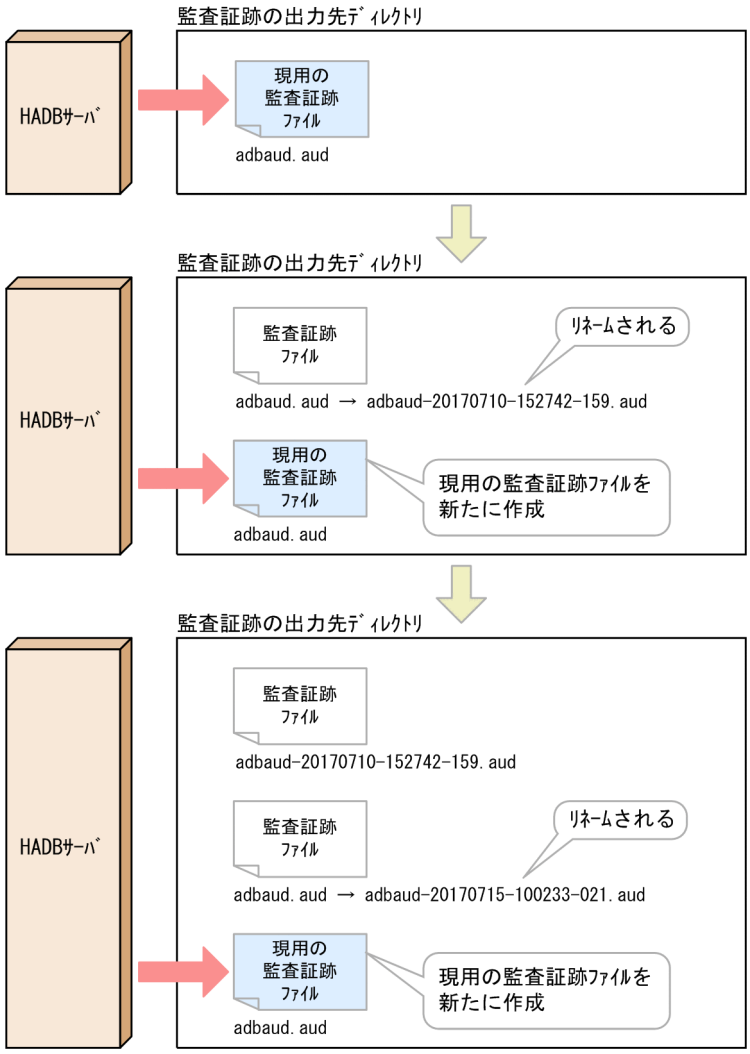
2.18.5　監査証跡の出力先（監査証跡ファイル）

監査証跡は指定したディレクトリ下（監査証跡の出力先ディレクトリ下）のファイルに出力されます。このファイルを**監査証跡ファイル**といいます。

また，現在，監査証跡の出力対象となっているファイルを**現用の監査証跡ファイル**といいます。現用の監査証跡ファイルの容量が上限に達すると，HADB サーバは，現用の監査証跡ファイルをリネームします。そのあと，新たに現用の監査証跡ファイルを作成し，そのファイルに監査証跡を出力します。これを**現用の監査証跡ファイルの切り替え**といいます。

現用の監査証跡ファイルの切り替えの仕組みを次の図に示します。

図 2-64　現用の監査証跡ファイルの切り替えの仕組み



[説明]

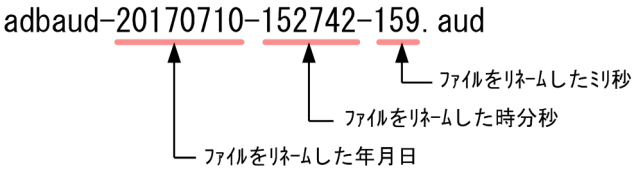
- 監査証跡の出力先ディレクトリ下の現用の監査証跡ファイルに監査証跡が出力されます。現用の監査証跡ファイルの名称は，adbaud.aud 固定です。
- 現用の監査証跡ファイルの容量が上限（256 メガバイト）に達すると，現用の監査証跡ファイルがリネームされます。

> **メモ**
>
> 256 メガバイトの上限値は，サーバ定義のadb_audit_log_max_size オペランドで変更できます。

- 新たに現用の監査証跡ファイルが作成され，そのファイルに監査証跡が出力されます。

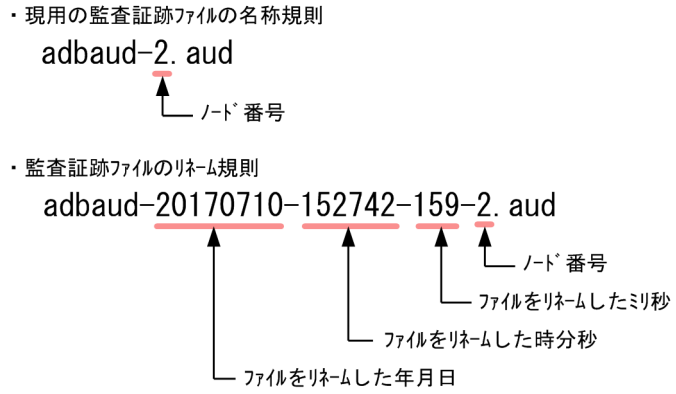
■監査証跡ファイルのリネーム規則

現用の監査証跡ファイルが切り替わる際，現用だった監査証跡ファイルは，次に示す規則に従ってリネームされて保存されます。



[マルチノード機能の場合]

現用の監査証跡ファイル名と，監査証跡ファイルのリネーム規則を次に示します。



メモ

- 上記の名称規則に該当しない監査証跡ファイル，または監査証跡の出力先ディレクトリ以外に格納されている監査証跡ファイルは，監査証跡の出力が完了したファイルとして扱われます。
- リネーム時に同じ名称の監査証跡ファイルが監査証跡の出力先ディレクトリにある場合は，ファイル名が重複しないようにリネームされます。

■現用の監査証跡ファイルの切り替わり契機

現用の監査証跡ファイルは，次の契機で切り替わります。

- 現用の監査証跡ファイルの容量が上限に達したとき
- `adbauditrail --swap` コマンドで現用の監査証跡ファイルを切り替えたとき

メモ

次の操作をした場合，この操作に関する監査証跡の出力後，現用の監査証跡ファイルがリネームされます。このとき，新しい現用の監査証跡ファイルは作成されません。

- HADB サーバの終了
- `adbauditrail --stop` コマンドによる監査証跡の無効化

次回の HADB サーバの開始時，または監査証跡の有効化時に，現用の監査証跡ファイルが作成されます。

■監査証跡の出力先ディレクトリ下に格納できる監査証跡ファイルの上限数

監査証跡の出力先ディレクトリ下に格納できる監査証跡ファイル数に上限を設定できます。上限数は，サーバ定義のadb_audit_log_max_num オペランドで指定します。例えば，このオペランドに100 を指定した場合，監査証跡の出力先ディレクトリ下に格納できる監査証跡ファイルの上限が 100 個になります。この監査証跡ファイルには，現用の監査証跡ファイルも含まれます。HADB サーバが 101 個目の監査証跡ファイルを作成しようとした場合，いちばん古い監査証跡ファイルが削除されます。

重要

デフォルトは，ファイル数の上限の設定なしとなります。この場合，監査証跡の出力先ディレクトリ下に格納される監査証跡ファイルは単調増加していくため，ディスクの空き容量に注意する必要があります。

2.18.6 監査証跡の参照

監査証跡は，監査証跡ファイルにバイナリ形式で出力されます。そのため，OS 上でファイルを開いて参照することはできません。監査証跡を参照する場合，次の 2 つの方法があります。

- ADB_AUDITREAD 関数を指定したSELECT 文を実行する

  ADB_AUDITREAD 関数とは，監査証跡ファイル中の監査証跡を表形式のデータ集合（表関数導出表）に変換する関数です。ADB_AUDITREAD 関数を指定したSELECT 文を実行すると，監査証跡を参照できます。

- adbconvertaudittrailfile コマンドで監査証跡を変換する

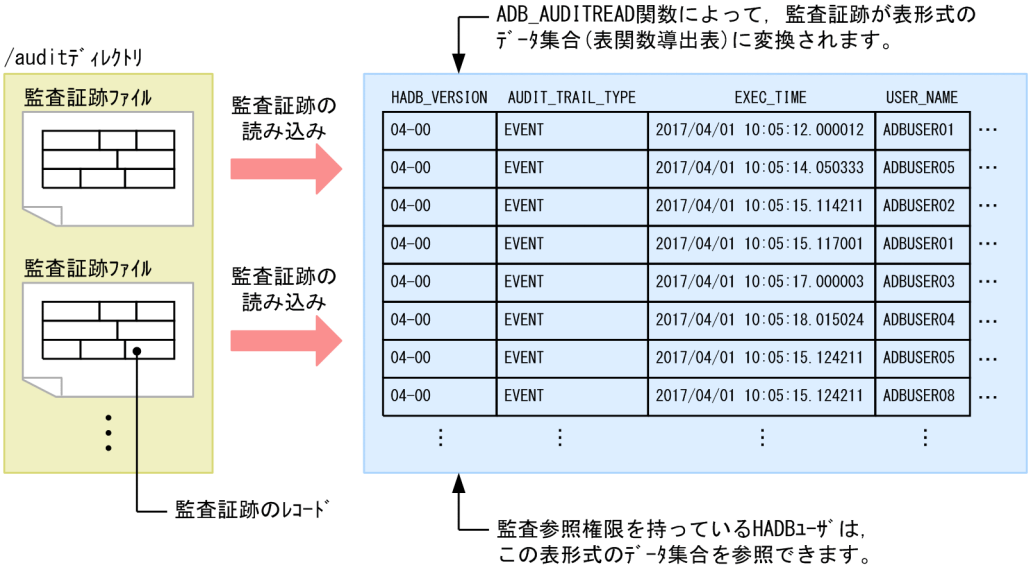
  監査証跡ファイルに出力された監査証跡をadbconvertaudittrailfile コマンドで変換すると，他製品である JP1/Audit で監査証跡を参照できます。監査証跡の変換については，「2.18.9 監査証跡の変換(JP1/Audit Management - Manager との連携)」を参照してください。

ここでは，ADB_AUDITREAD 関数を指定したSELECT 文を実行して，監査証跡を参照する方法について説明します。

ADB_AUDITREAD 関数中に，調査対象の監査証跡が格納されている監査証跡ファイルのパスを指定すると，HADB サーバはその監査証跡ファイル中の監査証跡を読み込み，表形式のデータ集合に変換します。

ADB_AUDITREAD 関数による監査証跡ファイル中の監査証跡の読み込みの例を次の図に示します。

図 2-65　ADB_AUDITREAD 関数による監査証跡ファイル中の監査証跡の読み込みの例



[説明]

下記のSELECT 文を実行した場合，下線部分に指定した監査証跡ファイル中の監査証跡が読み込まれて，表形式のデータ集合に変換されます。

```
SELECT * FROM
  TABLE(ADB_AUDITREAD(MULTISET['/audit/*.aud'])) "DT"
```

重要

ADB_AUDITREAD 関数中に監査証跡ファイルのパス（/audit/*.aud）を指定しています。この例では，特殊文字*を指定しています。この例のように指定すると，/audit ディレクトリ下のすべての監査証跡ファイルが，ADB_AUDITREAD 関数の読み込み対象になります。

ADB_AUDITREAD 関数の詳細については，マニュアル『HADB　SQL リファレンス』の『ADB_AUDITREAD 関数』を参照してください。

ADB_AUDITREAD 関数によって変換された表形式のデータ集合の各列の列名やデータ型は，HADB サーバによって決められています。各列の列名やデータ型，および格納される情報については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

■監査証跡の検索例

アクセス履歴を調査したい期間などの検索条件を，SELECT 文の探索条件に指定して，監査証跡を検索することができます。監査証跡の検索例を次に示します。

（例）

2017 年 4 月 1 日～2017 年 4 月 30 日の間に，HADB サーバにアクセスした HADB ユーザを検索します。

```
SELECT DISTINCT "USER_NAME"                                          ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/audit/*.aud'])) "DT"
```

```
    WHERE "EXEC_TIME" BETWEEN TIMESTAMP'2017/04/01 00:00:00.000000'   ...2
                          AND TIMESTAMP'2017/04/30 23:59:59.999999'
```

検索結果の例

```
USER_NAME
---------
ADBUSER02
ADBUSER06
ADBUSER07
```

[説明]

1. ADB_AUDITREAD 関数によって変換された表形式のデータ集合のUSER_NAME 列には，HADB サーバにアクセスした HADB ユーザの認可識別子が格納されています。
2. EXEC_TIME 列には，HADB ユーザが操作を実行した時刻（イベント完了時刻）が格納されています。

重要

監査参照権限を持っている HADB ユーザだけが，監査証跡を参照できます（ADB_AUDITREAD 関数を指定したSELECT 文を実行できます）。

2.18.7 監査証跡機能の有効化と無効化

監査証跡の出力は，HADB サーバの性能に与える影響が大きいため，デフォルトでは監査証跡を出力しない設定になっています。この状態を監査証跡機能が無効になっているといいます。

監査証跡機能の環境設定が完了したあとにadbauditttrail --start コマンドを実行すると，監査証跡機能が有効になり，監査証跡が出力されるようになります。監査証跡機能が有効な場合，次に示す監査証跡機能に関する操作が実行できます。

- ADB_AUDITREAD 関数を使用した監査証跡の参照
- CREATE AUDIT 文による監査対象の定義
- DROP AUDIT 文による監査対象定義の削除
- REVOKE 文による監査権限の取り消し

一度有効にした監査証跡機能を無効にする場合は，adbauditttrail --stop コマンドを実行します。監査証跡機能が無効になると，監査証跡の出力が停止します。

重要

監査管理権限を持っている HADB ユーザだけが，監査証跡機能の有効化と無効化を操作できます。

2.18.8　監査証跡機能で使用するディスクの準備

監査対象イベントの発生数によって，監査証跡の出力量が非常に多くなる可能性があります。その結果，監査証跡を出力するディスクに対する入出力が多く発生し，HADB サーバの処理性能に影響を及ぼすおそれがあります。そのため，監査証跡を出力する専用のディスク（監査証跡の出力先ディスク）を準備することを推奨します。また，そのディスクとは別に，監査証跡ファイルを保存しておくディスク（監査証跡の保存先ディスク，監査証跡の長期保存先ディスク）を準備することを推奨します。これら 3 つのディスクを準備して，監査証跡機能を運用することを推奨します。各ディスクの用途を次に説明します。

- 監査証跡の出力先ディスク

  監査証跡の出力先ディスクには，監査証跡の出力先ディレクトリだけを作成し，そのディレクトリ下に監査証跡を出力します。基本的に，このディスクに保存されている監査証跡ファイルは参照しません。

  監査証跡の出力先ディスクから監査証跡の保存先ディスクに，監査証跡ファイルを定期的に移動させます。ファイルの移動を実行するバッチプログラムを作成して，定期的にそのバッチプログラムを実行することを推奨します。

- 監査証跡の保存先ディスク

  監査証跡の保存先ディスクには，監査の際に使用する監査証跡ファイルを保存します。監査の際には，監査証跡の保存先ディスクに保存している監査証跡ファイルのパスを ADB_AUDITREAD 関数中に指定して，監査証跡を参照します。例えば，監査時に過去 1 年分の監査証跡を使用する場合は，過去 1 年分の監査証跡ファイルをこのディスクに保存しておきます。

  監査期間を過ぎた監査証跡ファイルは，監査証跡の長期保存先ディスクに移動させます。

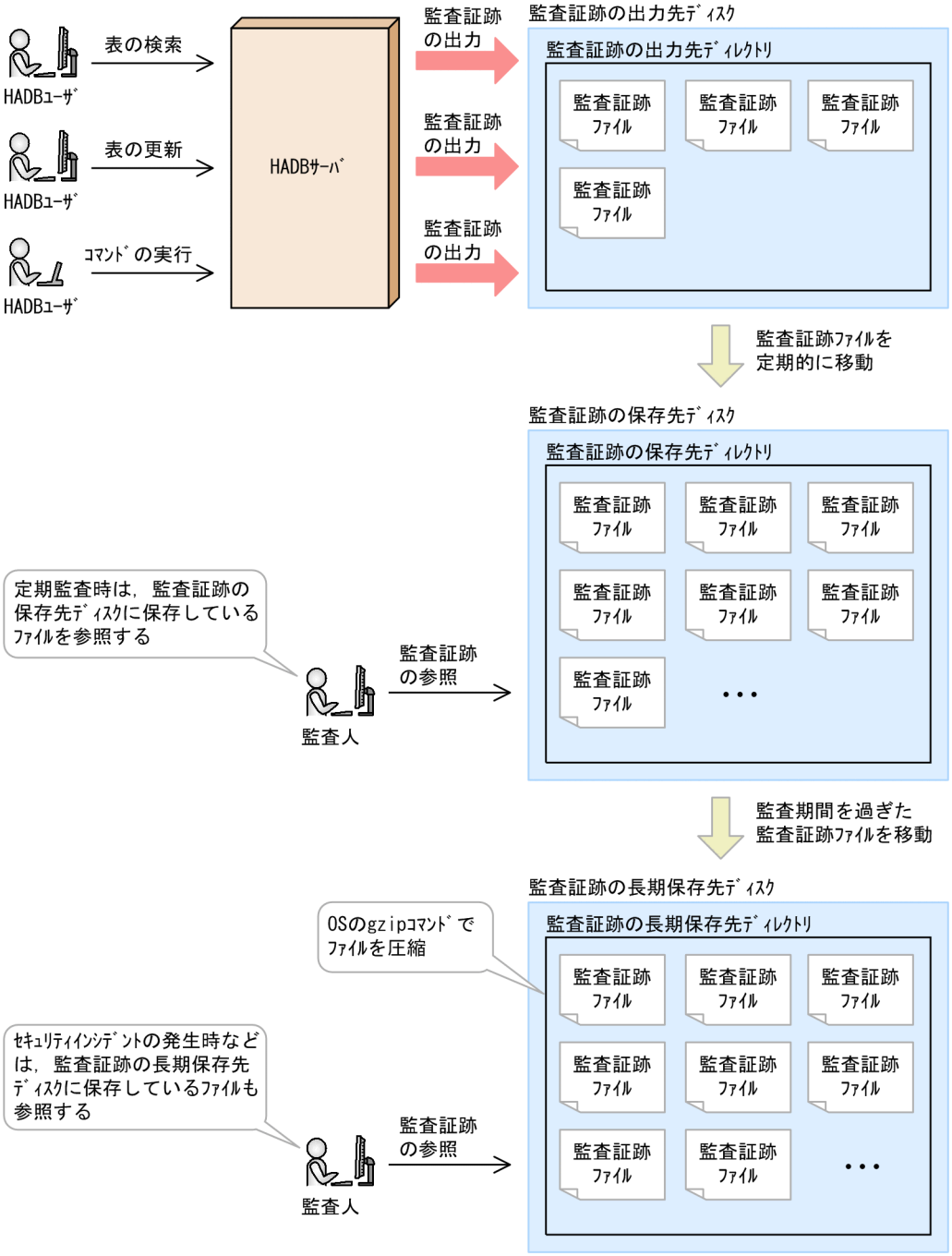
- 監査証跡の長期保存先ディスク

  監査証跡の長期保存先ディスクには，監査期間が過ぎた監査証跡ファイルを保存します。例えば，監査が 1 年ごとに実施される場合は，過去 1 年を過ぎた監査証跡ファイルを保存しておきます。

  通常は，監査証跡の長期保存先ディスクに保存している監査証跡ファイル中の監査証跡を参照することはありません。セキュリティインシデントが発生した場合や，過去に遡って監査証跡を調査する場合に備えて，監査証跡ファイルを保存しておくためのディスクです。

監査証跡機能の推奨運用の流れを次の図に示します。

図 2-66　監査証跡機能の推奨運用の流れ



上記の図の，監査証跡の長期保存先ディスクに保存している監査証跡ファイルは，参照頻度が少ないため，OS のgzip コマンドで圧縮することを推奨します。ディスク容量に応じてファイルの圧縮を検討してください。監査証跡の保存先ディスクに保存している監査証跡ファイルは，監査の際に参照するため，圧縮しないで保存しておくことを推奨します。

重要

監査証跡ファイルが圧縮されている状態であっても，ADB_AUDITREAD 関数を指定したSELECT 文で監査証跡を参照することができます。

2.18.9　監査証跡の変換（JP1/Audit Management - Manager との連携）

ここでは，監査証跡の変換について説明します。HADB サーバの監査証跡を JP1/Audit Management - Manager（JP1/Audit）で収集・管理する場合，監査証跡の変換が必要になります。

監査証跡ファイル内の監査証跡を，JP1/Audit が扱える形式（統一フォーマット形式）に adbconvertaudittrailfile コマンドで変換することを，**監査証跡の変換**といいます。

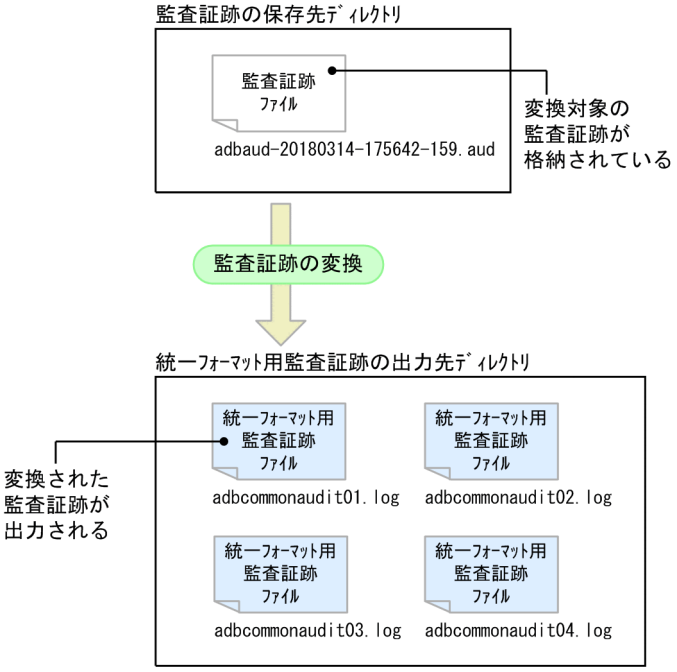
変換された監査証跡は，adbconvertaudittrailfile コマンド実行時に指定したディレクトリ下（統一フォーマット用監査証跡の出力先ディレクトリ下）のファイルに出力されます。このファイルを**統一フォーマット用監査証跡ファイル**といいます。統一フォーマット用監査証跡ファイル内の監査証跡は，JP1/Audit で収集・管理することができます。

「2.18.6　監査証跡の参照」で説明しているとおり，監査証跡は，監査証跡ファイルにバイナリ形式で出力されます。この監査証跡は JP1/Audit で扱えない形式であるため，HADB サーバと JP1/Audit を連携しても，JP1/Audit で監査証跡を収集・管理できません。そのため，JP1/Audit で監査証跡を収集・管理したい場合は，adbconvertaudittrailfile コマンドで監査証跡を変換する必要があります。

> **メモ**
>
> JP1/Audit で HADB サーバの監査証跡を収集・管理できるようになると，他製品の監査証跡とあわせた一元管理を実現できるようになります。

監査証跡の変換の概要を次の図に示します。

図 2-67　監査証跡の変換の概要



［説明］

adbconvertaudittrailfile コマンドに，**変換対象の監査証跡が格納されている監査証跡ファイル**，および**統一フォーマット用監査証跡の出力先ディレクトリ**を指定して，監査証跡の変換を行います。

adbconvertaudittrailfile コマンドで変換された監査証跡は，統一フォーマット用監査証跡の出力先ディレクトリに格納されている統一フォーマット用監査証跡ファイルに出力されます。

HADB サーバと JP1/Audit を連携している場合，監査証跡の変換を行うことで，統一フォーマット用監査証跡ファイルに出力された監査証跡は，JP1/Audit で収集・管理されるようになります。

監査証跡機能と JP1/Audit を連携する方法については，「12.8　監査証跡機能と JP1/Audit Management - Manager との連携」を参照してください。また，JP1/Audit の詳細については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。

adbconvertaudittrailfile コマンドの詳細については，マニュアル『HADB　コマンドリファレンス』の『adbconvertaudittrailfile（監査証跡ファイルの変換)』を参照してください。

重要

監査参照権限を持っている HADB ユーザだけが，adbconvertaudittrailfile コマンドを実行できます。

メモ

統一フォーマット用監査証跡の出力先ディレクトリには，統一フォーマット用監査証跡ファイルが最大で 4 つ格納されます。詳細については，「12.8.3　監査証跡機能と JP1/Audit を連携する際の環境設定」の「(1)　HADB サーバの環境設定」を参照してください。

2.19　マルチノード機能

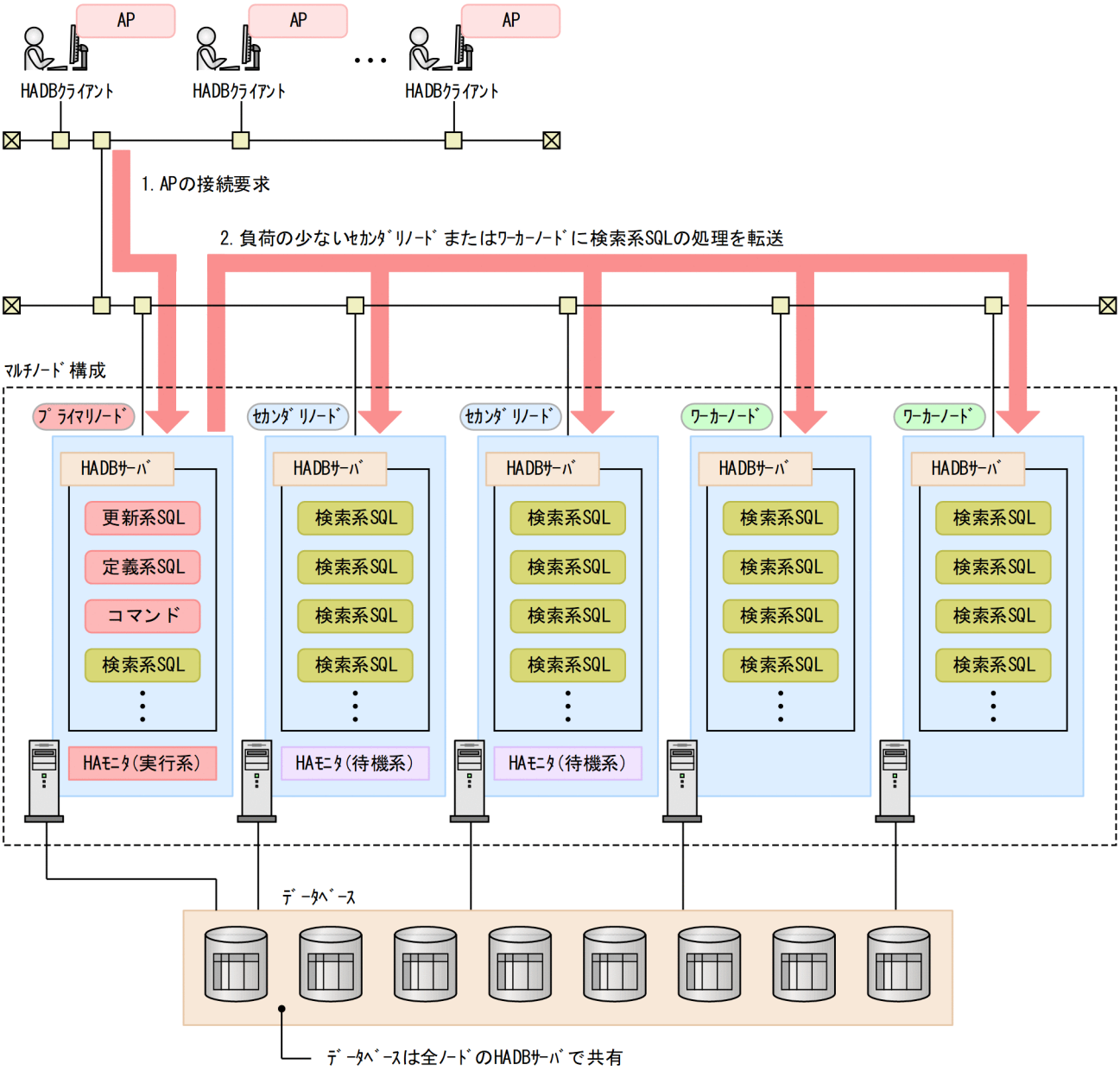
ここでは，マルチノード機能の特長，およびマルチノード機能を使用する際のシステム構成などについて説明します。

2.19.1　マルチノード機能とは

マルチノード機能とは，複数の HADB サーバを連携することによって，検索系 SQL（SELECT 文）の処理の負荷分散を実現する機能のことです（スケールアウトによる負荷分散）。

マルチノード機能を使用した負荷分散の例を次の図に示します。

図 2-68　マルチノード機能を使用した負荷分散の例



［説明］

上記の例では，5 つの HADB サーバで検索系 SQL の処理を実行し，負荷分散を行っています。

1. HADB クライアントからの AP の接続要求は，プライマリノードの HADB サーバが受け付けます。
2. プライマリノードの HADB サーバが，負荷分散の制御を行います。プライマリノードの HADB サーバが負荷の少ないセカンダリノードまたはワーカーノードのうちの 1 つのノードを選択し，そのノードで検索系 SQL を実行します。検索系 SQL の処理をセカンダリノードまたはワーカーノードに振り分けることで負荷分散を実現しています。

上記の図中に出てくる用語について説明します。

- **マルチノード構成**

  マルチノード機能を使用している複数の HADB サーバから成るシステム構成を**マルチノード構成**といいます。

- **プライマリノード**

  マルチノード機能の制御を行う HADB サーバがあるノードを**プライマリノード**といいます。プライマリノードは必ず 1 つ必要になります。

  プライマリノードの HADB サーバでは，更新系 SQL，定義系 SQL，一部のコマンドなど，データベースの更新が発生する処理などが実行されます。

  また，AP またはコマンドの接続要求は，プライマリノードの HADB サーバが受け付けます。

- **セカンダリノード**

  検索系 SQL を処理する HADB サーバがあるノードを**セカンダリノード**といいます。セカンダリノードは，1～3 つ設定できます。

  セカンダリノードの HADB サーバでは，検索系 SQL の処理が実行されます。

  また，セカンダリノードは，プライマリノードの予備のノードになります。プライマリノードで障害が発生した場合，HA モニタの制御によってセカンダリノードの 1 つをプライマリノードに変更して業務を続行します。詳細については，「2.19.4　マルチノード構成からのノードの切り離し（プライマリノード，セカンダリノード）」の「(1)　プライマリノードでノード障害が発生した場合」を参照してください。

- **ワーカーノード**

  検索系 SQL を処理する HADB サーバがあるノードを**ワーカーノード**といいます。ワーカーノードは，0～8 つ設定できます。

  ワーカーノードの HADB サーバもセカンダリノードの HADB サーバと同様に，検索系 SQL の処理が実行されます。

  ただし，ワーカーノードは，セカンダリノードとは異なり，プライマリノードの予備のノードにはなりません。そのため，プライマリノードで障害が発生しても，ワーカーノードがプライマリノードになることはありません。

メモ

- セカンダリノードおよびワーカーノードの負荷が高い場合は，プライマリノードでも検索系 SQL が実行されます。
- セカンダリノードおよびワーカーノードの HADB サーバでは，データベースの更新が発生する処理を実行することはできません。そのため，更新系 SQL，定義系 SQL，およびデータベースの更新が発生するコマンド（adbimport コマンドなど）は，セカンダリノードおよびワーカーノードの HADB サーバでは実行されません。データベースの更新が発生しない一部のコマンドについては，セカンダリノードおよびワーカーノードの HADB サーバで実行できます。

- HA モニタ

  マルチノード機能を使用する場合，HA モニタが必要になります。HA モニタは，プライマリノードおよびセカンダリノードの状態を監視し，プライマリノードまたはセカンダリノードに障害が発生した場合，ノードの切り離しを実行する製品です。HA モニタの機能については，「2.19.3　ノードの管理」を参照してください。

  HA モニタは，プライマリノードおよびセカンダリノードにインストールします。プライマリノードは HA モニタの実行系として動作し，セカンダリノードは HA モニタの待機系として動作します。

  なお，ワーカーノードには，HA モニタをインストールしません（ワーカーノードは HA モニタの管理対象外です）。

メモ

HADB サーバがインストールされているサーバマシンと，そのサーバマシンにインストールされている OS，HADB サーバ，および HA モニタを包括したものを**ノード**といいます。

マルチノード機能使用時の処理性能は，ハードウェア環境や処理件数に依存します。また，マルチノード機能を使用する場合は，マルチノード機能を使用しない場合に比べて，ノード間の通信処理や同期処理のオーバヘッドによって，SQL 文およびコマンドの実行処理に時間が掛かることがあります。

2.19.2　トランザクションおよびコマンドが実行されるノード

(1)　トランザクションが実行されるノード

トランザクションが実行されるノードは，トランザクションアクセスモードとトランザクション隔離性水準の指定によって決まります。次に示す条件をすべて満たすトランザクションは，セカンダリノードまたはワーカーノードで実行されます。

- トランザクションアクセスモードが読み取り専用モードである
- トランザクション隔離性水準がREAD COMMITTED である

ただし，セカンダリノードまたはワーカーノードのスレッド数の空き状況などによっては，上記の条件を満たしていても，プライマリノードでトランザクションが実行されることがあります。プライマリノードの HADB サーバが負荷の少ないノードを選択し，そのノードでトランザクションが実行されます。

重要

セカンダリノードおよびワーカーノードのリソースを有効に活用するために，検索処理だけを実行する AP は，できる限り次の指定をして実行することを推奨します。

- トランザクションアクセスモードを読み取り専用モードにする
- トランザクション隔離性水準を`READ COMMITTED` にする

トランザクションアクセスモードについては，「2.9.5　トランザクションアクセスモード」を参照してください。

トランザクション隔離性水準については，「2.9.2　HADB がサポートしているトランザクション隔離性水準」を参照してください。

なお，読み書き可能モードのトランザクションがプライマリノードで実行されている間は，すべてのトランザクションがプライマリノードで実行されます（セカンダリノードまたはワーカーノードではトランザクションが実行されません）。そのため，トランザクションアクセスモードが読み取り専用モードであり，かつトランザクション隔離性水準が`READ COMMITTED` であるトランザクションもプライマリノードで実行されます。したがって，セカンダリノードおよびワーカーノードのリソースを有効に活用するには，読み書き可能モードのトランザクションはできる限り細かい頻度で決着させるようにしてください。

重要

マルチノード機能を使用している場合，次に示す条件をすべて満たすトランザクションは，カーソルをクローズした際に自動的にトランザクションが正常終了します。

- トランザクションアクセスモードが読み取り専用モードである
- トランザクション隔離性水準が`READ COMMITTED` である

複数の文ハンドルで同時にカーソルをオープンしている場合は，オープンしているカーソルがすべてクローズされたときに，自動的にトランザクションが正常終了します。

(2)　SQL 文が実行されるノード

SQL 文が実行されるノードを次の表に示します。

表 2-22　SQL 文が実行されるノード

項番	SQL 文の種類	SQL 文が実行されるノード
1	定義系 SQL	プライマリノード

項番	SQL 文の種類			SQL 文が実行されるノード
2	操作系 SQL	更新系 SQL	INSERT 文	プライマリノード
3			UPDATE 文	
4			DELETE 文	
5			TRUNCATE TABLE 文	
6			PURGE CHUNK 文	
7		検索系 SQL	SELECT 文	プライマリノード，セカンダリノード，またはワーカーノードのどれか

■定義系 SQL，TRUNCATE TABLE 文，または PURGE CHUNK 文実行時の留意事項

セカンダリノードまたはワーカーノードでトランザクションが実行されている間は，定義系 SQL，TRUNCATE TABLE 文，またはPURGE CHUNK 文はすぐに実行されません（WAIT 状態になります）。セカンダリノードまたはワーカーノードで実行中のすべてのトランザクションが終了したあとに，定義系 SQL，TRUNCATE TABLE 文，またはPURGE CHUNK 文が実行されます。

また，定義系 SQL，TRUNCATE TABLE 文，またはPURGE CHUNK 文が実行されている間は，すべてのトランザクションがプライマリノードで実行されます（セカンダリノードまたはワーカーノードではトランザクションが実行されません）。

■INSERT 文，UPDATE 文，または DELETE 文実行時の留意事項

次に示すどちらかの条件を満たす場合，INSERT 文，UPDATE 文，またはDELETE 文はすぐに実行されません（WAIT 状態になります）。

- INSERT 文，UPDATE 文，DELETE 文の処理対象表が格納されている DB エリアをアクセスするトランザクションが，セカンダリノードまたはワーカーノードで実行されている
- INSERT 文，UPDATE 文，DELETE 文の処理対象表のインデクスが格納されている DB エリアをアクセスするトランザクションが，セカンダリノードまたはワーカーノードで実行されている

上記のどちらかの条件を満たすトランザクションが終了したあとに，INSERT 文，UPDATE 文，または DELETE 文が実行されます。

また，INSERT 文，UPDATE 文，またはDELETE 文が実行された場合，セカンダリノードまたはワーカーノードで実行中の SQL 文のうち，次のどちらかの条件を満たす SQL 文の処理をプライマリノードに切り替えます。

- INSERT 文，UPDATE 文，DELETE 文の処理対象表が格納されている DB エリアをアクセスする SQL 文
- INSERT 文，UPDATE 文，DELETE 文の処理対象表のインデクスが格納されている DB エリアをアクセスする SQL 文

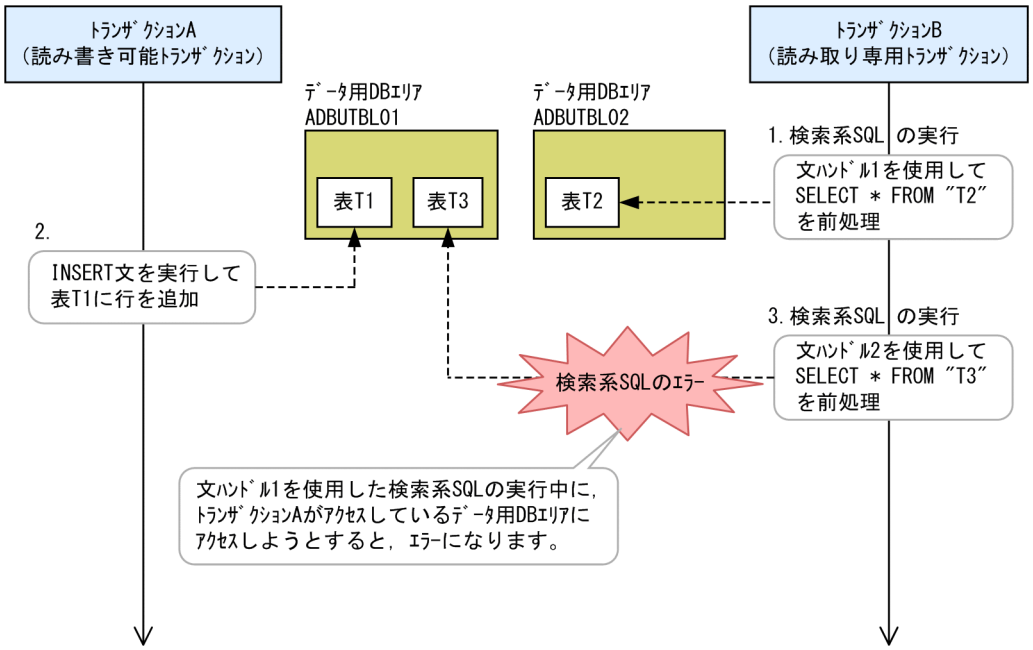
■検索系 SQL がエラーになるケース

マルチノード機能を使用している場合は，次のすべての条件を満たす検索系 SQL は実行できません。

1. 同一トランザクションで，複数の文ハンドルを使用して同時に検索系 SQL を実行しようとしている
2. 検索系 SQL の 1 つが，データ用 DB エリアにアクセスしている最中である

3. 更新系 SQL がアクセスしているデータ用 DB エリアに，2.とは別の検索系 SQL がアクセスしようとした（この検索系 SQL がエラーになります）

（例）



［説明］

読み書き可能トランザクションであるトランザクション A と，読み取り専用トランザクションであるトランザクション B が同時に実行されているとします。

また，トランザクション B では，2 つの文ハンドルを使用して検索系 SQL を同時に実行しようとしています（図中の 1.と 3.）。

1. 表T2 への検索系 SQL の前処理が，文ハンドル 1 を使用して実行されています。
2. トランザクション A によって，表T1 へのINSERT 文が実行されています。
3. 2.のINSERT 文の実行後（トランザクション A は未決着状態），トランザクション B が文ハンドル 2（1.の文ハンドルとは別の文ハンドル）を使用して，表T3 への検索系 SQL の前処理を実行します。

上記の 1.と 3.の処理が同時に実行された場合，3.の検索系 SQL がエラーになります。このとき，KFAA31898-E メッセージが出力されます。対処方法については，KFAA31898-E メッセージで説明されている対処方法に従ってください。

(3) コマンドを実行できるノード

コマンドを実行できるノードは，コマンドによって異なります。コマンドを実行できるノードについては，マニュアル『HADB　コマンドリファレンス』の『マルチノード機能を使用している場合にコマンドを実行できるノード』を参照してください。

(4)　トランザクションとコマンドの同時実行についての制限

トランザクションと一部のコマンドは同時に実行することはできません。トランザクションとコマンドの同時実行可否を次の表に示します。

表 2-23　トランザクションとコマンドの同時実行可否

項番	コマンド名		同時実行可否
1	adbarchivechunk		△
2	adbchgchunkcomment		△
3	adbchgchunkstatus		△
4	adbgetcst		△
5	adbidxrebuild		△
6	adbimport	バックグラウンドインポート	○
7		上記以外のインポート	△
8	adbinit		×
9	adbmergechunk		△
10	adbmodarea		×
11	adbreorgsystemdata		△
12	adbsql		○※
13	adbunarchivechunk		△
14	上記以外のコマンド		○

（凡例）

○：ほかのトランザクションと同時に実行できるコマンドです。

△：ほかのトランザクションとの同時実行に制限があるコマンドです。詳細については，このあとで説明している「■トランザクションとの同時実行に制限があるコマンド」を参照してください。

×：ほかのトランザクションと同時に実行できないコマンドです。

注※

クライアント定義のadb_clt_trn_access_mode オペランドにREAD_ONLY を指定している場合は，セカンダリノードまたはワーカーノードで実行されているほかのトランザクションと同時に実行できます。

■トランザクションとの同時実行に制限があるコマンド

「表 2-23　トランザクションとコマンドの同時実行可否」の「同時実行可否」欄が△のコマンドについては，次に示す制限があります。

- △のコマンドが実行されている間は，すべてのトランザクションがプライマリノードで実行されます（セカンダリノードおよびワーカーノードではトランザクションが実行されません）。そのため，

トランザクションアクセスモードが読み取り専用モードであり，かつトランザクション隔離性水準がREAD COMMITTED のトランザクションもプライマリノードで実行されます。

- セカンダリノードまたはワーカーノードでトランザクションが実行されている間は，△のコマンドはすぐに実行されません（WAIT 状態になります）。セカンダリノードまたはワーカーノードで実行中のすべてのトランザクションが終了したあとに，△のコマンドが実行されます。

2.19.3　ノードの管理

プライマリノードとセカンダリノードの状態は HA モニタによって管理され，ワーカーノードの状態は HADB によって管理されます。

HA モニタの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』を参照してください。

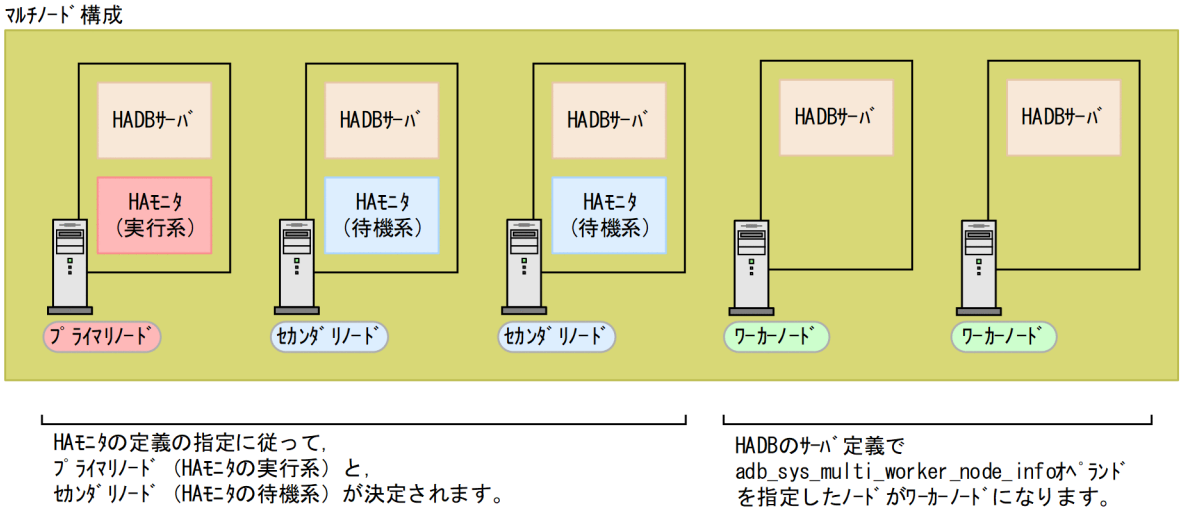
(1)　ノード種別の決定方法

プライマリノードとセカンダリノードは，HA モニタの定義の指定に従って決定されます。HA モニタの環境設定定義のservers ファイルのinitial オペランドにonline を指定すると，そのノードはプライマリノード（HA モニタの実行系）になり，standby を指定すると，そのノードはセカンダリノード（HA モニタの待機系）になります。

ワーカーノードは HA モニタの管理対象外のため，HA モニタの定義を指定しません。HADB のサーバ定義のadb_sys_multi_worker_node_info オペランドを指定したノードがワーカーノードになります。

ノード種別の決定方法を次の図に示します。

図 2-69　ノード種別の決定方法



メモ

- プライマリノードおよびセカンダリノードを開始するには，プライマリノードおよび各セカンダリノードでadbstart コマンドを実行し，そのあと HA モニタのmonbegin コマンドを実行します。
- ワーカーノードを開始するには，プライマリノードおよびセカンダリノードの開始処理の完了後に各ワーカーノードでadbstart コマンドを実行します。

■セカンダリノードとワーカーノードの差異（ワーカーノードの特徴）

セカンダリノードとワーカーノードは，読み取り専用モードのトランザクションが実行されるという同じ特徴を持っていますが，仕様が異なる部分もあります。ここでは，セカンダリノードとワーカーノードの主な差異と，ワーカーノードの特徴を説明します。

- ワーカーノードは，プライマリノードの切り替え対象のノードにはならない

  セカンダリノードとは異なり，ワーカーノードには HA モニタをインストールしません。そのため，ワーカーノードは，プライマリノードの切り替え対象のノードにはなりません。

- マルチノード構成にワーカーノードを動的に追加または削除できる

  マルチノード構成の HADB サーバが稼働中のときに，マルチノード構成にワーカーノードを動的に追加または削除できます。例えば，マルチノード構成の HADB サーバの開始時には，プライマリノードと全セカンダリノードの HADB サーバだけを開始し，任意のタイミングでワーカーノードの HADB サーバを開始してマルチノード構成にワーカーノードを追加することができます。これによって，システムの負荷が大きくなる時間帯は，ワーカーノードを追加してトランザクションの処理性能を増強し，その時間帯が過ぎたあとは，ワーカーノードを削除するような運用が可能になります。

(2) ノード障害の監視

HA モニタは，次の表に示す障害がプライマリノードとセカンダリノードに発生していないかどうかを監視しています。HA モニタが，これらの障害の発生を検知した場合，障害が発生したプライマリノードまたはセカンダリノードを HA モニタの制御によってマルチノード構成から切り離します。

表 2-24　HA モニタが監視している障害

項番	HA モニタが監視している障害
1	ハードウェア障害
2	ハードウェアの電源断
3	カーネル障害
4	HA モニタの異常終了
5	HADB サーバの異常終了（サーバプロセスの異常終了）

上記の表に示す障害を**ノード障害**といいます。

■マニュアル『HA モニタ Linux(R)(x86)編』を読む際のポイント

HA モニタの監視対象となる障害には，系障害（上記の表の項番 1～4 の障害）とサーバ障害（上記の表の項番 5 の障害）があります。マルチノード機能は，HA モニタのモニタモードで動作するため，サーバ障害を検知する場合は，サーバの監視コマンドの作成が必要になります。サーバの監視コマンドの作成方法については，「16.3.4　HA モニタの環境設定」の「(4)　コマンド用環境変数定義およびコマンドの作成」の「(e)　サーバの監視コマンドの作成」を参照してください。

なお，ワーカーノードについては，HA モニタによる障害の監視は行われません。そのため，ワーカーノードでノード障害が発生して，ワーカーノードの HADB サーバが異常終了した場合，ワーカーノードを HADB の制御によってマルチノード構成から削除します。

(3)　ノード障害が発生したときのシステムの処理

「(2)　ノード障害の監視」で説明したノード障害が発生した場合，HA モニタの機能を使用して，障害が発生したノード（プライマリノードまたはセカンダリノード）をマルチノード構成から切り離し，残ったノードで業務を継続します。プライマリノードまたはセカンダリノードにノード障害が発生したときのノードの切り離し処理は，HA モニタが管理しています。詳細については，「2.19.4　マルチノード構成からのノードの切り離し（プライマリノード，セカンダリノード）」を参照してください。

ワーカーノードでノード障害が発生し，ワーカーノードの HADB サーバが異常終了した場合，HADB の制御によってワーカーノードをマルチノード構成から削除します。詳細については，「2.19.6　マルチノード構成からのワーカーノードの削除と追加」を参照してください。

2.19.4　マルチノード構成からのノードの切り離し（プライマリノード，セカンダリノード）

プライマリノードまたはセカンダリノードでノード障害が発生した場合，障害が発生したノードをマルチノード構成から切り離し，残ったノードで業務を継続します。

メモ

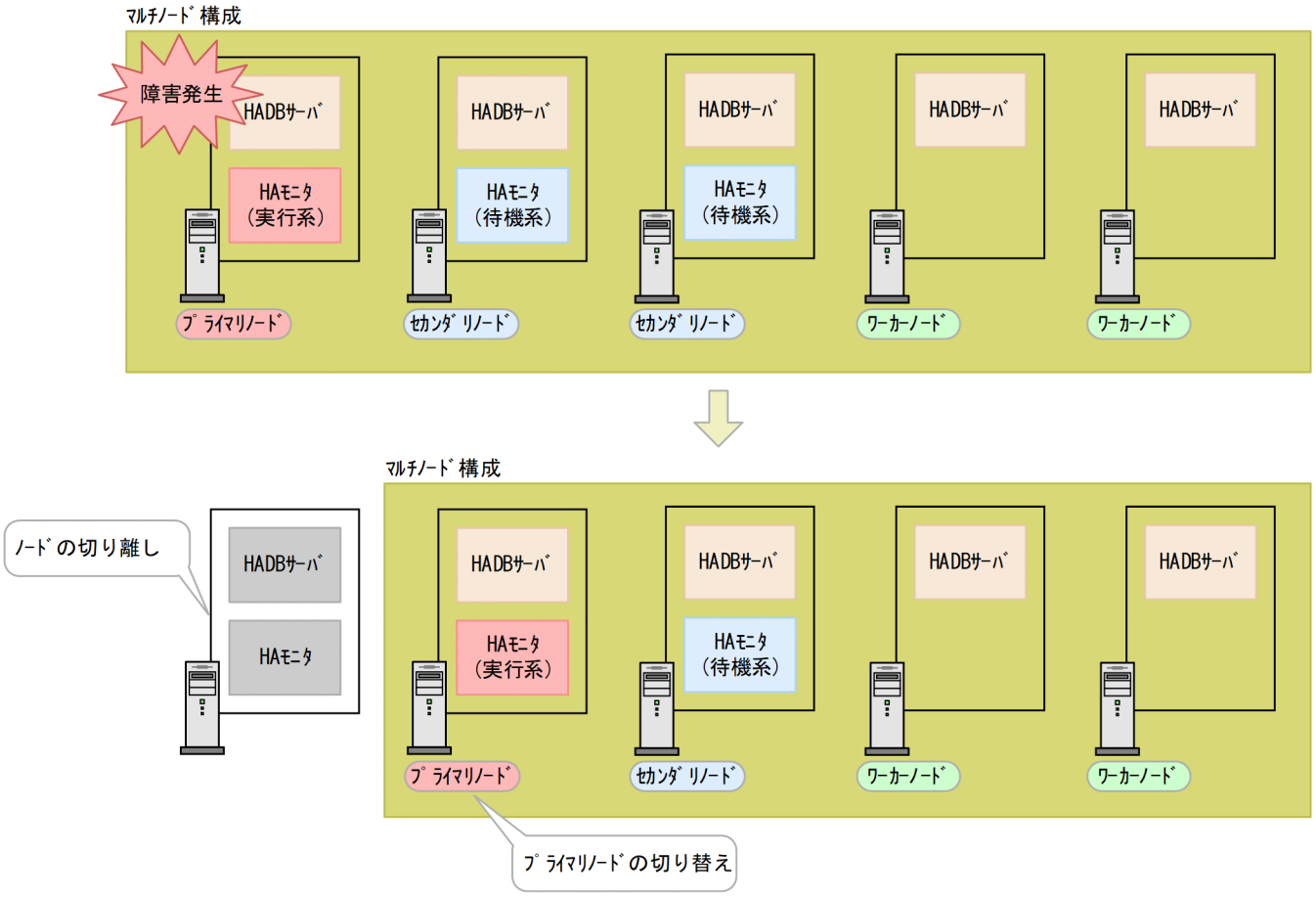
- ノードの切り離しは，HA モニタによって制御されます。
- 障害が発生したプライマリノードまたはセカンダリノードをマルチノード構成から切り離し，残ったノードで処理を継続することを，HA モニタでは縮退運用と呼んでいます。

(1)　プライマリノードでノード障害が発生した場合

プライマリノードでノード障害が発生した場合，そのプライマリノードをマルチノード構成から切り離し，セカンダリノードの 1 つをプライマリノードに切り替えて，残ったノードで業務を継続します。

プライマリノードでノード障害が発生した場合のノードの切り離しを次の図に示します。

図 2-70　プライマリノードでノード障害が発生した場合のノードの切り離し



[説明]

- ノード障害が発生したプライマリノードをマルチノード構成から切り離し，セカンダリノードの 1 つをプライマリノードに切り替えます。これをプライマリノードの切り替えといいます。

  重要

  ワーカーノードは，プライマリノードの切り替え先にはなりません。

- ノード障害が発生したプライマリノードで実行中の SQL 文はエラーになり，トランザクションを実行していた AP は，HADB サーバから切り離されます。また，ノード障害が発生した時点で，HADB サーバに接続だけしていてトランザクションを実行していない AP も，HADB サーバから切り離されます。

メモ

障害が発生したプライマリノードで実行されていた更新処理に対する回復処理（ロールバック）が，新たにプライマリノードになるセカンダリノードで実行されます。回復処理の完了後，プライマリノードの切り替え処理が行われます。

ノード障害発生時のノードの切り離し，およびプライマリノードの切り替えは，HA モニタが管理しています（HA モニタの系切り替え機能で実現しています）。プライマリノードは HA モニタの実行系として動作し，セカンダリノードは HA モニタの待機系として動作します。

セカンダリノードの 1 つをプライマリノードに切り替える際，共有するディスクの切り替えや，エイリアス IP アドレスの追加および削除も，HA モニタが実行します。

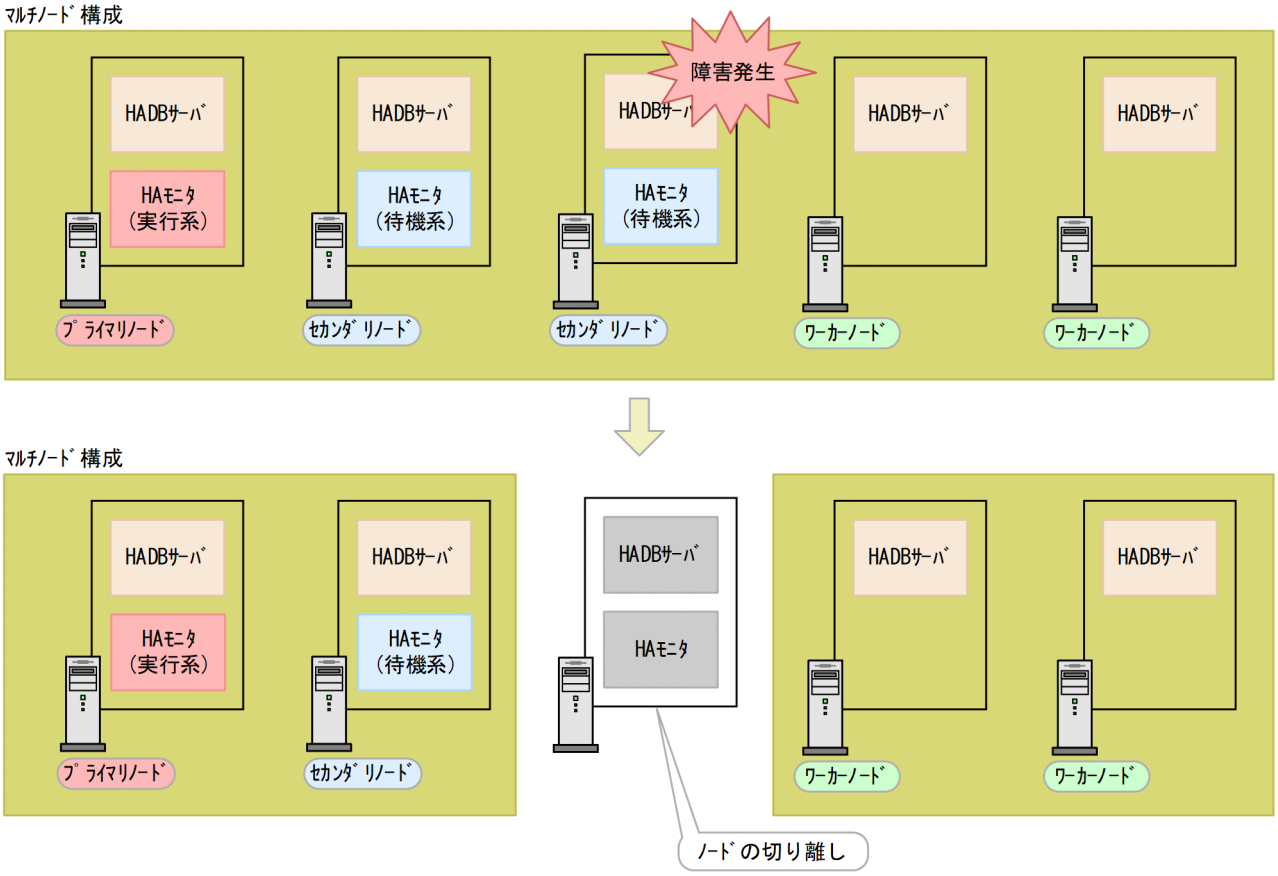
> **メモ**
>
> プライマリノードに切り替えられるセカンダリノードは，HA モニタの系切り替え機能によって，新たに実行系となるノードです。系の切り替え先（プライマリノードに切り替えられるセカンダリノード）の決定方法は HA モニタの仕様に依存します。

(2)　セカンダリノードでノード障害が発生した場合

セカンダリノードでノード障害が発生した場合，そのセカンダリノードをマルチノード構成から切り離し，残ったノードで業務を継続します。

セカンダリノードでノード障害が発生した場合のノードの切り離しを次の図に示します。

図 2-71　セカンダリノードでノード障害が発生した場合のノードの切り離し



［説明］

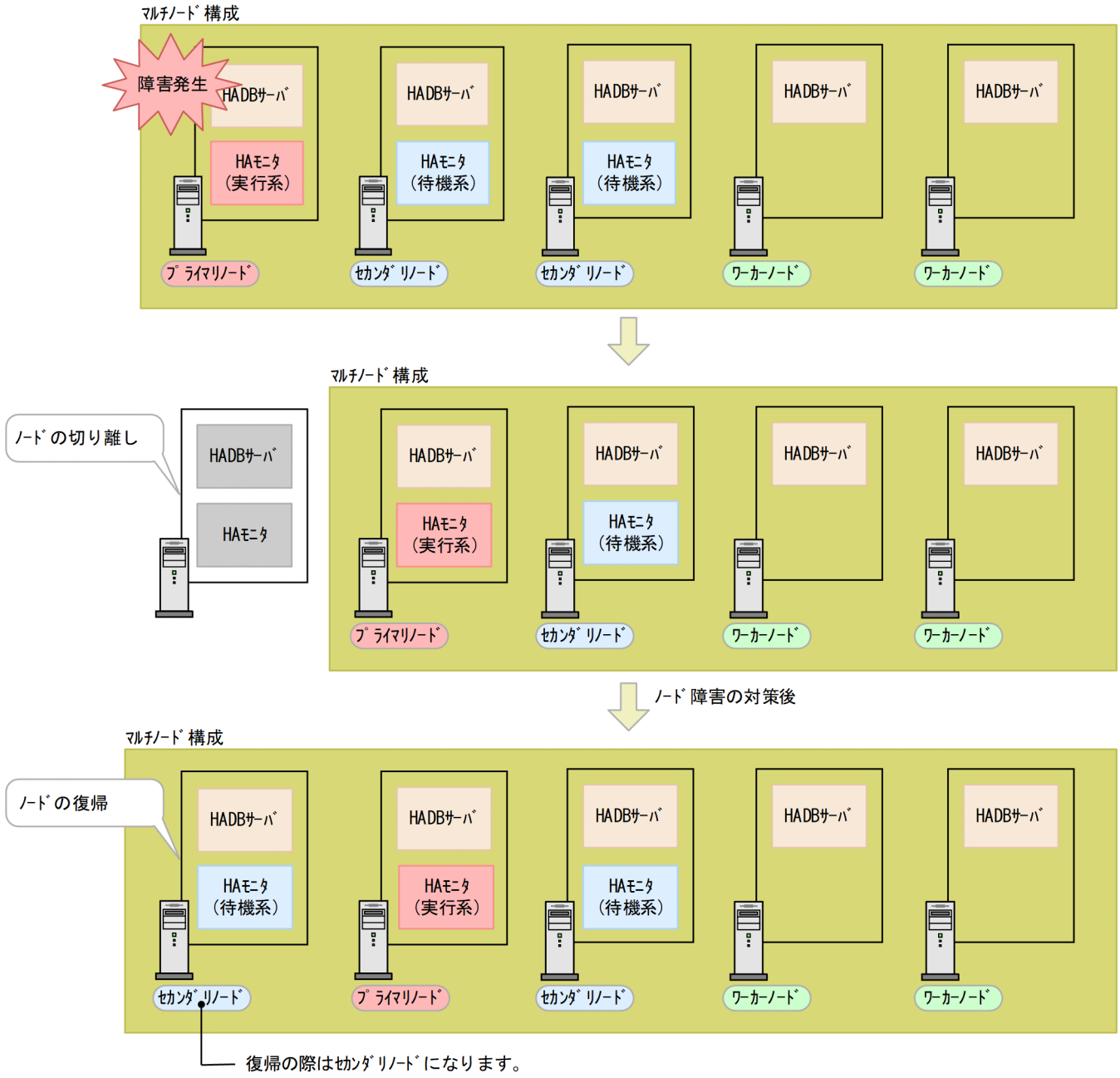
ノード障害が発生したセカンダリノードで実行中の SQL 文はエラーになり，トランザクションを実行していた AP は，HADB サーバから切り離されます。

> **メモ**
>
> 障害が発生したセカンダリノードで実行していたトランザクションのロールバックが，プライマリノードで実行されます。

2.19.5 マルチノード構成へのノードの復帰（プライマリノード，セカンダリノード）

ノード障害が発生してマルチノード構成から切り離されたノード（プライマリノードまたはセカンダリノード）は，障害対策後にマルチノード構成に復帰させることができます。その際，マルチノード構成の HADB サーバを停止する必要はありません。マルチノード構成へのノードの復帰の例を次の図に示します。

図 2-72　マルチノード構成へのノードの復帰の例



[説明]

マルチノード構成に復帰する際，ノード種別は必ずセカンダリノードになります。

マルチノード構成にノードを復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

メモ

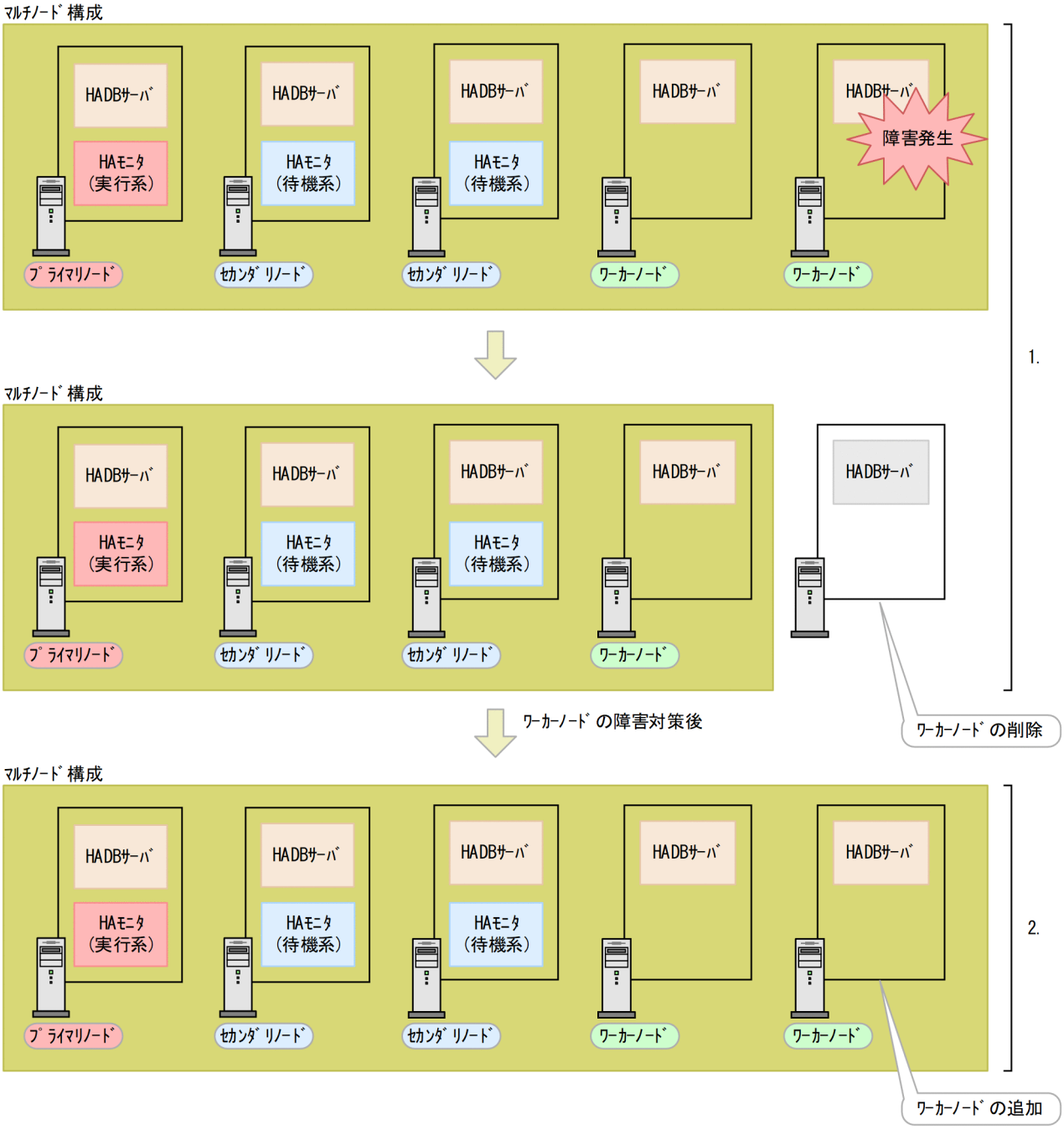
サーバマシンのメンテナンスをする場合は，HA モニタのコマンドを実行して，マルチノード構成からノード（プライマリノードまたはセカンダリノード）を切り離すことができます。サーバマシンのメンテナンスの終了後に，マルチノード構成に復帰させることができます。

2.19.6　マルチノード構成からのワーカーノードの削除と追加

ワーカーノードでノード障害が発生して HADB サーバが異常終了した場合，そのワーカーノードはマルチノード構成から削除されます。障害を対策したあとに，そのワーカーノードをマルチノード構成に再び追加することができます。

マルチノード構成からのワーカーノードの削除と追加を次の図に示します。

図 2-73　マルチノード構成からのワーカーノードの削除と追加



［説明］

1. HADB サーバが異常終了したワーカーノードはマルチノード構成から削除されます。ワーカーノードで実行中の SQL 文はエラーになり，トランザクションを実行していた AP は，HADB サーバから切り離されます。

   残った 4 つのノードで業務は継続されます。

2. 障害を対策したあとに，ワーカーノードをマルチノード構成に再び追加します。

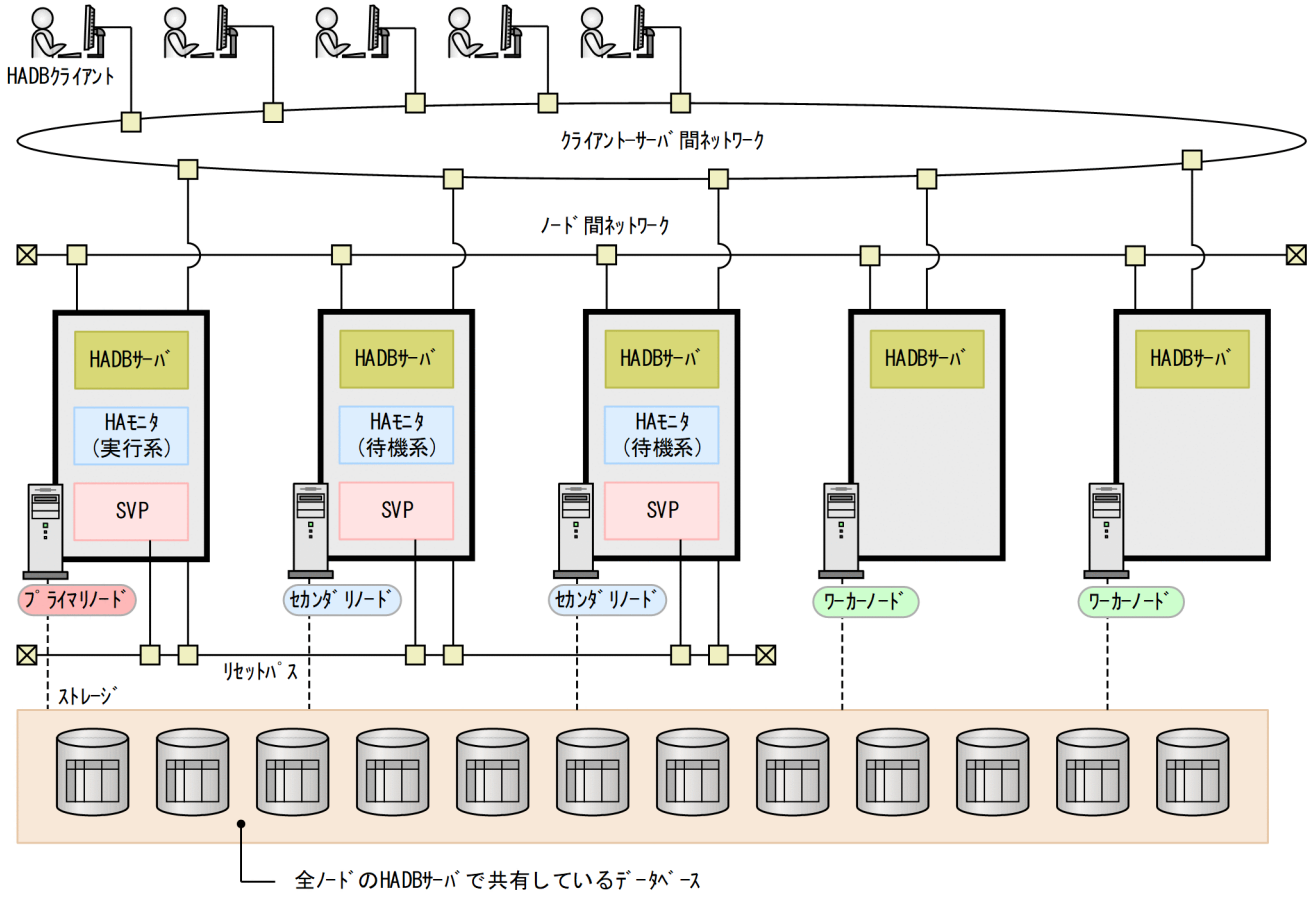
メモ

- HADB サーバが異常終了したワーカーノードで実行していたトランザクションのロールバックが，プライマリノードで実行されます。
- ワーカーノードは HA モニタの管理対象外のため，マルチノード構成からのワーカーノードの削除と追加は HADB によって制御されます。

2.19.7　マルチノード機能を使用する場合のシステム構成例

マルチノード機能を使用する場合のシステム構成例を次の図に示します。

図 2-74　マルチノード機能を使用する場合のシステム構成例



［説明］

- 各ノードに必要な製品

  プライマリノードおよびセカンダリノードには，HADB サーバおよび HA モニタをインストールする必要があります。

  ワーカーノードには，HADB サーバをインストールする必要があります。

- 必要なネットワーク

  マルチノード機能では，次に示す 3 つのネットワークを使用します。各ネットワークは物理的に分ける必要があります。

  - クライアントーサーバ間ネットワーク

    HADB クライアントと，プライマリノード，セカンダリノード，およびワーカーノードの HADB サーバ間の通信で使用するネットワークです。

  - ノード間ネットワーク

    各ノードの HADB サーバ間の通信，および HA モニタの監視パスとして使用するネットワークです。

  - リセットパス

    HA モニタが障害を検知した際，障害が発生したノードの入出力をリセットするために使用するネットワークです。HA モニタの機能である系のリセットを使用する場合は，プライマリノードとセカンダリノード間でリセットパスが必要になります。

- プライマリノードおよびセカンダリノードに必要なハードウェア

  系のリセットを使用する場合は，プライマリノードおよびセカンダリノードに SVP が必要になります。SVP とは障害管理プロセサのことです。障害管理プロセサとは，システム構成の設定や CPU の制御などができるハードウェアのことです。

監視パス，リセットパス，SVP，および障害管理プロセサの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『必要なハードウェア』を参照してください。

重要

HA モニタによる共有ディスクのデータ保護方法には，次のどちらかを選択する必要があります。

- 系のリセット
- 共有ディスクの SCSI リザーブ

推奨する方法は，系のリセットです。系のリセットを使用する場合は，プライマリノードおよびセカンダリノードのサーバマシンに SVP が搭載されている必要があります。

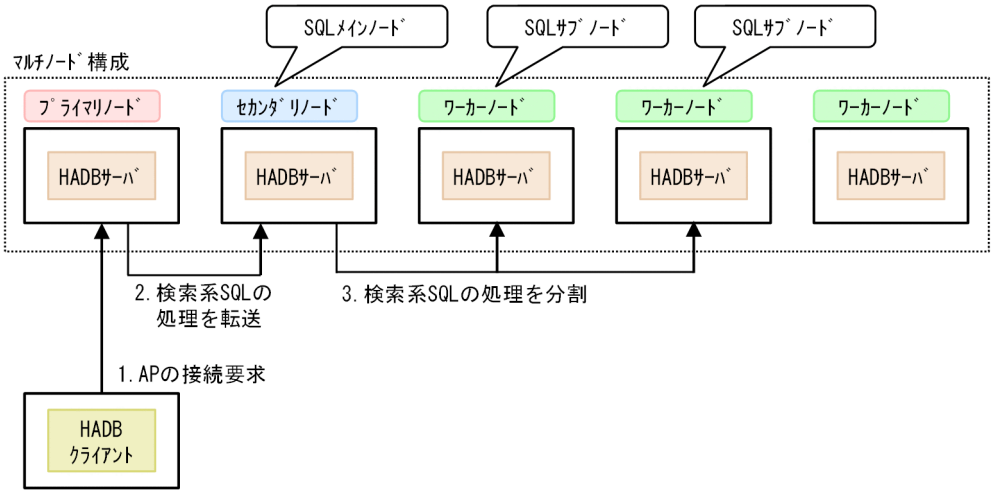
共有ディスクの SCSI リザーブを使用する場合は，SVP およびリセットパスは必要ありません。プライマリノードおよびセカンダリノードのサーバマシンに SVP が搭載されていない場合は，共有ディスクの SCSI リザーブを選択してください。

系のリセットおよび共有ディスクの SCSI リザーブの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『系のリセット』および『共有ディスクの SCSI リザーブ』を参照してください。

2.19.8　SQL パラレル実行機能

検索系 SQL（SELECT 文）の処理を複数のノードで並列実行できます。この機能を **SQL パラレル実行機能**といいます。この機能を使用することによって，検索系 SQL の処理時間が短縮できます。SQL パラレル実行機能の概要を次の図に示します。

図 2-75　SQL パラレル実行機能の概要



[説明]

上記の図の場合，3 つのノード（セカンダリノードと 2 つのワーカーノード）で検索系 SQL の処理を並列実行しています。

1. HADB クライアントからの AP の接続要求は，プライマリノードの HADB サーバが受け付けます。
2. プライマリノードの HADB サーバが，検索系 SQL を実行するトランザクションを割り当てるノードを決定します。トランザクションが割り当てられたノードを **SQL メインノード**といいます。
3. SQL メインノードの HADB サーバが検索系 SQL の処理を分割して，負荷の少ないノードに処理を依頼します。検索系 SQL の処理を割り振られたノードを **SQL サブノード**といいます。

メモ

- SQL メインノードと SQL サブノードで検索系 SQL の処理が並列実行されます。検索結果は，SQL メインノードから HADB クライアントに返されます。
- 各ノードの負荷の状況などを踏まえて，プライマリノードが SQL メインノードと SQL サブノードを決定します。上記の例では，セカンダリノードが SQL メインノードになっていますが，ワーカーノードが SQL メインノードになることもあります。

- 各ノードの負荷の状況によっては，プライマリノードが SQL メインノードまたは SQL サブノードになることもあります。
- SQL サブノードの数は，検索系 SQL を実行するときの各ノードの負荷によって変わります。負荷が高いノードは，SQL サブノードに選ばれません。そのため，負荷が高いノードがある場合は，検索系 SQL を並列実行するノード数が少なくなります。SQL サブノードが 1 つも確保できない場合は，検索系 SQL がエラーになります（KFAA31898-E メッセージが出力されます）。

■SQL パラレル実行機能が適用される検索系 SQL の条件

SQL パラレル実行機能が適用される検索系 SQL については条件があります。条件については，マニュアル『HADB　AP 開発ガイド』の『SQL パラレル実行機能が適用される条件』を参照してください。

■SQL パラレル実行機能を適用すると効果的なケース

大量のデータを検索するときに，SQL パラレル実行機能を適用すると効果があります。例えば，次のどれかの条件に該当する表を検索するときに SQL パラレル実行機能を適用すると，検索処理時間の短縮が見込まれます。

- データ容量が大きい表
- 1 回のデータインポートで格納するデータの容量が大きい表
  目安として，入力データファイルのサイズが，「1 ギガバイト×*ノード数*」以上の表が該当します。
- 検索対象のデータをあまり絞り込まないでアクセスすることが多い表

■SQL パラレル実行機能を使用するときの準備作業および運用

SQL パラレル実行機能を使用するときの準備作業および運用については，「16.29　SQL パラレル実行機能の運用」を参照してください。

■留意事項

次の 1.～3.のどれかの条件に該当する場合，SQL パラレル実行機能を適用した検索系 SQL はエラーになります（KFAA31898-E メッセージが出力されます）。

1. 読み書き可能トランザクションが実行中の場合
2. 次のどれかのコマンドが実行中の場合
   - adbimport コマンド（-b オプションを指定したバックグラウンドインポートは除きます）
   - adbidxrebuild コマンド
   - adbgetcst コマンド
   - adbmergechunk コマンド
   - adbchgchunkcomment コマンド
   - adbchgchunkstatus コマンド
   - adbarchivechunk コマンド
   - adbunarchivechunk コマンド

- adbreorgsystemdata コマンド

3. INSERT 文，UPDATE 文，またはDELETE 文が実行中であり，かつ次のどちらかの条件を満たす場合
   - INSERT 文，UPDATE 文，またはDELETE 文の処理対象表が格納されているデータ用 DB エリアと，検索系 SQL の処理対象表が格納されているデータ用 DB エリアが同じ場合
   - INSERT 文，UPDATE 文，またはDELETE 文の処理対象表のインデクスが格納されているデータ用 DB エリアと，検索系 SQL の処理対象表のインデクスが格納されているデータ用 DB エリアが同じ場合

2.20　HADB サーバのメモリ構成

ここでは，HADB サーバが使用するメモリの構成と，サーバ定義のオペランドとの関係について説明します。

2.20.1　HADB サーバが使用するメモリ

HADB サーバは，共有メモリとプロセスメモリを使用します。HADB サーバが使用する各メモリについて説明します。

■共有メモリ

共有メモリ内には，次に示すメモリが確保されます。

- **プロセス共通メモリ**

  プロセス共通メモリには，HADB サーバプロセスを起動，制御するために必要な情報が格納されます。

- **共有メモリ管理用のメモリ**

  共有メモリ管理用のメモリには，HADB サーバが使用する共有メモリの管理情報が格納されます。

- **グローバルバッファ用のメモリ**

  グローバルバッファ用のメモリには，次に示すバッファが格納されます。

  ・データベースの入出力処理で使用されるグローバルバッファ

  ・グローバル作業表の入出力処理で使用されるグローバル作業表用のグローバルバッファ

  > **メモ**
  >
  > グローバル作業表とは，1 つの SQL 文を複数の処理リアルスレッドで処理する際に，作業表のデータを複数の処理リアルスレッド間で共有するために作成される作業表のことです。

- **リアルスレッド固有メモリ**

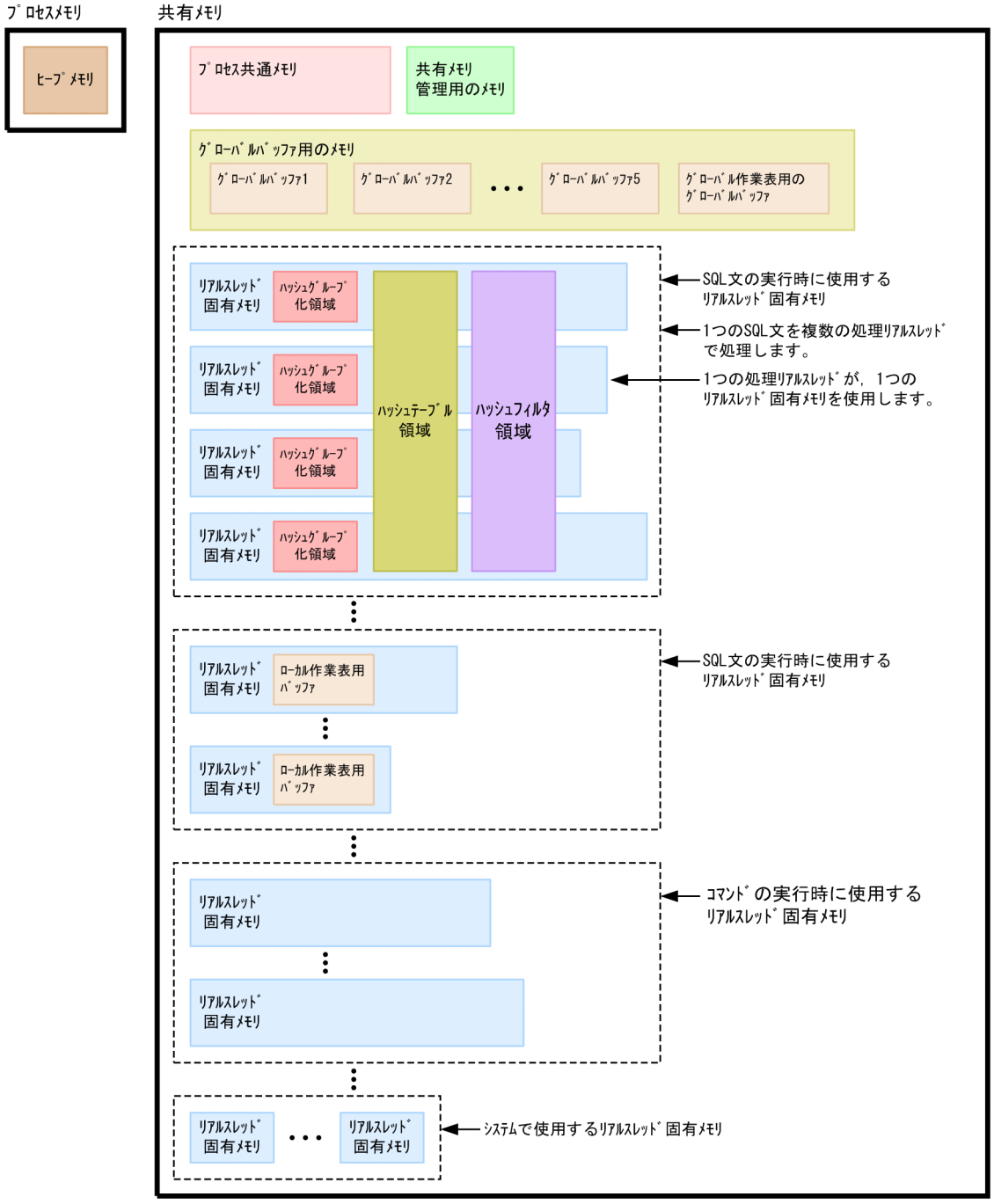
  リアルスレッド固有メモリには，SQL 文またはコマンドを処理する処理リアルスレッドが必要とする情報が格納されます。1 つの処理リアルスレッドに対して，1 つのリアルスレッド固有メモリが確保されます。HADB サーバプロセス内では，複数の処理リアルスレッドが実行され，リアルスレッド固有メモリには，処理リアルスレッドごとに必要な情報が格納されます。

■プロセスメモリ

プロセスメモリ内には，ヒープメモリが確保されます。HADB サーバは，コマンドの実行時にヒープメモリを使用します。

HADB サーバのメモリ構成を次の図に示します。

図 2-76　HADB サーバのメモリ構成



[説明]

- 共有メモリ内には，複数のリアルスレッド固有メモリが確保されます。
- 1 つの SQL 文または 1 つのコマンドは，複数の処理リアルスレッドで処理されます。このとき，1 つの処理リアルスレッドが，1 つのリアルスレッド固有メモリを使用します。
- SQL 文およびコマンドの実行時に使用するリアルスレッド固有メモリ以外に，システムで使用するリアルスレッド固有メモリが確保されます。

リアルスレッド固有メモリ内に確保される領域とバッファについて説明します。

- **ハッシュグループ化領域**

  ハッシュグループ化領域には，ローカルハッシュグループ化で使用されるハッシュテーブルが格納されます。ローカルハッシュグループ化が適用される SQL 文を実行した場合，各リアルスレッド固有メモリにハッシュグループ化領域が確保されます。

- **ハッシュテーブル領域**

  ハッシュテーブル領域には，複数の処理リアルスレッド間で共有するハッシュテーブルが格納されます。次に示す処理が適用される SQL 文を実行した場合に，ハッシュテーブル領域が確保されます。

  - 表の結合方式のハッシュジョイン
  - グループ化の処理方式のグローバルハッシュグループ化
  - 副問合せの処理方式のハッシュ実行
  - SELECT DISTINCT の処理方式のハッシュ実行
  - 集合演算の処理方式のハッシュ実行

- **ハッシュフィルタ領域**

  ハッシュフィルタ領域には，複数の処理リアルスレッド間で共有する，ハッシュ検索で使用するハッシュフィルタが格納されます。次に示す処理が適用される SQL 文を実行した場合に，ハッシュフィルタ領域が確保されます。

  - 表の結合方式のハッシュジョイン
  - 副問合せの処理方式のハッシュ実行

  メモ

  ハッシュフィルタについては，マニュアル『HADB　AP 開発ガイド』の『表の結合方式』または『副問合せの処理方式』を参照してください。

- **ローカル作業表用バッファ**

  ローカル作業表用バッファとは，ローカル作業表の入出力処理で使用されるバッファのことです。ローカル作業表とは，処理リアルスレッドごとに作成される作業表のことです。ORDER BY 句を指定した場合や，ローカルハッシュグループ化のグループ分けの場合に，ローカル作業表が作成されます。

メモ

次の処理については，マニュアル『HADB　AP 開発ガイド』の該当個所を参照してください。

- ローカルハッシュグループ化，グローバルハッシュグループ化：『グループ化の処理方式』
- ハッシュジョイン：『表の結合方式』
- 副問合せの処理方式のハッシュ実行：『副問合せの処理方式』
- SELECT DISTINCT の処理方式のハッシュ実行：『SELECT DISTINCT の処理方式』
- 集合演算の処理方式のハッシュ実行：『集合演算の処理方式』

2.20.2　サーバ定義のオペランドとの関係

HADB サーバが使用する各種メモリや領域のサイズを，サーバ定義のオペランドで指定します。ここでは，HADB サーバが使用するメモリとサーバ定義のオペランドの関係を説明します。

HADB サーバが使用するメモリの最大値をサーバ定義で指定する際，次に示すどちらかの方法があります。HADB では，1 の方法で指定することを推奨します。

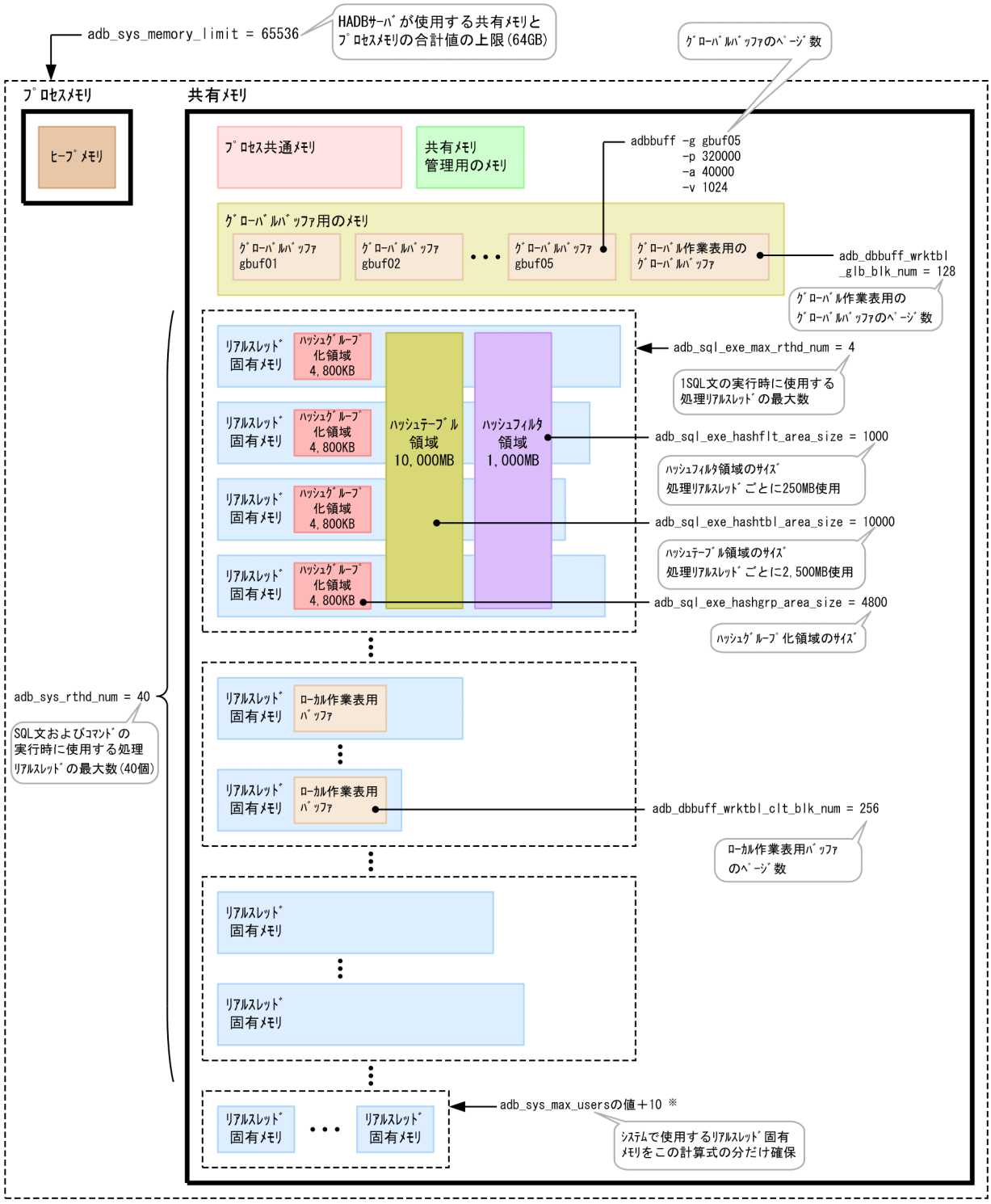
1. HADB サーバが使用するメモリ所要量の最大値（共有メモリとプロセスメモリの合計値の上限）を adb_sys_memory_limit オペランドに指定する
2. プロセス共通メモリの最大値をadb_sys_proc_area_max オペランドに指定し，リアルスレッド固有メモリの最大値をadb_sys_rthd_area_max オペランドに指定する

上記のそれぞれの方法ごとに，HADB サーバが使用するメモリとサーバ定義のオペランドの関係を説明します。

(1)　1 の方法の場合（adb_sys_memory_limit オペランドを指定する場合）

HADB サーバが使用するメモリとサーバ定義のオペランドの関係を次の図に示します。

図 2-77　HADB サーバが使用するメモリとサーバ定義のオペランドの関係
(adb_sys_memory_limit オペランドを指定する場合)



注

上記の図のサーバ定義のオペランドの数値は例です。見積もった値を指定してください。

注※

マルチノード機能を使用する場合は，リアルスレッド固有メモリが次の計算式の分だけ確保されます。

- ワーカーノードを使用しない場合

  adb_sys_max_usersオペランドの値
  ＋*セカンダリノード数*×（*adb_sys_max_parallel_exec_numオペランドの値*×2＋2）
  ＋15

- ワーカーノードを使用する場合
  - プライマリノードまたはセカンダリノードの場合

    adb_sys_max_usersオペランドの値
    ＋（*セカンダリノード数*＋*adb_sys_multi_max_worker_nodesオペランドの値*）
    ×（*adb_sys_max_parallel_exec_numオペランドの値*×2＋2）
    ＋15

  - ワーカーノードの場合

    adb_sys_max_usersオペランドの値
    ＋*セカンダリノード数*×（*adb_sys_max_parallel_exec_numオペランドの値*×2＋2）
    ＋*adb_sys_multi_max_worker_nodesオペランドの値*
    ×*adb_sys_max_parallel_exec_numオペランドの値*×2
    ＋17

重要

HADB サーバが使用するメモリ所要量の最大値（共有メモリとプロセスメモリの合計値の上限）をサーバ定義のadb_sys_memory_limit オペランドに指定します。

上記の図中の各メモリや領域とサーバ定義の関係を次の表に示します。

表 2-25　HADB サーバの各メモリおよび各領域の説明（adb_sys_memory_limit オペランドを指定する場合）

メモリまたは領域の種類		関係するサーバ定義のオペランド	見積もり方法の記載個所	HADB サーバの稼働中にメモリの使用量を確認する方法
共有メモリ	プロセス共通メモリ	－	「6.3.1　共有メモリの所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※2 を実行したときに出力されるPROCESS_USE の値で確認できます。
	共有メモリ管理用のメモリ	－	「6.3.1　共有メモリの所要量の求め方」の「(1)　共有メモリ管理の所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※2 を実行したときに出力されるOTHER_USE の値で確認できます。
	グローバルバッファ用のメモリ	• adbbuff グローバルバッファのページ数などを指定するオペランドです。	• 「6.3.1　共有メモリの所要量の求め方」の「(2)　グローバルバッ	-a オプションを指定して adbls -d mem コマンド※2 を実行したときに出力さ

メモリまたは領域の種類			関係するサーバ定義のオペランド	見積もり方法の記載個所	HADB サーバの稼働中にメモリの使用量を確認する方法
			• adb_dbbuff_wrktbl_glb_blk_num グローバル作業表用のグローバルバッファのページ数を指定するオペランドです。	ファページの所要量の求め方」 • 「6.25.1　グローバル作業表用のグローバルバッファのページ数の見積もり」	れるGBUFFER_USE の値で確認できます。
	リアルスレッド固有メモリ	―	• adb_sys_rthd_num SQL 文およびコマンドの実行時に使用する処理リアルスレッドの最大数を指定するオペランドです。1 つの処理リアルスレッドに対して，1 つのリアルスレッド固有メモリが確保されます。 • adb_sql_exe_max_rthd_num※1 1SQL 文の実行時に使用する処理リアルスレッドの最大数を指定します。	「6.3.1　共有メモリの所要量の求め方」の「(4)　リアルスレッド固有メモリの所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※2 を実行したときに出力されるTHREAD_USE の値で確認できます。
		ハッシュグループ化領域	• adb_sql_exe_hashgrp_area_size※1 1 つ当たりのハッシュグループ化領域のサイズを指定するオペランドです。	「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashgrp_area_size オペランド	
		ハッシュテーブル領域	• adb_sql_exe_hashtbl_area_size※1 ハッシュテーブル領域のサイズを指定するオペランドです。	「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashtbl_area_size オペランド	
		ハッシュフィルタ領域	• adb_sql_exe_hashflt_area_size※1 ハッシュフィルタ領域のサイズを指定するオペランドです。	「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashflt_area_size オペランド	
		ローカル作業表用バッファ	• adb_dbbuff_wrktbl_clt_blk_num※1 ローカル作業表用バッファのページ数を指定するオペランドです。	「6.25.2　ローカル作業表用バッファのページ数の見積もり」	

メモリまたは領域の種類		関係するサーバ定義のオペランド	見積もり方法の記載個所	HADB サーバの稼働中にメモリの使用量を確認する方法
プロセスメモリ	ヒープメモリ	－	「6.3.2　プロセスメモリの所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※2 を実行したときに出力されるHEAP_USE の値で確認できます。

（凡例）

－：該当なし

注※1

クライアント定義の同じオペランドで，サーバ定義で指定した値を変更できます。

注※2

-a オプションを指定して実行したadbls -d mem コマンドの出力項目については，マニュアル『HADB コマンドリファレンス』の『adbls -d mem（メモリの使用状況の表示)』を参照してください。

メモ

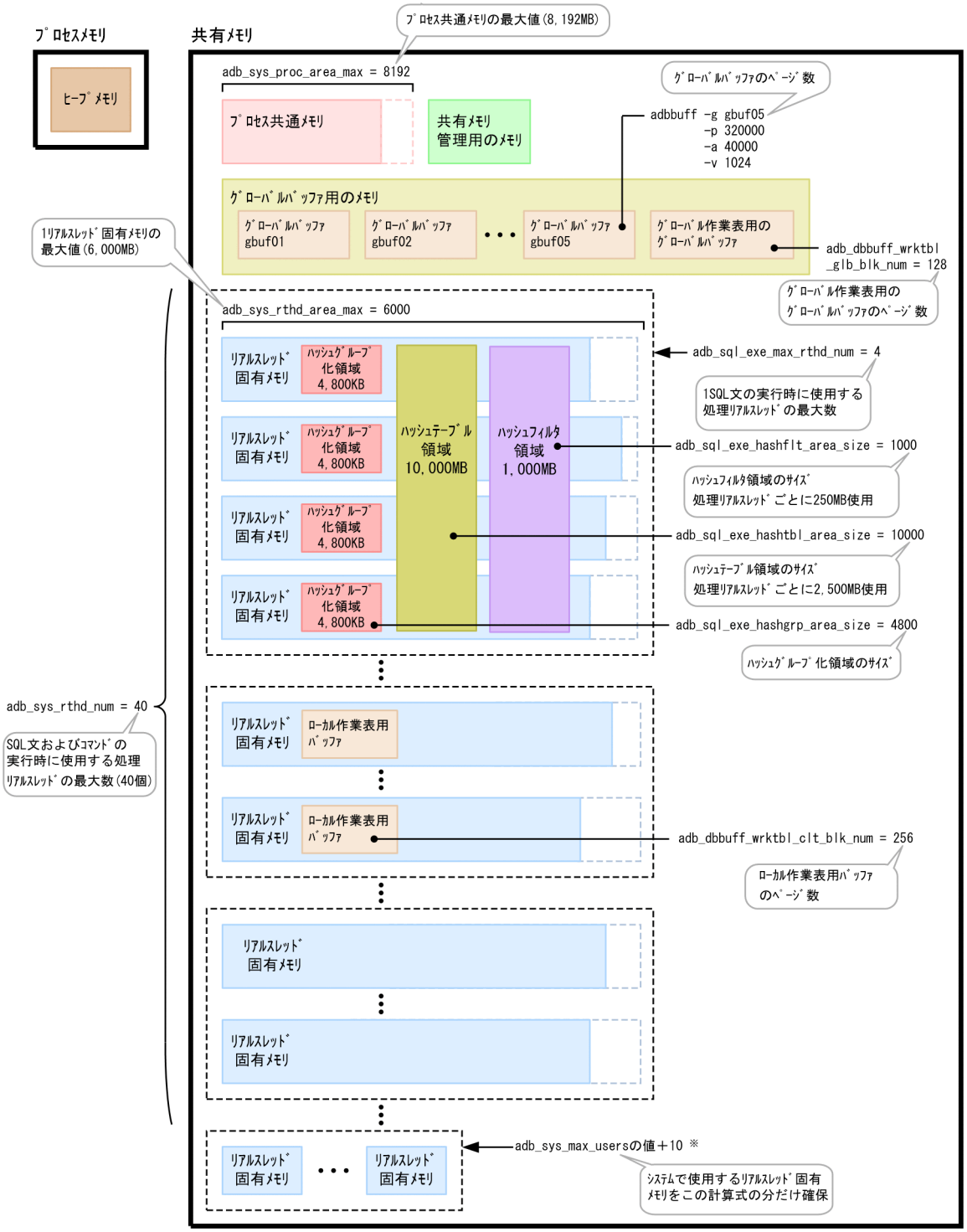
HADB サーバが使用するメモリ所要量の見積もり方法については，次の個所を参照してください。

- 「4.3　メモリ所要量の事前見積もり」
- 「6.3　HADB サーバのメモリ所要量の見積もり」

(2)　2 の方法の場合（adb_sys_proc_area_max および adb_sys_rthd_area_max オペランドを指定する場合）

HADB サーバが使用するメモリとサーバ定義のオペランドの関係を次の図に示します。

図 2-78　HADB サーバが使用するメモリとサーバ定義のオペランドの関係
(adb_sys_proc_area_max および adb_sys_rthd_area_max オペランドを指定する場合)



注

上記の図のサーバ定義のオペランドの数値は例です。見積もった値を指定してください。

注※

マルチノード機能を使用する場合は，リアルスレッド固有メモリが次の計算式の分だけ確保されます。

- ワーカーノードを使用しない場合

  adb_sys_max_usersオペランドの値
  ＋*セカンダリノード数*×（*adb_sys_max_parallel_exec_numオペランドの値*×2＋2）
  ＋15

- ワーカーノードを使用する場合
  - プライマリノードまたはセカンダリノードの場合

    adb_sys_max_usersオペランドの値
    ＋（*セカンダリノード数*＋*adb_sys_multi_max_worker_nodesオペランドの値*）
    ×（*adb_sys_max_parallel_exec_numオペランドの値*×2＋2）
    ＋15

  - ワーカーノードの場合

    adb_sys_max_usersオペランドの値
    ＋*セカンダリノード数*×（*adb_sys_max_parallel_exec_numオペランドの値*×2＋2）
    ＋*adb_sys_multi_max_worker_nodesオペランドの値*
    ×*adb_sys_max_parallel_exec_numオペランドの値*×2
    ＋17

上記の図中の各メモリや領域とサーバ定義の関係を次の表に示します。

表 2-26　HADB サーバの各メモリおよび各領域の説明（adb_sys_proc_area_max および adb_sys_rthd_area_max オペランドを指定する場合）

メモリまたは領域の種類		関係するサーバ定義のオペランド	見積もり方法の記載個所	HADB サーバの稼働中にメモリの使用量を確認する方法
共有メモリ	プロセス共通メモリ※1	• adb_sys_proc_area_max プロセス共通メモリの最大値を指定するオペランドです。	「6.3.1　共有メモリの所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※3 を実行したときに出力されるPROCESS_USE の値で確認できます。
	共有メモリ管理用のメモリ	－	「6.3.1　共有メモリの所要量の求め方」の「(1)　共有メモリ管理の所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※3 を実行したときに出力されるOTHER_USE の値で確認できます。
	グローバルバッファ用のメモリ	• adbbuff グローバルバッファのページ数などを指定するオペランドです。 • adb_dbbuff_wrktbl_glb_blk_num グローバル作業表用のグローバルバッファのページ数を指定するオペランドです。	• 「6.3.1　共有メモリの所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方」 • 「6.25.1　グローバル作業表用のグローバルバッファのページ数の見積もり」	-a オプションを指定して adbls -d mem コマンド※3 を実行したときに出力されるGBUFFER_USE の値で確認できます。

メモリまたは領域の種類			関係するサーバ定義のオペランド	見積もり方法の記載個所	HADB サーバの稼働中にメモリの使用量を確認する方法
	リアルスレッド固有メモリ※1	―	• adb_sys_rthd_area_max 1 つ当たりのリアルスレッド固有メモリの最大値を指定するオペランドです。 • adb_sys_rthd_num SQL 文およびコマンドの実行時に使用する処理リアルスレッドの最大数を指定するオペランドです。1 つの処理リアルスレッドに対して，1 つのリアルスレッド固有メモリが確保されます。 • adb_sql_exe_max_rthd_num※2 1SQL 文の実行時に使用する処理リアルスレッドの最大数を指定します。	「6.3.1　共有メモリの所要量の求め方」の「(4)　リアルスレッド固有メモリの所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※3 を実行したときに出力される THREAD_USE の値で確認できます。
		ハッシュグループ化領域	• adb_sql_exe_hashgrp_area_size※2 1 つ当たりのハッシュグループ化領域のサイズを指定するオペランドです。	「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashgrp_area_size オペランド	
		ハッシュテーブル領域	• adb_sql_exe_hashtbl_area_size※2 ハッシュテーブル領域のサイズを指定するオペランドです。	「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashtbl_area_size オペランド	
		ハッシュフィルタ領域	• adb_sql_exe_hashflt_area_size※2 ハッシュフィルタ領域のサイズを指定するオペランドです。	「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashflt_area_size オペランド	
		ローカル作業表用バッファ	• adb_dbbuff_wrktbl_clt_blk_num※2 ローカル作業表用バッファのページ数を指定するオペランドです。	「6.25.2　ローカル作業表用バッファのページ数の見積もり」	

メモリまたは領域の種類		関係するサーバ定義のオペランド	見積もり方法の記載個所	HADB サーバの稼働中にメモリの使用量を確認する方法
プロセスメモリ	ヒープメモリ	－	「6.3.2　プロセスメモリの所要量の求め方」	-a オプションを指定して adbls -d mem コマンド※3 を実行したときに出力されるHEAP_USE の値で確認できます。

(凡例)

－：該当なし

注※1

「(1)　1 の方法の場合（adb_sys_memory_limit オペランドを指定する場合)」と差異がある個所です。

HADB サーバが使用するメモリ所要量の最大値を指定するadb_sys_memory_limit オペランドを指定しない場合，プロセス共通メモリの最大値（adb_sys_proc_area_max オペランド）と，1 つ当たりのリアルスレッド固有メモリの最大値（adb_sys_rthd_area_max オペランド）を指定する必要があります。

注※2

クライアント定義の同じオペランドで，サーバ定義で指定した値を変更できます。

注※3

-a オプションを指定して実行したadbls -d mem コマンドの出力項目については，マニュアル『HADB コマンドリファレンス』の『adbls -d mem（メモリの使用状況の表示)』を参照してください。

メモ

HADB サーバが使用するメモリ所要量の見積もり方法については，次の個所を参照してください。

- 「4.3　メモリ所要量の事前見積もり」
- 「6.3　HADB サーバのメモリ所要量の見積もり」

3 体験環境構築ナビ

この章では，HADB の導入・操作を体験するために使用する，体験環境の構築手順について説明します。

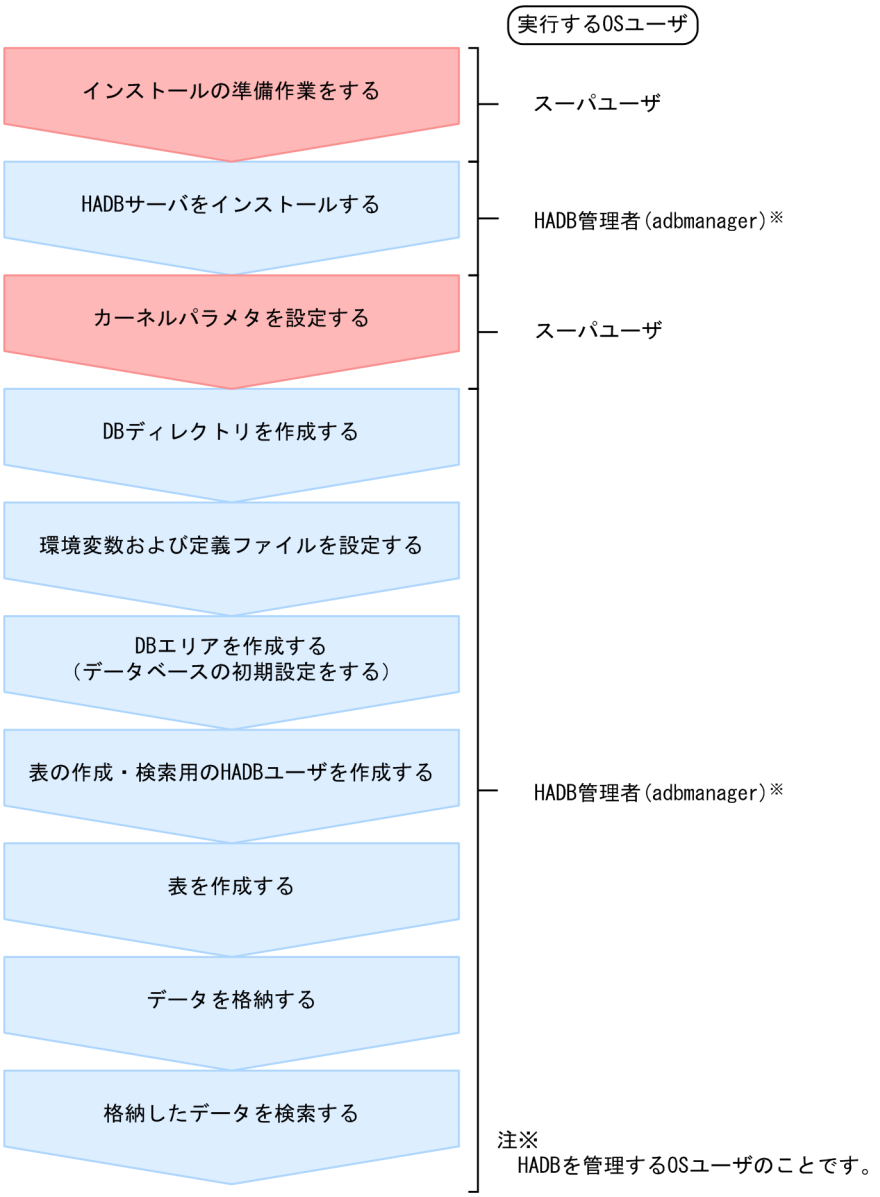
HADB サーバの構築イメージを把握するために，この章を読みながら実機で構築することを推奨します。作業時間の目安は 1 時間～2 時間です。

なお，この章で説明する作業手順および指定値は，あくまで体験環境を構築するためのものです。実際の業務で使用する環境を構築するためのものではありません。実際の業務で使用する環境を構築する場合は，このマニュアルの『第 3 編　設計編』を参照して必要な値を見積もった上で，『第 4 編　構築編』を参照して，環境を構築してください。

3.1　体験環境を構築する流れ

HADB サーバの体験環境を構築する流れを次の図に示します。体験環境を構築する手順の詳細については，次節以降で説明します。

図 3-1　HADB サーバの体験環境を構築する流れ



3.1.1　構築する HADB サーバのシステム構成

HADB サーバの体験環境を構築するマシン環境は，次に示す環境を前提としています。

- 搭載しているメモリサイズ：8 ギガバイト
- ディスクの空き容量：2 ギガバイト以上

- OS のバージョン

  次に示すどれかの OS が，サーバマシンにインストールされている必要があります。

  - RHEL 6

    6.2 以降のバージョンで動作します。

  - RHEL 7

    7.1 以降のバージョンで動作します。

  - RHEL 8

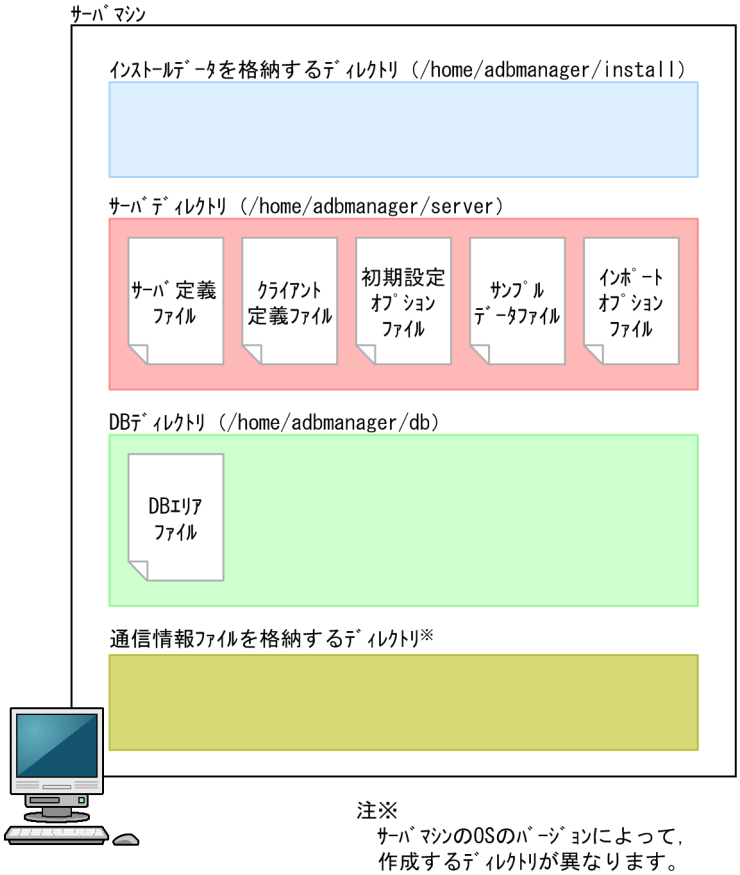
    8.1 以降のバージョンで動作します。

  OS には，HADB サーバが正しく動作するための前提パッケージがインストールされている必要があります。前提パッケージについては，「8.2.1　インストール前に実施する作業」の「(1)　前提パッケージの確認」を参照してください。

  なお，HADB は Intel64 アーキテクチャだけで動作します。AMD64 アーキテクチャでは動作しません。

体験環境で構築する HADB サーバのシステム構成を次の図に示します。

図 3-2　構築する HADB サーバのシステム構成



(1)　インストールデータを格納するディレクトリ

CD-ROM ファイルシステムに格納されている HADB サーバのインストールデータのコピー先となるディレクトリです。インストールデータを格納するディレクトリには，次のインストールデータを格納します。

- HADB サーバのインストールコマンド（adbinstall コマンド）
- HADB サーバのプログラムの圧縮ファイル（hitachi_advanced_data_binder_server-*$VER*.tar.gz ファイル）

注

$VER は，HADB のバージョンおよびリリース番号を示します。

(2) サーバディレクトリ

サーバディレクトリとは，HADB サーバを運用するために必要な定義ファイルや各種コマンドなどが格納されるディレクトリです。

HADB サーバのインストール時にインストールコマンド（adbinstall コマンド）で，サーバディレクトリとして使用するディレクトリ（/home/adbmanager/server）を指定します。サーバディレクトリ下には，HADB サーバのプログラムの圧縮ファイル（hitachi_advanced_data_binder_server-*$VER*.tar.gz ファイル）が展開され，ディレクトリおよびファイルが格納されます。

体験環境を構築するときに使用するファイルを次の表に示します。

表 3-1 体験環境で使用するサーバディレクトリ内のファイルの一覧

項番	ファイル名	説明
1	サーバ定義ファイル	HADB サーバを稼働するために必要な定義が設定されているファイルです。
2	クライアント定義ファイル	AP やコマンドを実行するために必要な定義が設定されているファイルです。
3	初期設定オプションファイル	DB エリアを作成するために必要なオプションが設定されているファイルです。
4	サンプルデータファイル	体験環境の表に格納するデータが入力されているテキストファイルです。
5	インポートオプションファイル	表にデータを格納するために必要なオプションが設定されているファイルです。

(3) DB ディレクトリ

DB ディレクトリとは，データベースの初期設定をして DB エリアを作成する際，DB エリアを格納するディレクトリです。データベースの初期設定をする前に，ユーザが DB ディレクトリ（/home/adbmanager/db）を作成します。

データベースの初期設定をすると，5 つの DB エリアが作成されます。作成される DB エリアを次の表に示します。ユーザが明示的に作成する DB エリアは**データ用 DB エリア**だけです。それ以外の DB エリアは自動的に作成されます。

表 3-2 作成される DB エリアの一覧

項番	DB エリア名	説明
1	データ用 DB エリア	表やインデクスを格納する DB エリアです。

項番	DB エリア名	説明
2	作業表用 DB エリア	SQL 文の実行時に HADB が作成する作業表を格納する DB エリアです。
3	マスタディレクトリ用 DB エリア	システムの内部情報を格納する DB エリアです。
4	ディクショナリ用 DB エリア	ディクショナリ表，およびディクショナリ表の B-tree インデクスを格納する DB エリアです。
5	システム表用 DB エリア	システム表，およびシステム表の B-tree インデクスを格納する DB エリアです。

(4) 通信情報ファイルを格納するディレクトリ

通信情報ファイルを格納するディレクトリとは，HADB サーバを運用するために必要な通信情報ファイルが格納されるディレクトリです。

HADB サーバをインストールする前に，ユーザが手動で，通信情報ファイルを格納するディレクトリを作成します。通信情報ファイルを格納するディレクトリは，サーバマシンの OS のバージョンによって異なります。

サーバマシンの OS が RHEL 6 の場合

- /dev/HADB/pth
- /lib/udev/devices/HADB/pth

サーバマシンの OS が RHEL 7 以降の場合

- /dev/HADB/pth

また，次に示す設定ファイルを作成する必要があります。

- /etc/tmpfiles.d/dev-HADB-pth.conf

3.1.2 体験環境の表に格納するデータ

体験環境の構築で表に格納するデータは，HADB サーバが提供するデータ（**サンプルデータ**）になります。サンプルデータの内容は，米国 50 州の州番号，州名，郵便番号，州議事堂の住所，および面積になります。表に格納するサンプルデータのイメージを次の図に示します。

図 3-3　体験環境の表に格納するサンプルデータ

STATECODE	STATENAME	ZIPCODE	ADDRESS	AREA
1	Alabama	36130-2751	State Capitol N-104 600 Dexter Avenue Montgomery	135765000000
2	Alaska	99811	State Capitol Juneau	1717854000000
3	Arizona	85007	State Capitol West Wing 1700 W. Washington, 9th Fl. Phoenix	295254000000
:	:	:	:	:

（凡例）
　STATECODE：州番号
　STATENAME：州名
　ZIPCODE：郵便番号
　ADDRESS：州議事堂の住所
　AREA：面積

3.2　HADB サーバをインストールする

ここでは，HADB サーバをインストールするために行う作業について説明します。

3.2.1　インストール前に準備すること

HADB サーバをインストールする前に，HADB を管理する OS ユーザ（**HADB 管理者**）と HADB 管理者が所属するグループ（**HADB 管理グループ**）を OS に設定します。

また，HADB サーバを動作させるために必要な**通信情報ファイル**を格納するディレクトリを作成します。

(1)　HADB 管理者と HADB 管理グループを設定する

HADB 管理者と HADB 管理グループには，任意の名称を設定できます。ここでは，次の名称で設定します。

- HADB 管理者：adbmanager
- HADB 管理グループ：adbgroup

1. スーパユーザで OS にログインします。
   ログインしたら，コマンドを入力する「端末」を開いてください。
2. HADB 管理グループ（adbgroup）を OS に設定します。
   次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
groupadd adbgroup
```

3. HADB 管理者（adbmanager）を OS に設定します。
   次に示す OS のコマンドを入力して，[Enter] キーを押してください。
   -g オプションには，手順 2.で設定した HADB 管理グループ（adbgroup）を指定してください。

```
useradd -g adbgroup adbmanager
```

4. HADB 管理者のパスワードを OS に設定します。
   次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
passwd adbmanager
```

   passwd コマンドを実行すると，adbmanager の新しいパスワードを 2 回入力するよう求められます。任意のパスワードを設定してください。

(2)　設定した HADB 管理者と HADB 管理グループを確認する

OS に設定した HADB 管理者と HADB 管理グループが，正しく設定されたかどうかを確認します。

スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
id adbmanager
```

id コマンドを実行すると，adbmanager に対して設定した内容が確認できます。実行結果の例を次に示します。

■実行結果の例

```
uid=501(adbmanager) gid=501(adbgroup) 所属グループ=501(adbgroup)
```

uid にadbmanager，gid および所属グループにadbgroup が設定されていれば，設定完了です。

> **メモ**
>
> HADB 管理者を HADB 管理グループに所属する OS ユーザとして OS に設定することで，HADB 管理者は HADB サーバの各種ファイルおよびディレクトリの所有者になります。これによって，ほかの OS ユーザからの書き込みを禁止でき，セキュリティを強化できます。また，HADB 管理者には，HADB のすべてのコマンドを実行する権限があります。

(3) 通信情報ファイルを格納するディレクトリを作成する

通信情報ファイルを格納するディレクトリは，サーバマシンの OS のバージョンによって異なります。

(a) RHEL 6 の場合

通信情報ファイルを格納するために，次に示すディレクトリを作成します。

- /dev/HADB/pth
- /lib/udev/devices/HADB/pth

ディレクトリに付与する権限は，777 になります。

1. /dev/HADB/pth ディレクトリを作成します。
   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

   ```
   mkdir -p -v -m 777 /dev/HADB/pth
   ```

2. /lib/udev/devices/HADB/pth ディレクトリを作成します。
   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

   ```
   mkdir -p -v -m 777 /lib/udev/devices/HADB/pth
   ```

(b) RHEL 7 以降の場合

通信情報ファイルを格納するために，次に示すディレクトリおよびファイルを作成します。

- /dev/HADB/pth

ディレクトリに付与する権限は，777 です。

- /etc/tmpfiles.d/dev-HADB-pth.conf

  設定ファイルに付与する権限は，644（デフォルト）です。デフォルトの権限であるため，特に指定は不要です。

1. /dev/HADB/pth ディレクトリを作成します。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

   ```
   mkdir -p -v -m 777 /dev/HADB/pth
   ```

2. 設定ファイル（dev-HADB-pth.conf）を作成します。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。設定ファイルが新規に作成されます。

   ```
   vi /etc/tmpfiles.d/dev-HADB-pth.conf
   ```

3. 設定ファイルを編集します。

   [I] キーを押して，編集モードに移行させてください。編集モードに移行させたら，次に示す内容を入力してください。

   ```
   # Type Path Mode UID  GID  Age Argument
   d /dev/HADB/pth 0777 root root - -
   ```

4. 設定ファイルの編集を終了します。

   入力が完了したら，[Esc] キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，[Enter] キーを押してください。

   ```
   :wq
   ```

   編集モードで設定した内容が保存されます。この手順が完了すると，設定ファイルの作成が完了します。

(4) 作成した通信情報ファイルを格納するディレクトリを確認する

通信情報ファイルを格納するディレクトリが，正しく作成されたかどうかを確認します。

(a) RHEL 6 の場合

通信情報ファイルを格納する 2 つのディレクトリが，正しく作成されたかどうかを，次の手順で確認します。

1. 作成した/dev/HADB/pth ディレクトリを確認します。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

   ```
   ls -d -l /dev/HADB/pth
   ```

   ls コマンドを実行すると，/dev/HADB/pth ディレクトリが作成されているかどうか，権限が付与されているかどうかが確認できます。実行結果の例を次に示します。

   ■実行結果の例

```
drwxrwxrwx 2 root root 40  8月 13 19:28 2013 /dev/HADB/pth
```

先頭にdrwxrwxrwx，末尾に/dev/HADB/pth が表示されていることを確認してください。

2. 作成した/lib/udev/devices/HADB/pth ディレクトリを確認します。

スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls -d -l /lib/udev/devices/HADB/pth
```

ls コマンドを実行すると，/lib/udev/devices/HADB/pth ディレクトリが作成されているかどうか，権限が付与されているかどうかが確認できます。実行結果の例を次に示します。

■実行結果の例

```
drwxrwxrwx 2 root root 4096  8月 13 19:29 2013 /lib/udev/devices/HADB/pth
```

先頭にdrwxrwxrwx，末尾に/lib/udev/devices/HADB/pth が表示されていることを確認してください。

(b) RHEL 7 以降の場合

通信情報ファイルを格納するディレクトリおよび設定ファイルが，正しく作成されたかどうかを，次の手順で確認します。

1. 作成した/dev/HADB/pth ディレクトリを確認します。

スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls -d -l /dev/HADB/pth
```

ls コマンドを実行すると，/dev/HADB/pth ディレクトリが作成されているかどうか，権限が付与されているかどうかが確認できます。実行結果の例を次に示します。

■実行結果の例

```
drwxrwxrwx 2 root root 40  8月 13 19:28 2013 /dev/HADB/pth
```

先頭にdrwxrwxrwx，末尾に/dev/HADB/pth が表示されていることを確認してください。

2. 作成した設定ファイル（dev-HADB-pth.conf）を確認します。

スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls -l /etc/tmpfiles.d/dev-HADB-pth.conf
```

ls コマンドを実行すると，/etc/tmpfiles.d/dev-HADB-pth.conf ファイルが作成されているか，デフォルトの権限が付与されているかどうかが確認できます。実行結果の例を次に示します。

■実行結果の例

```
-rw-r--r-- 2 root root 40  8月 13 19:29 2013 /etc/tmpfiles.d/dev-HADB-pth.conf
```

先頭に-rw-r--r--，末尾に/etc/tmpfiles.d/dev-HADB-pth.conf が表示されていることを確認してください。

(5) 関連項目

「8.2.1 インストール前に実施する作業」

3.2.2 インストールする

HADB 管理者と HADB 管理グループを OS に設定し，通信情報ファイルを格納するディレクトリを作成したら，HADB サーバをインストールします。

HADB サーバをインストールする場合，**HADB 管理者（adbmanager）**で実行します。スーパユーザでログインしている場合は，いったんログアウトして，HADB 管理者（adbmanager）でログインし直してください。

(1) インストールデータを格納するディレクトリを作成する

HADB サーバのインストールデータを格納するディレクトリを作成します。作成するディレクトリのパス名を次に示します。

- インストールデータを格納するディレクトリの絶対パス：/home/adbmanager/install

HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
mkdir /home/adbmanager/install
```

(2) 作成したインストールデータを格納するディレクトリを確認する

mkdir コマンドでインストールデータを格納するディレクトリが作成されているかどうかを確認するため，次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
ls -d /home/adbmanager/install
```

ls コマンドを実行すると，インストールデータを格納するディレクトリが正しく作成されたかどうかを確認できます。実行結果の例を次に示します。

■実行結果の例

```
/home/adbmanager/install
```

上記のディレクトリが表示されていれば，インストールデータを格納するディレクトリは作成されています。

(3) インストールデータを格納するディレクトリに書き込み権限を付与する

/home/adbmanager/install ディレクトリ下に，HADB 管理者がインストールデータを格納できるように，書き込み権限を付与してください。次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
chmod 755 /home/adbmanager/install
```

これで，インストールデータを格納するディレクトリに書き込み権限が付与されました。

(4) CD-ROM ファイルシステムをマウントする

HADB サーバのインストールコマンド（adbinstall コマンド），および HADB サーバのプログラムの圧縮ファイル（tar.gz ファイル）が格納された CD-ROM ファイルシステムを自動マウントしてください。

CD-ROM ファイルシステムが自動マウントされない場合，CD-ROM ファイルシステムをマウントする必要があります。次に示すコマンドを入力して，［Enter］キーを押してください。

```
mount /dev/cdrom /media
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

重要

CD-ROM のディレクトリ名やファイル名は，ご使用のサーバマシンによっては，ここの説明内容と表示が異なることがあります。OS の ls コマンドを実行して，表示されたディレクトリ名をそのまま入力してください。

(5) インストールコマンド（adbinstall コマンド）をコピーする

マウントした CD-ROM ファイルシステムに格納されている HADB サーバのインストールコマンド（adbinstall コマンド）を，/home/adbmanager/install ディレクトリ下にコピーしてください。

次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
cp /media/adbinstall /home/adbmanager/install
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

(6) HADB サーバのプログラムの圧縮ファイル（tar.gz ファイル）をコピーする

マウントした CD-ROM ファイルシステムに格納されている HADB サーバの HADB サーバのプログラムの圧縮ファイル（tar.gz ファイル）を，/home/adbmanager/install ディレクトリ下にコピーしてください。

次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
cp /media/hitachi_advanced_data_binder_server-$VER.tar.gz /home/adbmanager/install
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

$VER は，HADB のバージョンおよびリリース番号を示します。

(7) コピーしたインストールデータを確認する

adbinstall コマンドおよびtar.gz ファイルを/home/adbmanager/install ディレクトリ下にコピーできたことを確認します。次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls /home/adbmanager/install
```

ls コマンドを実行すると，adbinstall コマンドおよびtar.gz ファイルを/home/adbmanager/install ディレクトリ下にコピーできているかどうかを確認できます。実行結果の例を次に示します。

■実行結果の例

```
adbinstall hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，HADB のバージョンおよびリリース番号を示します。

(8) コピーしたインストールデータの完全性を確認する

adbinstall コマンドおよびtar.gz ファイルが正しくコピーできたこと（破損していないこと）を，OS の sha256sum コマンドで確認します。次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
cd /home/adbmanager/install
sha256sum -c /media/adbhashfile_server_sha256.txt
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

実行結果の例を次に示します。

■実行結果の例

```
hitachi_advanced_data_binder_server-$VER.tar.gz: 完了                  ...1
adbinstall: 完了                                                       ...1
sha256sum: !8A9_HADB: そのようなファイルやディレクトリはありません     ...2
!8A9_HADB: オープンまたは読み込みに失敗しました                        ...2
sha256sum: 警告: 一覧にある1個のファイルが読み込めませんでした         ...2
```

$VER は，HADB のバージョンおよびリリース番号を示します。

[説明]

1. チェック結果に「完了」が表示されている場合は，正しくコピーされています。
2. このエラーは無視してください。

   体験環境構築ナビでは，意図的に!8A9_HADB ファイルを/home/adbmanager/install ディレクトリ下にコピーさせていません。そのため，このエラーメッセージが出力されても問題ありません。

(9) 権限を付与する

/home/adbmanager/install ディレクトリ下にコピーした HADB サーバのインストールコマンド（adbinstall コマンド）に対して，HADB 管理者が実行できるように，実行権限を付与してください。

また，HADB サーバのプログラムの圧縮ファイル（tar.gz ファイル）に対して，読み取り権限を付与してください。

次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
chmod 500 /home/adbmanager/install/adbinstall
chmod 400 /home/adbmanager/install/hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，HADB のバージョンおよびリリース番号を示します。

これで，インストールコマンド（adbinstall コマンド）に対する実行権限が付与され，HADB サーバのプログラムの圧縮ファイル（tar.gz ファイル）に対する読み取り権限が付与されました。

(10) HADB サーバをインストールする

実行権限を付与した HADB サーバのインストールコマンド（adbinstall コマンド）を実行して，HADB サーバをインストールしてください。

次に示す HADB のコマンドを入力して，[Enter] キーを押してください。

```
/home/adbmanager/install/adbinstall -s /home/adbmanager/server
```

-s /home/adbmanager/server

adbinstall コマンドで HADB サーバをインストールする際，サーバディレクトリとして使用するディレクトリを，-s オプションに指定します。ここでは，/home/adbmanager/server ディレクトリを指定してください。

adbinstall コマンドを実行すると，HADB サーバのプログラムの圧縮ファイル（tar.gz ファイル）が展開されて，/home/adbmanager/server ディレクトリ下に HADB サーバを構成するディレクトリとファイルが格納されます。

> **重要**
>
> HADB 管理者（adbmanager）ではなく，スーパユーザ（root）でadbinstall コマンドを実行した場合は，KFAA91558-W メッセージが出力されます。この場合は，KFAA91558-W メッセージが出力されたあとに出力されるKFAA91559-Q メッセージの入力要求に対して，n またはN を入力してください。そのあとで，いったんログアウトして，HADB 管理者（adbmanager）でログインし直してください。そして，HADB 管理者（adbmanager）でadbinstall コマンドを実行してください。

(11) HADB サーバがインストールされたことを確認する

HADB サーバが正しくインストールされたかどうかを確認します。

次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
cat /home/adbmanager/server/adbinstsv.log
```

cat コマンドを実行して，HADB サーバのインストール実行結果ログファイル（adbinstsv.log）の内容を表示します。実行結果の例を次に示します。

■実行結果の例

```
Hitachi Advanced Data Binder vv-rr (yyyymmddhhmmss)
```

vv-rr は，HADB サーバのバージョンを示しています。
yyyymmddhhmmss は，HADB サーバをインストールした日付と時刻を示しています。
上記の情報が表示された場合は，HADB サーバは正しくインストールされています。

また，サーバディレクトリ下のディレクトリおよびファイル構成が正しいことを確認します。

次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls /home/adbmanager/server
```

実行結果の例を次に示します。

■実行結果の例

```
adbinstsv.log  bin  client  conf  include  lib  sample  spool
```

上記のディレクトリおよびファイルが表示された場合は，サーバディレクトリ下の構成が正しい状態になっています。

(12) 関連項目

「8.2.2 インストール手順」

3.3　カーネルパラメタを設定する

ここでは，HADB サーバを構築するために必要なカーネルパラメタ（Linux のオペレーティングシステムパラメタ）を設定する手順について説明します。

3.3.1　カーネルパラメタの設定および確認をする

次に示すファイルに対して，カーネルパラメタを設定します。

- sysctl.conf ファイル
- limits.conf ファイル

カーネルパラメタの設定は**スーパユーザ**で行います。HADB 管理者（adbmanager）でログインしている場合は，いったんログアウトして，スーパユーザでログインし直してください。

> **メモ**
>
> ここではvi コマンドを使用して，各ファイルにカーネルパラメタを設定する方法について説明します。なお，テキストエディタで各ファイルを開いて編集することもできます。

(1)　sysctl.conf ファイルにカーネルパラメタを設定する

カーネルパラメタをsysctl.conf ファイルに設定します。sysctl.conf ファイルの格納先を次に示します。

- sysctl.conf ファイル：/etc/sysctl.conf

1. sysctl.conf ファイルを開きます。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。vi コマンドを実行することで，sysctl.conf ファイルを編集できるようになります。

   ```
   vi /etc/sysctl.conf
   ```

2. カーネルパラメタを設定します。

   [I] キーを押して，編集モードに移行させてください。編集モードに移行させたら，カーソルキーを押して，カーネルパラメタを入力する位置にカーソルを移動させてください。カーソルを移動させたら，次に示すカーネルパラメタを入力します。

   ```
   net.core.rmem_default = 33554432
   net.core.rmem_max = 33554432
   net.core.wmem_default = 33554432
   net.core.wmem_max = 33554432
   fs.aio-max-nr = 65536
   ```

メモ

sysctl.conf ファイルに，すでに上記のカーネルパラメタが指定されている場合は，元に戻せるようにするため，変更前の指定値をコメントとして残しておいてください。行の先頭に「#」を入力すると，その行はコメントとして扱われます。

例

「#net.core.rmem_default = 10000」はコメントとして扱われ，指定値は無視されます。

```
#net.core.rmem_default = 10000
net.core.rmem_default = 33554432
```

入力するカーネルパラメタの詳細を次の表に示します。

表 3-3　sysctl.conf ファイルに指定するカーネルパラメタの一覧

項番	カーネルパラメタ	説明
1	net.core.rmem_default	受信用ウィンドウサイズのデフォルト値を指定します。33,554,432 を指定してください。
2	net.core.rmem_max	受信用ウィンドウサイズの上限値を指定します。33,554,432 を指定してください。
3	net.core.wmem_default	送信用ウィンドウサイズのデフォルト値を指定します。33,554,432 を指定してください。
4	net.core.wmem_max	送信用ウィンドウサイズの上限値を指定します。33,554,432 を指定してください。
5	fs.aio-max-nr	65,536 を指定してください。

3. カーネルパラメタの設定を終了します。

編集モードで必要なカーネルパラメタを入力したら，［Esc］キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，［Enter］キーを押してください。

```
:wq
```

編集モードで設定した内容が保存されます。これで，sysctl.conf ファイルにカーネルパラメタが設定されます。

(2)　sysctl.conf ファイルに設定したカーネルパラメタを確認する

カーネルパラメタがsysctl.conf ファイルに設定されているかどうかを確認するため，次に示すコマンドを入力して，［Enter］キーを押してください。sysctl.conf ファイルに設定したカーネルパラメタを確認できます。

```
sysctl -p
```

sysctl -p コマンドを実行することで，sysctl.conf ファイルに設定されているカーネルパラメタの一覧が表示されます。表示されるカーネルパラメタの一覧は情報量が多いため，抜粋したものを実行結果の例として次に示します。

■実行結果の例

```
          :
net.core.rmem_default = 33554432
net.core.rmem_max = 33554432
net.core.wmem_default = 33554432
net.core.wmem_max = 33554432
fs.aio-max-nr = 65536
          :
```

(3) limits.conf ファイルにカーネルパラメタを設定する

カーネルパラメタをlimits.conf ファイルに設定します。limits.conf ファイルの格納先を次に示します。

- limits.conf ファイル：/etc/security/limits.conf

1. limits.conf ファイルを開きます。

   次に示す OS のコマンドを入力して，[Enter] キーを押してください。vi コマンドを実行することで，limits.conf ファイルを編集できるようになります。

   ```
   vi /etc/security/limits.conf
   ```

2. カーネルパラメタを設定します。

   [I] キーを押して，編集モードに移行させてください。編集モードに移行させたら，カーソルキーを押して，カーネルパラメタを入力する位置にカーソルを移動させてください。カーソルを移動させたら，次に示すカーネルパラメタを入力します。

   ```
   adbmanager   soft   nofile    6400
   adbmanager   hard   nofile    6400
   adbmanager   soft   memlock   unlimited
   adbmanager   hard   memlock   unlimited
   ```

   メモ

   limits.conf ファイルに，すでに上記のカーネルパラメタが指定されている場合は，元に戻せるようにするため，変更前の指定値をコメントとして残しておいてください。行の先頭に「#」を入力すると，その行はコメントとして扱われます。

   例

   ```
   #adbmanager soft nofile 1000
   adbmanager soft nofile 6400
   ```

   入力するカーネルパラメタの詳細を次の表に示します。

表 3-4　limits.conf ファイルに指定するカーネルパラメタの一覧

項番	カーネルパラメタ	説明
1	soft nofile	open できるファイルディスクリプタ数を指定します。HADB 管理者（adbmanager）に対して，soft nofile を指定してください。soft nofile には，6400 を指定してください。
2	hard nofile	open できるファイルディスクリプタ数を指定します。HADB 管理者（adbmanager）に対して，hard nofile を指定してください。hard nofile には，6400 を指定してください。
3	soft memlock	共有メモリのメモリロックの上限値を指定します。HADB 管理者（adbmanager）に対して，soft memlock を指定してください。soft memlock には，unlimited を指定してください。
4	hard memlock	共有メモリのメモリロックの上限値を指定します。HADB 管理者（adbmanager）に対して，hard memlock を指定してください。hard memlock には，unlimited を指定してください。

重要

soft nofile およびhard nofile に指定する6400 は，体験環境を構築するための値です。体験環境ではなく，実際の業務で使用する環境を構築する場合は，「6.2　カーネルパラメタの見積もり」を参照して，必要な値を見積もった上で指定してください。

また，実際の業務で使用する環境を構築する場合，この章でsysctl.conf ファイルおよびlimits.conf ファイルに設定したカーネルパラメタ以外にも，指定が必要なものがあります。

3. カーネルパラメタの設定を終了します。

編集モードで必要なカーネルパラメタを入力したら，［Esc］キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，［Enter］キーを押してください。

```
:wq
```

編集モードで設定した内容が保存されます。これで，limits.conf ファイルにカーネルパラメタが設定されます。

(4)　limits.conf ファイルに設定したカーネルパラメタを確認する

カーネルパラメタがlimits.conf ファイルに設定されているかどうかを確認するため，次に示す OS のコマンドを入力して，［Enter］キーを押してください。スーパユーザから HADB 管理者（adbmanager）に切り替えることで，limits.conf ファイルに設定したカーネルパラメタを確認できます。

```
su adbmanager
```

一時的に，HADB 管理者（adbmanager）に切り替わります。続いて，次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
ulimit -a
```

ulimit -a コマンドを実行することで，HADB 管理者（adbmanager）に設定したカーネルパラメタの一覧が表示されます。表示されるカーネルパラメタの一覧は情報量が多いため，抜粋したものを実行結果の例として次に示します。

■実行結果の例

```
            :
max locked memory      (kbytes, -l) unlimited
open files                     (-n) 6400
            :
```

実行結果の例の内容を確認したら，HADB 管理者（adbmanager）からスーパユーザに切り替えます。次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
exit
```

(5) 設定したカーネルパラメタを有効にする

sysctl.conf ファイル，および limits.conf ファイルに設定したカーネルパラメタを有効にするために，OS を再起動します。次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
reboot
```

(6) 関連項目

「6.2 カーネルパラメタの見積もり」

3.4　DB ディレクトリを作成する

ここでは，DB エリアを格納する DB ディレクトリを作成する手順について説明します。

3.4.1　DB ディレクトリの作成および確認をする

ここでは，DB エリアを格納する **DB ディレクトリ**を作成します。DB エリアとは，表を格納する論理的な領域のことです。DB エリアはデータベースを初期設定するときに作成され，DB ディレクトリに格納されます。DB エリアは「3.6　DB エリアを作成する（データベースの初期設定をする）」で作成します。

作成する DB ディレクトリの名称は任意です。ここでは，次に示すディレクトリに作成します。

- DB ディレクトリの絶対パス：/home/adbmanager/db

DB ディレクトリの作成は **HADB 管理者（adbmanager）**で行います。スーパユーザでログインしている場合は，いったんログアウトして，HADB 管理者（adbmanager）でログインし直してください。

(1)　DB ディレクトリの作成コマンドを実行する

HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
mkdir /home/adbmanager/db
```

(2)　作成した DB ディレクトリを確認する

mkdir コマンドで DB ディレクトリが作成されているかどうかを確認するため，次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls -d /home/adbmanager/db
```

ls コマンドに-d オプションを指定することで，指定したディレクトリが作成されているかどうかを確認できます。実行結果の例を次に示します。

■実行結果の例

```
/home/adbmanager/db
```

DB ディレクトリが作成されていることを確認できたら，DB ディレクトリの作成は完了です。

3.5　環境変数および定義ファイルを設定する

ここでは，HADB サーバを運用するために必要な環境変数，および定義ファイルを設定する手順について説明します。

環境変数および定義ファイルの設定は HADB 管理者（adbmanager）で行います。

3.5.1　環境変数を設定する

HADB サーバを運用するために必要な環境変数を.bashrc ファイルに設定します。.bashrc ファイルの格納先を次に示します。

- .bashrc ファイル：/home/adbmanager/.bashrc

ここでは，vi コマンドを使用して.bashrc ファイルに環境変数を設定する方法について説明します。

> **メモ**
>
> .bashrc ファイルは，テキストエディタで開いて編集することもできます。

(1)　.bashrc ファイルに環境変数を設定する

1. .bashrc ファイルを開きます。

   HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，[Enter] キーを押してください。vi コマンドを実行することで，.bashrc ファイルを編集できるようになります。

   ```
   vi /home/adbmanager/.bashrc
   ```

2. 環境変数を設定します。

   [I] キーを押して，編集モードに移行させてください。編集モードに移行させたら，カーソルキーを押して，環境変数を入力する位置にカーソルを移動させてください。カーソルを移動させたら，次に示す環境変数を入力します。

   ```
   export ADBDIR=/home/adbmanager/server
   export ADBCLTDIR=$ADBDIR
   export LD_LIBRARY_PATH=$ADBDIR/lib:$ADBDIR/client/lib
   export PATH=$PATH:$ADBDIR/bin:$ADBDIR/client/bin
   export ADBLANG=UTF8
   export ADBCLTLANG=UTF8
   ```

   入力する環境変数の詳細を次の表に示します。

表 3-5　指定する環境変数の一覧

項番	環境変数	指定する値	説明
1	ADBDIR	/home/adbmanager/server	サーバディレクトリを絶対パスで指定します。
2	ADBCLTDIR	$ADBDIR	HADB サーバ上で AP を実行するため，ADBDIR と同じパスを指定します。左記の値を指定してください。
3	LD_LIBRARY_PATH	• $ADBDIR/lib • $ADBDIR/client/lib	• サーバディレクトリ下のlib ディレクトリ（$ADBDIR/lib）を指定します。 • HADB サーバ上で AP を実行する場合，サーバディレクトリ下のclient/lib ディレクトリ（$ADBDIR/client/lib）を指定します。
4	PATH	• $PATH • $ADBDIR/bin • $ADBDIR/client/bin	• OS に設定されているパスの設定を保持するため，$PATH を指定します。 • サーバディレクトリ下のbin ディレクトリ（$ADBDIR/bin）を指定します。 • HADB サーバ上で AP を実行するため，サーバディレクトリ下のclient/bin ディレクトリ（$ADBDIR/client/bin）を指定します。
5	ADBLANG	UTF8	使用する文字コードを指定します。
6	ADBCLTLANG	UTF8	

3. 環境変数の設定を終了します。

   編集モードで必要な環境変数を入力したら，［Esc］キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，［Enter］キーを押してください。

```
:wq
```

   編集モードで設定した内容が保存されます。これで，.bashrc ファイルに環境変数が設定されます。

4. 設定した環境変数を有効にします。

   次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
source /home/adbmanager/.bashrc
```

   source コマンドを実行することで，.bashrc ファイルに設定した環境変数を有効にできます。

(2)　.bashrc ファイルに設定した環境変数を確認する

設定した環境変数が設定されているかどうかを確認するため，次に示す OS のコマンドを入力して，［Enter］キーを押してください。

```
env
```

env コマンドを実行することで，OS に設定されている環境変数の一覧が表示されます。表示される環境変数の一覧は情報量が多いため，抜粋したものを実行結果の例として次に示します。

■実行結果の例

```
          :
LD_LIBRARY_PATH=/home/adbmanager/server/lib:/home/adbmanager/server/client/lib
ADBCLTDIR=/home/adbmanager/server
PATH=/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin:/
home/adbmanager/bin:/home/adbmanager/server/bin:/home/adbmanager/server/client/bin
ADBLANG=UTF8
ADBDIR=/home/adbmanager/server
ADBCLTLANG=UTF8
          :
```

(3) 関連項目

「8.4 環境変数の設定」

3.5.2 定義ファイルを編集する

HADB サーバを稼働させるために必要な**サーバ定義ファイル**，および AP やコマンドを実行するために必要な**クライアント定義ファイル**を編集します。

各定義ファイルには，ファイルのひな形があります。そのため，ファイルのひな形をコピーしてから編集します。

各定義ファイルのひな形の格納先を次に示します。

■各定義ファイルのひな形の格納先

- サーバ定義ファイルのひな形：/home/adbmanager/server/sample/conf/server.def
- クライアント定義ファイルのひな形：/home/adbmanager/server/sample/conf/client.def

ひな形をコピーして編集した各定義ファイル（server.def，client.def）の格納先を次に示します。

- 各定義ファイルの格納先：/home/adbmanager/server/conf/
  編集した各定義ファイルを上記に格納することで，HADB サーバが稼働できます。

ここでは，vi コマンドを使用して各定義ファイルを編集する方法について説明します。

メモ

各定義ファイルは，テキストエディタで開いて編集することもできます。

(1) サーバ定義ファイルを編集する

1. サーバ定義ファイルのひな形をコピーします。

   HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，[Enter] キーを押してください。サーバ定義ファイルのひな形をコピーします。

```
cp /home/adbmanager/server/sample/conf/server.def /home/adbmanager/server/conf/
```

2. コピーしたサーバ定義ファイルのひな形を開きます。

   次に示す OS のコマンドを入力して，[Enter] キーを押してください。vi コマンドを実行することで，サーバ定義ファイルが編集できるようになります。

```
vi /home/adbmanager/server/conf/server.def
```

3. データベースを格納する DB ディレクトリを指定します。

   [I] キーを押して，編集モードに移行させてください。編集モードに移行させたら，カーソルキーを押して，次に示すオペランドまでカーソルを移動させてください。

```
set adb_db_path = XXXXX
```

   カーソルを移動させたら，「XXXXX」を削除してください。そのあとで，**DB ディレクトリの絶対パス**（/home/adbmanager/db）を入力してください。

   ■入力結果

```
set adb_db_path = /home/adbmanager/db
```

4. グローバルバッファ TBLBUF01 に指定する値を変更します。

   カーソルキーを押して，次に示すオペランドまでカーソルを移動させてください。

```
adbbuff -g TBLBUF01 ¥
        -n ADBUTBL01 ¥
        -p 1000000 ¥
        -v 1024
```

   注

   「¥」は，半角のバックスラッシュに読み替えてください。

   カーソルを移動させたら，-p オプションの「1000000」を削除してください。そのあとで，「1000」を入力してください。

   また，-v オプションの「1024」を削除してください。そのあとで，「64」を入力してください。

   ■入力結果

```
adbbuff -g TBLBUF01 ¥
        -n ADBUTBL01 ¥
        -p 1000 ¥
        -v 64
```

5. グローバルバッファ IDXBUF01 に指定する値を変更します。

   カーソルキーを押して，次に示すオペランドまでカーソルを移動させてください。

```
adbbuff -g IDXBUF01 ¥
        -n ADBUIDX01 ¥
        -p 2500000
```

注

「¥」は，半角のバックスラッシュに読み替えてください。

カーソルを移動させたら，-p オプションの「2500000」を削除してください。そのあとで，「250」を入力してください。

■入力結果

```
adbbuff -g IDXBUF01 ¥
        -n ADBUIDX01 ¥
        -p 250
```

メモ

グローバルバッファとは，DB エリアに格納されている表のデータを入出力するために使用される領域です。

6. サーバ定義ファイルの編集を終了します。

ここでは，上記以外のオペランドの編集は不要です。[Esc] キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，[Enter] キーを押してください。

```
:wq
```

編集モードで設定した内容が保存されます。これで，HADB サーバを稼働させるために必要な情報がサーバ定義ファイルに設定されます。

(2) クライアント定義ファイルを編集する

1. クライアント定義ファイルのひな形をコピーします。

次に示す OS のコマンドを入力して，[Enter] キーを押してください。クライアント定義ファイルのひな形をコピーします。

```
cp /home/adbmanager/server/sample/conf/client.def /home/adbmanager/server/conf/
```

2. コピーしたクライアント定義ファイルのひな形を開きます。

次に示す OS のコマンドを入力して，[Enter] キーを押してください。vi コマンドを実行することで，クライアント定義ファイルが編集できるようになります。

```
vi /home/adbmanager/server/conf/client.def
```

3. HADB サーバのホスト名を指定します。

カーソルキーを押して，次に示すオペランドまでカーソルを移動させてください。

```
set adb_clt_rpc_srv_host = XXXXX
```

カーソルを移動したら，「XXXXX」を削除してください。そのあとで，localhost を入力してください。

■入力結果

```
set adb_clt_rpc_srv_host = localhost
```

4. クライアント定義ファイルの編集を終了します。

ここでは，上記以外のオペランドの編集は不要です。編集モードでホスト名を入力したら，［Esc］キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，［Enter］キーを押してください。

```
:wq
```

編集モードで設定した内容が保存されます。これで，AP やコマンドを実行するために必要な情報がクライアント定義ファイルに設定されます。

> **メモ**
>
> クライアント定義ファイルは AP を実行する HADB サーバ，または HADB クライアントで定義します。ここでは HADB サーバで AP を実行するため，HADB サーバ内にあるクライアント定義ファイルを編集しています。

(3) 関連項目

- 「7. サーバ定義の設計」
- 「8.5 サーバ定義の作成および変更方法」
- マニュアル『HADB AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』の『クライアント定義の作成』

3.6　DB エリアを作成する（データベースの初期設定をする）

データベースの初期設定をして，DB エリアを作成します。

DB エリアとは，表やインデクスなどを格納する論理的な領域のことです。DB エリアは，「3.4　DB ディレクトリを作成する」で作成した DB ディレクトリに格納されます。

また，DB エリアを作成する場合，どのような DB エリアを作成するかを指定した**初期設定オプションファイル**が必要です。初期設定オプションファイルの編集方法，および DB エリアの作成方法について説明します。

DB エリアの作成は HADB 管理者（adbmanager）で行います。

> **メモ**
>
> ここでは，データベースの初期設定を行って，DB エリアを作成する方法を説明します。データベースの初期設定を行う場合，DB エリアの作成コマンド（adbinit コマンド）を使用します。DB エリアの作成コマンド（adbinit コマンド）は，HADB サーバのインストール後，最初に DB エリアを作成する場合に使用します。
>
> なお，データベースの初期設定後に，DB エリアの追加・変更を行いたい場合，DB エリアの作成コマンド（adbinit コマンド）は使用しません。その場合は，DB エリアの追加・変更コマンド（adbmodarea コマンド）を使用します。DB エリアの追加・変更コマンド（adbmodarea コマンド）については，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』で説明しています。

3.6.1　初期設定オプションファイルを編集する

ここでは，vi コマンドを使用して初期設定オプションファイルを編集する方法について説明します。

初期設定オプションファイルには，ファイルのひな形があります。そのため，ファイルのひな形をコピーしてから編集します。

初期設定オプションファイルのひな形の格納先を次に示します。

- 初期設定オプションファイルのひな形：/home/adbmanager/server/sample/conf/adbinit.opt

ひな形をコピーして編集した初期設定オプションファイルの格納先を次に示します。

- 初期設定オプションファイルの格納先：/home/adbmanager/server/conf/
  編集した初期設定オプションファイルを上記に格納することで，DB エリアを作成できます。

> **メモ**
>
> 初期設定オプションファイルは，テキストエディタで開いて編集することもできます。

(1) DB エリアのページサイズを編集する

1. 初期設定オプションファイルのひな形をコピーします。

   HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，[Enter] キーを押してください。初期設定オプションファイルのひな形をコピーします。

   ```
   cp /home/adbmanager/server/sample/conf/adbinit.opt /home/adbmanager/server/conf/
   ```

2. コピーした初期設定オプションファイルのひな形を開きます。

   HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，[Enter] キーを押してください。vi コマンドを実行することで，初期設定オプションファイルが編集できるようになります。

   ```
   vi /home/adbmanager/server/conf/adbinit.opt
   ```

3. 作成する DB エリアのページサイズを編集します。

   [I] キーを押して，編集モードに移行させてください。編集モードに移行させたら，カーソルキーを押して，次に示すオペランドまでカーソルを移動させてください。

   ```
   adbinitdbarea -n ADBUIDX01 -p 8
   ```

   カーソルを移動させたら，デフォルトで入力されている-p オプションの「8」を削除してください。そのあとで，「4」を入力してください。

   ■入力結果

   ```
   adbinitdbarea -n ADBUIDX01 -p 4
   ```

4. 初期設定オプションファイルの編集を終了します。

   ここでは，上記以外のオペランドの編集は不要です。編集モードで作成する DB エリアのページサイズを入力したら，[Esc] キーを押してコマンドモードに移行させてください。コマンドモードに移行させたら，次に示すコマンドを入力して，[Enter] キーを押してください。

   ```
   :wq
   ```

   編集モードで設定した内容が保存されます。これで，DB エリアを作成するために必要な情報が初期設定オプションファイルに設定されます。

(2) 関連項目

マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』の『adbinit コマンドの指定形式』の『初期設定オプションの形式』

3.6.2　DB エリアを作成する

初期設定オプションファイルを編集したら，データベースの初期設定をして，DB エリアを作成します。

メモ

ここでは，データベースの初期設定を行って，DB エリアを作成する方法を説明します。データベースの初期設定を行う場合，DB エリアの作成コマンド（adbinit コマンド）を使用します。DB エリアの作成コマンド（adbinit コマンド）は，HADB サーバのインストール後，最初に DB エリアを作成する場合に使用します。

なお，データベースの初期設定後に，DB エリアの追加・変更を行いたい場合，DB エリアの作成コマンド（adbinit コマンド）は使用しません。その場合は，DB エリアの追加・変更コマンド（adbmodarea コマンド）を使用します。DB エリアの追加・変更コマンド（adbmodarea コマンド）については，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』で説明しています。

(1)　DB エリアの作成コマンドを実行する

HADB 管理者（adbmanager）で次に示す HADB のコマンドを入力して，［Enter］キーを押してください。DB エリアが作成されます。

```
adbinit -u ADBUSER01 -p '#HelloHADB_01' /home/adbmanager/server/conf/adbinit.opt /home/adbmanager/db
```

メモ

adbinit コマンドを実行時に「bash: adbinit: コマンドが見つかりません」と出力された場合は，設定した環境変数が誤っているおそれがあります。「3.5.1　環境変数を設定する」の「(2)　.bashrc ファイルに設定した環境変数を確認する」を参照して，設定した環境変数が正しいかどうかを確認してください。

adbinit コマンドに指定する各オプションについて，次に示します。

- -u ADBUSER01

  HADB ユーザのユーザ ID（認可識別子）を指定します。DB エリアが作成されるときに，-u オプションに指定した HADB ユーザもあわせて作成されます。ここでは，HADB ユーザのユーザ ID（認可識別子）としてADBUSER01 を指定してください。

  ここで作成される HADB ユーザ（ADBUSER01）は，表の作成・検索で必要な HADB ユーザを作成するときに使用します。

> **メモ**
>
> この章では，HADB ユーザを作成する方法を説明するために，ADBUSER01 を使用して表の作成・検索で必要な HADB ユーザ（ADBUSER02）を新規に作成する手順としています。詳細については，「3.7　表の作成・検索用の HADB ユーザを作成する」を参照してください。

- -p '#HelloHADB_01'

  -u オプションに指定したユーザ ID（ADBUSER01）のパスワードを指定します。ここでは，パスワードとして#HelloHADB_01 を指定してください。

- /home/adbmanager/server/conf/adbinit.opt

  初期設定オプションファイルの格納先を絶対パスで指定します。

- /home/adbmanager/db

  作成される DB エリアを格納する DB ディレクトリ名を絶対パスで指定します。なお，ここで指定する DB ディレクトリ名と，サーバ定義のadb_db_path オペランドに指定した DB ディレクトリ名は同じにしてください。

DB エリアを作成すると，メッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA90000-I adbinit processing started.
KFAA96201-I Database initialization started.
KFAA96204-I The DB directory "/home/adbmanager/db" is initialized.
KFAA96233-I Initialization of file "ADBUIDX01" is complete. (size = 5456 KB, information
= 7fd0dc15-f7b8-45ca-a178-1a1292c80fcc)
KFAA96233-I Initialization of file "ADBUTBL01" is complete. (size = 5440 KB, information
= df7978f1-9f7a-4bcf-a477-61056297a71c)
KFAA96202-I Database initialization ended. (return code = 0)
KFAA90001-I adbinit processing ended. (return code = 0)
```

(2)　作成した DB エリアを確認する

adbinit コマンドで DB エリアが作成されているかどうかを確認するため，次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls /home/adbmanager/db
```

ls コマンドを実行することで，指定したディレクトリ下にディレクトリおよびファイルが作成されているかどうかを確認できます。実行結果の例を次に示します。

■実行結果の例

```
ADBDIC  ADBMST  ADBSTBL  ADBSYS  ADBUIDX01  ADBUTBL01  ADBWORK  ADBWRK  SPOOL
```

DB エリアが作成されていることを確認できたら，DB エリアの作成は完了です。

(3)　関連項目

■データベースの初期設定を行って DB エリアを作成する場合

DB エリアの作成コマンド（adbinit コマンド）を使用する場合の関連項目を次に示します。

- マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』
- 「9.2　データベースを初期設定する（データ用 DB エリアを作成する)」

■データベースの初期設定後に DB エリアの追加・変更を行う場合

DB エリアの追加・変更コマンド（adbmodarea コマンド）を使用する場合の関連項目を次に示します。

- マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』
- 「11.10　データ用 DB エリアの運用」

3.7　表の作成・検索用の HADB ユーザを作成する

データベースの初期設定をして DB エリアを作成したら，表の作成・検索用の HADB ユーザを作成します。

ここでは，DB エリア作成時に作成された HADB ユーザ（ADBUSER01）を使用して，表の作成・検索で必要な HADB ユーザ（ADBUSER02）の作成方法，および作成した HADB ユーザ（ADBUSER02）に権限を付与する方法について説明します。

ADBUSER01 は，2 つの権限（DBA 権限およびCONNECT 権限）を持っているため，HADB ユーザの作成および権限の付与ができます。

HADB ユーザの作成および権限の付与をする場合，OS ユーザは HADB 管理者（adbmanager）を使用します。

> **メモ**
>
> この章では，HADB ユーザを作成する方法を説明するために，ADBUSER01 を使用して表の作成・検索で必要な HADB ユーザ（ADBUSER02）を新規に作成する手順としています。
>
> なお，HADB ユーザ（ADBUSER02）を新規に作成しないで，ADBUSER01 に表の作成・検索で必要な権限（スキーマ定義権限）を付与することもできます。
>
> 権限の詳細については，「2.7　権限」を参照してください。

3.7.1　HADB サーバを開始する

SQL 文を実行して HADB ユーザを作成するために，HADB サーバを開始します。

(1)　HADB サーバの開始コマンドを実行する

HADB 管理者（adbmanager）で次に示す HADB のコマンドを入力して，[Enter] キーを押してください。HADB サーバが開始されます。

```
adbstart
```

HADB サーバが開始されると，メッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA90000-I adbstart processing started.
KFAA91105-I The HADB system was started normally. (HADB server operation mode = "NORMAL")
KFAA90001-I adbstart processing ended. (return code = 0)
```

(2) 関連項目

- 「10.2.1 HADB サーバの開始方法」
- マニュアル『HADB コマンドリファレンス』の『adbstart（HADB サーバの開始）』

3.7.2 HADB サーバに接続する

HADB サーバを開始したら，HADB ユーザを作成するためのクライアントプログラム（adbsql コマンド）から HADB サーバに接続します。

(1) 接続コマンドを実行する

1. adbsql コマンドを実行します。

   HADB 管理者（adbmanager）で次に示す HADB のコマンドを入力し，［Enter］キーを押してください。

```
adbsql
```

2. ユーザ ID（認可識別子）の入力要求に応答します。

   adbsql コマンドを実行すると，HADB ユーザのユーザ ID（認可識別子）を入力するよう要求されます。ここで指定したユーザ ID（認可識別子）で，HADB サーバに接続します。ここでは，ADBUSER01 を指定して［Enter］キーを押してください。

```
  USER-ID？
ADBUSER01
```

3. パスワードの入力要求に応答します。

   ユーザ ID（認可識別子）を入力したら，パスワードを入力するよう要求されます。ここでは，#HelloHADB_01 を指定して［Enter］キーを押してください。

```
  PASSWORD？
#HelloHADB_01     入力したパスワードは表示されません。
```

(2) HADB サーバに接続したことを確認する

パスワードを指定して［Enter］キーを押すと，HADB サーバに接続され，SQL 文を入力できるようになります。実行結果の例を次に示します。

■実行結果の例

```
COMMAND ？      +----2----+----3----+----4----+----5----+----6----+----7----+
```

上記の実行結果が表示されていれば，SQL 文を入力できます。

重要

入力したHADBユーザのユーザID（認可識別子）およびパスワードが正しくない場合，HADBサーバとの接続に失敗してKFAA30561-Eメッセージが出力されます。その場合は，「3.12.4 HADBサーバの接続時にKFAA30561-Eメッセージが出力された」を参照してください。

(3) 関連項目

マニュアル『HADB　コマンドリファレンス』の『adbsql（SQLの実行)』

3.7.3 HADBユーザを作成する

SQL文を入力できるようになったら，表の作成・検索で必要なHADBユーザを作成します。

(1) HADBユーザを作成するSQL文を実行する

HADBユーザを作成するためには，定義系SQLのCREATE USER文を使用します。ここでは，表の作成・検索用のHADBユーザとして，次に示すHADBユーザを作成します。

- ユーザID（認可識別子）：ADBUSER02
- パスワード：#HelloHADB_02

次に示すSQL文を入力して，[Enter]キーを押してください。

なお，末尾に必ず半角のセミコロン（;）を入力してください。SQL文が実行されません。

```
CREATE USER "ADBUSER02" IDENTIFIED BY '#HelloHADB_02';
```

これで，HADBユーザ「ADBUSER02」が作成されます。なお，作成されたADBUSER02には，表の作成・検索するために必要な2つの権限（CONNECT権限およびスキーマ定義権限）を持っていません。そのため，必要な権限を付与する必要があります。

(2) HADBユーザを作成した結果を確認する

HADBユーザ「ADBUSER02」が作成された場合，次に示すメッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA96403-I SQL processing completed.
```

HADBユーザが作成されたことを確認したら，HADBユーザの作成は完了です。

(3)　関連項目

「9.4　実表を定義する HADB ユーザを作成する」

3.7.4　作成した HADB ユーザに権限を付与する

ここでは，HADB ユーザ「ADBUSER02」に対して，表の作成・検索に必要な権限を付与します。

(1)　権限を付与する SQL 文を実行する

HADB ユーザに権限を付与するためには，定義系 SQL のGRANT 文を使用します。ここでは，表の作成・検索で必要な権限として，次に示す 2 つの権限を HADB ユーザ「ADBUSER02」に付与します。

- CONNECT 権限
- スキーマ定義権限

次に示す SQL 文を入力して，［Enter］キーを押してください。

なお，末尾に必ず半角のセミコロン（ ; ）を入力してください。SQL 文が実行されません。

```
GRANT CONNECT,SCHEMA TO ”ADBUSER02”;
```

これで，HADB ユーザ「ADBUSER02」に，CONNECT 権限およびスキーマ定義権限が付与されます。

(2)　権限を付与した結果を確認する

HADB ユーザ「ADBUSER02」に権限を付与した場合，次に示すメッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA96403-I SQL processing completed.
```

権限が付与されたことを確認したら，権限の付与は完了です。

(3)　HADB サーバへの接続を終了する

HADB ユーザを作成して権限を付与したら，adbsql コマンドを終了します。次に示すadbsql サブコマンドを入力して，［Enter］キーを押してください。

なお，末尾に必ず半角のセミコロン（ ; ）を入力してください。SQL 文が実行されません。

```
#EXIT;
```

adbsql コマンドが終了し，ほかのコマンドを入力できる状態になります。実行結果の例を次に示します。

■実行結果の例

```
[adbmanager@localhost ~]$
```

(4) 関連項目

「9.4 実表を定義する HADB ユーザを作成する」

3.8　表を作成する

表の作成・検索用の HADB ユーザ（ADBUSER02）を作成したら，サンプルデータを格納するための表を作成します。

3.8.1　HADB サーバに接続する

HADB サーバを開始したら，表を作成するためのクライアントプログラム（adbsql コマンド）から HADB サーバに接続します。

(1)　接続コマンドを実行する

1. adbsql コマンドを実行します。

   HADB 管理者（adbmanager）で次に示す HADB のコマンドを入力し，［Enter］キーを押してください。

```
adbsql
```

2. ユーザ ID（認可識別子）の入力要求に応答します。

   adbsql コマンドを実行すると，HADB ユーザのユーザ ID（認可識別子）を入力するよう要求されます。ここで指定したユーザ ID（認可識別子）で，HADB サーバに接続します。ここでは，ADBUSER02 を指定して［Enter］キーを押してください。

```
  USER-ID？
ADBUSER02
```

3. パスワードの入力要求に応答します。

   ユーザ ID（認可識別子）を入力したら，パスワードを入力するよう要求されます。ここでは，#HelloHADB_02 を指定して［Enter］キーを押してください。

```
  PASSWORD？
#HelloHADB_02     入力したパスワードは表示されません。
```

(2)　HADB サーバに接続したことを確認する

パスワードを指定して［Enter］キーを押すと，HADB サーバに接続され，SQL 文を入力できるようになります。実行結果の例を次に示します。

■実行結果の例

```
COMMAND ？      +----2----+----3----+----4----+----5----+----6----+----7----+
```

上記の実行結果が表示されていれば，SQL 文を入力できます。

(3) 関連項目

マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』

3.8.2 スキーマを定義する

SQL 文を入力できるようになったら，スキーマを定義します。スキーマは，表を包括する論理的な概念のことです。スキーマを定義しないと，表を作成できません。

(1) スキーマを定義する SQL 文を実行する

スキーマを定義するためには，定義系 SQL のCREATE SCHEMA 文を使用します。次に示す SQL 文を入力して，[Enter] キーを押してください。

なお，末尾に必ず半角のセミコロン（ ; ）を入力してください。SQL 文が実行されません。

```
CREATE SCHEMA ADBUSER02;
```

これで，「ADBUSER02」ユーザのスキーマが定義されます。ここで定義するスキーマ名は，adbsql コマンドのユーザ ID（認可識別子）と同じにしてください。

(2) スキーマを定義した結果を確認する

スキーマ「ADBUSER02」が定義された場合，次に示すメッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA96403-I SQL processing completed.
```

スキーマが定義されたことを確認したら，スキーマの定義は完了です。

(3) 関連項目

マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE SCHEMA（スキーマの定義)』

3.8.3 表を作成する

ここでは，サンプルデータを格納する表として，「SAMPLE」という名称の表を作成します。

メモ

サンプル AP（$ADBDIR/sample）の認可識別子（SAMPLE）と，この章で説明している認可識別子（ADBUSER02）は異なります。もし，サンプル AP を使用する場合は，認可識別子を変更してから使用してください。

(1) 表を作成する SQL 文を実行する

表を作成するには，定義系 SQL のCREATE TABLE 文を使用します。次に示す SQL 文を入力して，[Enter] キーを押してください。

```
CREATE TABLE ”SAMPLE” (”STATECODE” SMALLINT, ”STATENAME” VARCHAR(15),
```

NEXT ?と表示されます。続けて，次に示す SQL 文を入力して，[Enter] キーを押してください。

```
”ZIPCODE” CHAR(15), ”ADDRESS” VARCHAR(100), ”AREA” DECIMAL(19))
```

NEXT ?と表示されます。続けて，次に示す SQL 文を入力して，[Enter] キーを押してください。

```
IN ADBUTBL01;
```

入力したCREATE TABLE 文の詳細について，次の表に示します。

表 3-6　指定する CREATE TABLE 文の詳細

項番	入力内容	説明	
1	”SAMPLE”	作成する表名を指定します。	
2	”STATECODE” SMALLINT	表の列名と，列のデータ型を指定します。	列名として「STATECODE」（州番号）を指定します。STATECODE 列には 1～50 の数字を格納するため，データ型として「SMALLINT」を指定します。
3	”STATENAME” VARCHAR(15)		列名として「STATENAME」（州名）を指定します。STATENAME 列には最大 15 バイトの可変長の文字列を格納するため，データ型として「VARCHAR」を指定します。
4	”ZIPCODE” CHAR(15)		列名として「ZIPCODE」（郵便番号）を指定します。ZIPCODE 列には，15 バイトの固定長の文字列を格納するため，データ型として「CHAR」を指定します。
5	”ADDRESS” VARCHAR(100)		列名として「ADDRESS」（州議事堂の住所）を指定します。ADDRESS 列には，最大 100 バイトの可変長の文字列を格納するため，データ型として「VARCHAR」を指定します。
6	”AREA” DECIMAL(19)		列名として「AREA」（面積）を指定します。AREA 列には，19 桁の数字を格納するため，データ型として「DECIMAL」を指定します。
7	IN ADBUTBL01	表を格納する DB エリア名を指定します。	

メモ

ファイルに記述したSQL文から表を作成することもできます。入力するSQL文（CREATE TABLE文）が長文のため，adbsqlコマンドの入力を誤ってしまうような場合は，「(3)　HADBサーバへの接続を終了する」を実行したあとで，「3.12.2　表を作成できない」を参照してください。

(2)　表を作成した結果を確認する

表「SAMPLE」が作成された場合，次に示すメッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA96403-I SQL processing completed.
```

表が作成されたことを確認したら，表の作成は完了です。

(3)　HADBサーバへの接続を終了する

表を作成したら，adbsqlコマンドを終了します。次に示すadbsqlサブコマンドを入力して，[Enter]キーを押してください。

なお，末尾に必ず半角のセミコロン（;）を入力してください。SQL文が実行されません。

```
#EXIT;
```

adbsqlコマンドが終了し，ほかのコマンドを入力できる状態になります。実行結果の例を次に示します。

■実行結果の例

```
[adbmanager@localhost ~]$
```

(4)　関連項目

マニュアル『HADB　SQLリファレンス』の『定義系SQL』の『CREATE TABLE（表の定義）』

3.9　データを格納する

表「SAMPLE」を作成したら，サンプルデータを表に格納します。

表にサンプルデータを格納する場合，adbimport コマンドを実行します。その際，表に格納するサンプルデータそのものは指定しないで，サンプルデータの格納先を示したテキストファイル（**入力データパスファイル**）を指定します。そのため，データを格納する前に，入力データパスファイルを作成する必要があります。

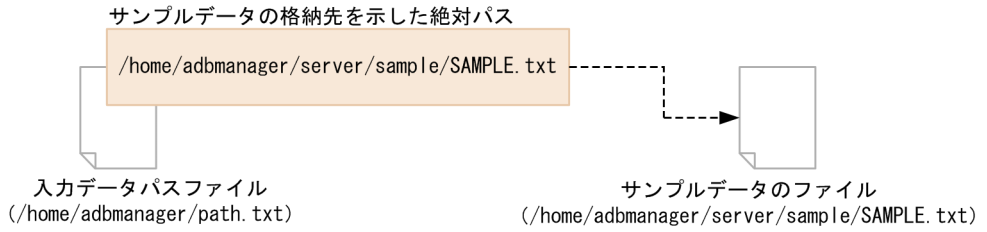
ここでは，入力データパスファイルの作成方法，および表にデータを格納する方法について説明します。

3.9.1　データの格納前に準備すること

表にデータを格納する前の準備作業として，サンプルデータの格納先を示したテキストファイル（**入力データパスファイル**）を作成します。

入力データパスファイルの詳細を次の図に示します。

図 3-4　入力データパスファイルの詳細



ここでは，vi コマンドを使用して入力データパスファイルを作成する方法について説明します。

(1)　入力データパスファイルを作成する

1. 入力データパスファイルを作成します。

   HADB 管理者（adbmanager）で次に示す OS のコマンドを入力して，[Enter] キーを押してください。ここでは，adbmanager のホームディレクトリ下にファイル名「path.txt」として作成します。

   ```
   vi /home/adbmanager/path.txt
   ```

2. サンプルデータの格納先を入力します。

   [I] キーを押してから，次に示す絶対パスを入力してください。この絶対パスが，サンプルデータの格納先になります。

   ```
   /home/adbmanager/server/sample/SAMPLE.txt
   ```

3. 入力データパスファイルの作成を終了します。

   [Esc] キーを押してから，次に示すコマンドを入力して，[Enter] キーを押してください。

```
:wq
```

これで，入力データパスファイルが作成されます。

(2) 作成した入力データパスファイルを確認する

vi コマンドで入力データパスファイルが作成されているかどうかを確認するため，次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
ls /home/adbmanager/path.txt
```

ls コマンドで指定したファイルが作成されているかどうかを確認できます。実行結果の例を次に示します。

■実行結果の例

```
/home/adbmanager/path.txt
```

入力データパスファイルが作成されていることを確認できたら，入力データパスファイルの作成は完了です。

(3) 関連項目

マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート）』

3.9.2 表にデータを格納する

入力データパスファイルを作成したら，表にサンプルデータを格納します

サンプルデータを格納する際，作成した**入力データパスファイル**および**インポートオプションファイル**を使用します。

インポートオプションファイルには，ファイルのひな形があるため，ひな形をコピーして使用します。ひな形の格納先を次に示します。

- インポートオプションファイルのひな形：/home/adbmanager/server/sample/conf/adbimport.opt

作成した入力データパスファイルと，ひな形をコピーしたインポートオプションファイルの格納先を次に示します。

- 入力データパスファイル：/home/adbmanager/path.txt
- インポートオプションファイルの格納先：/home/adbmanager/server/conf/
  インポートオプションファイルは，デフォルトの設定で使用します。

(1) インポートオプションファイルのひな形をコピーする

表にサンプルデータを格納するために，インポートオプションファイルのひな形をコピーしてください。

1. インポートオプションファイルのひな形をコピーします。

   次に示す OS のコマンドを入力して，[Enter] キーを押してください。インポートオプションファイルのひな形をコピーします。

```
cp /home/adbmanager/server/sample/conf/adbimport.opt /home/adbmanager/server/conf/
```

2. コピーしたインポートオプションファイルのひな形を確認します。

   次に示す OS のコマンドを入力して，[Enter] キーを押してください。インポートオプションファイルのひな形をコピーできているかどうかを確認します。

```
ls /home/adbmanager/server/conf/
```

   ■実行結果の例

```
adbimport.opt  adbinit.opt  client.def  server.def
```

   インポートオプションファイルのひな形をコピーできていれば，ls コマンドを実行すると，adbimport.opt が表示されます。

(2) データを格納するコマンドを実行する

HADB 管理者（adbmanager）で次に示す HADB のコマンドを入力して，[Enter] キーを押してください。

なお，下記のコマンドは読みやすさを考慮して，SAMPLE の前で改行しております。実際にコマンドを入力するときには，改行は不要です。adbimport.opt のあとに半角スペースを入力してから，続けてSAMPLE 以降を入力してください。

```
adbimport -u ADBUSER02 -p '#HelloHADB_02' -z /home/adbmanager/server/conf/adbimport.opt
          SAMPLE /home/adbmanager/path.txt
```

メモ

adbimport コマンドは，HADB サーバが稼働していないと実行できません。KFAA90004-E メッセージが出力された場合は，「3.7.1 HADB サーバを開始する」の「(1) HADB サーバの開始コマンドを実行する」を参照して HADB サーバを開始してください。

adbimport コマンドに指定する各オプションについて，次に示します。

表 3-7 adbimport コマンドに指定するオプションの詳細

項番	オプション	説明
1	-u ADBUSER02	adbimport コマンドを実行する HADB ユーザのユーザ ID（認可識別子）を指定します。ここでは，ADBUSER02 を指定します。
2	-p '#HelloHADB_02'	-u オプションに指定した HADB ユーザのパスワードを指定します。ここでは，#HelloHADB_02 を指定します。

項番	オプション	説明
3	-z /home/adbmanager/server/conf/adbimport.opt	データを格納するときにオプションを指定したい場合，インポートオプションファイルを指定します。 サンプルデータを格納する場合，デフォルトの内容が設定されているインポートオプションファイルを指定しないと，エラーになります。
4	SAMPLE	データを格納する表の表名を指定します。サンプルデータは表「SAMPLE」に格納するため，SAMPLE を指定します。
5	/home/adbmanager/path.txt	入力データパスファイルの格納先を絶対パスで指定します。

(3) データを格納した結果を確認する

adbimport コマンドを実行すると，メッセージが表示されます。表示結果の例を次に示します。

■表示結果の例

```
KFAA90000-I adbimport processing started.
KFAA80202-I Import processing started. The table is ”ADBUSER02”.”SAMPLE”.
KFAA80203-I 50 rows loaded.
KFAA80204-I Import processing ended. (return code = 0)
KFAA90001-I adbimport processing ended. (return code = 0)
```

これで，表へのサンプルデータの格納は完了です。

(4) 関連項目

マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』

3.10　格納したデータを検索する

サンプルデータを表に格納したら，表が検索できるようになります。HADB サーバに接続してから，格納したデータを検索する SQL 文（SELECT 文）を実行する手順を次に示します。

格納したデータの検索は HADB 管理者（adbmanager）で行います。

3.10.1　HADB サーバに接続してデータを検索する

(1)　接続コマンドを実行する

1. adbsql コマンドを実行します。

   HADB 管理者（adbmanager）で次に示す HADB のコマンドを入力して，［Enter］キーを押してください。

```
adbsql
```

2. ユーザ ID（認可識別子）の入力要求に応答します。

   ここでは，ADBUSER02 を入力して［Enter］キーを押してください。

```
  USER-ID？
ADBUSER02
```

3. パスワードの入力要求に応答します。

   ここでは，#HelloHADB_02 を指定して［Enter］キーを押してください。

```
  PASSWORD？
#HelloHADB_02     入力したパスワードは表示されません。
```

パスワードを入力して［Enter］キーを押すと，HADB サーバに接続され，SQL 文を入力できるようになります。実行結果の例を次に示します。

■実行結果の例

```
COMMAND ？     +----2----+----3----+----4----+----5----+----6----+----7----+
```

上記の実行結果が表示されていれば，SQL 文を入力できます。

(2)　格納したすべてのデータを検索する

データを格納した表のすべての行を検索する SQL 文（SELECT 文）を入力して，［Enter］キーを押してください。

なお，末尾に必ず半角のセミコロン（；）を入力してください。SQL 文が実行されません。

```
SELECT * FROM "SAMPLE";
```

表のすべての行を検索すると，米国 50 州の州番号，州名，郵便番号，州議事堂の住所，および面積が出力されます。

検索が完了すると，メッセージが出力されます。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA96404-I 50 rows were selected.
```

(3) HADB サーバへの接続を終了する

次に示すadbsql サブコマンドを入力して，［Enter］キーを押してください。adbsql コマンドが終了します。

なお，末尾に必ず半角のセミコロン（；）を入力してください。SQL 文が実行されません。

```
#EXIT;
```

adbsql コマンドが終了し，ほかのコマンドを入力できる状態になります。実行結果の例を次に示します。

■実行結果の例

```
[adbmanager@localhost ~]$
```

(4) HADB サーバを終了する

次に示す HADB のコマンドを入力して，［Enter］キーを押してください。HADB サーバが終了します。

```
adbstop
```

adbstop コマンドを実行するとメッセージが出力されて，HADB サーバが終了します。出力されるメッセージの例を次に示します。

■出力されるメッセージの例

```
KFAA90000-I adbstop processing started.
KFAA91154-I The HADB system was terminated normally.
KFAA90001-I adbstop processing ended. (return code = 0)
```

これで，体験環境を使用した HADB サーバの操作体験は終了です。体験環境が不要な場合は，「3.11 HADB サーバをアンインストールする」を参照して，体験環境の HADB サーバをアンインストールしてください。

重要

この章に従って構築した体験環境は，あくまで HADB サーバの導入・操作を体験するためのものです。実際の業務で使用するための環境ではありません。そのため，この章で説明しているコマンドおよび SQL 文以外を体験環境で実行した場合は，エラーが発生するなど，正常に動作しないおそれがあります。

実際の業務で使用する環境を構築する場合は，このマニュアルの『第 3 編　設計編』を参照して必要な値を見積もった上で，『第 4 編　構築編』を参照して，環境を構築してください。

(5)　関連項目

- マニュアル『HADB　SQL リファレンス』の『SELECT 文の例題集』
- マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』の『adbsql サブコマンド』
- 「10.2.2　HADB サーバの終了方法」
- マニュアル『HADB　コマンドリファレンス』の『adbstop（HADB サーバの終了)』

3.11　HADB サーバをアンインストールする

ここでは，HADB サーバをアンインストールする方法について説明します。構築した HADB サーバの体験環境が不要になった場合，次に示す手順に従ってアンインストールします。

HADB サーバのアンインストールは，**スーパユーザ**および **HADB 管理者（adbmanager）**で行います。

> **重要**
>
> HADB サーバが終了している状態でアンインストールしてください。HADB サーバが稼働しているときには，アンインストールできません。

3.11.1　サーバディレクトリを削除する

HADB サーバをアンインストールするために，サーバディレクトリを削除します。

(1)　サーバディレクトリの削除コマンドを実行する

HADB 管理者（adbmanager）で，次に示す OS のコマンドを入力して，[Enter] キーを押してください。サーバディレクトリが削除されます。

```
rm -rf /home/adbmanager/server
```

(2)　サーバディレクトリが削除できたことを確認する

HADB 管理者（adbmanager）で，次に示す OS のコマンドを入力して，[Enter] キーを押してください。サーバディレクトリが正しく削除できたかどうかを確認できます。

```
ls -d /home/adbmanager/server
```

■実行結果の例

```
ls: cannot access /home/adbmanager/server: そのようなファイルやディレクトリはありません
```

上記のような実行結果が表示されていれば，サーバディレクトリが削除されています。

3.11.2　インストールデータを格納するディレクトリを削除する

HADB サーバのインストールデータを格納するディレクトリを削除します。

(1) インストールデータを格納するディレクトリの削除コマンドを実行する

HADB 管理者（adbmanager）で，次に示す OS のコマンドを入力して，[Enter] キーを押してください。インストールデータを格納するディレクトリが削除されます。

```
rm -rf /home/adbmanager/install
```

(2) インストールデータを格納するディレクトリが削除できたことを確認する

HADB 管理者（adbmanager）で，次に示す OS のコマンドを入力して，[Enter] キーを押してください。インストールデータを格納するディレクトリが正しく削除できたかどうかを確認できます。

```
ls -d /home/adbmanager/install
```

■実行結果の例

```
ls: cannot access /home/adbmanager/install/: そのようなファイルやディレクトリはありません
```

上記のような実行結果が表示されていれば，インストールデータを格納するディレクトリが削除されています。

3.11.3 通信情報ファイルを格納するディレクトリを削除する

HADB サーバをアンインストールしたら，通信情報ファイルを格納するディレクトリを削除します。

(1) 通信情報ファイルを格納するディレクトリの削除コマンドを実行する

削除が必要な通信情報ファイルを格納するディレクトリは，サーバマシンの OS のバージョンによって異なります。

(a) RHEL 6 の場合

次に示すディレクトリを削除します。

- /dev/HADB/pth
- /lib/udev/devices/HADB/pth

1. /dev/HADB/pth ディレクトリを削除します。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。/dev/HADB/pth ディレクトリが削除されます。

   ```
   rm -rf /dev/HADB/pth
   ```

2. /lib/udev/devices/HADB/pth ディレクトリを削除します。

スーパユーザで次に示すOSのコマンドを入力して，[Enter]キーを押してください。/lib/udev/devices/HADB/pthディレクトリが削除されます。

```
rm -rf /lib/udev/devices/HADB/pth
```

(b) RHEL 7以降の場合

次に示すディレクトリ，およびファイルを削除します。

- /dev/HADB/pth
- /etc/tmpfiles.d/dev-HADB-pth.conf

1. /dev/HADB/pthディレクトリを削除します。
   スーパユーザで次に示すOSのコマンドを入力して，[Enter]キーを押してください。/dev/HADB/pthディレクトリが削除されます。

   ```
   rm -rf /dev/HADB/pth
   ```

2. /etc/tmpfiles.d/dev-HADB-pth.confファイルを削除します。
   スーパユーザで次に示すOSのコマンドを入力して，[Enter]キーを押してください。/etc/tmpfiles.d/dev-HADB-pth.confファイルが削除されます。

   ```
   rm -f /etc/tmpfiles.d/dev-HADB-pth.conf
   ```

(2) 通信情報ファイルを格納するディレクトリを削除できたことを確認する

通信情報ファイルを格納するディレクトリが，正しく削除されたかどうかを確認します。

確認する内容は，サーバマシンのOSのバージョンによって異なります。

(a) RHEL 6の場合

1. 削除した/dev/HADB/pthディレクトリを確認します。
   スーパユーザで次に示すOSのコマンドを入力して，[Enter]キーを押してください。

   ```
   ls -d -l /dev/HADB/pth
   ```

   lsコマンドを実行すると，/dev/HADB/pthディレクトリが削除されているかどうかが確認できます。実行結果の例を次に示します。

   ■実行結果の例

   ```
   ls: cannot access /dev/HADB/pth: そのようなファイルやディレクトリはありません
   ```

   上記のような実行結果が表示されていれば，/dev/HADB/pthディレクトリは削除されています。

2. 削除した/lib/udev/devices/HADB/pthディレクトリを確認します。
   スーパユーザで次に示すOSのコマンドを入力して，[Enter]キーを押してください。

```
ls -d -l /lib/udev/devices/HADB/pth
```

ls コマンドを実行すると，/lib/udev/devices/HADB/pth ディレクトリが削除されているかどうかが確認できます。実行結果の例を次に示します。

■実行結果の例

```
ls: cannot access /lib/udev/devices/HADB/pth: そのようなファイルやディレクトリはありません
```

上記のような実行結果が表示されていれば，/lib/udev/devices/HADB/pth ディレクトリは削除されています。

(b) RHEL 7 以降の場合

1. 削除した/dev/HADB/pth ディレクトリを確認します。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

   ```
   ls -d -l /dev/HADB/pth
   ```

   ls コマンドを実行すると，/dev/HADB/pth ディレクトリが削除されているかどうかが確認できます。実行結果の例を次に示します。

   ■実行結果の例

   ```
   ls: cannot access /dev/HADB/pth: そのようなファイルやディレクトリはありません
   ```

   上記のような実行結果が表示されていれば，/dev/HADB/pth ディレクトリは削除されています。

2. 削除した/etc/tmpfiles.d/dev-HADB-pth.conf ファイルを確認します。

   スーパユーザで次に示す OS のコマンドを入力して，[Enter] キーを押してください。

   ```
   ls -l /etc/tmpfiles.d/dev-HADB-pth.conf
   ```

   ls コマンドを実行すると，/etc/tmpfiles.d/dev-HADB-pth.conf ファイルが削除されているかどうかが確認できます。実行結果の例を次に示します。

   ■実行結果の例

   ```
   ls: cannot access /etc/tmpfiles.d/dev-HADB-pth.conf: そのようなファイルやディレクトリはありません
   ```

   上記のような実行結果が表示されていれば，/etc/tmpfiles.d/dev-HADB-pth.conf ファイルは削除されています。

3.11.4 HADB 管理者と HADB 管理グループを削除する

HADB サーバをインストールする際に，HADB を管理する OS ユーザ（**HADB 管理者**）と HADB 管理者が所属するグループ（**HADB 管理グループ**）を OS に設定しました。これらを削除する方法について説明します。

削除する HADB 管理者と HADB 管理グループを次に示します。

- HADB 管理者：adbmanager
- HADB 管理グループ：adbgroup

(1) HADB 管理者を削除する

次に示す OS のコマンドを入力し，［Enter］キーを押してください。

```
userdel -r adbmanager
```

userdel コマンドの各オプションの詳細について，次に示します。

- -r

  指定すると，ユーザを削除する際，ホームディレクトリも合わせて削除します。そのため，adbmanager のホームディレクトリ（/home/adbmanager/）が削除されます。

- adbmanager

  削除するユーザ名を指定します。

これで，HADB 管理者の削除は完了です。

(2) HADB 管理グループを削除する

スーパユーザで次に示す OS のコマンドを入力し，［Enter］キーを押してください。

```
groupdel adbgroup
```

groupdel コマンドのオプションの詳細について，次に示します。

- adbgroup

  削除するグループ名を指定します。

これで，HADB 管理グループの削除は完了です。

3.12　よくある質問と対処方法

ここでは，HADB サーバの体験環境を構築したときによくある質問と対処方法について説明します。

3.12.1　DB ディレクトリをスーパユーザで作成してしまった

HADB 管理者（adbmanager）で作成する DB ディレクトリ（/home/adbmanager/db）を，スーパユーザで作成してしまった場合，作成した DB ディレクトリを削除してください。

削除方法を次に示します。

1. スーパユーザでログインします。
   HADB 管理者（adbmanager）でログインしていた場合，スーパユーザでログインし直してください。
2. DB ディレクトリを削除します。
   次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
rmdir /home/adbmanager/db
```

これで，DB ディレクトリを削除できます。HADB 管理者（adbmanager）でログインし直してから，DB ディレクトリを作成してください。

なお，rmdir コマンドでは削除できない場合は，次に示す OS のコマンドを入力して，[Enter] キーを押してください。

```
rm -rf /home/adbmanager/db
```

3.12.2　表を作成できない

実行する SQL 文（CREATE TABLE 文）が長文のため，adbsql コマンドの入力を誤ってしまうような場合，次の方法で表を作成してみてください。

ファイルに記述した SQL 文の実行方法を次に示します。

> **メモ**
>
> 手順 1.～手順 3.で作成するファイル（infile）は，テキストエディタを使用して作成することもできます。

1. SQL 文を記述するファイル（infile）を作成します。
   次に示す OS のコマンドを入力して，[Enter] キーを押してください。ファイルの作成先はホームディレクトリ下（/home/adbmanager），ファイル名は「infile」としてください。

```
vi /home/adbmanager/infile
```

2. 実行する SQL 文を入力します。

   [I] キーを押してから，次に示す SQL 文を入力します。

   1 行目にADBUSER02 を入力して，2 行目に#HelloHADB_02 を入力してください。3 行目からCREATE TABLE 文を入力してください。

```
ADBUSER02
#HelloHADB_02
CREATE TABLE ”SAMPLE” (”STATECODE” SMALLINT, ”STATENAME” VARCHAR(15),
”ZIPCODE” CHAR(15), ”ADDRESS” VARCHAR(100), ”AREA” DECIMAL(19))
IN ADBUTBL01;
```

3. SQL 文を記述するファイル（infile）の作成を終了します。

   [Esc] キーを押してから，次に示すコマンドを入力して，[Enter] キーを押してください。

```
:wq
```

   これで，SQL 文を記述するファイル（infile）が作成されます。

4. SQL 文を記述するファイル（infile）を実行します。

   HADB サーバを開始した状態で，次に示す HADB のコマンドを実行してください。

```
adbsql -V < infile
```

   上記のコマンドを実行すると，手順 2.で入力した SQL 文が一括で実行されます。実行結果の例を次に示します。

   ■実行結果の例

```
  USER-ID ?
ADBUSER02

  PASSWORD ?
#HelloHADB_02

COMMAND ? +----2----+----3----+----4----+----5----+----6----+----7----+
CREATE TABLE ”SAMPLE” (”STATECODE” SMALLINT, ”STATENAME” VARCHAR(15),
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+
”ZIPCODE” CHAR(15), ”ADDRESS” VARCHAR(100), ”AREA” DECIMAL(19))
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+
IN ADBUTBL01;
KFAA96403-I SQL processing completed.
```

   これで，表が作成されます。

3.12.3 HADB サーバの開始時に KFAA91104-Q メッセージが出力された

HADB サーバの開始コマンド（adbstart コマンド）を実行したときに，KFAA91104-Q メッセージが出力された場合の対処方法について説明します。

HADB サーバの稼働中に，HADB サーバの終了コマンド（adbstop コマンド）を実行しないで OS をシャットダウンしたあとで，adbstart コマンドを実行した場合に，KFAA91104-Q メッセージ（トラブルシュート情報を削除してよいかどうかの確認メッセージ）が出力されます。

■KFAA91104-Q メッセージの出力例

```
KFAA90000-I adbstart processing started.
KFAA91104-Q Are you sure you want to discard trouble shoot information? (y/N)
```

HADB サーバを正しい手順で終了していないため，HADB サーバが異常終了した結果，トラブルシュート情報が出力されています。

この章で構築する体験環境については，出力されたトラブルシュート情報は不要であるため，削除します。

重要

体験環境ではなく実際の業務で使用する環境で，adbstart コマンド実行時にKFAA91104-Q メッセージが出力された場合は，トラブルシュート情報を取得する必要があるかどうかを確認してください。詳細については，マニュアル『HADB　コマンドリファレンス』の『adbstart (HADB サーバの開始)』の注意事項を参照してください。

KFAA91104-Q メッセージに対して，[Y] キーを押してください。[Y] キーを押すと，トラブルシュート情報が削除されたあとで，HADB サーバが開始されます。

■ [Y] キーを押したあとのメッセージの出力例

```
KFAA91105-I The HADB system was started normally. (HADB server operation mode = "NORMAL")
KFAA90001-I adbstart processing ended. (return code = 0)
```

リターンコードが0 なら，HADB サーバが開始されています。

3.12.4　HADB サーバの接続時に KFAA30561-E メッセージが出力された

HADB サーバへの接続コマンド（adbsql コマンド）を実行して，HADB ユーザのユーザ ID（認可識別子）およびパスワードを入力するときに，ユーザ ID（認可識別子）またはパスワードを誤り，KFAA30561-E メッセージが出力された場合の対処方法について説明します。

■KFAA30561-E メッセージの出力例

```
KFAA30561-E The specified authorization identifier or password is invalid.
```

この場合，HADB サーバへの接続に失敗しています。

HADB サーバの接続に失敗した場合，HADB サーバに再接続する必要があります。次に示すadbsql サブコマンドを入力して，[Enter] キーを押してください。

なお，末尾には必ず半角のセミコロン（；）を入力してください。SQL 文が実行されません。

```
#CONNECT;
```

ユーザ ID（認可識別子）およびパスワードの入力を求められるため，正しいユーザ ID（認可識別子）およびパスワードを入力してください。入力した内容に問題なければ，HADB サーバに再接続できます。

4 システム設計

この章では，システム構築するためのシステム設計（データベース，リソース，およびサーバ定義の設計）の流れについて説明します。

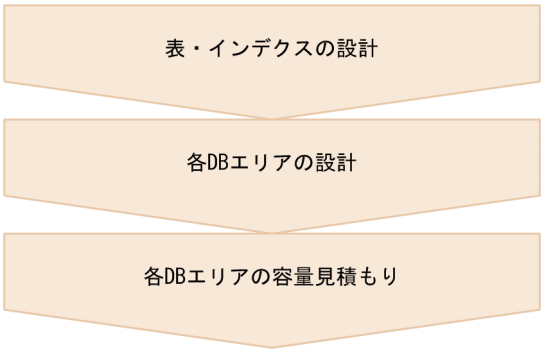
また，HADB サーバに格納するデータ（入力データ）の容量を基に，リソースとメモリ所要量を事前に見積もる方法について説明します。

4.1　システム設計の流れ

システム設計で行う次の設計の流れを図に示します。

- データベースの設計
- リソースの設計
- サーバ定義の設計

図 4-1　システム設計の流れ（データベースの設計）



システム設計で行うデータベースの設計については，「5.　データベースの設計」を参照してください。

メモ

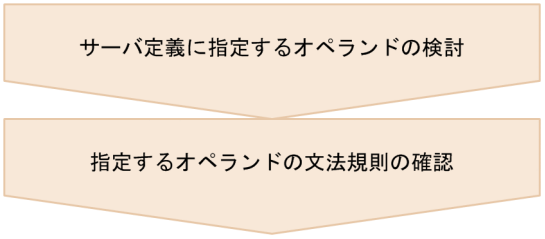
データベースを設計する前に，HADB サーバに格納するデータ（入力データ）の容量を基に，おおよそのデータベースの容量およびメモリ所要量を見積もることができます。詳細については，次の項目を参照してください。

- 「4.2　リソースの事前見積もり」
- 「4.3　メモリ所要量の事前見積もり」

図 4-2　システム設計の流れ（リソースの設計）

システム設計で行うリソースの設計については，「6.　リソースの設計」を参照してください。

図 4-3　システム設計の流れ（サーバ定義の設計）



システム設計で行うサーバ定義の設計については，「7.　サーバ定義の設計」を参照してください。

4.2　リソースの事前見積もり

HADB サーバに格納するデータ（入力データ）の容量を基に，各リソースの容量を見積もる方法について説明します。ここでは，次に示すディレクトリ下のディレクトリやファイルの容量を見積もります。

- DB ディレクトリ
- サーバディレクトリ
- アーカイブディレクトリ
- アンロードファイルの格納ディレクトリ
- 同義語辞書ファイルの格納ディレクトリ
- マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ

データベースを設計する前に，入力データを基におおよその容量を把握したい場合，次項以降の計算式を利用してください。ただし，詳細な容量は見積もれないため，あくまで目安としてください。

詳細な容量を見積もる場合は，「5.　データベースの設計」を参照してデータベースを設計してから，容量を見積もってください。

4.2.1　DB ディレクトリの事前見積もり

入力データを基に，DB ディレクトリ下のディレクトリやファイルの容量を見積もる計算式を次の表に示します。項番 1～項番 14 で見積もった値を合計した値が，DB ディレクトリ全体の容量になります。

ここで見積もった容量を基に，十分に空き容量のあるディスクを用意してください。

表 4-1　計算式（DB ディレクトリの容量）

項番	ディレクトリ名およびファイル名	説明	計算式（単位：ギガバイト）
1	$DBDIR/ADBMST	マスタディレクトリ用 DB エリアファイル	0.01
2	$DBDIR/ADBDIC	ディクショナリ用 DB エリアファイル	0.5
3	$DBDIR/ADBSTBL	システム表用 DB エリアファイル	0.5
4	$DBDIR/ADBWRK	作業表用 DB エリアファイル※12	*org_data*※1×1※2
5	$DBDIR/DBAREA※3	ローストア表を格納するデータ用 DB エリアファイル	*org_data*※1×2
6		カラムストア表を格納するデータ用 DB エリアファイル	*org_data*※1
7		B-tree インデクスを格納するデータ用 DB エリアファイル	*org_data*※1×1※4

項番	ディレクトリ名およびファイル名	説明	計算式（単位：ギガバイト）
8		テキストインデクスを格納するデータ用 DB エリアファイル	*org_data*※1×1.2※5
9		レンジインデクスを格納するデータ用 DB エリアファイル	*org_data*※1×0.00002＋0.2※6
10	$DBDIR/ADBSYS/ADBSLG	システムログファイル用ディレクトリ	1※7，※8
11	$DBDIR/ADBSYS/ADBSTS	ステータスファイル用ディレクトリ	
12	$DBDIR/ADBSYS/ADBUTL	コマンドステータスファイル用ディレクトリ	
13	$DBDIR/ADBWORK※9	作業用ディレクトリ※12	*db_idx*×4※10
14	$DBDIR/SPOOL※11	障害情報（コアファイル）の出力先ディレクトリ※12	2

（凡例）

org_data：入力データファイルの容量（単位：ギガバイト）

db_idx：項番 7『B-tree インデクスを格納するデータ用 DB エリアファイル』で求めた値（単位：ギガバイト）

注※1

入力データファイルを圧縮している場合は，入力データファイル内のデータを伸長したデータ量を代入してください。

注※2

作業表を使用する SQL の結果に大きく依存するため，データ量によっては計算式で求めた値以上の容量が必要になります。

注※3

adbinit コマンドで HADB 管理者が指定した名称になります。

注※4

B-tree インデクスを定義する列のデータ型およびデータ件数がわかっている場合，次に示す計算式で求めた方が，詳細に容量を見積もることができます。データ用 DB エリアに格納する B-tree インデクス数だけ計算してください。計算結果は，「表 4-1　計算式（DB ディレクトリの容量）」の『B-tree インデクスを格納するデータ用 DB エリアファイル』で求めた値に加算してください。

計算式（単位：ギガバイト）

$$db_idx = \left\lceil \frac{(20+KEYSZ) \times row_num \times \frac{ENT_NUM}{ENT_NUM-1}}{1,073,741,824} \right\rceil$$

変数の説明

- *KEYSZ*

  B-tree インデクスを定義する列の長さ（単位：バイト）

  「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を参照してください。

- *row_num*

  B-tree インデクスを定義した表のデータ件数

- *ENT_NUM*

  次に示す計算式から求めてください。

  計算式

$$ENT_NUM = MAX\left\{2, \left\downarrow 4,000 \div (20 + KEYSZ) \right\downarrow\right\}$$

注※5

テキストインデクスの定義時に「テキストインデクス表記ゆれ補正指定」，または，「テキストインデクスワード検索指定」を指定した場合は，次に示す計算式で求めてください。

計算式（単位：ギガバイト）

- 「テキストインデクス表記ゆれ補正指定」を指定した場合

$$org_data \times 2.4$$

- 「テキストインデクスワード検索指定」を指定した場合

$$org_data \times 2.8$$

- 「テキストインデクス表記ゆれ補正指定」，および「テキストインデクスワード検索指定」の両方を指定した場合

$$org_data \times 4$$

注※6

対象のレンジインデクスがマルチチャンク表に定義されている場合は，次に示す計算式で求めてください。

計算式（単位：ギガバイト）

$$\sum_{i=1}^{chunk_num} (chunk_data(i) \times 0.00002 + 0.2)$$

変数の説明

- *chunk_num*

  総チャンク数

- *chunk_data（i）*

  i 番目のチャンクのデータ量（単位：ギガバイト）

注※7

次に示す SQL 文およびコマンドで扱うデータ量に大きく依存するため，データ量によってはこの値以上の容量が必要になります。

- 更新系 SQL
- adbimport コマンド
- adbidxrebuild コマンド
- adbmergechunk コマンド

注※8

項番 10～項番 12 を合わせて，1 ギガバイトとして見積もってください。

注※9

次に示すコマンドに-w オプションを指定しない場合に，このディレクトリが使用されます。

- adbimport コマンド
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド

上記のコマンドに-w オプションを指定した場合，指定したディレクトリが作業用ディレクトリとして使用されます。

注※10

次に示すコマンドを実行するときだけ使用されます。

- adbimport コマンド
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド

B-tree インデクスを定義した列のデータ型および B-tree インデクスを定義した数に大きく依存するため，データ量によっては計算式で求めた値以上の容量が必要になります。

また，テキストインデクスを定義した場合は，次に示す計算式で求めてください。

計算式（単位：ギガバイト）

```
org_data×5
```

注※11

サーバ定義のadb_core_path オペランドを指定した場合は，$DBDIR/SPOOL ディレクトリ下ではなく，adb_core_path オペランドに指定したディレクトリ下にコアファイルが出力されます。

なお，コアファイルの出力が systemd-coredump によって制御されている場合，コアファイルは/var/lib/systemd/coredump ディレクトリ下に出力されます（デフォルトの設定では LZ4 形式で圧縮されて出力されます）。この場合，$DBDIR/SPOOL ディレクトリは，トラブルシュート情報の収集時にコアファイル展開用の一時ディレクトリとして使用されます。adb_core_path オペランドを指定したと

きは，adb_core_path オペランドに指定したディレクトリがコアファイル展開用の一時ディレクトリとして使用されます。

注※12

マルチノード機能を使用する場合は，全ノードで見積もりが必要となるディレクトリおよびファイルです。

4.2.2　サーバディレクトリの事前見積もり

入力データを基に，サーバディレクトリ下のファイル容量の見積もり値を次の表に示します。項番 1～項番 6 で見積もった値を合計した値が，サーバディレクトリ全体の容量になります。

表 4-2　見積もり値（サーバディレクトリの容量）

項番	ディレクトリ名およびファイル名	説明	見積もり値（単位：ギガバイト）
1	$ADBDIR/spool/adbmessage*XX*.log※1	サーバメッセージログファイル※5	0.064
2	$ADBDIR/spool/adbstatlog*XX*※1，※2	統計ログファイル※5	16
3	$ADBDIR/spool/adbdump*YYYYMMDDhhmmss*.*サーバプロセスのプロセス ID*※3	HADB ダンプファイル※5	100
4	$ADBDIR/spool/adbsqltrc*XX*.log※4	SQL トレースファイル※5	2
5	$ADBDIR/spool/adbcolumnize01	更新行のカラム化機能で使用するログファイル※5	0.1
6	$ADBDIR/spool/adboptlog*XX*※6	アクセスパス探索情報ログファイル※5	0.2

注※1

ファイル名の *XX* は通番であり，01～04 のどれかになります。

注※2

サーバ定義adb_sta_log_path オペランドを指定している場合は，サーバ定義adb_sta_log_path オペランドに指定したディレクトリ下に出力されます。

サーバ定義adb_sta_log_path オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式）」の adb_sta_log_path オペランドを参照してください。

注※3

ファイル名の *YYYYMMDDhhmmss* は，ファイルを生成した時刻になります。

注※4

ファイル名の *XX* は通番であり，01～08 のどれかになります。

注※5

マルチノード機能を使用する場合は，全ノードで見積もりが必要となるファイルです。

注※6

ファイル名の*XX*は通番であり，01～02のどちらかになります。

4.2.3　アーカイブディレクトリの事前見積もり

アーカイブディレクトリ下に格納されるアーカイブファイルの容量を見積もる計算式を，次に示します。見積もった値が，アーカイブディレクトリ全体の容量になります。

アーカイブマルチチャンク表を定義して，adbarchivechunkコマンドでチャンク内のデータをアーカイブする場合に，次に示す計算式で求めてください。

計算式（単位：ギガバイト）

$$\text{アーカイブディレクトリの事前見積もり} = \sum_{i=1}^{archive_chunk_num} (archive_chunk_data(i) \times 0.5)※$$

注※

アーカイブマルチチャンク表に格納されるデータによって，アーカイブ時の圧縮率が変わります。そのため，計算式で求めた値よりも，実際には大きな容量が必要になることがあります。

変数の説明

archive_chunk_num

アーカイブ対象のチャンク数（単位：個）

archive_chunk_data(i)

i番目のアーカイブ対象のチャンクに格納される元データのサイズ（単位：ギガバイト）

4.2.4　アンロードファイルの格納ディレクトリの事前見積もり

アンロードファイルの格納ディレクトリ下に格納されるアンロードファイルの容量の見積もり値を，次に示します。

アンロードファイルは，adbreorgsystemdataコマンドでシステム表を再編成する場合に作成されます。

値（単位：ギガバイト）

0.5※

注※

実際のアンロードファイルの容量は，再編成の対象とするシステム表の容量によって変わります。

4.2.5　同義語辞書ファイルの格納ディレクトリの事前見積もり

同義語辞書ファイルの格納ディレクトリ下に格納される同義語辞書ファイルの容量を見積もる計算式を，次に示します。見積もった値が，同義語辞書ファイルの格納ディレクトリ全体の容量になります。

adbsyndict コマンドで同義語辞書を作成する場合に，次に示す計算式で求めてください。

計算式（単位：ギガバイト）

$$\text{同義語辞書ファイルの格納ディレクトリの事前見積もり} = \sum_{i=1}^{syndict_num} (syndict_data(i) \times 24)$$

変数の説明

syndict_num

作成する同義語辞書の数（単位：個）

syndict_data(i)

各同義語辞書の同義語一覧定義ファイルのサイズ（単位：ギガバイト）

> **メモ**
>
> 同義語辞書ファイルの格納ディレクトリとは，サーバ定義のadb_syndict_storage_path オペランドに指定するディレクトリのことです。

4.2.6　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの事前見積もり

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの事前見積もり式は，「4.2.5　同義語辞書ファイルの格納ディレクトリの事前見積もり」と同じになります。見積もった値が，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ全体の容量になります。

なお，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリは，全ノードで見積もりが必要になります。

> **メモ**
>
> マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリとは，サーバ定義の adb_syndict_node_storage_path オペランドに指定するディレクトリのことです。

4.3　メモリ所要量の事前見積もり

ここでは，「4.2　リソースの事前見積もり」で求めた値を基に，メモリ所要量を見積もる方法について説明します。HADB サーバが使用するメモリは，**共有メモリ**および**プロセスメモリ**です。共有メモリには次の 4 種類があります。

- 共有メモリ管理
- グローバルバッファページ
- プロセス共通メモリ
- リアルスレッド固有メモリ

データベースを設計する前に，おおよそのメモリ所要量を把握したい場合，次項以降の計算式を利用してください。しかし，詳細なメモリ所要量は見積もれないため，あくまで目安としてください。

HADB サーバが使用するメモリ所要量は，次に示す処理のうち，どれを実行するかで大きく異なります。

■HADB サーバ開始後に実行する処理

- 通常運用（データベースへの接続，SQL 文の実行）を行う場合
- adbimport コマンドを実行してデータインポートを行う場合
- adbidxrebuild コマンドを実行してインデクスを再作成する場合
- adbgetcst コマンドを実行してコスト情報を収集する場合
- adbexport コマンドを実行してデータエクスポートを行う場合
- adbmergechunk コマンドを実行して複数のチャンクをマージする場合
- adbarchivechunk コマンドを実行してチャンクをアーカイブする場合
- adbunarchivechunk コマンドを実行してチャンクのアーカイブ状態を解除する場合

そのため，それぞれの処理で使用するメモリの所要量を求めてください。求めたメモリ所要量のうち，最も大きい値を HADB サーバが使用するメモリ所要量として見積もってください。

- **通常運用（データベースへの接続，SQL 文の実行）を行う場合**
  「4.3.1　通常運用時のメモリ所要量の事前見積もり」を参照してください。
- **adbimport コマンドを実行する場合**
  「4.3.2　adbimport コマンド実行時のメモリ所要量の事前見積もり」を参照してください。
- **adbidxrebuild コマンドを実行する場合**
  「4.3.3　adbidxrebuild コマンド実行時のメモリ所要量の事前見積もり」を参照してください。
- **adbgetcst コマンドを実行する場合**
  「4.3.4　adbgetcst コマンド実行時のメモリ所要量の事前見積もり」を参照してください。
- **adbexport コマンドを実行する場合**

「4.3.5　adbexport コマンド実行時のメモリ所要量の事前見積もり」を参照してください。

- adbmergechunk コマンドを実行する場合

  「4.3.6　adbmergechunk コマンド実行時のメモリ所要量の事前見積もり」を参照してください。

- adbarchivechunk コマンドを実行する場合

  「4.3.7　adbarchivechunk コマンド実行時のメモリ所要量の事前見積もり」を参照してください。

- adbunarchivechunk コマンドを実行する場合

  「4.3.8　adbunarchivechunk コマンド実行時のメモリ所要量の事前見積もり」を参照してください。

なお，詳細なメモリ所要量を見積もる場合は，データベースの設計および DB エリアの容量見積もりのあとで，「6.3　HADB サーバのメモリ所要量の見積もり」を参照してください。

4.3.1　通常運用時のメモリ所要量の事前見積もり

通常運用時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-3　メモリ所要量の求め方（通常運用時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024＋（*db_idx* ×1,024）	adbbuff オペランド
3		プロセス共通メモリ	1,024＋*max_users*＋*GBUF*	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	*rthd_num* ×1,074	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024	なし

変数の説明

dbarea_num：データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF：グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

rthd_num：全リアルスレッド数

見積もれない場合，HADB サーバをインストールしたマシンの CPU のコア数を仮定してください。

db_idx：インデクスを格納するデータ用 DB エリアの容量（単位：ギガバイト）

「4.2.1　DB ディレクトリの事前見積もり」の「表 4-1　計算式（DB ディレクトリの容量）」にある次の項目で求めた値を代入してください。

- 『B-tree インデクスを格納するデータ用 DB エリアファイル』
- 『テキストインデクスを格納するデータ用 DB エリアファイル』
- 『レンジインデクスを格納するデータ用 DB エリアファイル』

4.3.2　adbimport コマンド実行時のメモリ所要量の事前見積もり

adbimport コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-4　メモリ所要量の求め方（adbimport コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024	adbbuff オペランド
3		プロセス共通メモリ	1,024＋*max_users*＋*GBUF*	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	512＋*SORTBUF*＋*BLKBUF*	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024＋*sort_rthd* ×0.5	なし

変数の説明

dbarea_num

データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF

グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

SORTBUF

ソートバッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
SORTBUF＝ sort_buff_size × sort_rthd
```

sort_buff_size

インポートオプションadb_import_sort_buff_size の指定値

sort_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

BLKBUF

インデクス作成用バッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
BLKBUF＝
    ↑MAX(((buff_blk_size×buff_blk_num)×(import_rthd×2)÷1,024) ,
        (((buff_blk_size×buff_blk_num)÷1,024＋200＋txt_sort_buff_size)
        ×(import_rthd×2)))↑
```

buff_blk_size

4,096 を仮定してください。

buff_blk_num

インポートオプションadb_import_buff_blk_num の指定値

import_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

txt_sort_buff_size

インポートオプションadb_import_txt_buff_size の指定値

4.3.3 adbidxrebuild コマンド実行時のメモリ所要量の事前見積もり

adbidxrebuild コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-5 メモリ所要量の求め方（adbidxrebuild コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024	adbbuff オペランド

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
3		プロセス共通メモリ	1,024＋*max_users*＋*GBUF*	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	512＋*SORTBUF*＋*BLKBUF*	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024＋*sort_rthd*×0.5	なし

変数の説明

dbarea_num

データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF

グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

SORTBUF

ソートバッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
SORTBUF＝ sort_buff_size × sort_rthd
```

sort_buff_size

インデクス再作成オプションadb_idxrebuild_sort_buff_size の指定値

sort_rthd

次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

BLKBUF

インデクス作成用バッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
BLKBUF＝
    ↑MAX(((buff_blk_size×buff_blk_num)×(scan_rthd＋dividx_rthd)÷1,024) ,
         (((buff_blk_size×buff_blk_num)÷1,024＋200＋txt_sort_buff_size)
         ×(scan_rthd＋dividx_rthd)))↑
```

buff_blk_size

4,096 を仮定してください。

buff_blk_num

インデクス再作成オプションadb_idxrebuild_buff_blk_num の指定値

scan_rthd

次に示す計算式から求めてください。

↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値*－1）÷2↓

dividx_rthd

次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

txt_sort_buff_size

インデクス再作成オプションadb_idxrebuild_txt_buff_size の指定値

4.3.4 adbgetcst コマンド実行時のメモリ所要量の事前見積もり

adbgetcst コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-6 メモリ所要量の求め方（adbgetcst コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num*×1,024	adbbuff オペランド
3		プロセス共通メモリ	1,024＋*max_users*＋*GBUF*	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	*rthd_num*×512	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024	なし

変数の説明

dbarea_num：データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF：グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

rthd_num

コスト情報収集オプションadb_getcst_rthd_num の指定値

4.3.5 adbexport コマンド実行時のメモリ所要量の事前見積もり

adbexport コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-7 メモリ所要量の求め方（adbexport コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024＋（*db_idx* ×1,024）	adbbuff オペランド
3		プロセス共通メモリ	1,024＋ *max_users* ＋ *GBUF* ＋（*scan_buff_size* ×*rthd_num* ×4）	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	*rthd_num* ×512	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024	なし

変数の説明

dbarea_num：データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF：グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

scan_buff_size

エクスポートオプションadb_export_scan_buff_size の指定値

rthd_num

エクスポートオプションadb_export_rthd_num の指定値

db_idx：インデクスを格納するデータ用 DB エリアの容量（単位：ギガバイト）

「4.2.1　DB ディレクトリの事前見積もり」の「表 4-1　計算式（DB ディレクトリの容量）」にある次の項目で求めた値を代入してください。

- 『B-tree インデクスを格納するデータ用 DB エリアファイル』
- 『テキストインデクスを格納するデータ用 DB エリアファイル』
- 『レンジインデクスを格納するデータ用 DB エリアファイル』

4.3.6　adbmergechunk コマンド実行時のメモリ所要量の事前見積もり

adbmergechunk コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-8　メモリ所要量の求め方（adbmergechunk コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024	adbbuff オペランド
3		プロセス共通メモリ	1,024＋*max_users*＋*GBUF*	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	512＋*SORTBUF*＋*BLKBUF*	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024＋*sort_rthd* ×0.5	なし

変数の説明

dbarea_num

データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF

グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

SORTBUF

ソートバッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
SORTBUF＝ sort_buff_size × sort_rthd
```

sort_buff_size

マージチャンクオプションadb_mergechunk_sort_buff_size の指定値

sort_rthd

次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

BLKBUF

インデクス作成用バッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
BLKBUF＝
  ↑MAX(((buff_blk_size×buff_blk_num)×(scan_rthd＋dividx_rthd)÷1,024) ,
       (((buff_blk_size×buff_blk_num)÷1,024＋200＋txt_sort_buff_size)
       ×(scan_rthd＋dividx_rthd)))↑
```

buff_blk_size

4,096 を仮定してください。

buff_blk_num

マージチャンクオプションadb_mergechunk_buff_blk_num の指定値

scan_rthd

次に示す計算式から求めてください。

↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

dividx_rthd

次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

txt_sort_buff_size

マージチャンクオプションadb_mergechunk_txt_buff_size の指定値

4.3.7　adbarchivechunk コマンド実行時のメモリ所要量の事前見積もり

adbarchivechunk コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-9　メモリ所要量の求め方（adbarchivechunk コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024＋（*db_idx* ×1,024）	adbbuff オペランド
3		プロセス共通メモリ	1,024＋ *max_users* ＋ *GBUF* ＋（*scan_buff_size* ×*rthd_num* ×4）	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	*rthd_num* ×512	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024	なし

変数の説明

dbarea_num：データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

db_idx：インデクスを格納するデータ用 DB エリアの容量（単位：ギガバイト）

「4.2.1　DB ディレクトリの事前見積もり」の「表 4-1　計算式（DB ディレクトリの容量）」にある次の項目で求めた値を代入してください。

- 『B-tree インデクスを格納するデータ用 DB エリアファイル』
- 『テキストインデクスを格納するデータ用 DB エリアファイル』
- 『レンジインデクスを格納するデータ用 DB エリアファイル』

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF：グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

scan_buff_size

アーカイブチャンクオプションadb_arcv_scan_buff_size の指定値

rthd_num

アーカイブチャンクオプションadb_arcv_rthd_num の指定値

4.3.8　adbunarchivechunk コマンド実行時のメモリ所要量の事前見積もり

adbunarchivechunk コマンド実行時に使用するメモリ所要量の計算式を，次の表に示します。

表 4-10　メモリ所要量の求め方（adbunarchivechunk コマンド実行時）

項番	メモリ種別		計算式（単位：メガバイト）	関連するサーバ定義
1	共有メモリ	共有メモリ管理	100	なし
2		グローバルバッファページ	*dbarea_num* ×1,024	adbbuff オペランド
3		プロセス共通メモリ	1,024＋ *max_users* ＋ *GBUF*	• adb_sys_proc_area_max オペランド • adbbuff オペランド
4		リアルスレッド固有メモリ	512＋ *SORTBUF* ＋ *BLKBUF*	adb_sys_rthd_area_max オペランド
5	プロセスメモリ	ヒープメモリ	1,024＋ *sort_rthd* ×0.5	なし

変数の説明

dbarea_num

データ用 DB エリアの数

見積もれない場合，定義する表とインデクスの合計値を仮定してください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

GBUF

グローバルバッファ

項番 2「グローバルバッファページ」の計算結果を基に 400 メガバイトにつき，20 メガバイトを加算してください。

SORTBUF

ソートバッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
SORTBUF＝ sort_buff_size × sort_rthd
```

sort_buff_size

アンアーカイブチャンクオプションadb_unarcv_sort_buff_size の指定値

sort_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値 －1

BLKBUF

インデクス作成用バッファ

次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
BLKBUF＝
  ↑MAX(((buff_blk_size×buff_blk_num)×(unarcv_rthd×2)÷1,024) ,
       (((buff_blk_size×buff_blk_num)÷1,024＋200＋txt_sort_buff_size)
       ×(unarcv_rthd×2)))↑
```

buff_blk_size

4,096 を仮定してください。

buff_blk_num

アンアーカイブチャンクオプションadb_unarcv_buff_blk_num の指定値

unarcv_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

txt_sort_buff_size

アンアーカイブチャンクオプションadb_unarcv_txt_buff_size の指定値

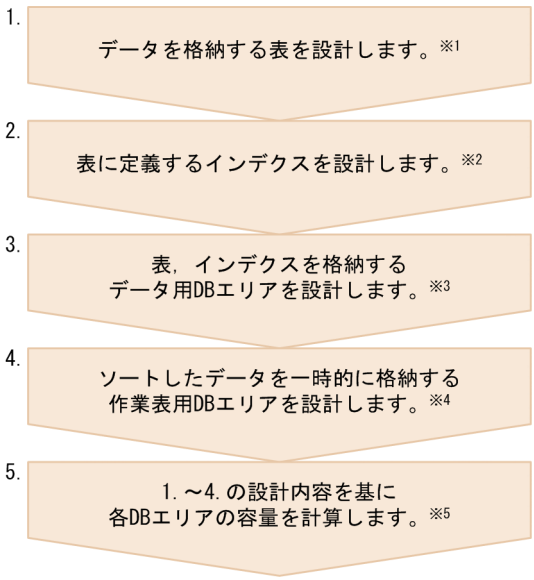
5 データベースの設計

この章では，表の設計，インデクスの設計，各 DB エリアの設計，および各 DB エリアの容量見積もりについて説明します。

5.1　データベースの設計手順

データベースの設計手順を，次の図に示します。

図 5-1　データベースの設計手順



注※1

手順の詳細については，「5.2　表の設計」を参照してください。

注※2

手順の詳細については，次の個所を参照してください。

- 「5.3　B-tree インデクスの設計」
- 「5.4　テキストインデクスの設計」
- 「5.5　レンジインデクスの設計」

注※3

手順の詳細については，「5.6　データ用 DB エリアの設計」を参照してください。

注※4

手順の詳細については，「5.7　作業表用 DB エリアの設計」を参照してください。

注※5

手順の詳細については，次の個所を参照してください。

- 「5.8　データ用 DB エリアの容量見積もり」
- 「5.9　作業表用 DB エリアの容量見積もり」
- 「5.10　マスタディレクトリ用 DB エリアの容量見積もり」
- 「5.11　ディクショナリ用 DB エリアの容量見積もり」
- 「5.12　システム表用 DB エリアの容量見積もり」

5.2　表の設計

ここでは，表を設計する際に検討する項目について説明します。

5.2.1　表を設計する際の流れ

表を設計する際の流れを次に示します。

1. 表をローストア表として定義するか，またはカラムストア表として定義するかを決める
   「5.2.2　ローストア表とカラムストア表の選択基準」を参照して，表をローストア表として定義するか，またはカラムストア表として定義するかを決めます。
2. 表をシングルチャンク表として定義するか，またはマルチチャンク表として定義するかを決める
   「5.2.3　シングルチャンク表とマルチチャンク表の選択基準」を参照して，表をシングルチャンク表として定義するか，またはマルチチャンク表として定義するかを決めます。
3. 表をローストア表として定義する場合は，表の正規化を検討する
   「5.2.6　表の正規化【ローストア表】」を参照して，表の正規化を検討します。
4. 表定義の際に指定する各項目を検討する（CREATE TABLE 文に指定するオプションを検討する）
   次の個所を参照して，表定義の際に指定する各項目を検討してください。
   - 「5.2.7　行長の固定（FIX 指定）【ローストア表】」
   - 「5.2.8　列の既定値の設定（DEFAULT 句）」
   - 「5.2.9　主キー（一意性制約定義）の指定（PRIMARY KEY）【シングルチャンク表】」
   - 「5.2.10　外部キーの指定（FOREIGN KEY）」
   - 「5.2.11　データページ内の未使用領域の確保（PCTFREE）【ローストア表】」
   - 「5.2.12　可変長データ型の列データの分岐指定（BRANCH）【ローストア表】」

メモ

- 【ローストア表】と記載されている項目は，表をローストア表として定義する場合に検討する項目です。表をカラムストア表として定義する場合は，検討不要な項目です。
- 【シングルチャンク表】と記載されている項目は，表をシングルチャンク表として定義する場合に検討する項目です。表をマルチチャンク表として定義する場合は，検討不要な項目です。

5.2.2　ローストア表とカラムストア表の選択基準

ここでは，表をローストア表として定義するか，カラムストア表として定義するかの選択基準について説明します。

(1) カラムストア表の制限事項

カラムストア表には，次に示す運用上の制限事項があります。これらの制限事項によって運用上の問題が発生する場合は，表をローストア表として定義してください。

- カラムストア表には，テキストインデクスを定義できません。
- カラムストア表は，アーカイブマルチチャンク表として定義できません。
- カラムストア表は，FIX 表にはできません。

(2) 表をローストア表として定義するか，またはカラムストア表として定義するかの選択基準

表をローストア表として定義するか，またはカラムストア表として定義するかは，次の表に示す選択基準を基に決定してください。

表 5-1　ローストア表とカラムストア表の選択基準

定義する表に対してどのような検索をするか，または定義する表の特徴		適している表の種類
定義する表に対して行われる検索	次のような検索が実行される表の場合 • 行単位でデータにアクセスするような検索をすることが多い場合 SELECT 文の選択式に*を指定したり，選択式にほぼすべての列を指定したりするような場合が該当します。 • 検索対象のデータを，B-tree インデクスを使用して絞り込むような検索をすることが多い場合	ローストア表
	次のような検索が実行される表の場合 • 検索対象のデータをあまり絞り込まないで，特定の列データ全体を検索することが多い場合 • 特定の範囲内（特定の年や月など）の，特定の列データにアクセスすることが多い場合 • 特定の列データに対する値の集計（平均や合計などを求める）をすることが多い場合 • 特定の列データのグループ化をすることが多い場合	カラムストア表
定義する表に対して行われる操作	• データの追加，更新，削除を行う表の場合 INSERT 文，UPDATE 文，およびDELETE 文に対する処理性能はローストア表の方が優れています。また，カラムストア表に対して，INSERT 文，UPDATE 文，およびDELETE 文を実行した場合，実行後の検索性能が低下※することがあります。そのため，定型的な業務で SQL 文によるデータの追加，更新，削除を行う場合は，ローストア表として定義してください。	ローストア表
定義する表のデータ容量	• データ容量が比較的小さい表の場合 表のデータ容量，または 1 回のデータインポートで格納するデータ容量が，比較的小さい表が該当します。目安としては，データインポートの際の入力データファイルのサイズが 1 ギガバイト未満の表が該当します。	ローストア表

定義する表に対してどのような検索をするか，または定義する表の特徴		適している表の種類
	なお，ローストア表のデータ容量は，データインポートの際の入力データファイルのサイズの約 2 倍になります。ただし，この数値は，インポートするデータの内容によって変動があります。 通常，ローストア表のデータ容量は，カラムストア表のデータ容量より大きくなります。	
	• データ容量が大きい表の場合 表のデータ容量，または 1 回のデータインポートで格納するデータ容量が大きい表が該当します。目安としては，入力データファイルのサイズが 1 ギガバイト以上の表が該当します。 なお，カラムストア表のデータ容量は，データインポートの際の入力データファイルのサイズの約 1/2 になります。ただし，この数値は，インポートするデータの内容によって変動があります。 通常，カラムストア表のデータ容量は，ローストア表のデータ容量より小さくなります。	カラムストア表
データウェアハウスでの使い方	• 解析対象のデータを格納する表と主キーで結合するような表の場合 データウェアハウスでのディメンジョン表やマスタ表などが該当します。	ローストア表
	• 解析対象のデータを格納する表の場合 データウェアハウスでのファクト表や，ログデータまたはセンサデータを格納する表などが該当します。 特に，解析対象のデータ（レンジインデクスによる絞り込みなどで検索対象となるデータ）が，数ギガバイト以上の場合は，カラムストア表として表を定義することを推奨します。 解析対象のデータのサイズは，データインポートの際の入力データファイルを基に見積もってください。 なお，解析対象のデータが数百メガバイト以下の場合は，解析対象のデータを格納する表であっても，ローストア表として表を定義することを推奨します。	カラムストア表
表に定義する列のデータ型	• 表に配列型の列を定義する場合 カラムストア表として定義してください。	カラムストア表

注※

更新行のカラム化機能を有効にすると，カラムストア表の検索性能の低下を防ぐことができます。更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

重要

カラムストア表に適した表は，マルチチャンク表に適するケースが多いため，カラムストア表かつマルチチャンク表として表を定義することを推奨します。表をマルチチャンク表として定義する際の考慮点については，「5.2.4　マルチチャンク表を定義する場合の考慮点」を参照してください。

表をローストア表として定義する場合は，CREATE TABLE 文のSTORAGE FORMAT オプションにROW を指定するか，またはSTORAGE FORMAT オプションの指定を省略してください。表をカラムストア表として定義する場合は，CREATE TABLE 文のSTORAGE FORMAT オプションにCOLUMN を指定してください。

CREATE TABLE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』を参照してください。

(3)　カラムストア表の検索時に使用される B-tree インデクス

カラムストア表に定義した B-tree インデクスが，カラムストア表の検索時に使用されるケースは限定されています。カラムストア表で，B-tree インデクスを使用する代表的な例を次に示します。次のような絞り込みが有効な場合にだけ，カラムストア表に B-tree インデクスを定義することを推奨します。

- SQL 文の実行時に，インデクス指定で使用するインデクスを指定する場合
- 集合関数MIN またはMAX を指定して，最小値や最大値を求める場合
- 探索条件を指定して，UPDATE 文またはDELETE 文を実行する場合

特定の年や月のデータにアクセスするような場合は，レンジインデクスを定義して検索範囲を絞り込むことを推奨します。

カラムストア表の検索時に使用される B-tree インデクスの規則については，マニュアル『HADB　AP 開発ガイド』の『SQL 文の実行時に使用される B-tree インデクスおよびテキストインデクス』を参照してください。

> **メモ**
>
> B-tree インデクスを定義したカラムストア表には，更新行のカラム化機能は適用されません。更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

(4)　カラムストア表の列データの圧縮方式

カラムストア表にデータをインポートする際，各列のデータは，次の表に示すどれかの方式で圧縮されます。そのため，ローストア表に比べてデータ容量が小さくなります。データインポートの実行時に，インポートするデータを基に，HADB サーバが次の表に示す圧縮方式のどれかを自動的に選択します。

表 5-2　カラムストア表の列データの圧縮方式の種類

項番	圧縮方式	説明
1	無圧縮（NONE）	データは圧縮されないで格納されます。
2	ランレングス圧縮（RUNLENGTH）	同じ値のデータが連続で出現した場合に，連続で出現したデータと繰り返し回数の組でデータを保存します。同じ値のデータが繰り返し連続して出現するような場合に有効な圧縮方式です。 圧縮の例：AAAABB → 4A2B

項番	圧縮方式	説明
3	差分圧縮（DELTA）	先頭のデータの値と隣り合うデータの差分値とで，データを保存します。ある特定の値の周辺にデータが集中している場合に有効な圧縮方式です。 圧縮の例：100,99,101,103 → 100,-1,2,2
4	差分ランレングス圧縮（DELTA_RUNLENGTH）	対象のデータを差分圧縮したあとに，差分値のデータをランレングス圧縮します。一定の値でインクリメントするデータ（等差数列をなすデータ）の場合に有効な圧縮方式です。伝票番号などのように，番号が 1 つずつ加算されていくようなデータを圧縮する際に有効です。
5	辞書圧縮（DICTIONARY）	よく出現するデータを辞書に格納しておき，データ部には辞書へのインデクス（添え字）だけを格納します。カーディナリティ（データの種類）が少ない文字列データなどで有効な圧縮方式です。 なお，辞書のサイズは最大 16 キロバイトです。

圧縮の方法には，上記の表に示すようにデータの重複を排除する方式と，データ間の差分を取る方式があります。そのため，128 バイト以上の文字列データや，128 バイト以上のバイナリデータなどのように，データの重複があまりない場合や，データ間の差分が少ない場合は，圧縮効果が少ないことがあります。

メモ

- データインポートの終了時に，HADB サーバが選択した圧縮方式がメッセージに出力されます。
- データインポート時に圧縮方式が決定されます。そのため，同じ表へのデータインポートの場合に，インポートするデータのパターンが異なると，同じ列であっても圧縮方式が異なることがあります。
- HADB サーバによる圧縮方式の選択は，データインポート時にデータを格納するスレッドごとに行われます。
- HADB サーバは，格納する列データの先頭 16 メガバイト分の列データに対して圧縮率を計算し，最も圧縮率がよい圧縮方式を選択します。圧縮率が悪く，圧縮後のデータサイズが，無圧縮のデータサイズ以上になった場合は，無圧縮が選択されます。

なお，次に示す処理の場合に，カラムストア表の列ごとに圧縮方式を指定することもできます。この場合，常に指定した圧縮方式で列データが圧縮されます。

- CREATE TABLE 文でカラムストア表を定義する
- ALTER TABLE 文でカラムストア表に列を追加する

■UPDATE 文または DELETE 文を実行するカラムストア表と圧縮方式の関係

更新を前提としたカラムストア表の場合，差分圧縮以外の圧縮方式を指定することを推奨※1 します。HADB による圧縮方式の自動選択※2 をしないようにしてください。

［理由］

カラムストア形式のデータに対して，UPDATE 文またはDELETE 文を実行した場合，HADB は対象の行データを全列分伸長します。差分圧縮（DELTA）の伸長の際，先頭のデータから伸長対象のデータまでのすべての差分値を計算するため，ほかの圧縮方式よりも列の伸長処理に時間が掛かります。そのため，カラムストア表に差分圧縮（DELTA）が適用された列があると，UPDATE 文またはDELETE 文の実行時間が長くなります。

注※1

CREATE TABLE 文の圧縮方式指定（COMPRESSION TYPE）で，差分圧縮（DELTA）および圧縮方式の自動選択（AUTO）以外を指定することを推奨します。

注※2

圧縮方式指定を省略した場合，または圧縮方式指定にAUTO を指定した場合，HADB が圧縮方式を自動選択します。その結果，圧縮方式に差分圧縮（DELTA）が適用される可能性があります。

(5) ローストア表とカラムストア表の使い分けの例

ローストア表とカラムストア表の使い分けの例を，次に示す運用例を基に説明します。

運用例

A 社では，各営業所からの売上データを集めた売上データ分析用のデータウェアハウスを構築しようとしている。売上データの分析軸として，日付，営業所，商品，顧客を考えている。このデータウェアハウスを構築するために，スキーマモデルとして，次の図に示すスタースキーマで構築することに決定した。

図 5-2　A 社の売上データ分析用のデータウェアハウスのスキーマモデル

上記のスタースキーマの場合，SALES 表がファクト表になります。また，PRODUCTS 表，CUSTOMERS 表，OFFICES 表，およびCALENDAR 表が，ディメンジョン表になります。各表には，次のデータが格納されています。

- SALES 表

  SALES 表には，売れた商品の商品 ID，商品個数，単価，顧客 ID，売上を計上した営業所の ID，売上発生日などの売上データが，過去 10 年分格納されています。1 日当たりおよそ 10 万件のデータが格納されるため，合計で 3 億 6 千万件くらいの売上データが格納されています。

  売上データ中の商品 ID，顧客 ID，営業所 ID，売上発生日には，それぞれPRODUCTS 表，CUSTOMERS 表，OFFICES 表，CALENDAR 表の主キーに当たるデータ（外部キー）が格納されています。

- PRODUCTS 表

  PRODUCTS 表には，A 社が扱っている商品のデータ（商品 ID，名称，カテゴリなど）が格納されています。

- CUSTOMERS 表

  CUSTOMERS 表には，A 社の顧客のデータ（顧客 ID，顧客名，顧客の業種，顧客の住所など）が格納されています。

- OFFICES 表

  OFFICES 表には，A 社の営業所のデータ（営業所 ID，営業所名，営業所の所在地など）が格納されています。

- CALENDAR 表

  CALENDAR 表には，日時に関するディメンジョンデータ（日付，月，四半期，年）が格納されています。

上記のようなスキーマモデルで，月次の売上データの集計，顧客ごとの売上分析，および営業ごとの売上分析などを行った場合，次のような検索が頻繁に実行されると予想されます。

- SALES 表に対する全件検索
- SALES 表中の，ある一定期間のデータの検索（今月のデータや前月のデータの検索など）

そのため，SALES 表は，カラムストア表として定義します。また，売上発生日に対応する列にレンジインデクスを定義します。

PRODUCTS 表，CUSTOMERS 表，OFFICES 表，およびCALENDAR 表は，SALES 表のデータにある外部キーによるジョインで使用されるケースが多くなると予想されます。これらの表は，データ件数が比較的多くないため，ローストア表として定義します。また，主キーに対応する列に B-tree インデクスを定義します。

5.2.3 シングルチャンク表とマルチチャンク表の選択基準

最初に，表をローストア表として定義するか，またはカラムストア表として定義するかを決めます。次に，その表をシングルチャンク表とするか，またはマルチチャンク表（レギュラーマルチチャンク表またはアーカイブマルチチャンク表）とするかを決めます。シングルチャンク表とマルチチャンク表の主な特徴を次に示します。

表 5-3 シングルチャンク表とマルチチャンク表（レギュラーマルチチャンク表またはアーカイブマルチチャンク表）の主な特徴

項目	表の種類		
	シングルチャンク表	マルチチャンク表	
		レギュラーマルチチャンク表	アーカイブマルチチャンク表
バックグラウンドインポート機能を使用できるかどうか	×	○	○
表に主キーを定義できるかどうか	○	×	×
表にユニークインデクスを定義できるかどうか	○	×	×

項目	表の種類		
	シングルチャンク表	マルチチャンク表	
		レギュラーマルチチャンク表	アーカイブマルチチャンク表
ローストア表のデータを圧縮できるかどうか	×	×	○
表のデータを更新または削除できるかどうか（UPDATE 文またはDELETE 文を実行できるかどうか）	○	○	△※1
検索に掛かる時間	○	○	△※2

（凡例）

○：実行できる，または性能への影響なし

△：更新または削除できないデータがある，または○に比べて性能が劣る

×：実行できない

注※1

アーカイブ状態のチャンク内のデータを更新または削除できません。

注※2

アーカイブ状態のチャンク内のデータを検索する場合が該当します。

なお，上記の表には，シングルチャンク表とマルチチャンク表の主な特徴だけを記載しています。シングルチャンク表とマルチチャンク表のメリット，デメリット，および制限事項などについては，次の個所を参照してください。次の個所を読んで表をシングルチャンク表とするか，またはマルチチャンク表（レギュラーマルチチャンク表またはアーカイブマルチチャンク表）とするかを決めてください。

- 「5.2.4　マルチチャンク表を定義する場合の考慮点」
- 「5.2.5　アーカイブマルチチャンク表を定義する場合の考慮点【ローストア表】」

重要

表をカラムストア表として定義する場合は，その表をアーカイブマルチチャンク表とすることはできません。

5.2.4　マルチチャンク表を定義する場合の考慮点

ここでは，実表をマルチチャンク表として定義する場合の考慮点について説明します。

ここをお読みいただいたあとに，「5.6.2　データ用 DB エリアにマルチチャンク表を格納する場合の考慮点」の説明もお読みください。

また，実表をアーカイブマルチチャンク表として定義する場合は，ここをお読みいただいたあとに，「5.2.5 アーカイブマルチチャンク表を定義する場合の考慮点【ローストア表】」の説明もお読みください。

(1) 実表をマルチチャンク表として定義する場合のメリット

実表をマルチチャンク表として定義した場合，次に示す運用ができます。

- データの検索とデータのインポートを同時に実行できます（バックグラウンドインポート機能を使用できます）。
- PURGE CHUNK 文で，チャンク単位にデータを一括削除できます。
- データを検索できない状態で事前にインポートだけしておき，任意のタイミングでインポートしたデータを検索することができます。運用例については，「11.4.24 バックグラウンドインポートとチャンクを意識した運用例 2（待機状態のチャンクを利用した運用）」を参照してください。

> **メモ**
>
> 4 メガバイト以下の少量のデータだけを頻繁に格納する場合は，バックグラウンドインポート機能ではなく，操作系 SQL のINSERT 文を使用することを検討してください。バックグラウンドインポートで少量のデータを頻繁に格納すると，むだな領域が発生してしまい，データの格納効率が悪くなります。バックグラウンドインポートを実行すると，すでに割り当てられているセグメントに空きがあっても，新規にセグメントが確保されて割り当てられるためです。

実表をカラムストア表として定義し，そのカラムストア表に対してINSERT 文，UPDATE 文，およびDELETE 文を実行する可能性が高い場合は，カラムストア表かつマルチチャンク表として実表を定義してください。カラムストア表に対してINSERT 文，UPDATE 文，およびDELETE 文を繰り返し実行すると，表の再編成が必要になることがあります。その際，マルチチャンク表であれば，チャンク単位で再編成が実行できるため，次のメリットがあります。

- 再編成に掛かる時間を短縮できる
- 再編成時に必要となるディスク容量を抑えることができる

また，実表をアーカイブマルチチャンク表として定義すると，チャンクアーカイブ機能も使用することができます。実表をアーカイブマルチチャンク表として定義する場合は，「5.2.5 アーカイブマルチチャンク表を定義する場合の考慮点【ローストア表】」の説明もお読みください。

(2) 実表をマルチチャンク表として定義する場合のデメリット

- マルチチャンク表には，主キーおよびユニークインデクスを定義できません。
- チャンクの管理，運用が必要になります。例えば，チャンクのマージを定期的にする必要があります。詳細については，「11.4.9 チャンクをマージする方法（チャンク数を減らす方法）」を参照してください。

(3) CREATE TABLE 文で必要な指定（チャンク数の最大値）

実表をマルチチャンク表として定義する場合は，CREATE TABLE 文のチャンク指定でチャンク数の最大値を指定してください。

(4) チャンク数の最大値の求め方

adbimport コマンドでバックグラウンドインポートを実行するたびに，チャンクが作成されます。また，adbmergechunk コマンドで，作成された複数のチャンクをマージできます。ただし，CREATE TABLE 文の*チャンク指定*で指定した*チャンク数の最大値*以上のチャンクは作成できません。

そのため，次の 3 点を踏まえて，*チャンク数の最大値*を見積もってください。

- adbimport コマンドでバックグラウンドインポートを実行する頻度
- adbmergechunk コマンドでチャンクをマージする頻度
- チャンクを保持する期間

なお，作成したチャンクはPURGE CHUNK 文で削除できます。削除したチャンクの数だけ，チャンクを作成できます。また，CREATE TABLE 文で指定した*チャンク数の最大値*は，ALTER TABLE 文で変更できます。

変数 *CHUNK_NUM* の値を，次の計算式で求めてください。計算式で求めた値以上の値を，CREATE TABLE 文の*チャンク数の最大値*に指定します。

計算式（単位：個数）

$$
\begin{aligned}
&CHUNK_NUM = \\
&\quad \uparrow years \times \uparrow (import_num \div merge_source_chunk_num) \uparrow \times safe_rate \uparrow \\
&\quad + (\uparrow merge_source_chunk_num \times safe_rate \uparrow)\text{※}
\end{aligned}
$$

注※：adbmergechunk コマンド実行時にマージ元チャンクが削除仕掛中のチャンクとして残るおそれがあるため，それを考慮して多めに見積もります。

> **重要**
>
> adbmergechunk コマンドの実行時に，マージ元チャンクが削除されなかった場合，マージ元チャンクは**削除仕掛中のチャンク**として残ります。その結果，削除仕掛中のチャンクの数だけ，使用できるチャンク数が減ります。そのため，変数 *CHUNK_NUM* の値を求める際は，削除仕掛中のチャンクの数も考慮する必要があります。削除仕掛中のチャンクについては，「11.4.9 チャンクをマージする方法（チャンク数を減らす方法）」の「(3) adbmergechunk コマンドによるチャンクのマージ」を参照してください。

変数の説明

years

チャンク（実表に格納したデータ）を保持する年数です。

adbimport コマンドでバックグラウンドインポートを実行して作成されたチャンクを，PURGE CHUNK 文で削除するまでの年数を代入してください。

import_num

1 年間にadbimport コマンドでバックグラウンドインポートを実行する回数を代入してください。

merge_source_chunk_num

adbmergechunk コマンドを実行するときに存在するマージ元チャンクの数です。

例えば，1 日 1 回adbimport コマンドでバックグラウンドインポートを実行して，5 日に 1 回 adbmergechunk コマンドでチャンクをマージする場合は，5 を代入してください。

safe_rate

安全率です。1.2 を代入してください。

変数 *CHUNK_NUM* の求め方の例を次に示します。

■変数 *CHUNK_NUM* の求め方の例

次の条件で，変数 *CHUNK_NUM* を計算します。

- チャンク（実表に格納したデータ）を 2 年間保持する
- 週に 1 回，adbimport コマンドでバックグラウンドインポートを実行する
- 月に 1 回，adbmergechunk コマンドで 4 つのチャンクをマージする
- 安全率を 1.2 とする

計算式（単位：個数）

$$CHUNK_NUM = \uparrow 2 \times \uparrow\left(\uparrow \frac{365}{7} \uparrow \div 4\right)\uparrow \times 1.2 \uparrow + \uparrow 4 \times 1.2 \uparrow = 39$$

この例の場合，変数 *CHUNK_NUM* の値は 39 となります。そのため，*チャンク数の最大値*には39 以上の値を指定してください。

5.2.5　アーカイブマルチチャンク表を定義する場合の考慮点【ロードストア表】

ここでは，実表をアーカイブマルチチャンク表として定義する場合の考慮点について説明します。

なお，ここの説明は，「5.2.4　マルチチャンク表を定義する場合の考慮点」をお読みいただいていることを前提としています。

(1)　実表をアーカイブマルチチャンク表として定義する場合のメリット

チャンク内のデータを圧縮することで，データベースの容量を削減できます。

(2) 実表をアーカイブマルチチャンク表として定義する場合のデメリット

- アーカイブされているデータは，UPDATE 文による更新またはDELETE 文による削除ができません。データを更新または削除する場合は，チャンク内のデータをアンアーカイブする必要があります。UPDATE 文またはDELETE 文がエラーになる条件については，マニュアル『HADB　SQL リファレンス』の次の個所を参照してください。
  - 『UPDATE 文の指定形式および規則』の『規則』
  - 『DELETE 文の指定形式および規則』の『規則』
- アーカイブされているデータを検索する場合，次に示す原因によって，アーカイブされていないデータを検索するときに比べて検索時間が長くなります。
  - データの伸長に時間が掛かる
  - インデクスを使用した検索ができない
- アーカイブマルチチャンク表を検索する際は，アーカイブレンジ列の情報をSELECT 文の探索条件に指定して，検索範囲を絞り込む必要があります。そのため，SELECT 文の探索条件にアーカイブレンジ列を指定した条件を指定できない場合，検索範囲を絞り込むことができないため，検索時間が長くなります。

  アーカイブマルチチャンク表を検索する際の留意事項については，マニュアル『HADB　AP 開発ガイド』の『アーカイブマルチチャンク表を検索する際の考慮点』を参照してください。

メモ

- アーカイブマルチチャンク表をシングルチャンク表に変更する場合，CREATE TABLE 文による表の再定義が必要になります。そのため，メリットおよびデメリットをよく検討してから，アーカイブマルチチャンク表を定義するかどうかを決定してください。
- アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合は，CREATE TABLE 文による表の再定義は必要ありません。ALTER TABLE 文で，変更することができます。

(3) アーカイブマルチチャンク表に格納するデータの条件

アーカイブマルチチャンク表には，日々発生する時系列データを格納してください。ここでいう時系列データとは，センサから発生したデータや，店舗の売り上げデータなど，データの発生日時順に並んでいるデータを意味しています。また，時系列データには，データの生成日時や，売り上げ発生日時のようなデータが含まれている必要があります。

(4) アーカイブディレクトリの作成

CREATE TABLE 文を実行する前にアーカイブディレクトリを作成してください。また，アーカイブディレクトリに対する各種権限を HADB 管理者に与える必要があります。詳細については，マニュアル『HADB SQL リファレンス』の『CREATE TABLE 文の指定形式および規則』の『指定形式の説明』の『チャンク指定』の『チャンクアーカイブ指定』を参照してください。

1 つのアーカイブマルチチャンク表に対して，専用のアーカイブディレクトリを作成することを推奨します。複数のアーカイブマルチチャンク表に対して，1 つのアーカイブディレクトリを共有することもできます。ただし，アーカイブディレクトリに対するアクセスが集中した場合，性能が低下するおそれがあります。

性能面を優先して考えた場合，次のようにアーカイブディレクトリを作成することを推奨します。

- アーカイブマルチチャンク表ごとにアーカイブディレクトリを作成する
- 専用のファイルシステムに 1 つのアーカイブディレクトリを作成する
- 専用のファイルシステムには，アーカイブディレクトリだけを作成する

(5) アーカイブレンジ列の指定

アーカイブマルチチャンク表を定義する場合，アーカイブレンジ列の指定は必須となります。次に示すデータ型の列は，アーカイブレンジ列に指定することはできません。

- 定義長が 33 バイト以上のCHARACTER 型
- VARCHAR 型
- BINARY 型
- VARBINARY 型

アーカイブレンジ列に格納されたデータは，アーカイブマルチチャンク表を検索する際，検索対象を絞り込むために使用されます。そのため，データの生成日時や，売り上げ発生日時などのデータを格納する列を，アーカイブレンジ列としてください。

なお，アーカイブレンジ列には，非ナル値制約が設定されます。

(6) アーカイブレンジ列に自動的に定義されるレンジインデクス

アーカイブレンジ列には自動的にレンジインデクスが定義されます。このレンジインデクスは，データ量が少ないため，専用のデータ用 DB エリアに格納する必要はありません。アーカイブマルチチャンク表に定義するほかのインデクスと同じデータ用 DB エリアに格納することを検討してください。

CREATE TABLE 文のチャンク指定でレンジインデクスを格納するデータ用 DB エリアを指定します。

なお，アーカイブレンジ列に自動的に定義されるレンジインデクスに，インデクス識別子を指定することができます。レンジインデクスのインデクス識別子を指定しない場合は，HADB サーバが自動的にインデクス識別子を決定します。

5.2.6 表の正規化【ローストア表】

表をローストア表として定義する場合は，表の正規化を検討してください。

表中の重複するデータを別表に移動することを，**表の正規化**といいます。表の格納効率や処理効率を向上させるために，表の正規化を行ってください。表の正規化を繰り返し行うことで，複雑なデータベースを最適な形にできます。

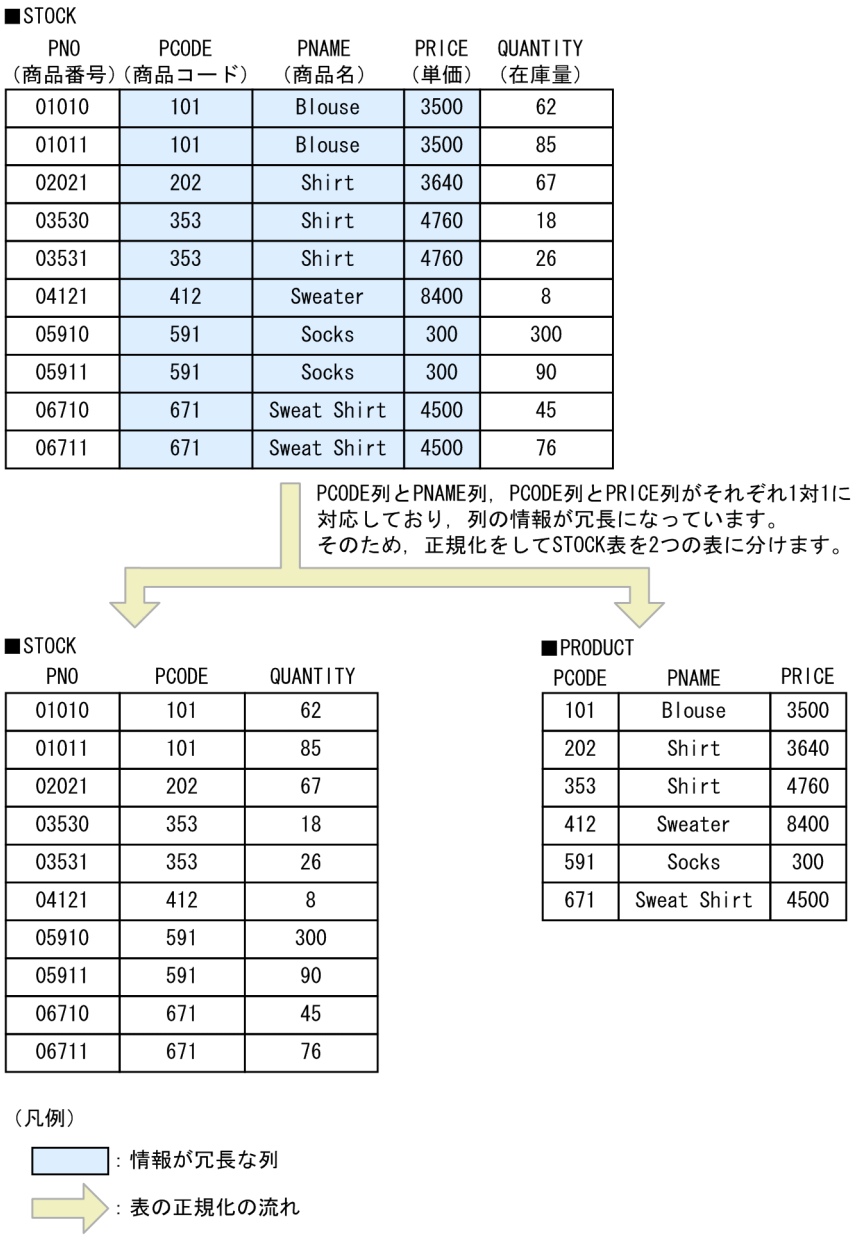
なお，表中に重複データが多く存在する場合は，表に配列型の列を定義することも検討してください。詳細については，「5.2.13　配列型の列の定義【カラムストア表】」を参照してください。

(1)　表の格納効率を向上させる場合

1 つの表に同じような情報を持つ列が複数ある場合は，この表を複数の表に分けて，それぞれの表に同じような情報を持つ列が 1 つだけになるように正規化します。これによって，表に対するデータの格納効率が向上します。

表に同じような情報を持つ列が複数ある場合の正規化を次の図に示します。

図 5-3　表に同じような情報を持つ列が複数ある場合の正規化の例



[説明]

正規化前のSTOCK 表のPCODE 列とPNAME 列，PCODE 列とPRICE 列はそれぞれ 1 対 1 で対応しているため，それぞれの列の情報は冗長になっています。このような場合は，STOCK 表から，PCODE 列，PNAME 列とPRICE 列で構成されるPRODUCT 表をほかに作成します。

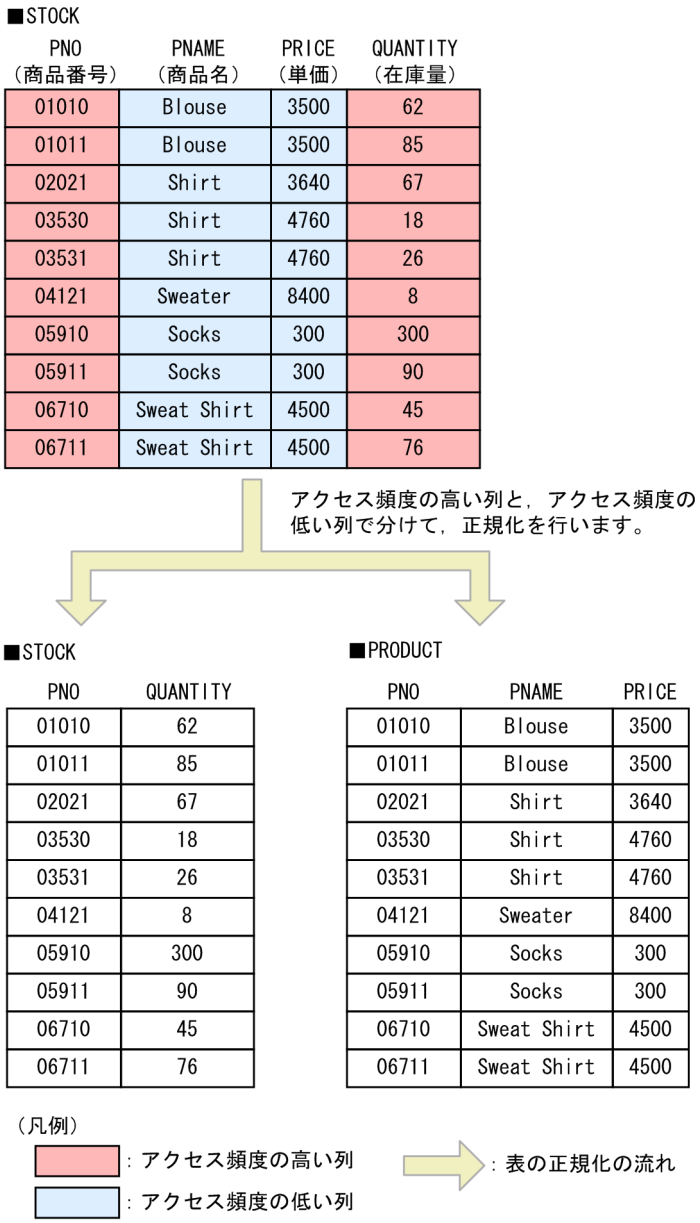
このとき，PRODUCT 表では，PCODE 列，PNAME 列およびPRICE 列に重複した情報を持たないようにします。

(2)　アクセス頻度の高い列と低い列がある場合

1 つの表に，アクセス頻度が高いと考えられる列と，アクセス頻度が低いと考えられる列がある場合は，この表をアクセス頻度が高い列で構成した表と，アクセス頻度が低い列で構成した表に正規化します。

アクセス頻度が高い列と低い列がある場合の正規化の例を次の図に示します。

図 5-4　アクセス頻度が高い列と低い列がある場合の正規化の例



[説明]

正規化前のSTOCK 表は，アクセス頻度の高い列（PNO 列，およびQUANTITY 列）と，アクセス頻度の低い列（PNAME 列，およびPRICE 列）が混在しています。このような場合は，STOCK 表をアクセス頻度の高い列を集めた表（STOCK 表）と，アクセス頻度の低い列を集めた表（PRODUCT 表）の 2 つの表に正規化します。

これによって，全体としての処理効率が向上します。

5.2.7　行長の固定（FIX 指定）【ローストア表】

表をローストア表として定義する場合は，ローストア表を FIX 表とするかどうかを検討してください。表をカラムストア表として定義する場合は，そのカラムストア表を FIX 表とすることはできません。

行長が固定となる表を**FIX 表**といいます。

固定長でナル値がないデータの表にFIX を指定すると，性能が向上します。また，FIX を指定していない表と比べて，物理的な行長が 1 列で 2 バイト短くなるため，列数の多い表の場合はディスク所要量を削減できます。

CREATE TABLE 文にFIX オプションを指定することで，FIX 表として定義されます。次に示す条件をすべて満たす場合は，FIX を指定してください。

- ナル値を持つ列がない場合
- 可変長の列がない場合

5.2.8　列の既定値の設定（DEFAULT 句）

データの追加や更新時に，決まった値を自動的に格納したい場合は，列の既定値の設定を検討してください。列の既定値とは，次のときに列に格納されるデフォルトの列値のことです。

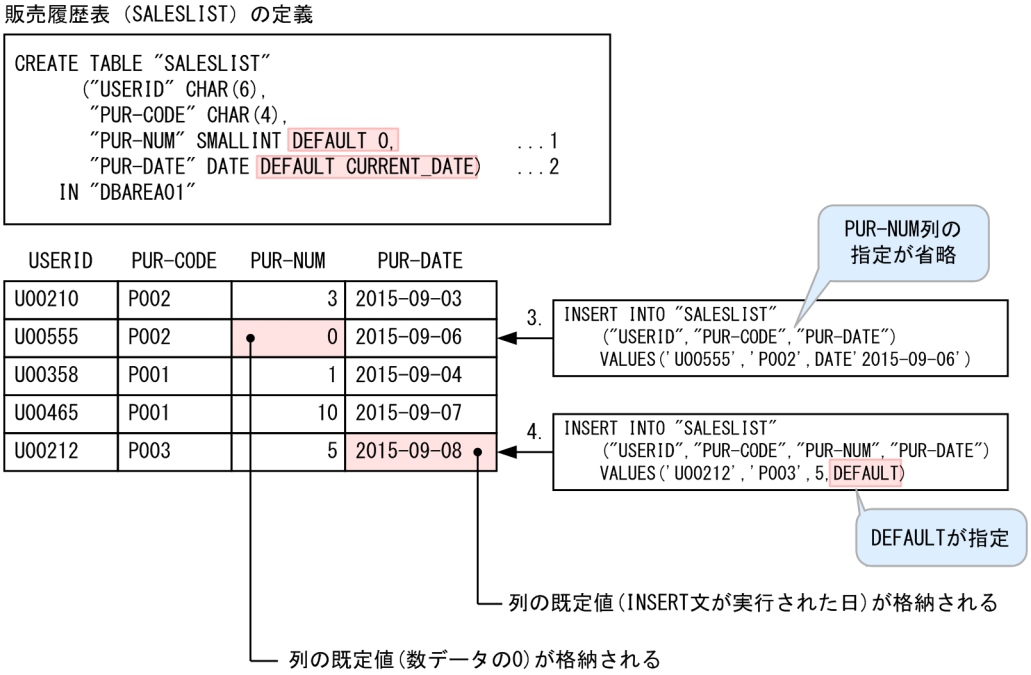
- INSERT 文による行の挿入時
  次の場合に列の既定値が格納されます。
  - 挿入値にDEFAULT を指定した場合
  - DEFAULT VALUES を指定した場合
  - データを挿入する列の列名の指定を省略した場合（すべての列名を省略した場合を除きます）
  - ビュー表に行を挿入した場合（ビュー表の列に対応しない基表の列には，列の既定値が格納されます）
- UPDATE 文による列値の更新時
  更新値にDEFAULT を指定した場合に，列の既定値が格納されます。
- adbimport コマンドによるデータインポート時
  adbimport コマンドでデータをインポートする際，入力データファイルのフィールドデータが空文字列の場合に，列の既定値が格納されます。
  ただし，インポートオプションadb_import_null_string にNULL を指定した場合は，列の既定値ではなくナル値が格納されます。

列の既定値を設定する場合は，CREATE TABLE 文でDEFAULT 句を指定します。DEFAULT 句については，マニュアル『HADB　SQL リファレンス』の『DEFAULT 句』を参照してください。

> **メモ**
>
> DEFAULT 句を省略した場合，ナル値が列の既定値になります。

DEFAULT 句を指定して列の既定値を設定したときのデータの格納例を次の図に示します。

図 5-5　DEFAULT 句を指定して列の既定値を設定したときのデータの格納例



[説明]

1. PUR-NUM 列にDEFAULT 句を指定して，数データの0 を列の既定値として定義します。
2. PUR-DATE 列にDEFAULT 句を指定して，日付を列の既定値として定義します。CURRENT_DATE は，INSERT 文またはUPDATE 文が実行された日，またはadbimport コマンドが起動された日を列の既定値とする指定です。
3. 挿入対象列にPUR-NUM 列が省略されているため，PUR-NUM 列には列の既定値（数データの0）が格納されます。
4. 挿入値にDEFAULT が指定されているため，PUR-DATE 列には列の既定値（INSERT 文が実行された日）が格納されます。

列の既定値として設定できる値については，マニュアル『HADB　SQL リファレンス』の『DEFAULT 句』を参照してください。

5.2.9　主キー（一意性制約定義）の指定（PRIMARY KEY）【シングルチャンク表】

表をシングルチャンク表として定義する場合は，主キーを定義するかどうかを検討してください。表をマルチチャンク表として定義する場合は，そのマルチチャンク表に主キーを定義できません。

実表中の行を一意（ユニーク）に識別するためのキーのことを，**主キー**といいます。主キーを定義することによって，実表のデータの一意性を保つことができます。

主キーを定義する場合，CREATE TABLE 文に一意性制約定義を指定します。

主キーを定義すると，主キーを構成する列には，次に示す 2 つの制約が適用されます。

- 一意性制約
  列値（または複数の列の列値の組み合わせ）が重複することを許さない制約のことです。
- 非ナル値制約
  列値にナル値を許さない制約（NOT NULL）のことです。

また，主キーを構成する列をインデクス構成列とするユニークインデクスが自動的に定義されます。ユニークインデクスのインデクス識別子の名称は，CREATE TABLE 文で指定した一意性制約定義中の制約名と同じになります。ユニークインデクスの詳細については，「5.3.5　ユニークインデクスを定義する列を決めるときの考慮点」を参照してください。

主キーを定義する場合は，実表中の行を一意に識別できる列，または列の組み合わせ（候補キー）の中から，主キーを選択してください。候補キーが複数ある場合は，候補キーの中で意味的に最も重要なものに主キーを定義してください。

重要

実表に主キーを定義していても，adbimport コマンドで重複する値をデータインポートした場合，一意性制約は保証されません。その場合，実表中から重複したキーを検索し，検索したキー値を持つ行を削除してください。重複したキーを検索する方法は，「15.9.2　一意性制約に違反した場合の対処方法（KFAA61205-W メッセージが出力された場合）」の「(2)　一意性制約違反状態を解除する方法」を参照してください。

5.2.10　外部キーの指定（FOREIGN KEY）

ほかの実表の主キーを参照する列（または複数の列の組み合わせ）のことを，**外部キー**といいます。主キーと外部キーによって，データベース内の表と表の関係を定義できます。

外部キーを定義する場合，CREATE TABLE 文に参照制約定義を指定します。

重要

CREATE TABLE 文に参照制約定義を指定しても，外部キーによる参照制約のチェックを行いません。そのため，表と表の間のデータの整合性を保つために，外部キーは利用できません。

メモ

- BI ツールなどで ER 図を出力したり，データベース内の表と表の関係を定義したりしたい場合に，外部キーを定義してください。

- 主キーについては，「5.2.9　主キー（一意性制約定義）の指定（PRIMARY KEY）【シングルチャンク表】」を参照してください。

5.2.11　データページ内の未使用領域の確保（PCTFREE）【ローストア表】

表をローストア表として定義する場合は，データページ内の未使用領域の比率の設定を検討してください。表をカラムストア表として定義する場合は，データページ内の未使用領域の比率の設定はできません。

CREATE TABLE 文のPCTFREE で，データページ（実表のデータを格納するページ）内の未使用領域の比率を指定できます。

次のコマンドを実行すると，PCTFREE に指定した未使用領域の比率に従って実表のデータが格納されます。

- adbimport コマンド
- adbunarchivechunk コマンド

実表に対して更新が頻繁に発生すると，同じページ内に行が格納できなくなります。その結果，行の配置に乱れが生じ，性能低下の原因になります。このため，実表に対して更新が頻繁に発生すると想定される場合は，データページ内の未使用領域の比率を高くしてください。

データページ内の未使用領域を確保するときの目安を次に示します。

実表に対して更新する行数が見積もれる場合

次に示す計算式で求めた結果を，PCTFREE に指定してください。

計算式（単位：%）

```
データページ内の未使用領域の比率＝
    ↑upd_row_num÷（row_num＋upd_row_num）×100↑
```

変数の説明

- *upd_row_num*：更新する行数（単位：個）
- *row_num*：実表全体の行数（単位：個）

実表に対して更新する行数が見積もれない場合

未使用領域には30%（PCTFREE の省略値）を指定してください。

実表に対して更新を行わない場合

未使用領域には0%を指定してください。

5.2.12　可変長データ型の列データの分岐指定（BRANCH）【ロースト ア表】

表をロースト ア表として定義する場合は，**可変長データ型の列データの分岐指定**を指定するかどうかを検討してください。表をカラムストア表として定義する場合は，**可変長データ型の列データの分岐指定**は指定できません。

ここでは，可変長データ型の列データの分岐指定について説明します。可変長データ型とは，次のデータ型のことです。

- VARCHAR 型
- VARBINARY 型

通常，1 行のデータは同じページに格納されますが，表に可変長データ型の列を定義した場合，可変長データ型の列データは分岐して別ページに格納されることがあります。

次に示す規則に従って，可変長データ型の列データは格納されます。

- **定義長が 255 バイト以下の可変長データ型の列の場合**
  分岐しません。可変長データ型の列データは，同じページに格納されます。
- **定義長が 256 バイト以上の可変長データ型の列の場合**
  基本行が 1 ページに収まるときは，可変長データ型の列データは分岐しません。同じページに格納されます。
  基本行が 1 ページに収まらないときは，可変長データ型の列のデータが分岐して別ページに格納されます。

CREATE TABLE 文の列定義のBRANCH の指定によって，可変長データ型の列データを，常に別のページに分岐するかどうかを指定できます。分岐するかどうかの目安を次に示します。

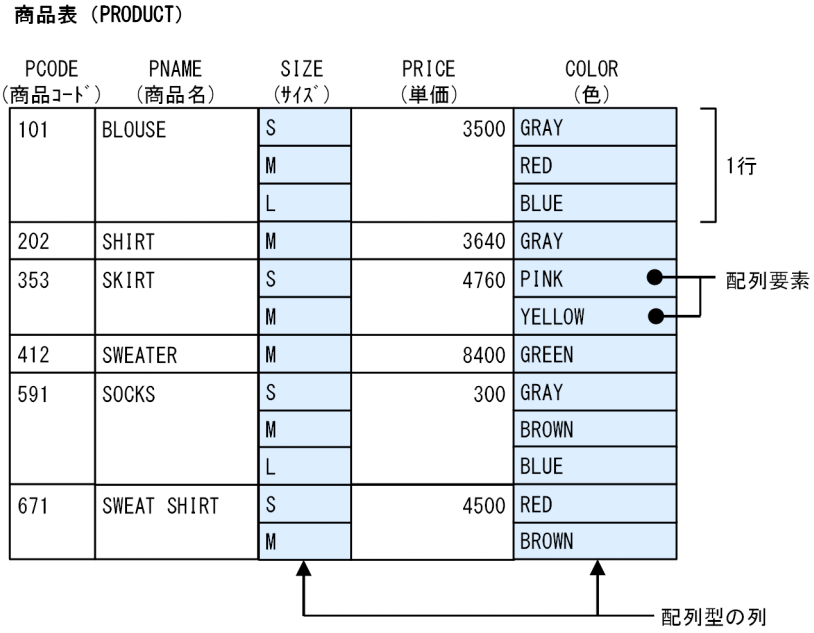
- **可変長データ型の列データの参照頻度が少ない場合**
  可変長データ型の列データを分岐して，別ページに格納するようにしてください。表の検索処理性能が向上することがあります。
  BRANCH にYES を指定すると，指定した可変長データ型の列のデータが分岐します。
- **可変長データ型（定義長が 256 バイト以上）の列データの参照頻度が多い場合**
  可変長データ型の列データを分岐しないようにしてください。表の検索処理性能が向上することがあります。
  BRANCH にNO を指定すると，指定した可変長データ型の列のデータは分岐しません。

なお，表オプションのBRANCH ALL を指定すると，表中の全可変長データ型の列データが分岐します。表中の全可変長データ型の列データの参照頻度が少ない場合に，BRANCH ALL を指定してください。

5.2.13　配列型の列の定義【カラムストア表】

HADB では，順序付けられた一連のデータを要素とする一次元の配列型のデータを扱うことができます(列のデータ型に配列型を定義できます)。表に配列型（ARRAY 型）の列を定義すると，複数の配列要素で構成されている配列データを 1 行にまとめて格納できます。配列型の列を定義した表の例を次の図に示します。

図 5-6　配列型の列を定義した表の例



[説明]

商品表（PRODUCT）に次の配列型の列を定義しています。

- SIZE（サイズ）
- COLOR（色）

配列型の列には，通常，1 行に 1 つ以上の配列要素が格納されます。この例の場合，SIZE 列には商品のサイズを示す配列要素（S，M，L）が，COLOR 列には商品の色を示す配列要素（RED，BLUE など）が格納されます。

なお，表を定義する際，配列要素のデータ型（**要素データ型**）を指定します。上記の例の場合，SIZE 列の要素データ型をCHAR(1)で定義し，COLOR 列の要素データ型をVARCHAR(8)で定義しています。要素データ型には，CHAR 型やVARCHAR 型のほかにINTEGER 型やDATE 型など，HADB でサポートしているデータ型を指定できます。

> **メモ**
>
> 配列型の列を定義した表の定義方法や，検索方法などについては，「11.1.18　配列型の列を定義した実表の運用」を参照してください。

■配列型の列を定義しない場合

配列型の列を定義しない場合，次のどちらかのように表を定義することになります。

- 重複データが多い表を定義する

  (例)

  商品表（PRODUCT）

PCODE（商品ｺｰﾄﾞ）	PNAME（商品名）	SIZE（ｻｲｽﾞ）	PRICE（単価）	COLOR（色）
101	BLOUSE	S	3500	GRAY
101	BLOUSE	M	3500	RED
101	BLOUSE	L	3500	BLUE
202	SHIRT	M	3640	GRAY
353	SKIRT	S	4760	PINK
353	SKIRT	M	4760	YELLOW
412	SWEATER	M	8400	GREEN
591	SOCKS	S	300	GRAY
591	SOCKS	M	300	BROWN
591	SOCKS	L	300	BLUE
671	SWEAT SHIRT	S	4500	RED
671	SWEAT SHIRT	M	4500	BROWN

  この場合，重複データがある行が増えるため，ディスク容量が増加します。

- 表の正規化を実施して複数の表を定義する

  (例)

  商品表（PRODUCT）

PCODE（商品ｺｰﾄﾞ）	PNAME（商品名）	PRICE（単価）
101	BLOUSE	3500
202	SHIRT	3640
353	SKIRT	4760
412	SWEATER	8400
591	SOCKS	300
671	SWEAT SHIRT	4500

  商品ｻｲｽﾞ表（PRODUCT_SIZE）

PCODE（商品ｺｰﾄﾞ）	SIZE（ｻｲｽﾞ）
101	S
101	M
101	L
202	M
353	S
353	M
412	M
591	S
591	M
591	L
671	S
671	M

  商品ｶﾗｰ表（PRODUCT_COLOR）

PCODE（商品ｺｰﾄﾞ）	COLOR（色）
101	GRAY
101	RED
101	BLUE
202	GRAY
353	PINK
353	YELLOW
412	GREEN
591	GRAY
591	BROWN
591	BLUE
671	RED
671	BROWN

  この場合，重複データを減らすことができますが，複数の表にアクセスする必要があるため，SQL 文の実行時に表の結合が必要になります。そのため，SQL 文が複雑になったり，検索性能が低下したりするおそれがあります。

■配列型の列を定義するメリット

- データの重複がなくなり，ディスク容量が削減されます。
- アクセス対象の表が 1 つになるため，表の結合が不要になり，SQL 文がシンプルになります。

- 関連するデータが物理的に近くに格納されるため，表の結合で複数の表にアクセスするよりも性能面で優れています。

■前提条件および制限事項

- 配列型の列を定義した表は，カラムストア表として定義する必要があります。ローストア表としては定義できません。

  カラムストア表の概要については，「2.2.1　ローストア表とカラムストア表」を参照してください。カラムストア表を設計する際の考慮点については，「5.2.2　ローストア表とカラムストア表の選択基準」を参照してください。
- 配列型の列を定義した表には，B-tree インデクスおよび一意性制約を定義できません。
- 配列型の列には，DEFAULT 句を指定できません。
- 配列型の列を定義した表は，SQL 文（INSERT 文，UPDATE 文，DELETE 文）による行の追加，更新，および削除ができません。

■最大要素数の決め方

CREATE TABLE 文で表を定義する際，配列型の列の配列要素の最大数（**最大要素数**）を指定します。最大要素数の指定値が，実際の配列要素の数よりも著しく大きい場合，リソースを必要以上に使用するため，メモリ不足や性能低下の原因となるおそれがあります。そのため，最大要素数には，想定される配列要素の最大数を目安にした値を指定するようにしてください。

なお，最大要素数は，ALTER TABLE 文で後から大きくすることができます。ただし，最大要素数を大きくした表に依存するビュー表が無効化されます。ALTER TABLE 文による最大要素数の変更方法，およびその際の留意事項については，「11.1.18　配列型の列を定義した実表の運用」の「(4)　最大要素数の変更」を参照してください。

重要

最大要素数を小さくする場合は，CREATE TABLE 文で表を再定義する必要があります。

■データ用 DB エリアに関する留意事項

- 配列型の列を定義した表を格納するデータ用 DB エリアのページサイズは，8,192 バイト以上にしてください。
- 配列型の列を定義した表を格納するデータ用 DB エリアの 1 セグメントの容量は，次の計算式を満たすようにしてください。

  1セグメントの容量（単位：バイト）≧*ARRAY_ROWSZ*（配列型の列を定義した表の行長）

  ARRAY_ROWSZ の求め方については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(3)　配列型の列を定義した表の行長（変数 ARRAY_ROWSZ）の求め方」を参照してください。

メモ

- データ用 DB エリアのページサイズおよび 1 セグメントの容量は，adbinit コマンドまたはadbmodarea コマンドでデータ用 DB エリアを作成する際に指定します。

- 1 セグメントの容量は，adbinit コマンドまたはadbmodarea コマンドの-s オプションで指定しますが，-s オプションの指定単位はメガバイト単位になっています。上記の計算式の単位はバイトのため，注意してください。

5.3 B-tree インデクスの設計

ここでは，B-tree インデクスを設計する際に検討する項目について説明します。

B-tree インデクスは AP の性能に大きく影響するため，どの列に B-tree インデクスを定義するかを十分に検討する必要があります。

重要

カラムストア表に定義した B-tree インデクスが，カラムストア表の検索時に使用されるケースは限定されています。カラムストア表で，B-tree インデクスを使用する代表的な例を次に示します。次のような絞り込みが有効な場合にだけ，カラムストア表に B-tree インデクスを定義することを推奨します。

- SQL 文の実行時に，インデクス指定で使用するインデクスを指定する場合
- 集合関数MIN またはMAX を指定して，最小値や最大値を求める場合
- 探索条件を指定して，UPDATE 文またはDELETE 文を実行する場合

特定の年や月のデータにアクセスするような場合は，レンジインデクスを定義して検索範囲を絞り込むことを推奨します。

カラムストア表の検索時に使用される B-tree インデクスの規則については，マニュアル『HADB　AP 開発ガイド』の『SQL 文の実行時に使用される B-tree インデクスおよびテキストインデクス』を参照してください。

5.3.1 B-tree インデクスを定義する場合の注意点（B-tree インデクスの未完状態）

B-tree インデクスを定義する場合は，実表を作成後，データを格納する前にCREATE INDEX 文で定義してください。

行を格納するセグメントが割り当てられている状態の実表に対して B-tree インデクスを定義した場合，定義された B-tree インデクスは**未完状態**になります。未完状態になると，定義した B-tree インデクスは無効な状態となり，使用できなくなります。

未完状態を解除するには，B-tree インデクスを再作成する必要があります。そのため，実表にデータを格納する前に，B-tree インデクスをどの列に定義するかを事前によく検討してください。

未完状態の B-tree インデクスに対して，次に示す操作を行うとエラーになります。

■エラーになる操作

- 未完状態の B-tree インデクスを使用するSELECT 文の実行

- 未完状態の B-tree インデクスが定義された表に対するINSERT 文，UPDATE 文，およびDELETE 文の実行
- 未完状態の B-tree インデクスが定義された実表に対する-d オプションを指定しないadbimport コマンドの実行
- 未完状態の B-tree インデクスが定義された実表に対するadbmergechunk コマンドの実行
- 未完状態の B-tree インデクスが定義されたアーカイブマルチチャンク表に対するadbunarchivechunk コマンドの実行

B-tree インデクスの未完状態の解除方法については，「15.9.1　B-tree インデクスが未完状態になったときの対処方法」を参照してください。

メモ

次に示すときなどは，行を格納するセグメントが割り当てられていない状態です。このときに，CREATE INDEX 文を実行すると，B-tree インデクスは正常に作成されます。

- 実表の定義直後
- TRUNCATE TABLE 文の実行直後

例えば，DELETE 文で表の全行を削除しても，行を格納するセグメントは割り当てられている状態のままです。そのため，DELETE 文で表の全行を削除したあとにCREATE INDEX 文を実行した場合，その B-tree インデクスは未完状態になります。

5.3.2　B-tree インデクスを定義する列を決めるときの考慮点

HADB は，追記型の更新処理方式を採用しています。そのため，列の値が更新されると，行全体が更新されます（更新前の行は無効になり，更新後の行が新たに挿入されます）。表に B-tree インデクスが定義されている場合，行が更新されると，B-tree インデクスのキー値も更新されます。行の更新が頻繁に発生するような表に B-tree インデクスを必要以上に定義すると，更新処理の性能がその分低下します。

B-tree インデクスを定義する列を決める場合，次に示す点を考慮してください。

(1)　B-tree インデクスを定義した方がよい列

B-tree インデクスを定義した方がよい列の例を次に示します。

データを絞り込むための条件に指定する列

WHERE 句に探索条件（検索条件）を指定して検索する場合，探索条件に指定する列に B-tree インデクスを定義しておくと，検索するページ数を削減できます（検索範囲を絞り込めます）。検索性能の向上につながるため，探索条件に指定する列の中で，データの検索範囲を絞り込める列に B-tree インデクスを定義することを検討してください。

表の結合の条件に指定する列（複数の表を指定して検索する場合）

複数の表に検索対象のデータが分散していて，共通の情報を持つ列を対応づけて検索する（表の結合を行う）場合，結合条件に指定する列に B-tree インデクスを定義しておくと，結合条件の評価回数（行の突き合わせ回数）を削減できます。検索性能の向上につながるため，結合条件に指定する列に B-tree インデクスを定義することを検討してください。

また，次の条件をすべて満たす場合は，結合条件に指定する列にレンジインデクスも定義することも検討してください。

- B-tree インデクスを定義する表がマルチチャンク表である
- 結合条件に指定する列が，「5.5.1　レンジインデクスを定義する際の考慮点」の「(1)　レンジインデクスを定義した方がよいケース」の条件を満たしている

(2)　B-tree インデクスを定義しない方がよい列

B-tree インデクスを定義しない方がよい列の例を次に示します。

データの重複が多い列

データの重複が多い列に B-tree インデクスを定義しても，データの検索範囲を絞り込めないため，検索するページ数を削減できません。検索性能が低下する原因になるため，データの重複が多い列には B-tree インデクスを定義しないようにしてください。

5.3.3　単一列インデクスと複数列インデクスの使い分け

B-tree インデクスには，単一列インデクスと複数列インデクスがあります。

- **単一列インデクス**

  表の 1 つの列で 1 つの B-tree インデクスを作成したものを指します。単一列インデクスは，1 つの列をキーにして検索する場合に指定します。
- **複数列インデクス**

  表の複数の列で 1 つの B-tree インデクスを作成したものを指します。

単一列インデクスと複数列インデクスの使い分けについて，次で説明します。

(1)　単一列インデクスを定義した方がよい場合

1 つの列をキーにして検索する場合は，単一列インデクスを定義してください。

例えば，次のような検索をする場合には，列C1 に単一列インデクスを定義します。

SQL の指定例

```
SELECT * FROM "T1"
     WHERE "C1" = 1
```

B-tree インデクスの定義例

```
CREATE INDEX ”T1IX_C1”
     ON ”T1” (”C1” ASC) IN DBAREA01 EMPTY
```

(2) 複数列インデクスを定義した方がよい場合

1 つの列だけでは検索する行を十分に絞り込めないため，複数の列をキーにして検索する行を絞り込む場合は，複数列インデクスを定義してください。

例えば，次のような検索をする場合に，列C1 と列C2 に対してそれぞれ単一列インデクスを定義していると，列C1 または列C2 のどちらか 1 つの B-tree インデクスしか使用されません。そのため，列C1 と列C2 を構成列とする複数列インデクス（T1IX_C1C2）を定義します。

SQL の指定例

```
SELECT * FROM ”T1”
      WHERE ”C1” = 1 AND ”C2” = ’A’
```

B-tree インデクスの定義例

```
CREATE INDEX ”T1IX_C1C2”
      ON ”T1”(”C1” ASC,”C2” ASC) IN DBAREA01 EMPTY
```

なお，複数列インデクスを使って探索条件を評価する場合，複数列インデクスを定義したときの列の指定順に値を評価します。そのため，複数列インデクスを定義する際は，検索時に指定する探索条件などを考慮して列の指定順序を決定してください。

例えば，上記の「SQL の指定例」を実行する場合，「B-tree インデクスの定義例」のとおりCREATE INDEX 文を実行することを推奨します。

(3) 複数列インデクスを使用するときの留意事項

探索条件によるデータの絞り込み範囲について説明します。

複数列インデクスを使用して検索する行を絞り込む場合，SELECT 文のWHERE 句に指定する探索条件によって検索範囲の絞り込み方が異なります。インデクス構成列の指定順序は，「*＝条件*」を指定することが多い列を先に指定する方が検索範囲を小さくできます。

WHERE 句の探索条件の指定と検索範囲の関係を次に示します。

B-tree インデクスの定義例

```
CREATE INDEX ”T1IX_C1C2C3”
        ON ”T1”(”C1” ASC,”C2” ASC,”C3” ASC) IN DBAREA01 EMPTY
```

図 5-7　WHERE 句の探索条件と B-tree インデクスの検索範囲

インデクス構成例	B-treeインデクスのキー値											
C1	1	1	1	1	2	2	2	2	3	3	3	3
C2	A	B	B	C	A	B	B	C	A	B	B	C
C3	10	10	20	10	10	10	20	10	10	10	20	10

WHERE句の探索条件	検索範囲
WHERE "C1" = 2 AND "C2" = 'B' AND "C3" = 20	
WHERE "C1" = 2 AND "C2" >= 'B' AND "C3" = 20※1	
WHERE "C1" = 2 AND "C2" = 'B'	
WHERE "C1" = 2 AND "C2" >= 'B'	
WHERE "C1" = 2 AND "C3" = 20※2	
WHERE "C1" = 2	
WHERE "C1" >= 2 AND "C2" = 'B' AND "C3" = 20※3	
WHERE "C1" >= 2	
WHERE "C2" = 'B' AND "C3" = 20※4	
WHERE "C2" = 'B'※4	
WHERE "C3" = 20※4	

（凡例） 網掛け部分が検索の範囲です。

注※1

列C2 に対する条件が「=*条件*」または「IS NULL *条件*」ではないため，列C3 に対する条件は検索範囲の絞り込みには使用されません。

注※2

列C2 に対する条件がないため，列C3 に対する条件は検索範囲の絞り込みには使用されません。

注※3

列C1 に対する条件が「=*条件*」または「IS NULL *条件*」ではないため，列C2 と列C3 に対する条件は検索範囲の絞り込みには使用されません。

注※4

列C1 に対する条件がないため，列C2 や列C3 に対する条件は検索範囲の絞り込みには使用されません。

5.3.4　B-tree インデクスのインデクスページ内の未使用領域の確保 (PCTFREE)

CREATE INDEX 文のPCTFREE で，B-tree インデクスのインデクスページ内の未使用領域をどれだけ確保するか指定できます。

B-tree インデクスが定義されている表に対して行を追加した場合，または行を更新した場合に，インデクスキーの格納対象となるインデクスページに未使用領域がないと，B-tree インデクスのインデクスページスプリットが発生します。B-tree インデクスのインデクスページスプリットについては，「(2)　B-tree インデクスのインデクスページスプリット」を参照してください。

B-tree インデクスのインデクスページスプリットが発生すると，ディスクの I/O 回数が増加するため，HADB の性能が低下するおそれがあります。そのため，操作系 SQL の INSERT 文による行の追加，および UPDATE 文による行の更新を行う表に定義する B-tree インデクスについては，インデクスページ内に未使用領域を確保してください。

(1)　B-tree インデクスのインデクスページ内に未使用領域を確保するときの目安

B-tree インデクスのインデクスページ内に未使用領域を確保するときの目安を次に示します。

表に対して追加および更新する行数が見積もれる場合

表に対して追加および更新する行数分，インデクスキーが増加します。未使用領域には，次に示す計算式で求めた値を指定してください。

計算式（単位：%）

```
未使用領域＝
    ↑(追加および更新する行数÷(データページ構築時の行数＋追加および更新する行数))×100↑
```

*データページ構築時の行数*とは，対象の表に初めて adbimport コマンドを実行して格納した表データの行数のことです。

表に対して追加および更新する行数が見積もれない場合

未使用領域には30%（PCTFREE の省略値）を指定してください。

表に対して追加および更新を行わない場合（参照だけを行う場合）

未使用領域には0%を指定してください。

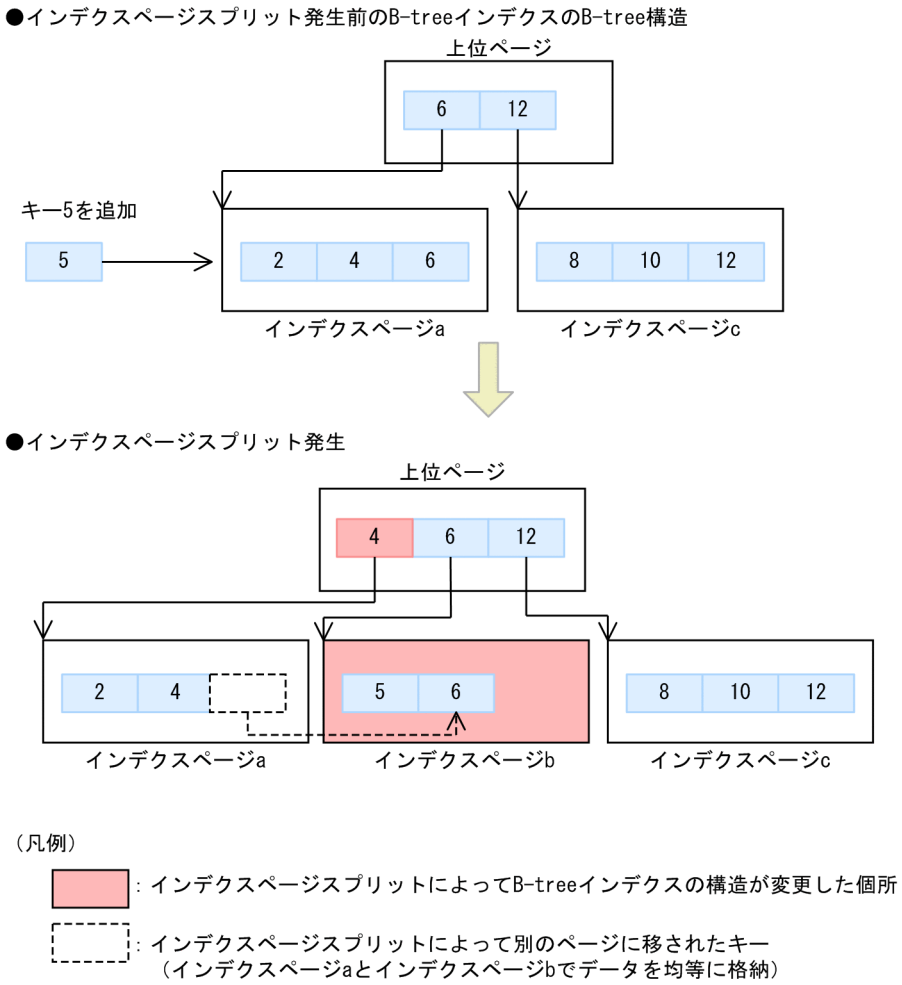
次に示すコマンドを実行すると，PCTFREE に指定した未使用領域の比率に従ってデータが格納されます。

- adbimport コマンド
  - 作成モードでデータインポートする場合（-d オプションを指定した場合）
  - バックグラウンドインポート機能を使用する場合（-b オプションを指定した場合）
  - 追加モードでデータインポートする場合（-d オプションと-b オプションを指定しない場合）
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド

(2)　B-tree インデクスのインデクスページスプリット

B-tree インデクスはページごとに管理するキーの範囲を決定しています。キー追加時に，本来そのキーを格納するためのインデクスページに空きがない場合，該当ページが管理しているキー群を 2 つに分割します。これを，**B-tree インデクスのインデクスページスプリット**といいます。B-tree インデクスのインデクスページスプリットの例を次の図に示します。

図 5-8　B-tree インデクスのインデクスページスプリットの例



5.3.5　ユニークインデクスを定義する列を決めるときの考慮点

ユニークインデクスとは，キー値（B-tree インデクスを定義した列の値）の重複を許さない B-tree インデクスのことです。キー値の重複を許さない制約を**一意性制約**といいます。

B-tree インデクスを定義する列に対して，重複したキー値がないことを保証したい場合に，ユニークインデクスを定義することを検討してください。

メモ

実表の定義時に主キー（一意性制約定義）を定義した場合も，ユニークインデクスが自動的に定義されます。主キーについては，「5.2.9　主キー（一意性制約定義）の指定（PRIMARY KEY）【シングルチャンク表】」を参照してください。

ただし，次に示す場合は重複したキー値があっても重複キー値にはなりません（一意性制約に違反しません）。

- ナル値を含むキー値の場合
- 複数列インデクスで，どれか 1 つの列の値が異なる場合

ユニークインデクスは，CREATE INDEX 文のUNIQUE を指定することで定義できます。UNIQUE を指定した場合，キー値が重複するようなデータの更新および追加はできません。また，あるキー値を持つ行を追加しようとした際に，実行中の別のトランザクションがすでに同じキー値を持つ行を追加していた場合も追加できません。

重要

ユニークインデクスを定義していても，次の場合には一意性制約は保証されません（キー値の重複が発生するおそれがあります）。

- adbimport コマンドでデータインポートした場合
- adbidxrebuild コマンドで B-tree インデクスを再作成した場合

5.3.6　ナル値除外指定の設定（EXCLUDE NULL VALUES）

B-tree インデクスの定義時にナル値除外指定を設定すると，ナル値だけで構成されるキーを含まない B-tree インデクスが作成されます（ナル値だけで構成されるキーは作成されません）。

例えば，列の大半の値がナル値だけで構成される列を，インデクス構成列にする場合は，ナル値除外指定を設定することを検討してください。ナル値除外指定を設定することで，ナル値だけで構成されるキーは作成されないため，B-tree インデクスの作成時間を短縮できます。また，B-tree インデクスのデータ容量を削減できます。

ナル値除外指定は，CREATE INDEX 文で B-tree インデクスを定義するときに，EXCLUDE NULL VALUES を指定することで設定できます。

なお，表の検索範囲にナル値が含まれることがある場合，ナル値除外指定を設定した B-tree インデクスは，検索に使用されるインデクスの候補にはならないため，注意してください。例えば，検索条件がない場合，IS NULL 述語が指定されている場合などが，これに該当します。

5.4　テキストインデクスの設計

ここでは，テキストインデクスを設計する際に検討する項目について説明します。

なお，テキストインデクスはCREATE INDEX 文で定義します。

> **重要**
>
> カラムストア表にはテキストインデクスを定義できません。

5.4.1　テキストインデクスを定義する列を決めるときの考慮点

テキストインデクスを定義する列を決める場合，次に示す点を考慮してください。

(1)　テキストインデクスを定義した方がよいケース

次に示す検索を行う場合に，テキストインデクスを定義することを検討してください。

- LIKE 述語による中間一致検索または後方一致検索を行う場合

  テキストインデクスを使用すると，B-tree インデクスを使用するよりも，読み込むページ数を削減できます。そのため，B-tree インデクスを定義するよりも，テキストインデクスを定義した方が，高速に検索できます。テキストインデクスは，LIKE 述語の一致値に指定する列に定義します。

- LIKE_REGEX 述語を指定した検索を実行する場合

  テキストインデクスを使用すると，読み込むページ数を削減できます。そのため，テキストインデクスを定義した方が，高速に検索できます。

- スカラ関数CONTAINS を指定した検索を実行する場合

  テキストインデクスを使用すると，読み込むページ数を削減できます。そのため，テキストインデクスを定義した方が，高速に検索できます。

- 表記ゆれ補正検索を行う場合

  表記ゆれ補正検索用のテキストインデクスを使用すると，読み込むページ数を削減できます。そのため，表記ゆれ補正検索用のテキストインデクスを定義した方が，高速に検索できます。表記ゆれ補正検索については，「2.17.1　表記ゆれ補正検索」を参照してください。

- ワード検索を行う場合

  ワード検索用のテキストインデクスを使用すると，読み込むページ数を削減できます。そのため，ワード検索用のテキストインデクスを定義した方が，高速に検索できます。ワード検索については，「2.17.4 ワード検索」を参照してください。

検索対象データとテキストインデクスの関係を次の表に示します。

表 5-4　検索対象データとテキストインデクスの関係

項番	検索対象データ	テキストインデクスの有効性	注意事項
1	文書データ（100 文字以上の文書資料やメールなど）	◎	文字列が長い場合は，テキストインデクスのデータ容量が大きくなるおそれがあります。 また，文書データの場合，検索条件に使用しないような文字列もテキストインデクスのデータに含まれるため，ほかの検索対象データに比べて，テキストインデクスのデータ容量が大きくなる傾向にあります。
2	英語で書かれた文書データ（100 文字以上の文書資料やメールなど）	◎	英語で書かれた文章に含まれる英単語を検索する場合は，テキストインデクスワード検索指定を指定して，テキストインデクスを定義してください。また，スカラ関数CONTAINS を使用してワード検索を行ってください。テキストインデクスワード検索指定については，「5.4.5　ワード検索用のテキストインデクスの指定（TEXT WORDCONTEXT）」を参照してください。 なお，文字列が長い場合は，テキストインデクスのデータ容量が大きくなるおそれがあります。また，文書データの場合，検索条件に使用しないような文字列もテキストインデクスのデータに含まれるため，ほかの検索対象データに比べて，テキストインデクスのデータ容量が大きくなる傾向にあります。 さらに，テキストインデクスワード検索指定を指定すると，指定しない場合に比べて，テキストインデクスのデータ容量が大きくなります。
3	英数字だけの文字列（URL やアクセスログなど）	○	次の場合は，テキストインデクスを定義しても，効果がない可能性があります。 • 極端にデータ長が短い場合 • 使われている文字のバリエーションが少ない場合
4	名称データ（10 文字前後の商品名など）	△	次の場合は，テキストインデクスを定義しても，効果がない可能性があります。 • 10 文字前後の文字列でも，購入履歴のような同じ名称データが大量に存在する場合
5	数字だけの文字列（ID やコードなど）	×	仮に，検索結果が一意に決まる場合であっても，数字だけの文字列のときは文字の組み合わせが少ないため，検索に時間が掛かる可能性があります。そのため，テキストインデクスを定義しても，効果を期待できません。

（凡例）

◎：有効である可能性が非常に高い

○：有効である可能性が高い

△：有効である可能性が低い

×：有効ではない

(2) テキストインデクスを定義しない方がよいケース

次のケースに該当する場合は，対象列にテキストインデクスを定義しないようにしてください。

- 列のデータに重複（部分一致または完全一致）が多い場合
- 数字だけの番号列（ID など）のような文字の種類が少ない場合
- 検索する文字列が「a」や「0」のような単純で短い文字列の場合
- 検索する文字列が長過ぎる場合（1,000 文字を超える場合）

仮に，テキストインデクスを定義しても，データの検索範囲を絞り込めないため，検索するページ数が削減できません。そのため，テキストインデクスの効果が期待できません。

(3) テキストインデクスを定義できる列

次の条件を満たす列だけに，テキストインデクスを定義できます。

テキストインデクスを定義できる列

- CHARACTER 型の列
- VARCHAR 型の列

5.4.2 テキストインデクスのインデクスページ内の未使用領域の確保 (PCTFREE)

CREATE INDEX 文のPCTFREE で，テキストインデクスのインデクスページ内の未使用領域をどれだけ確保するか指定できます。

テキストインデクスが定義されている表に対して行を追加した場合，または行を更新した場合に，インデクスキーの格納対象となるインデクスページに未使用領域がないと，テキストインデクスのインデクスページスプリットが発生します。テキストインデクスのインデクスページスプリットについては，「(2) テキストインデクスのインデクスページスプリット」を参照してください。

テキストインデクスのインデクスページスプリットが発生すると，ディスクの I/O 回数が増加するため，HADB の性能が低下するおそれがあります。そのため，操作系 SQL のINSERT 文による行の追加，およびUPDATE 文による行の更新を行う表に定義するテキストインデクスについては，インデクスページ内に未使用領域を確保してください。

(1) テキストインデクスのインデクスページ内に未使用領域を確保するときの目安

テキストインデクスのインデクスページ内に未使用領域を確保するときの目安を次に示します。

表に対して追加および更新する行数が見積もれる場合

表に対して追加および更新する行数分，インデクスキーが増加します。未使用領域には，次に示す計算式で求めた値を指定してください。

計算式（単位：%）

```
未使用領域＝
    ↑(追加および更新する行数÷(データページ構築時の行数＋追加および更新する行数))×100↑
```

*データページ構築時の行数*とは，対象の表に初めてadbimport コマンドを実行して格納した表データの行数のことです。

表に対して追加および更新する行数が見積もれない場合

未使用領域には30%（PCTFREE の省略値）を指定してください。

表に対して追加および更新を行わない場合（参照だけを行う場合）

未使用領域には0%を指定してください。

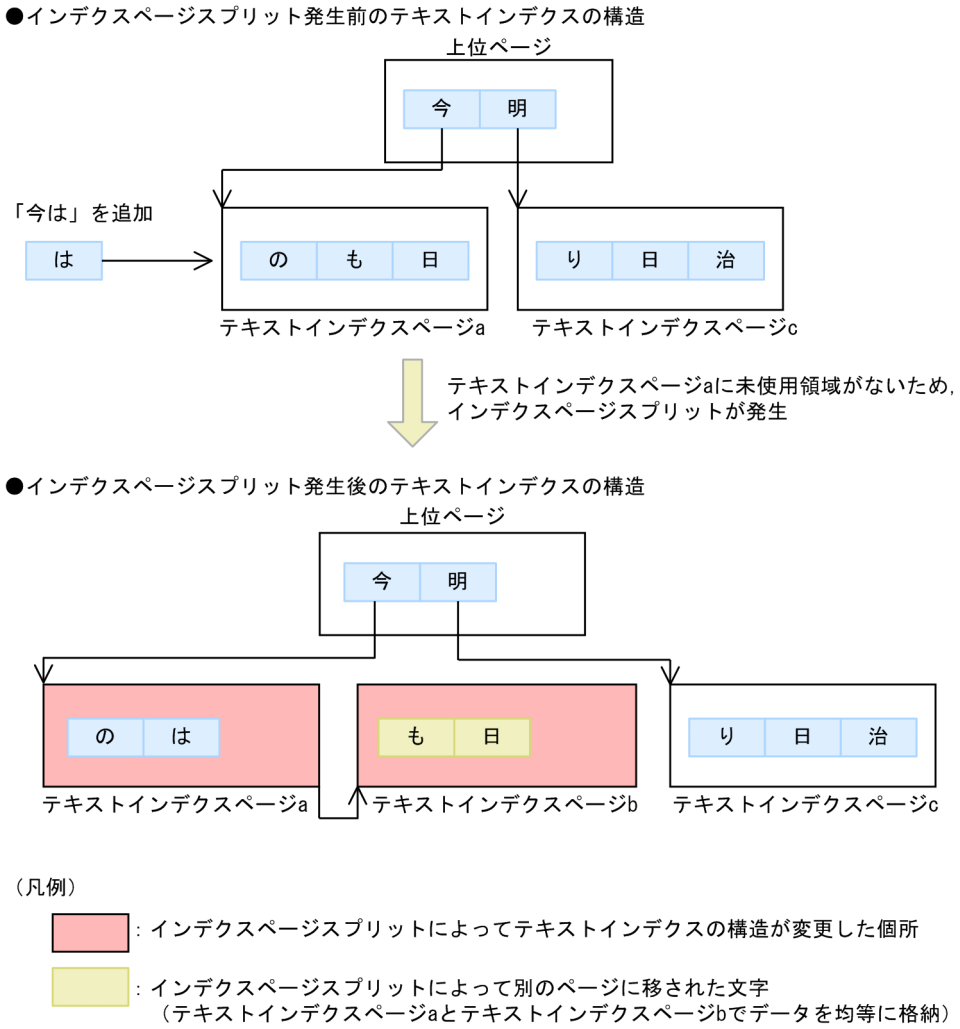
次に示すコマンドを実行すると，PCTFREE に指定した未使用領域の比率に従ってデータが格納されます。

- adbimport コマンド
  - 作成モードでデータインポートする場合（-d オプションを指定した場合）
  - バックグラウンドインポート機能を使用する場合（-b オプションを指定した場合）
  - 追加モードでデータインポートする場合（-d オプションと-b オプションを指定しない場合）
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド

(2) テキストインデクスのインデクスページスプリット

テキストインデクスを定義した実表にデータを追加したり，データを更新したりした場合，テキストインデクスのインデクスページにデータが追加されます。このとき，データを追加するインデクスページに空きがない場合，新規にテキストインデクスのインデクスページを確保します。これを，**テキストインデクスのインデクスページスプリット**といいます。テキストインデクスのインデクスページスプリットの例を次の図に示します。

図 5-9　テキストインデクスのインデクスページスプリットの例



5.4.3　テキストインデクスを定義する場合の注意点（テキストインデクスの未完状態）

テキストインデクスを定義する場合は，実表を作成後，データを格納する前にCREATE INDEX 文で定義してください。

行を格納するセグメントが割り当てられている状態の実表に対してテキストインデクスを定義した場合，定義されたテキストインデクスは**未完状態**になります。未完状態のテキストインデクスは無効な状態であるため，使用できません。

未完状態を解除するには，テキストインデクスを再作成する必要があります。そのため，実表にデータを格納する前に，テキストインデクスをどの列に定義するかを検討してください。

未完状態のテキストインデクスに対して，次に示す操作を行うとエラーになります。

■エラーになる操作

- 未完状態のテキストインデクスを使用するSELECT 文の実行
- 未完状態のテキストインデクスが定義された表に対するINSERT 文，UPDATE 文，およびDELETE 文の実行
- 未完状態のテキストインデクスが定義された実表に対する-d オプションを指定しないadbimport コマンドの実行
- 未完状態のテキストインデクスが定義された実表に対するadbmergechunk コマンドの実行
- 未完状態のテキストインデクスが定義されたアーカイブマルチチャンク表に対するadbunarchivechunk コマンドの実行

テキストインデクスの未完状態の解除方法については，「15.10.1　テキストインデクスが未完状態になったときの対処方法」を参照してください。

メモ

次に示すときなどは，行を格納するセグメントが割り当てられていない状態です。このときに，CREATE INDEX 文を実行すると，テキストインデクスは正常に作成されます。

- 実表の定義直後
- TRUNCATE TABLE 文の実行直後

例えば，DELETE 文で表の全行を削除しても，行を格納するセグメントは割り当てられている状態のままです。そのため，DELETE 文で表の全行を削除したあとにCREATE INDEX 文を実行した場合，そのテキストインデクスは未完状態になります。

5.4.4　テキストインデクスの表記ゆれ補正指定の指定（CORRECTIONRULE）

表記ゆれ補正検索を行う場合は，表記ゆれ補正検索に対応しているテキストインデクスを定義してください。表記ゆれ補正検索については，「2.17.1　表記ゆれ補正検索」を参照してください。

また，表記ゆれ補正検索に対応しているテキストインデクスを定義すると，次に示す検索を行う場合に読み込むページ数を少なくすることができます。その分，表の検索性能を向上させることができます。

- スカラ関数CONTAINS を指定して，表記ゆれ補正検索を実行する
- LIKE_REGEX 述語で，FLAG に「I」または「IGNORECASE」を指定して，正規表現を使用して検索を実行する（英大文字と英小文字を同一の文字と判断して検索する）

表記ゆれ補正検索に対応しているテキストインデクスを定義する場合は，CREATE INDEX 文でCORRECTIONRULE（テキストインデクス表記ゆれ補正指定）を指定します。

なお，表記ゆれ補正検索に対応しているテキストインデクスは，通常のテキストインデクスに比べて，インデクスデータの容量が増加します。そのため，テキストインデクスを定義する前に，インデクスデータ

の容量を求めて，問題がないことを確認してください。詳細については，「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」を参照してください。

重要

HADB サーバで使用する文字コードが Shift-JIS の場合（環境変数ADBLANG の指定値がSJIS の場合），表記ゆれ補正検索に対応しているテキストインデクスを定義することはできません。

表記ゆれ補正検索に対応しているテキストインデクスは，通常のテキストインデクスとしても使用できます。

5.4.5　ワード検索用のテキストインデクスの指定（TEXT WORDCONTEXT）

ワード検索を行う場合は，ワード検索用のテキストインデクスを定義してください。ワード検索については，「2.17.4　ワード検索」を参照してください。

また，ワード検索用のテキストインデクスを定義すると，スカラ関数CONTAINS を指定してワード検索を行う場合に読み込むページ数を少なくすることができます。その分，表の検索性能を向上させることができます。

ワード検索用のテキストインデクスを定義する場合は，CREATE INDEX 文のINDEXTYPE にTEXT WORDCONTEXT を指定します。TEXT WORDCONTEXT の指定を，テキストインデクスワード検索指定といいます。

メモ

ワード検索用のテキストインデクスを定義する場合は，DELIMITER（テキストインデクス区切り文字指定）も指定する必要があります。DELIMITER については，「5.4.6　ワード検索時の区切り文字の選択（DELIMITER）」を参照してください。

重要

1～2 文字のような短い文字列を指定してワード検索の前方一致検索を行う場合は，ワード検索用のテキストインデクスを定義しないでください。ワード検索用のテキストインデクスを使用しても，データの検索範囲を絞り込めないため，検索性能が低下します。1～2 文字のような短い文字列を指定して前方一致検索を行いたい場合は，ワード検索以外の検索を行うようにしてください。

なお，ワード検索用のテキストインデクスは，通常のテキストインデクスに比べて，インデクスデータの容量が増加します。そのため，テキストインデクスを定義する前に，インデクスデータの容量を求めて，問題がないことを確認してください。詳細については，「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」を参照してください。

ワード検索用のテキストインデクスは，通常のテキストインデクスとしても使用できます。

5.4.6　ワード検索時の区切り文字の選択（DELIMITER）

ワード検索に対応するテキストインデクスを定義する際，単語と単語の間を区切る区切り文字の種類を指定する必要があります。ワード検索に対応するテキストインデクスについては，「5.4.5　ワード検索用のテキストインデクスの指定（TEXT WORDCONTEXT）」を参照してください。

区切り文字は，CREATE INDEX 文のDELIMITER（テキストインデクス区切り文字指定）で指定します。DELIMITER は，INDEXTYPE にTEXT WORDCONTEXT を指定しているときに指定する必要があります。

区切り文字の種類には，DEFAULT とALL があります。DELIMITER の指定時に，どちらかを指定する必要があります。DEFAULT とALL が区切り文字として扱う文字を次に示します。

DEFAULT

ワード検索をする際，次の文字を区切り文字として扱います。

- 半角空白（0x20）
- タブ（0x09）
- 改行（0x0A）
- 復帰（0x0D）
- ピリオド（0x2E）
- 疑問符（0x3F）
- 感嘆符（0x21）

ALL

ワード検索をする際，次の文字を区切り文字として扱います。

- 半角空白（0x20）
- タブ（0x09）
- 改行（0x0A）
- 復帰（0x0D）
- ピリオド，疑問符，および感嘆符を含む 1 バイトの記号（0x21～0x2F，0x3A～0x40，0x5B～0x60，0x7B～0x7E）

記号を含む英語の文章を検索する場合，区切り文字によって検索結果が異なります。その点を考慮して，区切り文字の種類を選択してください。DEFAULT を指定した方がよいケースとALL を指定した方がよいケースを，それぞれ次に示します。

■DEFAULT を指定した方がよいケース

ワード検索で，記号を含めて検索を行いたい場合は，DEFAULT を指定することを推奨します。例えば，URL，メールアドレスなどを検索する場合が該当します。DEFAULT を指定すると，「taro@hitachi.com」で検索するときに，「taro@hitachi」が 1 つの単語として扱われます。そのため，「taro@hitachi」が含まれるデータだけを検索できます。

もし，ALL を指定した場合，「taro」と「hitachi」が別々の単語として扱われます。そのため，「xxxxx@taro.hitachi.com」や「taro.hitachi@com」のようなデータも一緒に検索されます。

■ALL を指定した方がよいケース

ワード検索で，記号を含めないで検索を行いたい場合は，ALL を指定することを推奨します。例えば，「high△speed」のような複数の単語（熟語）を検索する場合が該当します。ALL を指定すると，「high△speed」で検索するときに，「high△speed」だけでなく，「high-speed」，「high_speed」のような熟語もまとめて検索することができます。なお，「△」は半角空白を示します。

■留意事項

ワード検索の対象表に内部導出表が適用された場合，DELIMITER の指定が無効になることがあります。

- 導出表が展開される場合

  導出表が展開された結果の列に定義されているワード検索用のテキストインデクスのDELIMITER の指定が有効になります。

- 導出表が展開されない場合

  DELIMITER にALL を指定してもその指定は無効になります。DELIMITER にDEFAULT を指定したときに仮定される区切り文字で，ワード検索が実行されます。

なお，アーカイブマルチチャンク表を等価変換した内部導出表については，上記の規則は適用されません。

導出表の展開規則については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『内部導出表』を参照してください。

5.5　レンジインデクスの設計

ここでは，レンジインデクスを設計する際に検討する項目について説明します。

なお，レンジインデクスはCREATE INDEX 文で定義します。

5.5.1　レンジインデクスを定義する際の考慮点

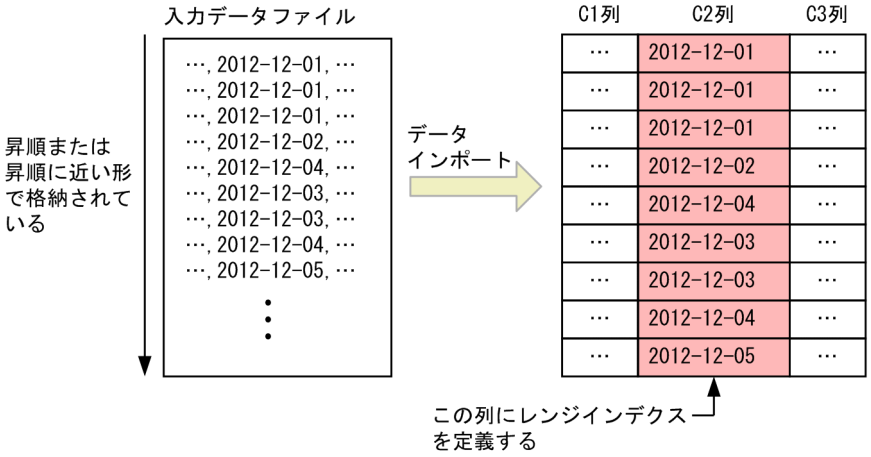
(1)　レンジインデクスを定義した方がよいケース

レンジインデクスを使用した検索では，レンジインデクスを定義した列の値域を基に検索データを絞り込みます。そのため，次に示すような場合にレンジインデクスを定義します。

- **入力データファイル中のデータが昇順（降順でもよい）に格納されている，またはそれに近い形で格納されている列がある場合**

  例えば，表にデータの取得日を格納する列があり，データの取得日が昇順または降順に近い形で入力データファイルに格納されている場合，その列にレンジインデクスを定義すると効果があります。例を次の図に示します。

  図 5-10　レンジインデクスを定義した方がよいケース（その 1）

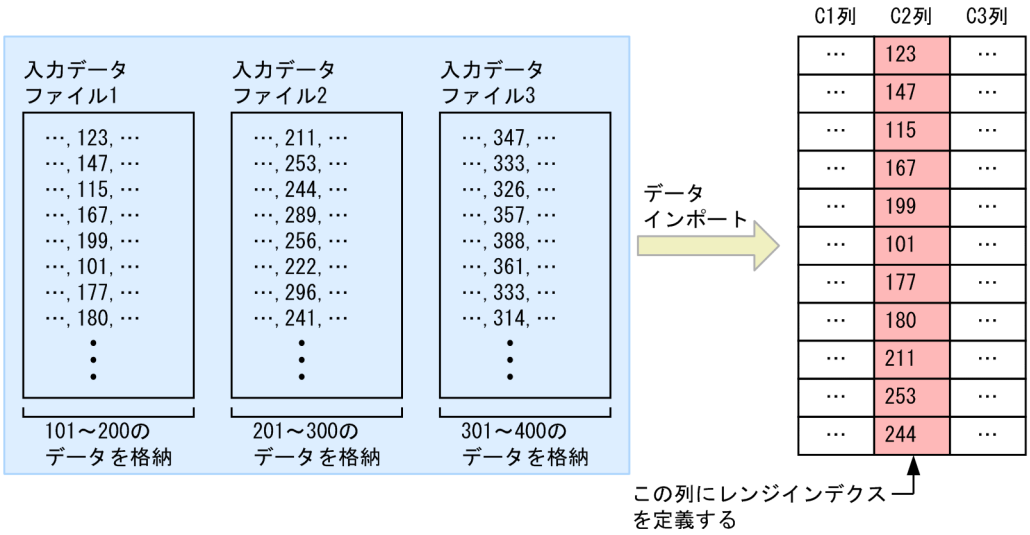


  表を検索する際，レンジインデクスを定義した列（上記の図のC2 列）を検索条件に指定します。ただし，指定した検索条件によっては，レンジインデクスが使用されないこともあります。レンジインデクスが使用される条件については，マニュアル『HADB　AP 開発ガイド』の『SQL 文の実行時に使用されるレンジインデクス』を参照してください。

- **入力データファイルごとに，ある決まった範囲のデータが格納されている列がある場合**

  例えば，表に商品番号を格納する列があり，入力データファイルごとに商品番号の値の範囲がある程度決まっている場合，その列にレンジインデクスを定義すると効果があります。例を次の図に示します。

図 5-11　レンジインデクスを定義した方がよいケース（その 2）

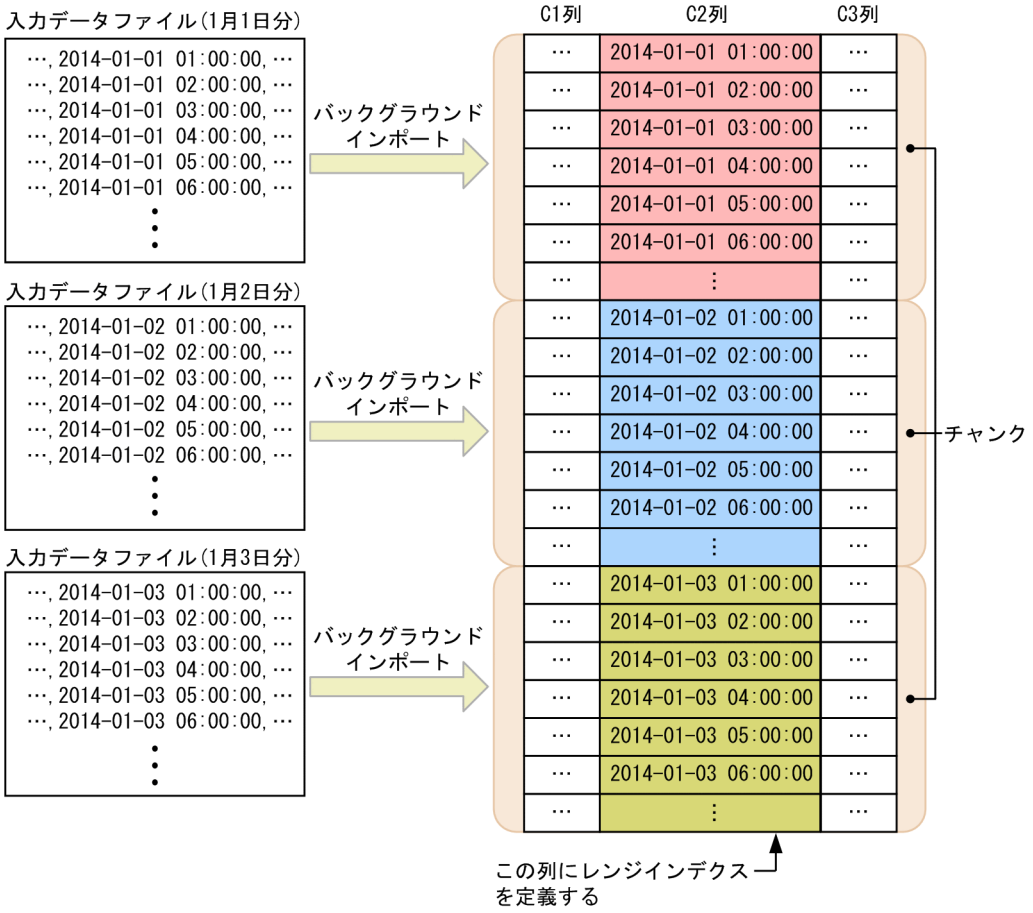


表を検索する際，レンジインデクスを定義した列（上記の図のC2 列）を検索条件に指定します。ただし，指定した検索条件によっては，レンジインデクスが使用されないこともあります。レンジインデクスが使用される条件については，マニュアル『HADB　AP 開発ガイド』の『SQL 文の実行時に使用されるレンジインデクス』を参照してください。

- **バックグラウンドインポートで格納するデータがチャンク間で重複しない場合**

  例えば，センサから発生したデータ（1 日分）を，毎日バックグラウンドインポートで表に格納するとします。バックグラウンドインポート対象の表に，データの発生日時を格納する列がある場合，その列にレンジインデクスを定義すると効果があります。例を次の図に示します。

図 5-12　レンジインデクスを定義した方がよいケース（その 3）



[説明]

1 つの入力データファイルには 1 日分のデータが格納されています。バックグラウンドインポートを実行した場合，C2 列に格納される「データの発生日時」のデータは，各チャンクで重複しません。

この場合，チャンクごとに値域が管理されているため，探索条件を満たさない値域のデータを格納しているチャンクの検索処理をスキップできます。そのため，効率良く検索できます。

表を検索する際，レンジインデクスを定義した列（上記の図のC2 列）を検索条件に指定します。ただし，指定した検索条件によっては，レンジインデクスが使用されないこともあります。レンジインデクスが使用される条件については，マニュアル『HADB　AP 開発ガイド』の『SQL 文の実行時に使用されるレンジインデクス』を参照してください。

(2)　レンジインデクスを定義しない方がよいケース

次に示すような場合は，各セグメントの値域が広がるため，レンジインデクスを定義しても，セグメントのスキップの効果が期待できません。

- 入力データファイル中のデータが，昇順または降順に近い形で並んでいる列がない場合
- 行の追加，更新，削除を頻繁に行う場合

なお，行を削除する場合は，セグメントの値域は広がりませんが，セグメント内に格納されているデータの値域に比べて，レンジインデクスの値域が広くなるため，レンジインデクスの効果が期待できません。

また，次に示すような場合は，各チャンクの値域が重複するため，レンジインデクスを定義しても，チャンクのスキップの効果が期待できません。

- バックグラウンドインポートを適用したデータインポートで格納するデータが，チャンク間で重複する場合

セグメントのスキップ，およびチャンクのスキップのどちらの効果も期待できない列には，レンジインデクスを定義しないようにしてください。検索性能が低下する原因になります。

(3) レンジインデクスを定義できない列

次に示す条件を満たす列にレンジインデクスは定義できません。

- 定義長が 33 バイト以上のCHAR 型の列
- VARCHAR 型の列
- BINARY 型の列
- VARBINARY 型の列

5.5.2 レンジインデクス専用のグローバルバッファのページ数の見積もり

レンジインデクスを格納する DB エリアに割り当てるグローバルバッファをサーバ定義のadbbuff オペランドで指定します。その際，adbbuff オペランドの-a オプションに指定するレンジインデクス専用のグローバルバッファのページ数を正しく見積もってください。見積もり方法については，「7.2.11 グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドの-a オプションの説明を参照してください。

5.5.3 レンジインデクスの一括先読み

HADB サーバの開始時にレンジインデクスをグローバルバッファ上に一括して読み込むことができます。これを**レンジインデクスの一括先読み**といいます。レンジインデクスの一括先読みをする場合は，サーバ定義のadb_sql_rngidx_preread オペランドを指定してください。

(1) レンジインデクスの一括先読みを行うメリット

HADB サーバの開始後，初回の表検索時，グローバルバッファ上にはレンジインデクスのデータがありません。そのため，グローバルバッファ上にレンジインデクスのデータを読み込むための I/O が必ず発生し，それが原因で表の検索性能が低下します。レンジインデクスの一括先読みを実行しておくと，この I/O をなくすことができます。

(2)　一括先読みしたレンジインデクスのデータの掃き出しを防ぐ方法

一括先読みしたレンジインデクスのデータも，グローバルバッファ上からの掃き出しの対象になります。一括先読みしたレンジインデクスのデータの掃き出しを防ぎたい場合は，次に示すことを実施してください。

- 掃き出しを防ぎたいレンジインデクスを専用のデータ用 DB エリアに格納する
- 掃き出しを防ぎたいレンジインデクスを格納するデータ用 DB エリアに割り当てるグローバルバッファのページ数を正しく見積もる（サーバ定義adbbuff オペランドの-a オプションの指定値を正しく見積もる）

-a オプションの見積もり式については，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドの-a オプションの説明を参照してください。

5.5.4　レンジインデクスを定義する場合の注意点（レンジインデクスの未完状態）

レンジインデクスを定義する場合，B-tree インデクスを定義するときと同様に，実表を作成後，データを格納する前にCREATE INDEX 文で定義してください。

行を格納するセグメントが割り当てられている状態の実表に対してレンジインデクスを定義した場合，定義されたレンジインデクスは**未完状態**になります。未完状態になると，定義したレンジインデクスは無効な状態となり，使用できなくなります。

未完状態を解除するには，レンジインデクスを再作成する必要があります。そのため，実表にデータを格納する前に，レンジインデクスをどの列に定義するかを事前によく検討してください。

未完状態のレンジインデクスに対して，次に示す操作を行うとエラーになります。

■エラーになる操作

- 未完状態のレンジインデクスを使用するSELECT 文の実行
- 未完状態のレンジインデクスが定義された表に対するINSERT 文，およびUPDATE 文の実行
- 未完状態のレンジインデクスが定義された実表に対する-d オプションを指定しないadbimport コマンドの実行
- 未完状態のレンジインデクスが定義された実表に対するadbmergechunk コマンドの実行
- 未完状態のレンジインデクスが定義されたアーカイブマルチチャンク表に対するadbunarchivechunk コマンドの実行

レンジインデクスの未完状態の解除方法については，「15.11.1　レンジインデクスが未完状態になったときの対処方法」を参照してください。

メモ

次に示すときなどは，行を格納するセグメントが割り当てられていない状態です。このときに，CREATE INDEX 文を実行すると，レンジインデクスは正常に作成されます。

- 実表の定義直後
- TRUNCATE TABLE 文の実行直後

例えば，DELETE 文で表の全行を削除しても，行を格納するセグメントは割り当てられている状態のままです。そのため，DELETE 文で表の全行を削除したあとにCREATE INDEX 文を実行した場合，そのレンジインデクスは未完状態になります。

5.6　データ用 DB エリアの設計

データ用 DB エリアはセグメントおよびページで構成されています。データの格納効率を上げるためには，ページサイズや 1 セグメントの容量を正しい指針に基づいて設計する必要があります。

また，ページサイズや 1 セグメントの容量が適切でないと，データへのアクセス効率が悪くなり，性能が低下します。

データ用 DB エリアを設計するときの考慮点を次に示します。

- どのデータ用 DB エリアにどの表（インデクス）をどれだけ格納するか
- 1 つのデータ用 DB エリアを何個のデータ用 DB エリアファイルで構成するか
- データ用 DB エリアにマルチチャンク表を格納するかどうか
- カラムストア表とローストア表の両方を同じデータ用 DB エリアに格納するかどうか
- 1 ページのサイズをどうするか
- 1 セグメントの容量をどうするか

各考慮点の詳細については，次項以降で説明します。

> **メモ**
>
> ページサイズや 1 セグメントの容量を設計してから，データ用 DB エリアの容量を見積もってください。データ用 DB エリアの容量見積もりについては，「5.8　データ用 DB エリアの容量見積もり」を参照してください。

5.6.1　データ用 DB エリアを設計する際の考慮点

データ用 DB エリアを設計する際の考慮点について説明します。

なお，データ用 DB エリアにマルチチャンク表を格納する場合は，「5.6.2　データ用 DB エリアにマルチチャンク表を格納する場合の考慮点」の説明もお読みください。

(1)　表とインデクスを格納するデータ用 DB エリアの考慮点

表とインデクスをデータ用 DB エリアに格納する際の考慮点を次に示します。

- データ量が多い表は，データ用 DB エリアにその表だけを格納することを推奨します。
- データ量が多いインデクスは，データ用 DB エリアにそのインデクスだけを格納することを推奨します。
- データ量が少ない（数メガバイト）表やインデクスの場合は，複数の表や複数のインデクスを 1 つのデータ用 DB エリアにまとめて格納しても問題ありません。

- カラムストア表を定義する場合，そのカラムストア表だけを格納する専用のデータ用 DB エリアを用意してください。

データ量が多い表やインデクスを 1 つのデータ用 DB エリアにまとめて格納した場合，次に示すような検索を実行すると，同一の DB エリアに対して I/O が集中し性能が低下するおそれがあります。

- 1 つのデータ用 DB エリアに複数の表を格納して，結合を行う SQL を使用した検索を実行した場合
- 1 つのデータ用 DB エリアに複数のインデクスを格納して，インデクスを利用する検索を実行した場合

また，次のような問題も発生するおそれがあります。

- データ用 DB エリアに掛かる排他によって，ある表に対する操作（SQL 文またはコマンド）が実行されている間は，同じデータ用 DB エリアに格納されている別の表に対する操作が実行できなくなる
- グローバルバッファを共有している場合，ある表への SQL 文の実行によって，同じデータ用 DB エリアに格納されている別の表に対する検索性能に影響を及ぼす

  特に，カラムストア表を格納するデータ用 DB エリアと，ローストア表やインデクスを格納するデータ用 DB エリアを同じグローバルバッファに割り当てると，ローストア表の検索性能や，インデクスを使用した検索性能が低下するおそれがあります。

(2) レンジインデクスを格納するデータ用 DB エリアの考慮点

レンジインデクス固有の考慮点を次に示します。

レンジインデクスを定義した表を格納するデータ用 DB エリアには，ほかの表またはインデクスを格納しないことを推奨します。ほかの表またはインデクスを格納すると，レンジインデクスの容量が増加することがあります。

(3) データ用 DB エリアを構成するデータ用 DB エリアファイルの考慮点

1 つのデータ用 DB エリアを，複数のデータ用 DB エリアファイルで構成すると，データ用 DB エリアファイルへのアクセスが並列化されるため，負荷を分散させることができます。そのため，アクセス頻度が高い表やインデクスを格納するデータ用 DB エリアは，複数のデータ用 DB エリアファイルで構成することを検討してください。1 つのデータ用 DB エリアを構成するデータ用 DB エリアファイル数の目安は，HADB サーバをインストールしたマシンのコア数の 20%～40%になります。

1 つのデータ用 DB エリアを，複数のデータ用 DB エリアファイルで構成した場合のメリットを次に示します。

- `adbimport` コマンドによるデータインポートを実行する場合

  データ用 DB エリアファイルごとに処理リアルスレッドを割り当てて，データの格納処理およびインデクスの作成処理を並列に行います。
- 割り当てられたセグメントが解放される場合

  データ用 DB エリアファイルごとに処理リアルスレッドを割り当てて，セグメントの解放処理を並列に行います。

- SQL パラレル実行機能を使用する場合

  データ用 DB エリアファイルごとに検索系 SQL の処理を並列に実行します。そのため，データ用 DB エリアファイル数が多い方が処理の並列度が上がります（ただし，マルチノード構成の全ノード数が上限となります）。SQL パラレル実行機能については，「2.19.8　SQL パラレル実行機能」を参照してください。

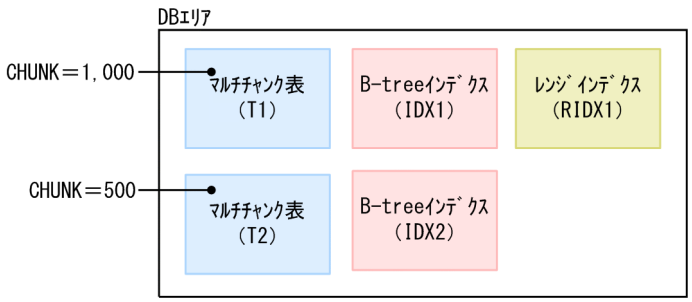
5.6.2　データ用 DB エリアにマルチチャンク表を格納する場合の考慮点

マルチチャンク表とマルチチャンク表に定義したインデクスを 1 つのデータ用 DB エリアにまとめて格納する場合，次のことを考慮してください。

- 1 つのデータ用 DB エリアが管理できるチャンク数の上限は 30,000 個です。
- 表のチャンク数だけではなく，インデクスのチャンク数も考慮してください。

例

1 つのデータ用 DB エリアに，次のようにマルチチャンク表とインデクスを格納したとします。



- マルチチャンク表T1 に，B-tree インデクスIDX1 とレンジインデクスRIDX1 を定義しています。マルチチャンク表T1 に指定したチャンク数の最大値を 1,000 とします。
- マルチチャンク表T2 に，B-tree インデクスIDX2 を定義しています。マルチチャンク表T2 に指定したチャンク数の最大値を 500 とします。

この場合，T1，IDX1 とRIDX1 で作成されるチャンク数の最大値は 3,000 個となります。また，T2 とIDX2 で作成されるチャンク数の最大値は 1,000 個となります。

```
1,000（T1）＋1,000（IDX1）＋1,000（RIDX1）＝3,000（個）
500（T2）＋500（IDX2）＝1,000（個）
```

このため，1 つのデータ用 DB エリアに作成されるチャンク数の最大値は 4,000 個となります。

この合計値が 30,000 個を超えることはできません。

1 つのデータ用 DB エリアに複数の表やインデクスを格納した場合，バックグラウンドインポートを実行すると，格納した表やインデクスの分だけチャンクが作成されます。複数の表やインデクスを格納する場合，1 つのデータ用 DB エリアが管理できるチャンク数の上限を超えないようにしてください。

> **メモ**
>
> チャンク数の最大値の求め方については，「5.2.4　マルチチャンク表を定義する場合の考慮点」を参照してください。

■アーカイブマルチチャンク表の場合

アーカイブマルチチャンク表の場合は，次に示すことにも考慮してください。

- **アーカイブレンジ列に自動的に定義されるレンジインデクスのチャンク数**

  アーカイブレンジ列に自動的に定義されるレンジインデクスについてもチャンクが作成されます。そのため，このレンジインデクスのチャンク数についても，計算する必要があります。

- **ロケーション表のチャンク数**

  ロケーション表は，アーカイブマルチチャンク表と同じデータ用 DB エリアに格納されます。このロケーション表についてもチャンクが作成されます。そのため，ロケーション表のチャンク数についても，計算する必要があります。ロケーション表のチャンク数は，次に示す計算式で求められます。

  計算式

  ```
  ロケーション表のチャンク数＝
          データ用DBエリアに格納されているアーカイブマルチチャンク表の数×10
  ```

5.6.3　データ用 DB エリアのページサイズを決める際の考慮点

データ用 DB エリアのページサイズを決めるときの考慮点を説明します。

データ用 DB エリアのページサイズの指定方法を次に示します。

- adbinit コマンドの初期設定オプションadbinitdbarea の-p オプションにキロバイト単位で指定する
- adbmodarea コマンドの DB エリア追加・変更オプションadbaddarea の-p オプションにキロバイト単位で指定する

(1)　表を格納するデータ用 DB エリアの場合

表を格納するデータ用 DB エリアのページサイズを決めるときの考慮点を次に示します。

- ページサイズは，基本行を格納できる長さが必要になります。1 つのデータ用 DB エリアに複数の表を格納する場合は，表の中で最大の基本行を格納できる長さにしてください。基本行を格納できる長さがないと，表定義時にエラーとなります。基本行の長さの求め方については，「5.8.2　行の種別ごとの格納ページ数の求め方」を参照してください。
- 1 ページ中に格納できる最大行数は 255 行です。そのため，行長の合計に比べてページサイズが大き過ぎると，使用できないむだな領域が発生し，データの格納効率が悪くなります。例えば，行長が 100 バイトの時に，1 ページに格納できるデータは，最大「100 バイト×255 行＝ 25,500 バイト」となり

ます。このとき，ページサイズを 32,768 バイト（32 キロバイト）にすると，「32,768 バイト－25,500 バイト－*管理領域*≒約 7,000 バイト」の領域が使用できないため，むだな領域になります。

- カラムストア表を格納するデータ用 DB エリアのページサイズには，32,768 バイト（32 キロバイト）を指定することを推奨します。それによって圧縮率の向上が見込まれます。
- 配列型の列が定義された表は，ページサイズが 4,096 バイト（4 キロバイト）のデータ用 DB エリアには格納できません。

(2) B-tree インデクスを格納するデータ用 DB エリアの場合

ページサイズには，8 キロバイトまたは 16 キロバイトを指定してください。

(3) テキストインデクスを格納するデータ用 DB エリアの場合

ページサイズには，8 キロバイトを指定してください。

(4) レンジインデクスを格納するデータ用 DB エリアの場合

ページサイズには，最大値である 32 キロバイトを指定してください。

5.6.4 データ用 DB エリアの 1 セグメントの容量を決める際の考慮点

データ用 DB エリアを作成する際に，データ用 DB エリアの 1 セグメントの容量を指定します。ここでは，1 セグメントの容量の推奨値，および 1 セグメントの容量を決める際の考慮点を説明します。

なお，データ用 DB エリアの 1 セグメントの容量は，次の方法で指定します。

- adbinit コマンドの初期設定オプションadbinitdbarea の-s オプションにメガバイト単位で指定する
- adbmodarea コマンドの DB エリア追加・変更オプションadbaddarea の-s オプションにメガバイト単位で指定する

(1) 表を格納するデータ用 DB エリアの場合

表を格納するデータ用 DB エリアの 1 セグメントの容量の推奨値と考慮点を説明します。

重要

1 つのデータ用 DB エリアに 1 つの表を格納するのが基本ですが，1 つのデータ用 DB エリアに複数の表を格納する場合は，1 セグメントの容量の推奨値が同じ表を格納することを推奨します。

なお，1 つのデータ用 DB エリアに 1 セグメントの容量の推奨値が異なる表を格納する場合は，格納する表の中でいちばん大きい推奨値を 1 セグメントの容量に指定してください。

■ローストア表を格納するデータ用 DB エリアの場合

1 セグメントの容量には 4 メガバイトを指定してください。

■カラムストア表を格納するデータ用 DB エリアの場合

- 1 セグメントの容量を大きくすると，辞書圧縮方式の圧縮率が向上することがあります。1 セグメントの容量の推奨値を求める計算式を次に示します。

  計算式

$$\sum_{k=1}^{col_num} elm_num(k) \times COLDATASIZE(k)$$

  変数の説明

  col_num

  表の列数（単位：個）

  elm_num(k)

  表の k 番目に定義した列の配列要素数の平均値（単位：個）

  配列型以外の列の場合は，1 としてください。

  COLDATASIZE(k)

  表の k 番目に定義した列のデータ長の平均値（単位：バイト）

  列のデータ長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(3)　配列型の列を定義した表の行長（変数 ARRAY_ROWSZ）の求め方」の変数 *COLDATASIZE(k)* を参照してください。

  1 セグメントの容量の推奨値

  1 セグメントの容量の推奨値は，計算結果によって異なります。

  - **計算結果が 640 を超える場合**

    16 メガバイトを指定することを推奨します。

  - **計算結果が 1,280 を超える場合**

    32 メガバイトを指定することを推奨します。

  - **計算結果が上記以外の場合**

    4 メガバイトを指定してください。

- 1 セグメントの容量が小さいデータ用 DB エリアに配列型の列を定義した表を格納する場合，表の定義時または表への列の追加時にコマンドがエラーになります。このとき，KFAA31737-E メッセージが出力されます。このメッセージの対処方法に従って 1 セグメントの容量を大きくしてください。配列型の列を定義した表を格納するデータ用 DB エリアの 1 セグメントの容量の条件については，「5.2.13　配列型の列の定義【カラムストア表】」の「■データ用 DB エリアに関する留意事項」を参照してください。

  なお，1 セグメントの容量が最大値である 32 メガバイトの場合にKFAA31737-E メッセージが出力されたときは，次のどれかの対処をしてください。

  - 表に定義する列数を減らす

- 表に定義する配列型の最大要素数を減らす
- 列定義の圧縮方式指定に，NONE（データ圧縮なし），RUNLENGTH（ランレングス圧縮方式），DELTA（差分圧縮方式），またはDELTA_RUNLENGTH（差分ランレングス圧縮方式）を指定する

- 表に格納するデータ量が少ないときに 1 セグメントの容量が大きいと，セグメントの格納効率や検索性能に影響を及ぼします。1 スレッドの平均格納量，または 1 スレッドの平均検索量が 1 セグメントの容量よりも小さいときは，1 セグメントの容量を小さくすることを検討してください。1 スレッドの平均格納量，および 1 スレッドの平均検索量の計算式を次に示します。

  計算式

  ・1スレッドの平均格納量（単位：バイト） ＝ $\frac{\text{import_csv_size} \times 0.1}{\text{import_rthd_num}}$

  ・1スレッドの平均検索量（単位：バイト） ＝ $\frac{\text{table_csv_size} \times 0.1}{\text{sql_rthd_num} \times \text{uthd_num}}$

  変数の説明

  import_csv_size

  1 回のインポートで格納するデータの総量（単位：バイト）

  import_rthd_num

  adbimport コマンドが使用する処理リアルスレッド数

  詳細については，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

  table_csv_size

  表に格納するデータの総量（単位：バイト）

  sql_rthd_num

  サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値

  uthd_num

  サーバ定義adb_sys_uthd_num オペランドの指定値

- adbimport コマンドを実行すると，セグメントの格納効率を確認できます。1 回のインポートに使用するデータでadbimport コマンドを実行し，KFAA82004-I メッセージに出力される次の項目を確認してください。
  - データが格納されたセグメント数
  - データが格納されたセグメント中の未使用ページ数

  これらの値でセグメントの空比率を計算してください。セグメントの空比率が 20%を超える場合は，1 セグメントの容量を小さくすることを検討してください。

  セグメントの空比率の計算式を次に示します。

  計算式（単位：%）

$$\text{セグメントの空比率} = \frac{\text{データが格納されたセグメント中の未使用ページ数} \times \text{page_size}}{\text{データが格納されたセグメント数} \times \text{1セグメントの容量（単位：バイト）}} \times 100$$

変数の説明

page_size

表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

メモ

1 セグメントの大きさを表す用語には，次の 2 つがあります。

- 1 セグメントの容量
  1 セグメントの大きさをメガバイト単位で表すときの用語です。
- セグメントサイズ
  1 セグメントの大きさをページ数単位で表すときの用語です。
  セグメントサイズは，「*1 セグメントの容量÷ページサイズ*」で求められます。

(2) インデクスを格納するデータ用 DB エリアの場合

1 セグメントの容量には，4 メガバイトを指定してください。

5.7　作業表用 DB エリアの設計

作業表用 DB エリアは，作業表を格納するために使用します。作業表は，作業表を作成する SQL 文を実行すると作成されます。

作業表用 DB エリアのページサイズは，adbinit コマンドの初期設定オプションのadb_init_wrk_page_size オペランドに，キロバイト単位で指定します。通常は，作業表用 DB エリアのページサイズには 256 キロバイトを指定してください。

なお，作業表用 DB エリアのページサイズが小さ過ぎる場合，行長が長い作業表を格納できません。実行する SQL 文によっては，エラーとなるおそれがあります。例えば，作業表の構成列に定義長が大きい列を含むために，作業表の行長が長くなる SQL 文を実行する場合などが該当します。そのため，作成される作業表の行長を考慮して，作業表用 DB エリアのページサイズを指定してください。作業表の行長の求め方については，「5.9.2　作業表を格納するために必要な基本行用ページ数の求め方」を参照してください。

作業表用 DB エリアのページサイズを変更するには，サーバ定義のadb_dbarea_wrk_page_size オペランドに，変更後のページサイズをキロバイト単位で指定してください。その後，HADB サーバを再起動してください。

また，作業表を作成する SQL 文については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『SQL を実行した場合に作成される作業表について』を参照してください。

5.8　データ用 DB エリアの容量見積もり

データ用 DB エリアの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

データ用DBエリアの容量 ＝ *DATA_PAGE_NUM* ×*page_size* ÷1,024

変数の説明

DATA_PAGE_NUM：データ用 DB エリアの総ページ数

page_size：データ用 DB エリアのページサイズ（単位：バイト）

DATA_PAGE_NUM については，「5.8.1　データ用 DB エリアの総ページ数の求め方」を参照してください。

5.8.1　データ用 DB エリアの総ページ数の求め方

データ用 DB エリアの総ページ数（*DATA_PAGE_NUM*）は，次に示す計算式から求めてください。

(1)　計算式

データ用 DB エリアの総ページ数（*DATA_PAGE_NUM*）の計算式を次に示します。

計算式

$$DATA_PAGE_NUM = 1 + dbarea_file_num \times \left(21 + \left\uparrow \frac{32,000}{\left\downarrow \frac{page_size}{160} \right\downarrow} \right\uparrow \right) + \left\uparrow \frac{SGDATA}{SEGBF} \right\uparrow + \left\uparrow \frac{SGDATA}{\left\downarrow \frac{\left\downarrow \frac{page_size \times 125}{16} \right\downarrow}{SEGBF} \right\downarrow \times SEGBF} \right\uparrow + SGDATA \times SEGSIZE$$

(2)　変数の説明

データ用 DB エリアの総ページ数（*DATA_PAGE_NUM*）を求める際に使用する変数を次に示します。

変数の説明

dbarea_file_num

データ用 DB エリアファイルの数（単位：個）

page_size

ページサイズ（単位：バイト）

SEGBF

セグメントブロック化係数

セグメントブロック化係数は，データ用 DB エリアのページサイズおよび 1 セグメントの容量によって決まります。次の表から求めてください。

表 5-5　データ用 DB エリアのページサイズおよび 1 セグメントの容量ごとのセグメントブロック化係数

項番	ページサイズ（単位：キロバイト）	1 セグメントの容量（単位：メガバイト）	セグメントブロック化係数
1	4	4	12
2	8		42
3	16		127
4	32		341
5		16	170
6		32	102

SEGSIZE

セグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

SEGSIZE＝ 1セグメントの容量（単位：バイト）÷*page_size*

SGDATA

データ用 DB エリアのセグメント数

次に示す計算式から求めてください。

```
SGDATA= SGROWTBL＋SGCOLUMNTBL＋SGIDX＋SGRIX＋SGTIX
```

なお，変数 *SGROWTBL*，*SGCOLUMNTBL*，*SGIDX*，*SGRIX*，および *SGTIX* は，シングルチャンク表とマルチチャンク表で求め方が異なります。また，1 つのデータ用 DB エリア内に，シングルチャンク表とマルチチャンク表の両方を格納する場合，それぞれの変数 *SGROWTBL*，*SGCOLUMNTBL*，*SGIDX*，*SGRIX*，および *SGTIX* を求めて，すべて合計する必要があります。変数 *SGROWTBL*，*SGCOLUMNTBL*，*SGIDX*，*SGRIX*，および *SGTIX* の求め方を次に示します。

■シングルチャンク表の場合

SGROWTBL（シングルチャンク表の場合）

データ用 DB エリア内のローストア表のデータページを管理するセグメント数

「(a)　変数 SGROWTBL の求め方（シングルチャンク表の場合）」を参照してください。

SGCOLUMNTBL（シングルチャンク表の場合）

データ用 DB エリア内のカラムストア表のデータページを管理するセグメント数

「(b)　変数 SGCOLUMNTBL の求め方（シングルチャンク表の場合）」を参照してください。

SGIDX（シングルチャンク表の場合）

データ用 DB エリア内の B-tree インデクスページを管理するセグメント数

「(c)　変数 SGIDX の求め方（シングルチャンク表の場合）」を参照してください。

SGRIX（シングルチャンク表の場合）

データ用 DB エリア内のレンジインデクスページを管理するセグメント数

「(d)　変数 SGRIX の求め方（シングルチャンク表の場合）」を参照してください。

SGTIX（シングルチャンク表の場合）

データ用 DB エリア内のテキストインデクスページを管理するセグメント数

「(e)　変数 SGTIX の求め方（シングルチャンク表の場合）」を参照してください。

■マルチチャンク表の場合

SGROWTBL（マルチチャンク表の場合）

データ用 DB エリア内のローストア表のデータページを管理するセグメント数

「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」を参照してください。

SGCOLUMNTBL（マルチチャンク表の場合）

データ用 DB エリア内のカラムストア表のデータページを管理するセグメント数

「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」を参照してください。

SGIDX（マルチチャンク表の場合）

データ用 DB エリア内の B-tree インデクスページを管理するセグメント数

「(h)　変数 SGIDX の求め方（マルチチャンク表の場合）」を参照してください。

SGRIX（マルチチャンク表の場合）

データ用 DB エリア内のレンジインデクスページを管理するセグメント数

「(i)　変数 SGRIX の求め方（マルチチャンク表の場合）」を参照してください。

SGTIX（マルチチャンク表の場合）

データ用 DB エリア内のテキストインデクスページを管理するセグメント数

「(j)　変数 SGTIX の求め方（マルチチャンク表の場合）」を参照してください。

(a)　変数 SGROWTBL の求め方（シングルチャンク表の場合）

次に示す計算式から求めてください。

$$SGROWTBL = \sum_{i=1}^{row_tbl_num_in_dbarea} \left(\left\uparrow \frac{BP(i)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{VP(i)}{SEGSIZE} \right\uparrow \right)$$

row_tbl_num_in_dbarea

データ用 DB エリア中のすべての表数（単位：個）

BP（i）

各表を格納するために必要な基本行用ページ数

「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」を参照してください。

VP（i）

各表を格納するために必要な分岐行用ページ数

「5.8.2　行の種別ごとの格納ページ数の求め方」の「(2)　分岐行用ページ数（変数 VP（i））の求め方」を参照してください。

(b)　変数 SGCOLUMNTBL の求め方（シングルチャンク表の場合）

カラムストア表にはデータが圧縮されて格納されるため，ローストア表のようにレコード長から順に見積もり式を計算していくことができません。そのため，実際に少量のデータを格納し，その圧縮率を基にセグメント数を見積もります。

次に示す計算式から求めてください。

$$SGCOLUMNTBL = \sum_{i=1}^{column_tbl_num_in_dbarea} \left(\left\uparrow \frac{COLUMNTBLSIZE(i)}{4} \right\uparrow + \left\uparrow \frac{ADDTBLPGNUM(i) + DELTBLPGNUM(i)}{SEGSIZE - 1} \right\uparrow \right)$$

column_tbl_num_in_dbarea

データ用 DB エリア中のすべてのカラムストア表の数（単位：個）

SEGSIZE

セグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

SEGSIZE＝ 1セグメントの容量（単位：バイト）÷*page_size*

COLUMNTBLSIZE（i）

i 番目のカラムストア表のデータサイズ（単位：メガバイト）

次に示す計算式から求めてください。

$$COLUMNTBLSIZE(i) = IMPORTDATASIZE(i) \times COMPRESSION_RATE + COLUMNIZESIZE(i)$$

IMPORTDATASIZE（i）

i 番目のカラムストア表に格納する元データ（CSV ファイル）のサイズ（単位：メガバイト）

COMPRESSION_RATE

元データの圧縮率

i 番目のカラムストア表に格納する元データの一部をadbimport コマンドで格納してください。そのあと，次に示す計算式から求めてください。

$$COMPRESSION_RATE = \frac{testdbsize}{testcsvsize}$$

testdbsize

元データの一部をadbimport コマンドでインポートした際の，データベースの使用量（単位：メガバイト）

計算式を求めるためにインポートする元データの一部には，100～500 メガバイト程度の実データに近いデータを用意することを推奨します。adbimport コマンドを実行するときは，インポートオプションadb_import_rthd_num に2 を指定してください。

adbimport コマンド実行後に，adbdbstatus コマンドで，**表のサマリ情報**に含まれるUsed_segments（表の使用中セグメント数）をメガバイト単位で出力してください。そして，データベースの使用量を確認してください。

adbimport コマンドおよびadbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

testcsvsize

adbimport コマンドでインポートした元データの一部の CSV ファイルのサイズ（単位：メガバイト）

COLUMNIZESIZE（i）

i 番目のカラムストア表に，更新行のカラム化機能のメンテナンス処理で格納されるカラムストア形式のデータのサイズ（単位：メガバイト）

変数*COLUMNIZESIZE（i）*は更新行のカラム化機能を有効にしている場合にだけ加算してください。なお，i 番目のカラムストア表に B-tree インデクスを定義している場合は，0 として見積もってください。

計算式を次に示します。

$$COLUMNIZESIZE(i) = \left\uparrow IMPORTDATASIZE(i) \times COMPRESSION_RATE \times \frac{insert_row_num(i)}{import_row_num(i)} \right\uparrow$$

IMPORTDATASIZE（i）

i 番目のカラムストア表に格納する元データ（CSV ファイル）のサイズ（単位：メガバイト）

COMPRESSION_RATE

元データの圧縮率

上記にある変数*COMPRESSION_RATE* を参照して求めてください。

insert_row_num（i）

i 番目のカラムストア表で，INSERT 文またはUPDATE 文によって，追加または更新される行数

import_row_num（i）

i 番目のカラムストア表に格納する元データ（CSV ファイル）の行数

ADDTBLPGNUM（i）

i 番目のカラムストア表にINSERT 文またはUPDATE 文で追加するデータのページ数

次に示す計算式から求めてください。

$$ADDTBLPGNUM(i) = BP(i) + VP(i)$$

BP（i）

i 番目のカラムストア表にINSERT 文またはUPDATE 文で追加するデータの基本行用ページ数

i 番目のカラムストア表にINSERT 文またはUPDATE 文で追加する行数を基に，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」を参照して求めてください。

VP（i）

i 番目のカラムストア表にINSERT 文またはUPDATE 文で追加するデータの分岐行用ページ数

i 番目のカラムストア表にINSERT 文またはUPDATE 文で追加する行数を基に，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(2)　分岐行用ページ数（変数 VP（i））の求め方」を参照して求めてください。

DELTBLPGNUM（i）

i 番目のカラムストア表にUPDATE 文またはDELETE 文で無効情報管理用ページに追加するデータのページ数

次に示す計算式から求めてください。

$$DELTBLPGNUM(i) = DELLEAFPGNUM(i) + DELUPPERPGNUM(i)$$

DELLEAFPGNUM（i）

次に示す計算式から求めてください。

$$DELLEAFPGNUM(i) = \left\uparrow \frac{12 \times (update_row_num + delete_row_num)}{0.5 \times (page_size - 80)} \right\uparrow$$

update_row_num

i 番目のカラムストア表に対してUPDATE 文で更新する行数

delete_row_num

i 番目のカラムストア表に対してDELETE 文で削除する行数

page_size

i 番目のカラムストア表が定義されている DB エリアのページサイズ（単位：バイト）

DELUPPERPGNUM（i）

次に示す計算式から求めてください。

$$DELUPPERPGNUM(i) = \left\uparrow \frac{12 \times DELLEAFPGNUM(i)}{0.5 \times (page_size - 80)} \right\uparrow$$

また，変数*SGCOLUMNTBL* については，すべての列で圧縮方式に**無圧縮（NONE）**が選択された場合を仮定し，そのサイズを求めることで見積もり値を求めることもできます。ただし，選択された圧縮方式や圧縮率によっては，実際に格納した場合のサイズに比べて見積もり値が大きくなることがあります。

すべての列で圧縮方式に**無圧縮（NONE）**が選択された場合の見積もり値は，次に示す計算式から求めてください。

$$SGCOLUMNTBL = \sum_{i=1}^{column_tbl_num_in_dbarea} (COLUMNDATASEGNUM(i) + ROWDATASEGNUM(i) + COLUMNIZESEGNUM(i))$$

COLUMNDATASEGNUM（i）

i 番目のカラムストア表のカラムデータ用セグメント数

次に示す計算式から求めてください。

$$COLUMNDATASEGNUM(i) = \frac{row_num}{MIN\left\{262,144 \ , \ \left(\left\downarrow \frac{page_size-80}{COL_MAX_SIZE \times \uparrow column_num \div COL_PAGE_NUM \uparrow}\right\downarrow\right) \times MAX\left(1, \left\downarrow \frac{COL_PAGE_NUM}{column_num}\right\downarrow\right)\right\}}$$

row_num

i 番目のカラムストア表に格納する行数

column_num

i 番目のカラムストア表の列数

page_size

データ用 DB エリアのページサイズ（単位：バイト）

COL_PAGE_NUM

変数*COL_PAGE_NUM* に代入する値は，データ用 DB エリアのページサイズおよび 1 セグメントの容量によって決まります。次の表のデータ用 DB エリアのページサイズおよび 1 セグメントの容量に対応する値を代入してください。

表 5-6　変数 COL_PAGE_NUM に代入する値

項番	ページサイズ （単位：キロバイト）	1 セグメントの容量 （単位：メガバイト）	*COL_PAGE_NUM* の値
1	4	4	1,021
2	8		510
3	16		255
4	32		127
5		16	511
6		32	1,023

COL_MAX_SIZE

i 番目のカラムストア表の列データの最大データ長

次に示す計算式から求めてください。

$$COL_MAX_SIZE = \max_{k=1,\cdots,column_num} COLUMNDATASIZE(k) \times elm_num(k)$$

例えば，i 番目のカラムストア表にC1 列，C2 列，C3 列が定義されている場合，次の計算をしてください。計算結果の最大値が *COL_MAX_SIZE* の値になります。

- *COLUMNDATASIZE(1)×elm_num(1)*
- *COLUMNDATASIZE(2)×elm_num(2)*
- *COLUMNDATASIZE(3)×elm_num(3)*

COLUMNDATASIZE（k）

k 列目の列データのデータ長（単位：バイト）

変数 *COLUMNDATASIZE* に代入する値は，列データのデータ型によって異なります。次の表で該当する値を代入してください。

表 5-7　各列データのデータ型のデータ長の一覧

項番	分類	データ型		データ長（単位：バイト）
1	数データ	INTEGER		10
2		SMALLINT		6
3		DECIMAL(*m*,*n*)※ NUMERIC(*m*,*n*)※	*m*≦4	4
4			5≦*m*≦8	6
5			9≦*m*≦16	10
6			17≦*m*	18
7		DOUBLE PRECISION		10
8		FLOAT		
9	文字データ	CHARACTER(*n*)	*n*≦127	$\uparrow\frac{n+1}{2}\uparrow \times 2$
10			128≦*n*	10
11		VARCHAR(*n*)	*n*≦127	$\uparrow\frac{n+1}{2}\uparrow \times 2$
12			128≦*n*≦255	128
13			256≦*n*	130
14	日時データ	DATE		6
15		TIME(*p*)		$\uparrow\frac{3+\uparrow p\div 2\uparrow}{2}\uparrow \times 2$

項番	分類	データ型		データ長（単位：バイト）
16		TIMESTAMP(*p*)		$\uparrow \frac{7 + \uparrow p \div 2 \uparrow}{2} \uparrow \times 2$
17	バイナリデータ	BINARY(*n*)	*n*≦127	$\uparrow \frac{n+1}{2} \uparrow \times 2$
18			128≦*n*	10
19		VARBINARY(*n*)	*n*≦127	$\uparrow \frac{n+1}{2} \uparrow \times 2$
20			128≦*n*≦255	128
21			256≦*n*	130

（凡例）

m，*n*：正の整数

p：0，3，6，9 または 12

注※

全体の桁数が *m* 桁で，小数点以下の桁数が *n* 桁の固定小数点数を示します。*m* を省略した場合は 38 が仮定されます。

elm_num（k）

k 列目の配列要素数の平均値（単位：個）

k 列目の列が配列型以外の列の場合は，1 としてください。

ROWDATASEGNUM（i）

i 番目のカラムストア表のローデータ用セグメント数

次に示す計算式から求めてください。

$$ROWDATASEGNUM(i) = \left\{ MAX\left(\uparrow \frac{row_num \times var_num}{255} \uparrow , \uparrow \frac{row_num \times (14 + var_size) \times var_num}{page_size - 80} \uparrow \right) + ADDTBLPGNUM(i) + DELTBLPGNUM(i) \right\} \div SEGSIZE$$

row_num

i 番目のカラムストア表に格納する行数

page_size

データ用 DB エリアのページサイズ（単位：バイト）

var_num

分岐行として管理される列の列数（単位：個）

CHAR 型，またはBINARY 型の場合，定義長が 128 バイト以上の列の列数を求めてください。

VARCHAR 型，またはVARBINARY 型の場合，実長が 128 バイト以上のデータを含む列の列数を求めてください。

var_size

分岐行の列のデータ長（単位：バイト）

変数*var_size* に代入する値は，データ型によって異なります。次の表で該当する値を代入してください。

表 5-8　変数 var_size に代入する値

項番	分類	データ型	データ長（単位：バイト）
1	文字データ	CHAR	定義長
2		VARCHAR	*d*
3	バイナリデータ	BINARY	定義長
4		VARBINARY	*d*

（凡例）

d：実際のデータ長

COLUMNIZESEGNUM（i）

i 番目のカラムストア表に，更新行のカラム化機能のメンテナンス処理で格納されるカラムデータ用セグメント数

変数*COLUMNIZESEGNUM（i）*は，更新行のカラム化機能を有効にしている場合にだけ加算してください。なお，i 番目のカラムストア表に B-tree インデクスを定義している場合は，0 として見積もってください。

計算式を次に示します。

COLUMNIZESEGNUM(i)＝ COLUMNDATASEGNUM(i)

変数*COLUMNDATASEGNUM（i）*の求め方については，上記にある変数*COLUMNDATASEGNUM（i）*を参照してください。その際，変数*row_num* には i 番目のカラムストア表で，INSERT 文またはUPDATE 文によって，追加または更新される行数を代入してください。

(c)　変数 SGIDX の求め方（シングルチャンク表の場合）

次に示す計算式から求めてください。

$$SGIDX = \sum_{i=1}^{idx_num_in_dbarea} \left(\left\uparrow \frac{IP_LOWER(i)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{IP_UPPER(i)}{SEGSIZE} \right\uparrow \right)$$

idx_num_in_dbarea

データ用 DB エリア中のすべての B-tree インデクス数（単位：個）

IP_LOWER(i)

各 B-tree インデクスの下位ページ用のセグメントで使用するページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」を参照してください。

IP_UPPER(i)

各 B-tree インデクスの上位ページ用のセグメントで使用するページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」を参照してください。

(d)　変数 SGRIX の求め方（シングルチャンク表の場合）

次に示す計算式から求めてください。

$$SGRIX = \sum_{i=1}^{rng_num_in_dbarea} RS(i)$$

rng_num_in_dbarea

データ用 DB エリア中のレンジインデクス数（単位：個）

RS（i）

各レンジインデクスを格納するために必要なセグメント数

「5.8.6　レンジインデクスを格納するセグメント数の求め方」を参照してください。

(e)　変数 SGTIX の求め方（シングルチャンク表の場合）

次に示す計算式から求めてください。

$$SGTIX = \sum_{i=1}^{idx_num_in_dbarea} \left(\left\uparrow \frac{TIP_STRSEG(i)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{TIP_APPSEG(i)}{SEGSIZE} \right\uparrow \right)$$

idx_num_in_dbarea

データ用 DB エリア中のすべてのテキストインデクス数（単位：個）

TIP_STRSEG(i)

各テキストインデクスの文字列管理部用のセグメントで使用するページ数

「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」の「(1)　文字列管理部用のセグメントで使用する格納ページ数（変数 TIP_STRSEG（i））の求め方」を参照してください。

TIP_APPSEG(i)

各テキストインデクスの出現情報管理部用のセグメントで使用するページ数

「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」の「(2)　出現情報管理部用のセグメントで使用する格納ページ数（変数 TIP_APPSEG（i））の求め方」を参照してください。

(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）

次に示す計算式から求めてください。

$$SGROWTBL = \sum_{i=1}^{row_tbl_num_in_dbarea} \left\{ \sum_{k=1}^{chunk_num} \left(\left\uparrow \frac{CHBP(i,k)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{CHVP(i,k)}{SEGSIZE} \right\uparrow \right) + SGLTBL(i) + SGLIDX(i) \right\}$$

row_tbl_num_in_dbarea

データ用 DB エリア中のすべてのローストア表数（単位：個）

chunk_num

作成されるアーカイブ状態ではないチャンク数（単位：個）

CHBP（i，k）

各チャンクの表を格納するために必要な基本行用ページ数

対象のチャンクに-b オプションを指定したadbimport コマンド，INSERT 文，およびUPDATE 文で格納するデータ量から見積もった値を代入してください。

CHVP（i，k）

各チャンクの表を格納するために必要な分岐行用ページ数

対象のチャンクに-b オプションを指定したadbimport コマンド，INSERT 文，およびUPDATE 文で格納するデータ量から見積もった値を代入してください。

SGLTBL(i)

i 番目のアーカイブマルチチャンク表に対応するロケーション表を格納するために必要なセグメント数

アーカイブマルチチャンク表の場合に求める必要があります。

次に示す計算式から求めてください。

計算式

$$SGLTBL(i) = \left\uparrow \frac{LBP(i)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{LVP(i)}{SEGSIZE} \right\uparrow$$

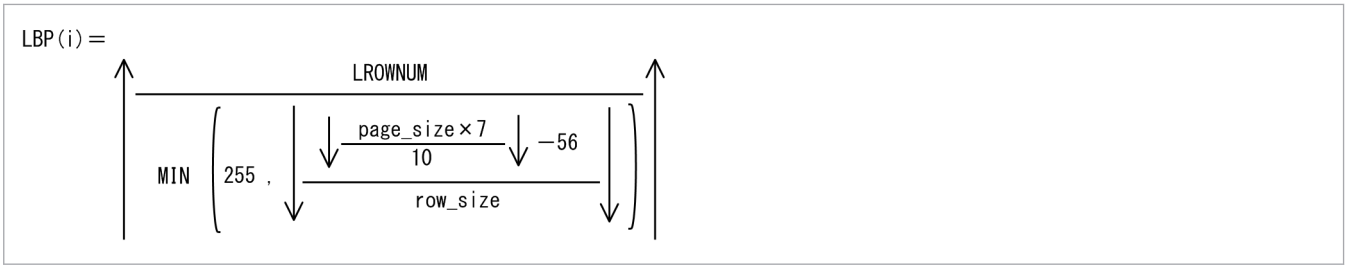
SEGSIZE

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある「*SEGSIZE*」を参照してください。

LBP(i)

次に示す計算式から求めてください。

計算式



page_size

アーカイブマルチチャンク表を定義したデータ用 DB エリアのページサイズ（単位：バイト）

row_size

1 行のサイズ（単位：バイト）

変数*row_size* に代入する値は，アーカイブレンジ列のデータ型によって異なります。変数*row_size* とアーカイブレンジ列のデータ型の関係を，次の表に示します。

表 5-9　変数 row_size とアーカイブレンジ列のデータ型の関係

項番	アーカイブレンジ列のデータ型		変数 row_size に代入する値
1	INTEGER		102
2	SMALLINT		94
3	DECIMAL(*m*,*n*) NUMERIC(*m*,*n*)	1≦*m*≦4	90
4		5≦*m*≦8	94
5		9≦*m*≦16	102
6		17≦*m*≦38	118
7	DOUBLE PRECISION		102
8	FLOAT		
9	CHARCTER(*p*)		88 + 2×*p*
10	DATE		94
11	TIME		104
12	TIMESTAMP		112

（凡例）

m：精度

n：位取り

p：文字列の長さ（単位：バイト）

LROWNUM

ロケーション表に格納する行数（単位：個）

次に示す計算式から求めてください。

計算式

$$LROWNUM = \sum_{i=1}^{archive_chunk_num} FILENUM(i) + \sum_{j=1}^{merge_chunk_num} FILENUM(j)$$

archive_chunk_num

adbarchivechunk コマンドの実行回数

merge_chunk_num

adbmergechunk コマンドの実行回数

FILENUM(*i*)

i 回目のadbarchivechunk コマンドで作成されるアーカイブファイル数の合計

次に示す計算式から求めてください。

計算式

$$FILENUM(i) = \sum_{i=1}^{chunk_num} (SGMTGRPNUM \times thd_num)$$

FILENUM(j)

j 回目のadbmergechunk コマンドでマージするアーカイブ状態のチャンクに対応するアーカイブファイル数の合計

変数*FILENUM(i)*の計算式を使用して求めてください。その際,「*i*」を「*j*」に置き換えてください。

chunk_num

コマンドの対象となるチャンク数（単位：個）

thd_num

次の値を代入してください。

- adbarchivechunk コマンドで作成されるアーカイブファイル数を求める場合

  次に示す計算式から求めてください。

  ↓（*adbarchivechunkコマンド実行時のアーカイブチャンクオプションadb_arcv_rthd_numの指定値*－1）÷2↓

- adbmergechunk コマンドでマージするアーカイブ状態のチャンクに対応するアーカイブファイル数を求める場合

  次の値を指定してください。

  adbmergechunkコマンドの処理対象チャンクに対して，adbarchivechunkコマンド実行時のアーカイブチャンクオプションadb_arcv_rthd_numの指定値

SGMTGRPNUM

変数*SGMTGRPNUM* は，変数*sgmtnum_in_chunk* の値で異なります。変数*sgmtnum_in_chunk* と変数*SGMTGRPNUM* の関係を，次の表に示します。

表 5-10　変数 sgmtnum_in_chunk と変数 SGMTGRPNUM の関係

項番	変数 *sgmtnum_in_chunk* の値	変数 *SGMTGRPNUM* の値（単位：個数）
1	*sgmtnum_in_chunk* ＜ 16	*SGMTGRPNUM* には，1 を代入してください。
2	16≦*sgmtnum_in_chunk* かつ *sgmtnum_in_chunk* ＜ 32	*SGMTGRPNUM* には，4 を代入してください。
3	32≦*sgmtnum_in_chunk* かつ *sgmtnum_in_chunk* ＜ 64	*SGMTGRPNUM* には，8 を代入してください。
4	64≦*sgmtnum_in_chunk* かつ *sgmtnum_in_chunk* ＜ 256	*SGMTGRPNUM* には，16 を代入してください。
5	256≦*sgmtnum_in_chunk*	*SGMTGRPNUM* は，次に示す計算式で求めてください。 SGMTGRPNUM＝ MAX(↓sgmtnum_in_chunk÷256↓ , 32)

sgmtnum_in_chunk

アーカイブ対象のチャンクの中で，最も大きな表のセグメント数

次に示す計算式から求めてください。

計算式

$$\text{sgmtnum_in_chunk} = \left\uparrow \frac{\text{CHBP(j)}}{\text{SEGSIZE}} \right\uparrow + \left\uparrow \frac{\text{CHVP(j)}}{\text{SEGSIZE}} \right\uparrow$$

CHBP(j)

「*CHBP（i, k)*」を参照してください。

CHVP(j)

「*CHVP（i, k)*」を参照してください。

SEGSIZE

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある「*SEGSIZE*」を参照してください。

page_size

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある「*page_size*」を参照してください。

LVP(i)

次に示す計算式から求めてください。

計算式

$$LVP(i) = MAX(A\ ,\ B)$$

変数の説明

A：次の計算式から求めてください。

$$\left\lceil \frac{LROWNUM}{255} \right\rceil$$

B：次の計算式から求めてください。

$$\left\lceil \frac{LROWNUM \times 524}{MAX\left(\left\lfloor \frac{page_size \times 7}{10} \right\rfloor - 56\ ,\ MIN(524\ ,\ page_size - 56) \right)} \right\rceil$$

SGLIDX(i)

i 番目のアーカイブマルチチャンク表に対応するロケーション表に定義されたインデクスを格納するために必要なセグメント数

アーカイブマルチチャンク表の場合に求める必要があります。

次に示す計算式から求めてください。

計算式

$$SGLIDX(i) = \left\lceil \frac{LICHKIDP_LOWER(i)}{SEGSIZE} \right\rceil + \left\lceil \frac{LICHKIDP_UPPER(i)}{SEGSIZE} \right\rceil + \left\lceil \frac{LIRNGO1P_LOWER(i)}{SEGSIZE} \right\rceil + \left\lceil \frac{LIRNGO1P_UPPER(i)}{SEGSIZE} \right\rceil + \left\lceil \frac{LIRNGO2P_LOWER(i)}{SEGSIZE} \right\rceil + \left\lceil \frac{LIRNGO2P_UPPER(i)}{SEGSIZE} \right\rceil$$

LICHKIDP_LOWER(i)

アーカイブファイルに対応するチャンクのチャンク ID に定義されたインデクスの下位ページ用セグメントで使用するページ数（単位：ページ）

LICHKIDP_UPPER(i)

アーカイブファイルに対応するチャンクのチャンク ID に定義されたインデクスの上位ページ用セグメントで使用するページ数（単位：ページ）

LIRNG01P_LOWER(i)

アーカイブファイルに格納されたデータについて，アーカイブレンジ列の値の範囲（上限値と下限値）に定義されたインデクスの下位ページ用セグメントで使用するページ数（単位：ページ）

LIRNG01P_UPPER(i)

アーカイブファイルに格納されたデータについて，アーカイブレンジ列の値の範囲（上限値と下限値）に定義されたインデクスの上位ページ用セグメントで使用するページ数（単位：ページ）

LIRNG02P_LOWER(i)

アーカイブファイルに格納されたデータについて，アーカイブレンジ列の値の下限値に定義されたインデクスの下位ページ用セグメントで使用するページ数（単位：ページ）

LIRNG02P_UPPER(i)

アーカイブファイルに格納されたデータについて，アーカイブレンジ列の値の下限値に定義されたインデクスの上位ページ用セグメントで使用するページ数（単位：ページ）

LICHKIDP_LOWER(i)，*LIRNG01P_LOWER(i)*，および*LIRNG02P_LOWER(i)*については，「5.8.3 B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の変数*IP_LOWER(i)*の計算式で求めてください。*LICHKIDP_UPPER(i)*，*LIRNG01P_UPPER(i)*，および*LIRNG02P_UPPER(i)*については，「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の変数*IP_UPPER(i)*の計算式で求めてください。その際，各変数には次の値を代入してください。

R

1.5 を代入してください。

key_num

0 を代入してください。

dup_key_num

次の計算式で求めた値を代入してください。

計算式

```
dup_key_num＝ ↑FILENUM(i)÷256↑
```

page_size

アーカイブマルチチャンク表を格納するデータ用 DB エリアのページサイズを，バイト単位で代入してください。

pctfree

30 を代入してください。

key_dup

0 を代入してください。

KEYSZDB

次の表で該当する値を代入してください。

表 5-11　変数 KEYSZDB に代入する値

項番	見積もり対象の変数	アーカイブレンジ列のデータ型	変数 KEYSZDB に代入する値
1	見積もり対象の変数は次のとおりです。 • *LICHKIDP_LOWER(i)* • *LICHKIDP_UPPER(i)*	－	8
2	見積もり対象の変数は次のとおりです。	INTEGER	16
3		SMALLINT	8

項番	見積もり対象の変数	アーカイブレンジ列のデータ型		変数 KEYSZDB に代入する値
4	• *LIRNG01P_LOWER(i)* • *LIRNG01P_UPPER(i)*	DECIMAL(*m*,*n*) NUMERIC(*m*,*n*)	1≦*m*≦4	4
5			5≦*m*≦8	8
6			9≦*m*≦16	16
7			17≦*m*≦38	32
8		DOUBLE PRECISION		16
9		FLOAT		
10		CHARACTER(*p*)		2×*p*
11		DATE		8
12		TIME		18
13		TIMESTAMP		26
14	見積もり対象の変数は次のとおりです。 • *LIRNG02P_LOWER(i)* • *LIRNG02P_UPPER(i)*	INTEGER		8
15		SMALLINT		4
16		DECIMAL(*m*,*n*) NUMERIC(*m*,*n*)	1≦*m*≦4	2
17			5≦*m*≦8	4
18			9≦*m*≦16	8
19			17≦*m*≦38	16
20		DOUBLE PRECISION		8
21		FLOAT		
22		CHARACTER(*p*)		*p*
23		DATE		4
24		TIME		9
25		TIMESTAMP		13

（凡例）

－：該当しません。

m：精度

n：位取り

p：文字列の長さ（単位：バイト）

(g) 変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）

カラムストア表にはデータが圧縮されて格納されるため，ローストア表のようにレコード長から順に見積もり式を計算していくことができません。そのため，実際に少量のデータを格納し，その圧縮率を基にセグメント数を見積もります。

次に示す計算式から求めてください。

$$SGCOLUMNTBL = \sum_{i=1}^{column_tbl_num_in_dbarea} \left(\left\uparrow \frac{COLUMNTBLSIZE(i)}{4} \right\uparrow + UPDATESEGNUM(i) \right)$$

column_tbl_num_in_dbarea

データ用 DB エリア中のすべてのカラムストア表の数（単位：個）

COLUMNTBLSIZE（i）

i 番目のカラムストア表のデータサイズ（単位：メガバイト）

次に示す計算式から求めてください。

$$COLUMNTBLSIZE(i) = \sum_{k=1}^{chunk_num} (IMPORTDATASIZE(i,k) \times COMPRESSION_RATE + COLUMNIZESIZE(i,k))$$

chunk_num

作成されるチャンク数（単位：個）

IMPORTDATASIZE（i，k）

i 番目のカラムストア表の，k 番目のチャンクに格納する元データ（CSV ファイル）のサイズ（単位：メガバイト）

COMPRESSION_RATE

元データの圧縮率

i 番目のカラムストア表に格納する元データの一部をadbimport コマンドで格納してください。そのあと，次に示す計算式から求めてください。

$$COMPRESSION_RATE = \frac{testdbsize}{testcsvsize}$$

testdbsize

元データの一部をadbimport コマンドでインポートした際の，データベースの使用量（単位：メガバイト）

計算式を求めるためにインポートする元データの一部には，100～500 メガバイト程度の実データに近いデータを用意することを推奨します。adbimport コマンドを実行するときは，インポートオプション adb_import_rthd_num に2 を指定してください。

adbimport コマンド実行後に，adbdbstatus コマンドで，**表のサマリ情報**に含まれるUsed_segments（表の使用中セグメント数）をメガバイト単位で出力してください。そして，データベースの使用量を確認してください。

adbimport コマンドおよびadbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

testcsvsize

adbimport コマンドでインポートした元データの一部の CSV ファイルのサイズ（単位：メガバイト）

COLUMNIZESIZE（i, k）

i 番目のカラムストア表の k 番目のチャンクに，更新行のカラム化機能のメンテナンス処理で格納されるカラムストア形式のデータのサイズ（単位：メガバイト）

*COLUMNIZESIZE（i, k）*は更新行のカラム化機能を有効にしている場合にだけ加算してください。なお，i 番目のカラムストア表に B-tree インデクスを定義している場合は，0 として見積もってください。

計算式を次に示します。

$$COLUMNIZESIZE(i,k) = \left\uparrow IMPORTDATASIZE(i,k) \times COMPRESSION_RATE \times \frac{insert_row_num(i,k)}{import_row_num(i,k)} \right\uparrow$$

IMPORTDATASIZE（i, k）

i 番目のカラムストア表の k 番目のチャンクに格納する元データ（CSV ファイル）のサイズ（単位：メガバイト）

COMPRESSION_RATE

元データの圧縮率

上記にある変数 *COMPRESSION_RATE* を参照して求めてください。

insert_row_num（i, k）

i 番目のカラムストア表の k 番目のチャンクで，INSERT 文またはUPDATE 文によって，追加または更新される行数

import_row_num（i, k）

i 番目のカラムストア表の k 番目のチャンクに格納する元データ（CSV ファイル）の行数

UPDATESEGNUM（i）

i 番目のカラムストア表に対して，更新系 SQL の実行で増えるセグメント数

次に示す計算式から求めてください。

$$UPDATESEGNUM(i) = \sum_{k=1}^{chunk_num} \left\uparrow \frac{ADDTBLPGNUM(i,k) + DELTBLPGNUM(i,k)}{SEGSIZE - 1} \right\uparrow$$

SEGSIZE

セグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

SEGSIZE＝ 1セグメントの容量（単位：バイト）÷*page_size*

page_size

i 番目のカラムストア表が定義されている DB エリアのページサイズ（単位：バイト）

ADDTBLPGNUM（i，k）

i 番目のカラムストア表の，k 番目のチャンクにINSERT 文またはUPDATE 文で追加するデータのページ数

次に示す計算式から求めてください。

$$ADDTBLPGNUM(i,k) = BP(i,k) + VP(i,k)$$

BP（i，k）

i 番目のカラムストア表の，k 番目のチャンクにINSERT 文またはUPDATE 文で追加するデータの基本行用ページ数

i 番目のカラムストア表の，k 番目のチャンクにINSERT 文またはUPDATE 文で追加する行数を基に，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」を参照して求めてください。

VP（i，k）

i 番目のカラムストア表の，k 番目のチャンクにINSERT 文またはUPDATE 文で追加するデータの分岐行用ページ数

i 番目のカラムストア表の，k 番目のチャンクにINSERT 文またはUPDATE 文で追加する行数を基に，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(2)　分岐行用ページ数（変数 VP（i））の求め方」を参照して求めてください。

DELTBLPGNUM（i，k）

i 番目のカラムストア表の，k 番目のチャンクにUPDATE 文またはDELETE 文で無効情報管理用ページに追加するデータのページ数

次に示す計算式から求めてください。

$$DELTBLPGNUM(i,k) = DELLEAFPGNUM(i,k) + DELUPPERPGNUM(i,k)$$

DELLEAFPGNUM（i，k）

次に示す計算式から求めてください。

$$DELLEAFPGNUM(i,k) = \left\uparrow \frac{12 \times (update_row_num + delete_row_num)}{0.5 \times (page_size - 80)} \right\uparrow$$

update_row_num

i 番目のカラムストア表の，k 番目のチャンクに対してUPDATE 文で更新する行数

delete_row_num

i 番目のカラムストア表の，k 番目のチャンクに対してDELETE 文で削除する行数

page_size

i 番目のカラムストア表が定義されている DB エリアのページサイズ（単位：バイト）

DELUPPERPGNUM（i，k）

次に示す計算式から求めてください。

$$DELUPPERPGNUM(i,k) = \left\uparrow \frac{12 \times DELLEAFPGNUM(i,k)}{0.5 \times (page_size - 80)} \right\uparrow$$

また，変数*SGCOLUMNTBL* については，すべての列で圧縮方式に**無圧縮（NONE）**が選択された場合を仮定し，そのサイズを求めることで見積もり値を求めることもできます。ただし，選択された圧縮方式や圧縮率によっては，実際に格納した場合のサイズに比べて見積もり値が大きくなることがあります。

すべての列で圧縮方式に**無圧縮（NONE）**が選択された場合の見積もり値は，次に示す計算式から求めてください。

$$SGCOLUMNTBL = \sum_{i=1}^{column_tbl_num_in_dbarea} COLUMNTBLSEGNUM(i)$$

COLUMNTBLSEGNUM（i）

i 番目のカラムストア表のセグメント数

次に示す計算式から求めてください。

$$COLUMNTBLSEGNUM(i) = \sum_{k=1}^{chunk_num} (COLUMNDATASEGNUM(i,k) + ROWDATASEGNUM(i,k) + COLUMNIZESEGNUM(i,k))$$

COLUMNDATASEGNUM（i，k）

i 番目のカラムストア表の，k 番目のチャンクのカラムデータ用セグメント数

次に示す計算式から求めてください。

$$COLUMNDATASEGNUM(i,k) = \frac{row_num}{MIN\left\{262,144 \ , \left(\left\downarrow \frac{page_size - 80}{COL_MAX_SIZE \times \uparrow column_num \div COL_PAGE_NUM \uparrow} \right\downarrow\right) \times MAX\left(1, \left\downarrow \frac{COL_PAGE_NUM}{column_num} \right\downarrow\right)\right\}}$$

row_num

i 番目のカラムストア表の，k 番目のチャンクに格納する行数

column_num

i 番目のカラムストア表の列数

page_size

データ用 DB エリアのページサイズ（単位：バイト）

COL_PAGE_NUM

変数*COL_PAGE_NUM* に代入する値は，データ用 DB エリアのページサイズおよび 1 セグメントの容量によって決まります。次の表のデータ用 DB エリアのページサイズおよび 1 セグメントの容量に対応する値を代入してください。

表 5-12　変数 COL_PAGE_NUM に代入する値

項番	ページサイズ（単位：キロバイト）	1 セグメントの容量（単位：メガバイト）	*COL_PAGE_NUM* の値
1	4	4	1,021
2	8		510
3	16		255
4	32		127
5		16	511
6		32	1,023

COL_MAX_SIZE

i 番目のカラムストア表の列データの最大データ長

次に示す計算式から求めてください。

$$COL_MAX_SIZE = \max_{s=1,\cdots,column_num} COLUMNDATASIZE(s) \times elm_num(s)$$

例えば，i 番目のカラムストア表にC1 列，C2 列，C3 列が定義されている場合，次の計算をしてください。計算結果の最大値が *COL_MAX_SIZE* の値になります。

- *COLUMNDATASIZE(1)×elm_num(1)*
- *COLUMNDATASIZE(2)×elm_num(2)*
- *COLUMNDATASIZE(3)×elm_num(3)*

COLUMNDATASIZE（s）

s 列目の列データのデータ長（単位：バイト）

変数 *COLUMNDATASIZE* に代入する値は，列データのデータ型によって異なります。次の表で該当する値を代入してください。

表 5-13　各列データのデータ型のデータ長の一覧

項番	分類	データ型		データ長（単位：バイト）
1	数データ	INTEGER		10
2		SMALLINT		6
3		DECIMAL(*m*,*n*)※ NUMERIC(*m*,*n*)※	$m \leqq 4$	4
4			$5 \leqq m \leqq 8$	6
5			$9 \leqq m \leqq 16$	10
6			$17 \leqq m$	18
7		DOUBLE PRECISION		10
8		FLOAT		

項番	分類	データ型		データ長（単位：バイト）
9	文字データ	CHARACTER(*n*)	*n*≦127	$\uparrow\frac{n+1}{2}\uparrow\times 2$
10			128≦*n*	10
11		VARCHAR(*n*)	*n*≦127	$\uparrow\frac{n+1}{2}\uparrow\times 2$
12			128≦*n*≦255	128
13			256≦*n*	130
14	日時データ	DATE		6
15		TIME(*p*)		$\uparrow\frac{3+\uparrow p\div 2\uparrow}{2}\uparrow\times 2$
16		TIMESTAMP(*p*)		$\uparrow\frac{7+\uparrow p\div 2\uparrow}{2}\uparrow\times 2$
17	バイナリデータ	BINARY(*n*)	*n*≦127	$\uparrow\frac{n+1}{2}\uparrow\times 2$
18			128≦*n*	10
19		VARBINARY(*n*)	*n*≦127	$\uparrow\frac{n+1}{2}\uparrow\times 2$
20			128≦*n*≦255	128
21			256≦*n*	130

（凡例）

m，*n*：正の整数

p：0，3，6，9または12

注※

全体の桁数が*m*桁で，小数点以下の桁数が*n*桁の固定小数点数を示します。*m*を省略した場合は38が仮定されます。

elm_num（s）

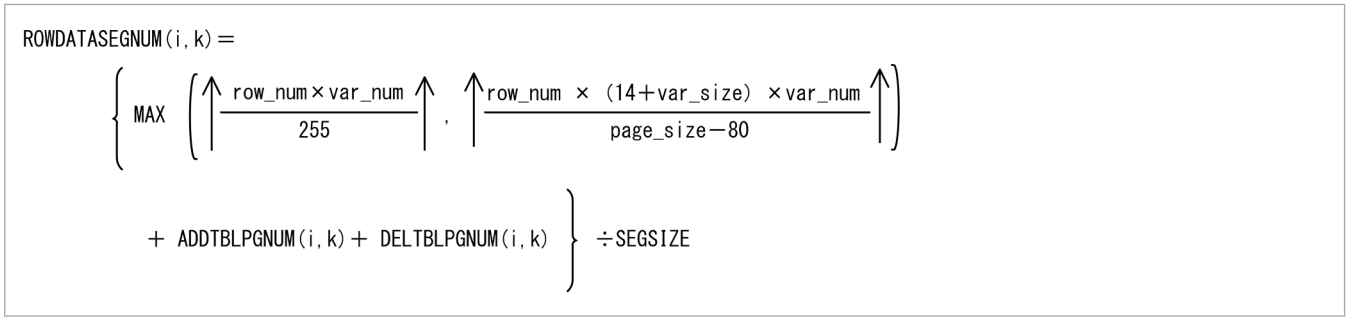
s列目の配列要素数の平均値（単位：個）

s列目の列が配列型以外の列の場合は，1としてください。

ROWDATASEGNUM（i，k）

i番目のカラムストア表の，k番目のチャンクのローデータ用セグメント数

次に示す計算式から求めてください。



row_num

i 番目のカラムストア表の，k 番目のチャンクに格納する行数

page_size

データ用 DB エリアのページサイズ（単位：バイト）

var_num

分岐行として管理される列の列数（単位：個）

CHAR 型，またはBINARY 型の場合，定義長が 128 バイト以上の列の列数を求めてください。

VARCHAR 型，またはVARBINARY 型の場合，実長が 128 バイト以上のデータを含む列の列数を求めてください。

var_size

分岐行の列のデータ長（単位：バイト）

変数 *var_size* に代入する値は，データ型によって異なります。次の表で該当する値を代入してください。

表 5-14　変数 var_size に代入する値

項番	分類	データ型	データ長（単位：バイト）
1	文字データ	CHAR	定義長
2		VARCHAR	*d*
3	バイナリデータ	BINARY	定義長
4		VARBINARY	*d*

（凡例）

d：実際のデータ長

COLUMNIZESEGNUM（i, k）

i 番目のカラムストア表の k 番目のチャンクに，更新行のカラム化機能のメンテナンス処理で格納されるカラムデータ用セグメント数

*COLUMNIZESEGNUM（i, k）*は，更新行のカラム化機能を有効にしている場合にだけ加算してください。なお，i 番目のカラムストア表に B-tree インデクスを定義している場合は，0 として見積もってください。

計算式を次に示します。

COLUMNIZESEGNUM(i, k)＝ COLUMNDATASEGNUM(i, k)

*COLUMNDATASEGNUM（i，k）*の求め方については，上記にある変数*COLUMNDATASEGNUM（i，k）*を参照してください。その際，変数*row_num* にはi番目のカラムストア表のk番目のチャンクで，INSERT文またはUPDATE文によって，追加または更新される行数を代入してください。

(h)　変数SGIDXの求め方（マルチチャンク表の場合）

次に示す計算式から求めてください。

$$SGIDX = \sum_{i=1}^{idx_num_in_dbarea} \left\{ \sum_{k=1}^{chunk_num} \left(\left\uparrow \frac{CHIP_LOWER(i,k)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{CHIP_UPPER(i,k)}{SEGSIZE} \right\uparrow \right) \right\}$$

idx_num_in_dbarea

データ用DBエリア中のすべてのB-treeインデクス数（単位：個）

chunk_num

作成されるアーカイブ状態ではないチャンク数（単位：個）

CHIP_LOWER（i，k）

各チャンクのB-treeインデクスの下位ページ用のセグメントで使用するページ数

「5.8.3　B-treeインデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数IP_LOWER（i））の求め方」を参照してください。

その際，対象のチャンクに-bオプションを指定したadbimportコマンド，INSERT文，およびUPDATE文で格納するデータ量から見積もった値を代入してください。

CHIP_UPPER（i，k）

各チャンクのB-treeインデクスの上位ページ用のセグメントで使用するページ数

「5.8.3　B-treeインデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数IP_UPPER（i））の求め方」を参照してください。

その際，対象のチャンクに-bオプションを指定したadbimportコマンド，INSERT文，およびUPDATE文で格納するデータ量から見積もった値を代入してください。

(i)　変数SGRIXの求め方（マルチチャンク表の場合）

次に示す計算式から求めてください。

$$SGRIX = \sum_{i=1}^{rng_num_in_dbarea} \left\{ \sum_{k=1}^{chunk_num} CHRS(i,k) \right\}$$

rng_num_in_dbarea

データ用DBエリア中のレンジインデクス数（単位：個）

chunk_num

作成されるアーカイブ状態ではないチャンク数（単位：個）

CHRS（i，k）

各チャンクのレンジインデクスを格納するために必要なセグメント数

対象のチャンクに-b オプションを指定したadbimport コマンド，INSERT 文，およびUPDATE 文で格納するデータ量から見積もった値を代入してください。

(j)　変数 SGTIX の求め方（マルチチャンク表の場合）

次に示す計算式から求めてください。

$$SGTIX = \sum_{i=1}^{idx_num_in_dbarea} \left\{ \sum_{k=1}^{chunk_num} \left(\left\uparrow \frac{CHTIP_STRSEG(i,k)}{SEGSIZE} \right\uparrow + \left\uparrow \frac{CHTIP_APPSEG(i,k)}{SEGSIZE} \right\uparrow \right) \right\}$$

idx_num_in_dbarea

データ用 DB エリア中のすべてのテキストインデクス数（単位：個）

chunk_num

作成されるアーカイブ状態ではないチャンク数（単位：個）

CHTIP_STRSEG（i，k）

各チャンクのテキストインデクスの文字列管理部用のセグメントで使用するページ数

「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」の「(1)　文字列管理部用のセグメントで使用する格納ページ数（変数 TIP_STRSEG（i））の求め方」を参照してください。

その際，対象のチャンクに-b オプションを指定したadbimport コマンド，INSERT 文，およびUPDATE 文で格納するデータ量から見積もった値を代入してください。

CHTIP_APPSEG（i，k）

各チャンクのテキストインデクスの出現情報管理部用のセグメントで使用するページ数

「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」の「(2)　出現情報管理部用のセグメントで使用する格納ページ数（変数 TIP_APPSEG（i））の求め方」を参照してください。

その際，対象のチャンクに-b オプションを指定したadbimport コマンド，INSERT 文，およびUPDATE 文で格納するデータ量から見積もった値を代入してください。

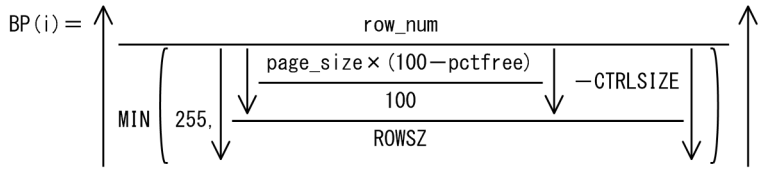
5.8.2　行の種別ごとの格納ページ数の求め方

ここでは，各表を格納するために必要な基本行用ページ数（変数*BP（i）*），および各表を格納するために必要な分岐行用ページ数（変数 *VP（i）*）の求め方について説明します。

(1) 基本行用ページ数（変数 BP（i））の求め方

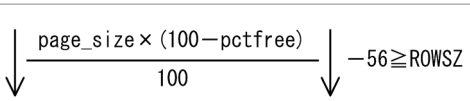
基本行用ページ数（変数*BP（i）*）は，次に示す計算式から求めてください。変数*BP（i）*の分母の括弧部は 1 ページに格納される行数であり，最小 1，最大 255 となります。

計算式



変数の説明

- *row_num*
  表に格納する行数（単位：個）
- *page_size*
  データ用 DB エリアのページサイズ（単位：バイト）
- *pctfree*
  CREATE TABLE 文のPCTFREE で指定する未使用領域の比率（単位：%）
  未使用領域の比率を指定しない場合は，30%と仮定して計算してください。
  カラムストア表にINSERT 文またはUPDATE 文で追加するデータの基本行用ページ数を求める場合は，0%と仮定して計算してください。
  なお，*pctfree* の値は，次に示す計算式を満たすように値を決めてください。
  計算式（pctfree の値）

- *CTRLSIZE*
  ローストア表の場合は56 を指定してください。
  カラムストア表にINSERT 文またはUPDATE 文で追加するデータの基本行用ページ数を求める場合は80 を指定してください。

また，行長*ROWSZ* は次に示す計算式から求めてください。非 FIX 表（表オプションBRANCH ALL の指定なし），非 FIX 表（表オプションBRANCH ALL の指定あり）および FIX 表で計算式が異なります。

カラムストア表にINSERT 文またはUPDATE 文で追加するデータの基本行用ページ数を求める場合には，「計算式（非 FIX 表で表オプション BRANCH ALL の指定なしの場合)」に従って求めてください。

計算式（非 FIX 表で表オプション BRANCH ALL の指定なしの場合）

$$ROWSZ = \left\uparrow \frac{28 + 2 \times col_num + \sum_{i=1}^{col_num} col_size(i)}{2} \right\uparrow \times 2$$

計算式（非 FIX 表で表オプション BRANCH ALL の指定ありの場合）

$$ROWSZ = \left\uparrow \frac{28 + \left\downarrow \frac{col_num + 7}{8} \right\downarrow + \sum_{i=1}^{col_num} col_size(i)}{2} \right\uparrow \times 2$$

計算式（FIX 表の場合）

$$ROWSZ = \left\uparrow \frac{26 + \sum_{i=1}^{col_num} col_size(i)}{2} \right\uparrow \times 2$$

変数の説明

- *col_num*

  表の列の総数（単位：個）

- *col_size（i）*

  各列のデータ長（単位：バイト）

  次に示す表を基に，各列のデータ長を求めてください。そのあとに，すべての列の合計を求めてください。

表 5-15　各データ型のデータ長の一覧

項番	分類	データ型		データ長（単位：バイト）
1	数データ	INTEGER		8
2		SMALLINT		4
3		DECIMAL(*m*,*n*)※ NUMERIC(*m*,*n*)※	1≦*m*≦4	2
4			5≦*m*≦8	4
5			9≦*m*≦16	8
6			17≦*m*≦38	16
7		DOUBLE PRECISION		8
8		FLOAT		
9	文字データ	CHARACTER(*n*)		*n*
10		VARCHAR(*n*)		*var_col_size*
11	日時データ	DATE		4

項番	分類	データ型	データ長（単位：バイト）
12		TIME(*p*)	3 ＋ ↑*p*÷2↑
13		TIMESTAMP(*p*)	7 ＋ ↑*p*÷2↑
14	バイナリデータ	BINARY(*n*)	*n*
15		VARBINARY(*n*)	*var_col_size*
16	配列データ	ARRAY	11

（凡例）

m，*n*：正の整数

p：0，3，6，9または12

注※

全体の桁数が*m*桁で，小数点以下の桁数が*n*桁の固定小数点数を示します。*m*を省略した場合は38が仮定されます。

var_col_size：VARCHAR型およびVARBINARY型の列のデータ長

次に示す表を基にデータ長を求めてください。

表5-16　VARCHAR型およびVARBINARY型の列のデータ長の一覧

項番	表オプション BRANCH ALLの指定有無	列定義 BRANCHの指定	定義長n （単位：バイト）	データ長 （単位：バイト）
1	指定あり	－	－	9
2	指定なし	YES	－	11
3		NO	1≦*n*≦255	*d* ＋ 2
4			256≦*n*≦32,000	*d* ＋ 3
5		AUTOまたは指定なし	1≦*n*≦255	*d* ＋ 2
6			256≦*n*≦32,000	11

（凡例）

n：正の整数

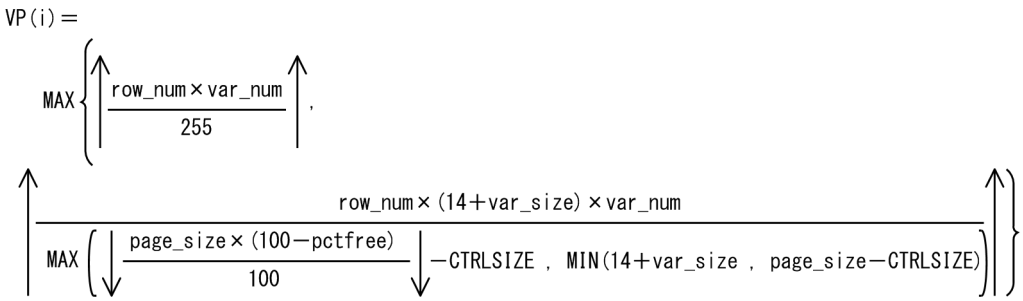
d：実際のデータ長

－：該当しません。

(2)　分岐行用ページ数（変数VP（i））の求め方

分岐行用ページ数（変数*VP（i）*）は，次に示す計算式から求めてください。

計算式



変数の説明

- *row_num*

  表に格納する行数（単位：個）

- *var_num*

  分岐行として管理される列の列数（単位：個）

  次に示す列の数を求めてください。

  表オプションBRANCH ALL の指定ありの場合

  ・VARCHAR 型，およびVARBINARY 型の列の数

  表オプションBRANCH ALL の指定なしの場合

  次のどれかに該当するVARCHAR 型，およびVARBINARY 型の列の数の合計

  ・列定義BRANCH にYES を指定した列

  ・定義長 256 バイト以上で列定義BRANCH にAUTO を指定した列

  ・定義長 256 バイト以上で列定義BRANCH を省略した列

- *page_size*

  データ用 DB エリアのページサイズ（単位：バイト）

- *pctfree*

  CREATE TABLE 文のPCTFREE で指定する未使用領域の比率（単位：%）

  未使用領域の比率を指定しない場合は，30%と仮定して計算してください。

  カラムストア表にINSERT 文またはUPDATE 文で追加するデータの分岐行用ページ数を求める場合は，0%と仮定して計算してください。

  なお，*pctfree* の値は，「(1)　基本行用ページ数（変数 BP（i））の求め方」の変数「*pctfree*」を参照して，計算式を満たすように値を決定してください。

- *var_size*

  分岐行の列の平均データ長（単位：バイト）

  次に示す表を基に求めてください。

表 5-17　分岐行となる列のデータ長の一覧

分類	データ型	データ長（単位：バイト）
文字データ	VARCHAR	*d*

分類	データ型	データ長（単位：バイト）
バイナリデータ	VARBINARY	*d*

（凡例）

d：実際のデータ長

- *CTRLSIZE*

  ローストア表の場合は56 を指定してください。

  カラムストア表にINSERT 文またはUPDATE 文で追加するデータの分岐行用ページ数を求める場合は80 を指定してください。

(3) 配列型の列を定義した表の行長（変数 ARRAY_ROWSZ）の求め方

配列型の列を定義した表の行長（変数*ARRAY_ROWSZ*）は，次に示す計算式から求めてください。

計算式

$$ARRAY_ROWSZ = \left(2 + \frac{\sum_{k=1}^{col_num} ARRAY_DICSIZE(k)}{page_size} + ARRAY_DATAPAGENUM \right) \times page_size$$

変数の説明

- *col_num*

  表の列の総数

- *page_size*

  データ用 DB エリアのページサイズ（単位：バイト）

- *ARRAY_DICSIZE(k)*

  表の k 番目に定義された列の圧縮用辞書のサイズ

  次に示す計算式から求めてください。なお，列定義の圧縮方式指定に，NONE（データ圧縮なし），RUNLENGTH（ランレングス圧縮方式），DELTA（差分圧縮方式），またはDELTA_RUNLENGTH（差分ランレングス圧縮方式）を指定する列については，*ARRAY_DICSIZE(k)* は 0 になります。

  計算式

$$ARRAY_DICSIZE = MIN\{16{,}000 , (COLDATASIZE(k)+2) \times elm_num(k)\} + COLDATASIZE(k) \times \left(\frac{MIN\{16{,}000 , (COLDATASIZE(k)+2) \times elm_num(k)\}}{page_size-56} \right)$$

- *elm_num(k)*

  表の k 番目に定義した列の最大要素数（単位：個）

配列型以外の列の場合は，1 としてください。

- *COLDATASIZE(k)*

  表の k 番目に定義した列のデータ長（単位：バイト）

  次に示す計算式で求めてください。

  計算式

$$COLDATASIZE(k) = \left\lfloor \frac{COLDATASIZE_NOBOUND(k) + 1}{2} \right\rfloor$$

- *COLDATASIZE_NOBOUND(k)*

  配列型の列の要素データ型および定義長を基に，次に示す表から求めてください。

表 5-18　変数 COLDATASIZE_NOBOUND の値

項番	分類	要素データ型		COLDATASIZE_NOBOUND の値（単位：バイト）
1	数データ	INTEGER		9
2		SMALLINT		5
3		DECIMAL(*m*,*n*)※ NUMERIC(*m*,*n*)※	*m*≦4	3
4			5≦*m*≦8	5
5			9≦*m*≦16	9
6			17≦*m*	17
7		DOUBLE PRECISION		9
8		FLOAT		
9	文字データ	CHARACTER(*n*)	*n*≦127	*n* ＋ 1
10			128≦*n*	9
11		VARCHAR(*n*)	*n*≦127	*n* ＋ 2
12			128≦*n*≦255	129
13			256≦*n*	130
14	日時データ	DATE		5
15		TIME(*p*)		4 ＋ ↑*p*÷2↑
16		TIMESTAMP(*p*)		8 ＋ ↑*p*÷2↑
17	バイナリデータ	BINARY(*n*)	*n*≦127	*n* ＋ 1
18			128≦*n*	9
19		VARBINARY(*n*)	*n*≦127	*n* ＋ 2
20			128≦*n*≦255	129

項番	分類	要素データ型		COLDATASIZE_NOBOUND の値（単位：バイト）
21			256≦*n*	130

（凡例）

m，*n*：正の整数

p：0，3，6，9 または 12

注※

全体の桁数が *m* 桁で，小数点以下の桁数が *n* 桁の固定小数点数を示します。*m* を省略した場合は38 が仮定されます。

- *ARRAY_DATAPAGENUM*

次の手順で求めてください。

■前提条件

- *ARRAY_DATAPAGENUM* の初期値は 1 です。
- 列の定義順を示す *k* の初期値は 1 です。
- 変数については，変数 *ARRAY_DICSIZE(k)* を参照してください。

■手順

1. *free_size* を求める

次の計算式で求めてください。

```
free_size ＝ page_size － 56
```

2. k 番目の列の次の値を求める

```
（COLDATASIZE(k)＋8）×elm_num(k)
```

*手順 2.の値*が「*page_size* － 56」（*手順 1.の値*）を超える場合は，「*page_size* － 56」（*手順 1.の値*）を*手順 2.の値*としてください。

3. k 番目の列の次の値を求める

```
手順1. の値 － 手順2. の値
```

4. 手順 3.の値を確認する

手順 3.の値＜ 0 の場合

ARRAY_DATAPAGENUM に 1 を加算して，手順 1.に戻って手順を繰り返してください。

0≦手順 3.の値の場合

k に 1 を加算してください。

k が表に定義した列数を超えたら計算は完了です。超えていない場合は手順 2.に戻って手順を繰り返してください。

5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方

ここでは，B-tree インデクスのセグメントごとの格納ページ数の求め方について説明します。

B-tree インデクスのセグメントは次のとおりです。

- 下位ページ用のセグメント
- 上位ページ用のセグメント

(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER (i)）の求め方

下位ページ用のセグメントで使用する格納ページ数（変数*IP_LOWER (i)*）は，次に示す計算式から求めてください。

なお，B-tree インデクスにナル値除外指定が指定されている場合，ナル値だけで構成されるキーについては，キーの種類の個数，およびキー値の重複数の平均値を求める計算式から除いてください。

計算式

IP_LOWER (i) ＝ (*PIDX_LEAF*＋*PDUP*) ×*R*

PIDX_LEAF は，計算式 1 から求めてください。

PDUP は，計算式 2 から求めてください。*PDUPLIST* はキー値ごとの行 ID リストページ数の平均，*PDUPDIR* はキー値ごとの行 ID ディレクトリページ数の平均になります。

R は，B-tree インデクスページごとにキーの重複数，またはキー長にばらつきがある場合には 1.2～1.5 とし，平均値近辺に収束している場合には 1 としてください。

計算式 1（PIDX_LEAF の求め方）

$$PIDX_LEAF = MAX\left(1, \uparrow \frac{key_num + dup_key_num}{KNPP} \uparrow\right)$$

計算式 2（PDUP の求め方）

$$PDUP = (PDUPLIST + PDUPDIR) \times dup_key_num$$

$$PDUPLIST = \left\lceil \frac{dup_key_dup}{MAX\left(1, \left\lfloor \frac{\left\lfloor \frac{page_size \times 50}{100} \right\rfloor - 94}{6} \right\rfloor\right)} \right\rceil$$

$$PDUPDIR = \left\lceil \frac{PDUPLIST}{MAX\left(1, \left\lfloor \frac{\left\lfloor \frac{page_size \times 50}{100} \right\rfloor - 84}{16} \right\rfloor\right)} \right\rceil$$

変数の説明

- *key_num*
  キー値の重複が 255 以下のキーの種類の個数（単位：個）
- *dup_key_num*
  キー値の重複が 256 以上のキーの種類の個数（単位：個）
- *dup_key_dup*
  キー値の重複が 256 以上のキーの重複数の平均値（単位：個）
- *page_size*
  データ用 DB エリアのページサイズ（単位：バイト）
- *KNPP*
  1B-tree インデクスページ当たりの平均キー格納件数
  次に示す計算式から求めてください。

  計算式（KNPP の求め方）

$$KNPP = MAX\left(1, \left\lfloor \frac{\left\lfloor \frac{page_size \times (100 - pctfree)}{100} \right\rfloor - 80}{8 + KEYSZDB + 6 \times MIN(dbarea_file_num,\ key_dup) + 6 \times key_dup} \right\rfloor\right)$$

- *pctfree*
  CREATE INDEX 文で指定する未使用領域の比率（単位：%）
  未使用領域の比率を指定しない場合は，30%と仮定して計算してください。
- *key_dup*
  キー値の重複が 255 以下のキーの重複数の平均値（単位：個）
- *dbarea_file_num*
  データ用 DB エリアファイルの数（単位：個）
- *KEYSZDB*
  データベース格納キー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」によって算出された値*KEYSZ* を使用して，次に示す計算式から求めてください。

計算式（KEYSZDB の求め方）

$$KEYSZDB = \left\lceil \frac{KEYSZ}{2} \right\rceil \times 2$$

(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方

上位ページ用のセグメントで使用する格納ページ数（変数*IP_UPPER（i）*）は，次に示す計算式から求めてください。

計算式

$$IP_UPPER(i) = \sum_{k=2}^{n} PIDX(k) \times R$$

*PIDX（k）*は，計算式 1 に示す漸化式から求めてください。*PIDX（2）*は必ず計算し，*PIDX（n）* ＝ 1 になるまで，*PIDX（k ＋ 1）*の計算式を繰り返してください。

R は，B-tree インデクスページごとにキーの重複数，またはキー長にばらつきがある場合には 1.2～1.5 とし，平均値近辺に収束している場合には 1 としてください。

計算式 1（PIDX（k）の求め方）

$$PIDX(1) = PIDX_LEAF$$

$$PIDX(k+1) = \left\lceil \frac{PIDX(k)}{MAX\left(2, \left\lfloor \frac{\left\lfloor \frac{page_size \times (100 - pctfree)}{100} \right\rfloor - 80}{12 + KEYSZDB} \right\rfloor\right)} \right\rceil$$

変数の説明

- *PIDX_LEAF*

  「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の『計算式 1（PIDX_LEAF の求め方)』に従って求めてください。

- *page_size*

  データ用 DB エリアのページサイズ（単位：バイト）

- *pctfree*

  `CREATE INDEX` 文で指定する未使用領域の比率（単位：%）

  未使用領域の比率を指定しない場合は，30%と仮定して計算してください。

- *KEYSZDB*

  データベース格納キー長（単位：バイト）

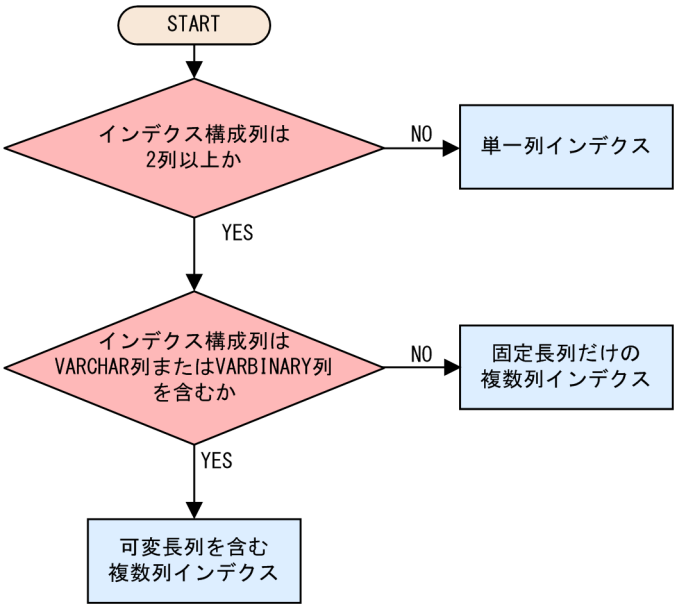
  「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の変数 *KEYSZDB* に従って求めてください。

5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方

B-tree インデクスのキー長（変数 *KEYSZ*）の計算方法を次に示します。

まず，次に示す図に従って B-tree インデクスの種別を特定してください。

図 5-13　B-tree インデクスの種別を特定する方法



次に，特定した B-tree インデクスの種別ごとの計算式によって，*KEYSZ* を求めてください。ただし，各インデクス構成列のキー長は，次に示す表を参照してください。

表 5-19　B-tree インデクスのキー長一覧

項番	分類	データ型		キー長（単位：バイト）
1	数データ	INTEGER		8
2		SMALLINT		4
3		DECIMAL（*m*，*n*） NUMERIC（*m*，*n*）	1≦*m*≦4	2
4			5≦*m*≦8	4
5			9≦*m*≦16	8
6			17≦*m*≦38	16
7		DOUBLE PRECISION		8

項番	分類	データ型	キー長（単位：バイト）
8		FLOAT	
9	文字データ	CHARACTER（*n*）	*n*
10		VARCHAR（*n*）	*d*
11	日時データ	DATE	4
12		TIME（*p*）	3＋↑*p*÷2↑
13		TIMESTAMP（*p*）	7＋↑*p*÷2↑
14	バイナリデータ	BINARY（*n*）	*n*
15		VARBINARY（*n*）	*d*

（凡例）

d：実際のデータ長

m，*n*：正の整数

p：0，3，6，9または12

(1) 単一列インデクスの場合

計算式（単一列インデクス）（単位：バイト）

KEYSZ＝ *flag_size*＋*key_size*

変数の説明

- *flag_size*
  次に示すどれかの値（単位：バイト）
  ・インデクス構成列が，VARCHAR列またはVARBINARY列以外の場合：0
  ・インデクス構成列が，定義長255バイト以下のVARCHAR列またはVARBINARY列の場合：1
  ・インデクス構成列が，定義長256バイト以上のVARCHAR列またはVARBINARY列の場合：2
- *key_size*
  インデクス構成列のキー長（単位：バイト）

(2) 固定長列だけの複数列インデクスの場合

計算式（固定長列だけの複数列インデクス）（単位：バイト）

$$KEYSZ = idx_col_num \times flag_null + \sum_{i=1}^{N} key_size(i)$$

変数の説明

- *idx_col_num*

インデクス構成列の数

- *flag_null*
  次に示すどちらかの値（単位：バイト）
  ・インデクス構成列中にナル値を含まないキーの場合：0
  ・インデクス構成列中にナル値を含むキーの場合：1
- *key_size*（*i*）
  i 番目のインデクス構成列のキー長（単位：バイト）

(3)　可変長列を含む複数列インデクスの場合

計算式（可変長列を含む複数列インデクス）（単位：バイト）

$$KEYSZ=(idx_col_num+1)\times flag_type+\sum_{i=1}^{N} key_size(i)$$

変数の説明

- *idx_col_num*
  インデクス構成列の数
- *flag_type*
  次に示すどちらかの値（単位：バイト）
  ・B-tree インデクスを構成する列の長さが 255 バイト以下の場合：1
  ・B-tree インデクスを構成する列の長さが 256 バイト以上の場合：2
  複数列インデクスを構成する列の長さについては，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義)』を参照してください。
- *key_size*（*i*）
  i 番目のインデクス構成列のキー長（単位：バイト）

5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方

ここでは，テキストインデクスのセグメントごとの格納ページ数の求め方について説明します。

テキストインデクスのセグメントは次のとおりです。

- 文字列管理部用のセグメント
- 出現情報管理部用のセグメント

(1) 文字列管理部用のセグメントで使用する格納ページ数（変数 TIP_STRSEG(i)）の求め方

文字列管理部用のセグメントで使用する格納ページ数（変数 *TIP_STRSEG(i)*）は，次に示す計算式から求めてください。

計算式

```
TIP_STRSEG(i)＝(TIXSTR＋TIXCOR＋TIXWORD)×1.5
```

変数の説明

TIXSTR

テキストインデクスの文字列管理部のページ総数

次に示す計算式から求めてください。

なお，1 文字 1 文字に関連性がない文字列（ランダムな文字列など）に対してテキストインデクスを追加した場合は，次の計算式に収束しないことがあります。

計算式

page_size

データ用 DB エリアのページサイズ（単位：バイト）

pctfree

CREATE INDEX 文で指定する未使用領域の比率（単位：%）

未使用領域の比率を指定しない場合は，30%と仮定して計算してください。

TIXCOR

テキストインデクスの表記ゆれ補正検索で必要なページ総数

テキストインデクスの定義時に，「テキストインデクス表記ゆれ補正指定」を指定した場合に求める必要があります。

次に示す計算式から求めてください。

なお，1 文字 1 文字に関連性がない文字列（ランダムな文字列など）に対してテキストインデクスを追加した場合は，次の計算式に収束しないことがあります。

計算式

page_size

データ用 DB エリアのページサイズ（単位：バイト）

pctfree

CREATE INDEX 文で指定する未使用領域の比率（単位：%）

未使用領域の比率を指定しない場合は，30%と仮定して計算してください。

TIXWORD

テキストインデクスのワード検索で必要なページ総数

テキストインデクスの定義時に，「テキストインデクスワード検索指定」を指定した場合に求める必要があります。

次に示す計算式から求めてください。

なお，1 文字 1 文字に関連性がない文字列（ランダムな文字列など）に対してテキストインデクスを追加した場合は，次の計算式に収束しないことがあります。

計算式

$$\left\uparrow \frac{(260+205\times subarea_num)\times 1,024^2+5,040\times datapage_num}{\left\downarrow \frac{page_size\times(100-pctfree)}{100} \right\downarrow} \right\uparrow$$

subarea_num

テキストインデクスを定義した実表が格納されているデータ用 DB エリアを構成するデータ用 DB エリアファイルの総数

datapage_num

テキストインデクスを定義した実表のデータページ数（基本行と分岐行のページ数を合計した値）

page_size

データ用 DB エリアのページサイズ（単位：バイト）

pctfree

CREATE INDEX 文で指定する未使用領域の比率（単位：%）

未使用領域の比率を指定しない場合は，30%と仮定して計算してください。

(2) 出現情報管理部用のセグメントで使用する格納ページ数（変数 TIP_APPSEG（i））の求め方

出現情報管理部用のセグメントで使用する格納ページ数（変数 *TIP_APPSEG（i）*）は，次に示す計算式から求めてください。

計算式

$$TIP_APPSEG(i) = TIXAPP \times 1.5$$

変数の説明

TIXAPP

テキストインデクスの出現情報管理部のページ総数

次に示す計算式から求めてください。

なお，1 文字 1 文字に関連性がない文字列（ランダムな文字列など）に対してテキストインデクスを追加した場合は，次の計算式に収束しないことがあります。

計算式

$$\left\uparrow \frac{198 \times 1{,}024^2 \times subarea_num + 5{,}040 \times datapage_num}{\left\downarrow \frac{page_size \times (100 - pctfree)}{100} \right\downarrow} \right\uparrow + 6$$

subarea_num

テキストインデクスを定義した実表が格納されているデータ用 DB エリアを構成するデータ用 DB エリアファイルの総数

datapage_num

テキストインデクスを定義した実表のデータページ数（基本行と分岐行のページ数を合計した値）

page_size

データ用 DB エリアのページサイズ（単位：バイト）

pctfree

CREATE INDEX 文で指定する未使用領域の比率（単位：%）

未使用領域の比率を指定しない場合は，30%と仮定して計算してください。

5.8.6　レンジインデクスを格納するセグメント数の求め方

各レンジインデクスを格納するために必要なセグメント数（変数*RS（i）*）は，次に示す計算式から求めてください。

計算式（単位：セグメント数）

$$RS(i) = SGRI + SGSAM$$

変数の説明

SGRI：「(1)　変数 SGRI の求め方」を参照してください。

SGSAM：「(2)　変数 SGSAM の求め方」を参照してください。

(1)　変数 SGRI の求め方

変数*SGRI* は，次に示す計算式から求めてください。

計算式（単位：セグメント数）

$$SGRI = \left\lceil \frac{(PGGRPRI + PGGRPSGM) \times PGGRPSIZE}{SEGSIZE} \right\rceil$$

変数の説明

SEGSIZE

レンジインデクスを定義したDBエリアのセグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

計算式

SEGSIZE＝1セグメントの容量（単位：バイト）÷*page_size*

page_size

レンジインデクスを格納するDBエリアのページサイズ（単位：バイト）

PGGRPRI

次に示す計算式から求めてください。

計算式（単位：ページグループ数）

$$PGGRPRI = \left\lceil \frac{DBASGM - SUBAREANUM + 1}{\left\lfloor \frac{(page_size - 64) \times 8}{col_size \times 16 + 1} \right\rfloor \times PGGRPSIZE} \right\rceil + SUBAREANUM - 1$$

DBASGM

次に示す計算式から求めてください。

計算式（単位：個）

$$DBASGM = SGTBL + SGIDX + SGTIX + \sum_{i=1}^{rng_num_in_table_dbarea} ASGRI(i)$$

SUBAREANUM

次に示す計算式から求めてください。

計算式（単位：個）

$$SUBAREANUM = \sum_{i=1}^{cmd_num} \left(tbl_dbareafile_num(i) \times \left\lceil \frac{tbl_dbarea_initsize(i)}{16,384} \right\rceil \right)$$

cmd_num

レンジインデクスを定義した表が格納されたDBエリアを，次のコマンドで初期化した回数，または追加した回数の合計

- adbinitコマンド

- adbmodarea コマンド

tbl_dbareafile_num(i)

レンジインデクスを定義した表が格納された DB エリアを，i 番目に実行した次のコマンドで初期化したとき，または追加したときに，作成された DB エリアファイルの数（単位：個）

- adbinit コマンド
- adbmodarea コマンド

tbl_dbarea_initsize(i)

レンジインデクスを定義した表が格納された DB エリアを，i 番目に実行した次のコマンドで初期化したとき，または追加したときに，初期確保サイズのオプションに指定したサイズ（単位：ギガバイト）

- adbinit コマンド
- adbmodarea コマンド

SGTBL

レンジインデクスを定義した表，およびその表と同じデータ用 DB エリアに格納されている表を格納するセグメント数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の次の個所を参照してください。なお，表の種類によって参照先が異なります。

- シングルチャンク表かつローストア表の場合
  「(a)　変数 SGROWTBL の求め方（シングルチャンク表の場合）」
- シングルチャンク表かつカラムストア表の場合
  「(b)　変数 SGCOLUMNTBL の求め方（シングルチャンク表の場合）」
- マルチチャンク表かつローストア表の場合
  「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」
- マルチチャンク表かつカラムストア表の場合
  「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」

SGIDX

レンジインデクスを定義した表と同じデータ用 DB エリアに格納されている B-tree インデクスのセグメント数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の次の個所を参照してください。なお，表の種類によって参照先が異なります。

- シングルチャンク表の場合
  「(c)　変数 SGIDX の求め方（シングルチャンク表の場合）」
- マルチチャンク表の場合
  「(h)　変数 SGIDX の求め方（マルチチャンク表の場合）」

SGTIX

レンジインデクスを定義した表と同じデータ用 DB エリアに格納されているテキストインデクスのセグメント数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の次の個所を参照してください。なお，表の種類によって参照先が異なります。

- シングルチャンク表の場合

  「(e)　変数 SGTIX の求め方（シングルチャンク表の場合）」
- マルチチャンク表の場合

  「(j)　変数 SGTIX の求め方（マルチチャンク表の場合）」

rng_num_in_table_dbarea

レンジインデクスを定義した表が格納されているデータ用 DB エリア内のレンジインデクス数（単位：個）

ASGRI（i）

レンジインデクスを定義した表と同じデータ用 DB エリアに格納されているレンジインデクスのセグメント数（単位：個）

計算式（単位：個）

$$ASGRI(i) = SGSAM + \left\lceil \frac{SUBAREANUM \times PGGRPSIZE \times 2}{SEGSIZE} \right\rceil + \left\lceil \frac{SGTBL}{65,536} \right\rceil$$

SGSAM

「(2)　変数 SGSAM の求め方」を参照してください。

SGTBL

レンジインデクスを定義した表と同じデータ用 DB エリアに格納されている i 番目のレンジインデクスを定義した表のセグメント数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある次の個所を参照してください。なお，表の種類によって参照先が異なります。

- シングルチャンク表かつローストア表の場合

  「(a)　変数 SGROWTBL の求め方（シングルチャンク表の場合）」
- シングルチャンク表かつカラムストア表の場合

  「(b)　変数 SGCOLUMNTBL の求め方（シングルチャンク表の場合）」
- マルチチャンク表かつローストア表の場合

  「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」
- マルチチャンク表かつカラムストア表の場合

  「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」

col_size

レンジインデクスを定義した列のデータ長（単位：バイト）

各列のデータ長を求める場合は，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の「表 5-15　各データ型のデータ長の一覧」を参照してください。

PGGRPSGM

次に示す計算式から求めてください。

計算式（単位：ページグループ数）

$$PGGRPSGM = \left\lceil \frac{PGGRPRI}{\left\lfloor \frac{(page_size - 64) \times 8}{65} \right\rfloor \times PGGRPSIZE} \right\rceil + SUBAREANUM - 1$$

PGGRPSIZE

変数 *PGGRPSIZE* は，次に示す計算式で求めた変数 *TMPPGGRPSIZE* 以上の，最小の 2 のべき乗数となります。

重要

- 変数 *PGGRPSIZE* の計算結果が255 より大きくなった場合は，変数 *PGGRPSIZE* に255 を代入してください。
- 対象のレンジインデクスが，バージョン 02-02 より前に定義されたレンジインデクスであり，かつadbidxrebuild コマンドを一度も実行したことがないレンジインデクスである場合，変数 *PGGRPSIZE* には変数 *SEGSIZE* の値を代入してください。

計算式

$$TMPPGGRPSIZE = \left\lceil \frac{4 \times \sqrt{TBLSGMTMAXNUM \times col_size}}{rng_page_size} \right\rceil$$

TBLSGMTMAXNUM

次に示す計算式から求めてください。

計算式

$$TBLSGMTMAXNUM = \left\lceil \frac{17,592,186,044,416}{TABLESGMTCAPA} \right\rceil$$

TABLESGMTCAPA

レンジインデクスを定義した表を格納するデータ用 DB エリアの 1 セグメントの容量（単位：バイト）

rng_page_size

レンジインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

(2)　変数 SGSAM の求め方

変数 *SGSAM* は，次に示す計算式から求めてください。

計算式（単位：セグメント数）

$$SGSAM = \left\lceil \frac{SUBAREAPTRNUM + \left\lceil \frac{col_size}{4} \right\rceil}{\left\lfloor \frac{(page_size - 64) \times 8}{65} \right\rfloor \times SEGSIZE} \right\rceil$$

変数の説明

SUBAREAPTRNUM

次に示す計算式から求めてください。

計算式（単位：個）

$$SUBAREAPTRNUM = \sum_{i=1}^{cmd_num} \left(tbl_dbareafile_num(i) \right) + \left\lfloor \frac{tbl_dbarea_max_initsize}{16,384} \right\rfloor \times 8,192$$

cmd_num

レンジインデクスを定義した表が格納されたDBエリアを，次のコマンドで初期化した回数，または追加した回数の合計

- adbinitコマンド
- adbmodareaコマンド

tbl_dbareafile_num(i)

レンジインデクスを定義した表が格納されたDBエリアを，i番目に実行した次のコマンドで初期化したとき，または追加したときに，作成されたDBエリアファイルの数（単位：個）

- adbinitコマンド
- adbmodareaコマンド

tbl_dbarea_max_initsize

レンジインデクスを定義した表が格納されたDBエリアを，次のコマンドで初期化したとき，または追加したときに，実行したすべてのコマンドの中で，初期確保サイズのオプションに指定したサイズが最も大きいサイズ（単位：ギガバイト）

- adbinitコマンド
- adbmodareaコマンド

col_size

レンジインデクスを定義した列のデータ長（単位：バイト）

各列のデータ長を求める場合は，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数BP（i））の求め方」の「表5-15　各データ型のデータ長の一覧」を参照してください。

なお，対象のレンジインデクスがチャンクのスキップができないレンジインデクスの場合，変数*col_size*には0 を代入してください。対象のレンジインデクスがチャンクのスキップができるかどうかを確認する場合は「11.3.6　レンジインデクスの確認（チャンクのスキップの可否）」を参照してください。

page_size

レンジインデクスを格納する DB エリアのページサイズ（単位：バイト）

SEGSIZE

セグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

計算式

SEGSIZE＝ *1セグメントの容量*（単位：バイト）÷*page_size*

5.9　作業表用 DB エリアの容量見積もり

作業表用 DB エリアの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

作業表用DBエリアの容量 ＝ *WRK_PAGE_NUM* ×*page_size* ÷1,024

変数の説明

WRK_PAGE_NUM：作業表用 DB エリアの総ページ数

「5.9.1　作業表用 DB エリアの総ページ数の求め方」を参照して求めてください。

page_size：作業表用 DB エリアのページサイズ

「5.7　作業表用 DB エリアの設計」を基に決めた，初期設定オプションadb_init_wrk_page_size の指定値を代入してください。

ただし，adb_init_wrk_page_size の指定値の単位はキロバイトですが，変数*page_size* の単位はバイトなので注意してください。

初期設定オプションadb_init_wrk_page_size については，マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』を参照してください。

5.9.1　作業表用 DB エリアの総ページ数の求め方

作業表用 DB エリアの総ページ数（変数 *WRK_PAGE_NUM*）は，次に示す計算式から求めてください。

計算式

$$WRK_PAGE_NUM = 1 + dbarea_file_num + \sum_{i=1}^{tbl_num} WP(i)$$

変数の説明

dbarea_file_num

作業表用 DB エリアファイルの数（単位：個）

作業表用 DB エリアファイルは1 個であるため，変数dbarea_file_num には 1 を代入してください。

tbl_num

同時に作成する作業表の総数（単位：個）

WP（i)

作業表を格納するために必要な基本行用ページ数

変数 *WP（i)* については，「5.9.2　作業表を格納するために必要な基本行用ページ数の求め方」を参照して求めてください。

5.9.2　作業表を格納するために必要な基本行用ページ数の求め方

作業表を格納するために必要な基本行用ページ数（変数 *WP（i）*）は，次に示す計算式から求めてください。

計算式

$$WP(i)=\left\lceil\frac{row_num}{\left\lfloor\frac{page_size-64}{ROWSZ}\right\rfloor}\right\rceil+\left\lceil\frac{WORK_ARRAY_ROWSZ\times row_num}{page_size-64}\right\rceil$$

変数の説明

row_num

表に格納する行数（単位：個）

page_size

作業表用 DB エリアのページサイズ（単位：バイト）

「5.7　作業表用 DB エリアの設計」を基に決めた，初期設定オプションadb_init_wrk_page_size の指定値を代入してください。

ただし，adb_init_wrk_page_size の指定値の単位はキロバイトですが，変数page_size の単位はバイトです。単位が異なるため，注意してください。

初期設定オプションadb_init_wrk_page_size については，マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』を参照してください。

変数*ROWSZ* については，次に示す計算式から求めてください。

なお，計算式で求めた値が「*ROWSZ* ＞ 32,704」となる場合は，計算式で求めた値に，「2×（*col_num* ＋ 1)」で求めた値を加算してください。

計算式

$$ROWSZ=\left\lceil\frac{28+2\times col_num+\sum_{i=1}^{col_num}col_size(i)}{2}\right\rceil\times 2$$

変数の説明

col_num：作業表を構成する列の総数（単位：個）

col_size（i）：各列のデータ長（単位：バイト）

変数*col_size（i）* については，次に示す表を基に各列のデータ長を求めてください。そのあと，すべての列の合計値を求めてください。

表 5-20　各データ型のデータ長の一覧

項番	分類	データ型		データ長（単位：バイト）
1	数データ	INTEGER		8
2		SMALLINT		4
3		DECIMAL(*m*,*n*)※ NUMERIC(*m*,*n*)※	1≦*m*≦4	2
4			5≦*m*≦8	4
5			9≦*m*≦16	8
6			17≦*m*≦38	16
7		DOUBLE PRECISION		8
8		FLOAT		
9	文字データ	CHARACTER(*n*)		*n*
10		VARCHAR(*n*)		*wrk_var_col_size*
11	日時データ	DATE		4
12		TIME(*p*)		3 + ↑*p*÷2↑
13		TIMESTAMP(*p*)		7 + ↑*p*÷2↑
14	バイナリデータ	BINARY(*n*)		*n*
15		VARBINARY(*n*)		*d* + 2
16	配列データ	ARRAY		14

（凡例）

m，*n*：正の整数

d：実際のデータ長

p：0，3，6，9 または 12

注※

全体の桁数が *m* 桁で，小数点以下の桁数が *n* 桁の固定小数点数を示します。*m* を省略した場合は 38 が仮定されます。

wrk_var_col_size：作業表に関するVARCHAR 型の列のデータ長

次に示す表を基にデータ長を求めてください。

表 5-21　作業表に関する VARCHAR 型の列のデータ長の一覧

項番	定義長 n（単位：バイト）	データ長（単位：バイト）
1	1≦*n*≦32,000	*d* + 2
2	32,001≦*n*≦64,000	*d* + 8

（凡例）

n：正の整数

d：実際のデータ長

変数 *WORK_ARRAY_ROWSZ* については，次に示す計算式から求めてください。

計算式

WORK_ARRAY_ROWSZ＝

$$\left\lceil \frac{\sum_{i=1}^{col_num} (14 + (col_size(i) + 1) \times elm_num(i))}{2} \right\rceil \times 2$$

変数の説明

col_num

作業表を構成する列の総数

col_size（i）

各列のデータ長（単位：バイト）

「表 5-20　各データ型のデータ長の一覧」を基に各列のデータ長を求めてください。そのあと，すべての列の合計値を求めてください。

elm_num（i）

列の最大要素数

配列型以外の列の場合は，0 としてください。

5.10　マスタディレクトリ用 DB エリアの容量見積もり

マスタディレクトリ用 DB エリアの容量は，8.1 メガバイトとして見積もってください。

なお，マスタディレクトリ用 DB エリアを構成するマスタディレクトリ用 DB エリアファイル（ファイル名：ADBMST）は，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに 1 つだけ作成されます。

5.11　ディクショナリ用 DB エリアの容量見積もり

ディクショナリ用 DB エリアの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
ディクショナリ用DBエリアの容量＝
        ADBDICSIZE_INIT＋ADBDICSIZE_OPERATION
```

変数の説明

ADBDICSIZE_INIT

データベースの大きさから求められるディクショナリ用 DB エリアの容量

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
ADBDICSIZE_INIT＝
        ↑8,200＋USERSCM×1.1＋USERTBL×9,000＋USERIDX×31＋USERVIEW×1,100
        ＋DBAREANUM＋USERNUM×71＋USERFKEYNUM×42＋USERARCHIVE×82↑
```

USERSCM

HADB ユーザが定義したスキーマの数

USERTBL

HADB ユーザが定義した実表の数

変数 *USERTBL* には，アーカイブマルチチャンク表の数も含みます。

USERIDX

HADB ユーザが定義したインデクスの数

USERVIEW

HADB ユーザが定義したビュー表の数

DBAREANUM

DB エリアの数

USERNUM

CREATE USER 文で作成した HADB ユーザの数＋ 1

1 は，adbinit コマンド実行時に作成された HADB ユーザになります。

USERFKEYNUM

HADB ユーザが定義した外部キーの数

USERARCHIVE

アーカイブマルチチャンク表の数

ADBDICSIZE_OPERATION

データベースの運用に従って求められるディクショナリ用 DB エリアの容量

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
ADBDICSIZE_OPERATION=
    ↑CREATE_USER＋CREATE_SCHEMA＋CREATE_TABLE＋CREATE_VIEW＋CREATE_INDEX＋CREATE_AUDIT
      ＋GRANT＋ALTER_USER＋ALTER_TABLE＋ALTER_VIEW＋DROP_USER＋DROP_SCHEMA＋DROP_TABLE
      ＋DROP_VIEW＋DROP_INDEX＋REVOKE↑
```

CREATE_USER

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CREATE_USER ＝ 4×*CREATE_USER_EXEC_NUM*

CREATE_USER_EXEC_NUM

CREATE USER 文を実行する回数を代入してください。

CREATE_SCHEMA

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CREATE_SCHEMA ＝ 4×*CREATE_SCHEMA_EXEC_NUM*

CREATE_SCHEMA_EXEC_NUM

CREATE SCHEMA 文を実行する回数を代入してください。

CREATE_TABLE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CREATE_TABLE ＝ 596×*CREATE_TABLE_EXEC_NUM*

CREATE_TABLE_EXEC_NUM

CREATE TABLE 文を実行する回数を代入してください。

CREATE_VIEW

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CREATE_VIEW ＝ 742×*CREATE_VIEW_EXEC_NUM*

CREATE_VIEW_EXEC_NUM

CREATE VIEW 文を実行する回数を代入してください。

CREATE_INDEX

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CREATE_INDEX ＝ 24×*CREATE_INDEX_EXEC_NUM*

CREATE_INDEX_EXEC_NUM

CREATE INDEX 文を実行する回数を代入してください。

CREATE_AUDIT

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CREATE_AUDIT ＝ 472 × *CREATE_AUDIT_EXEC_NUM*

CREATE_AUDIT_EXEC_NUM

CREATE AUDIT 文を実行する回数を代入してください。

GRANT

次に示す計算式から求めてください。

計算式（単位：キロバイト）

GRANT ＝ 656×*GRANT_EXEC_NUM*

GRANT_EXEC_NUM

GRANT 文を実行する回数を代入してください。

ALTER_USER

次に示す計算式から求めてください。

計算式（単位：キロバイト）

ALTER_USER ＝ 4×*ALTER_USER_EXEC_NUM*

ALTER_USER_EXEC_NUM

ALTER USER 文を実行する回数を代入してください。

ALTER_TABLE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
ALTER_TABLE =
   ADD_COLUMN + RENAME_TABLE + RENAME_COLUMN
   + CHANGE_CHUNK_MAX + CHANGE_CHUNK_ARCHIVE + CHANGE_CHUNK_UNARCHIVE
```

ADD_COLUMN

次に示す計算式から求めてください。

計算式（単位：キロバイト）

ADD_COLUMN ＝ 36×*ADD_COLUMN_EXEC_NUM*

ADD_COLUMN_EXEC_NUM

ALTER TABLE 文で，表に列を追加する回数を代入してください。

RENAME_TABLE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RENAME_TABLE ＝ 692×*RENAME_TABLE_EXEC_NUM*

RENAME_TABLE_EXEC_NUM

ALTER TABLE 文で，表名を変更する回数を代入してください。

RENAME_COLUMN

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RENAME_COLUMN ＝ 76×*RENAME_COLUMN_EXEC_NUM*

RENAME_COLUMN_EXEC_NUM

ALTER TABLE 文で，表の列名を変更する回数を代入してください。

CHANGE_CHUNK_MAX

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CHANGE_CHUNK_MAX ＝ 4×*CHANGE_CHUNK_MAX_EXEC_NUM*

CHANGE_CHUNK_MAX_EXEC_NUM

ALTER TABLE 文で，マルチチャンク表のチャンク数の最大値を変更する回数を代入してください。

CHANGE_CHUNK_ARCHIVE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CHANGE_CHUNK_ARCHIVE ＝ 84×*CHANGE_CHUNK_ARCHIVE_EXEC_NUM*

CHANGE_CHUNK_ARCHIVE_EXEC_NUM

ALTER TABLE 文で，レギュラーマルチチャンク表からアーカイブマルチチャンク表に変更する回数を代入してください。

CHANGE_CHUNK_UNARCHIVE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CHANGE_CHUNK_UNARCHIVE ＝ 48×*CHANGE_CHUNK_UNARCHIVE_EXEC_NUM*

CHANGE_CHUNK_UNARCHIVE_EXEC_NUM

ALTER TABLE 文で，アーカイブマルチチャンク表からレギュラーマルチチャンク表に変更する回数を代入してください。

ALTER_VIEW

次に示す計算式から求めてください。

計算式（単位：キロバイト）

ALTER_VIEW ＝ 678×*ALTER_VIEW_EXEC_NUM*

ALTER_VIEW_EXEC_NUM

ALTER VIEW 文を実行する回数を代入してください。

DROP_USER

次に示す計算式から求めてください。

計算式（単位：キロバイト）

DROP_USER ＝ 1,992×*DROP_USER_EXEC_NUM*

DROP_USER_EXEC_NUM

DROP USER 文を実行する回数を代入してください。

DROP_SCHEMA

次に示す計算式から求めてください。

計算式（単位：キロバイト）

DROP_SCHEMA ＝ 1,992×*DROP_SCHEMA_EXEC_NUM*

DROP_SCHEMA_EXEC_NUM

DROP SCHEMA 文を実行する回数を代入してください。

DROP_TABLE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

DROP_TABLE ＝ 204×*DROP_TABLE_EXEC_NUM*

DROP_TABLE_EXEC_NUM

DROP TABLE 文を実行する回数を代入してください。

DROP_VIEW

次に示す計算式から求めてください。

計算式（単位：キロバイト）

DROP_VIEW ＝ 36×*DROP_VIEW_EXEC_NUM*

DROP_VIEW_EXEC_NUM

DROP VIEW 文を実行する回数を代入してください。

DROP_INDEX

次に示す計算式から求めてください。

計算式（単位：キロバイト）

DROP_INDEX = 8×*DROP_INDEX_EXEC_NUM*

DROP_INDEX_EXEC_NUM

DROP INDEX 文を実行する回数を代入してください。

REVOKE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

REVOKE = 1,992×*REVOKE_EXEC_NUM*

REVOKE_EXEC_NUM

REVOKE 文を実行する回数を代入してください。

なお，ディクショナリ用 DB エリアを構成するディクショナリ用 DB エリアファイル（ファイル名：ADBDIC）は，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに 1 つだけ作成されます。

5.12　システム表用 DB エリアの容量見積もり

システム表用 DB エリアの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
システム表用DBエリアの容量＝
        10,000＋STBLSIZE_SUM＋REORGSDAREASIZE
```

変数の説明

STBLSIZE_SUM

各システム表の容量を合計した値

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STBLSIZE_SUM＝
    STBLTABLESSIZE＋STBLCOLUMNSSIZE＋STBLINDEXESSIZE
      ＋STBLCHUNKSSIZE＋STBLSYNONYMDICSIZE
```

STBLTABLESSIZE

STATUS_TABLES 表の容量

STBLCOLUMNSSIZE

STATUS_COLUMNS 表の容量

STBLINDEXESSIZE

STATUS_INDEXES 表の容量

STBLCHUNKSSIZE

STATUS_CHUNKS 表の容量

STBLSYNONYMDICSIZE

STATUS_SYNONYM_DICTIONARIES 表の容量

各システム表の容量は「5.12.1　システム表の容量見積もり」を参照して求めてください。

REORGSDAREASIZE

adbreorgsystemdata コマンドの実行用領域の容量

adbreorgsystemdata コマンドを実行する場合に加算してください。

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
REORGSDAREASIZE＝
    MAX（STBLTABLESSIZE, STBLCOLUMNSSIZE, STBLINDEXESSIZE
        , STBLCHUNKSSIZE, STBLSYNONYMDICSIZE）
```

次に示す各システム表の容量の中で，最も大きな値です。

STBLTABLESSIZE

STATUS_TABLES 表の容量

STBLCOLUMNSSIZE

STATUS_COLUMNS 表の容量

STBLINDEXESSIZE

STATUS_INDEXES 表の容量

STBLCHUNKSSIZE

STATUS_CHUNKS 表の容量

STBLSYNONYMDICSIZE

STATUS_SYNONYM_DICTIONARIES 表の容量

各システム表の容量は「5.12.1　システム表の容量見積もり」を参照して求めてください。

なお，システム表用 DB エリアを構成するシステム表用 DB エリアファイル（ファイル名：ADBSTBL）は，DB ディレクトリ下のレギュラーファイル，または DB ディレクトリ下以外のブロックスペシャルファイルに 1 つだけ作成されます。

5.12.1　システム表の容量見積もり

ここでは，次に示すシステム表の容量を見積もる方法について説明します。

- STATUS_TABLES 表
  「(1)　STATUS_TABLES 表の容量見積もり」を参照して求めてください。
- STATUS_COLUMNS 表
  「(2)　STATUS_COLUMNS 表の容量見積もり」を参照して求めてください。
- STATUS_INDEXES 表
  「(3)　STATUS_INDEXES 表の容量見積もり」を参照して求めてください。
- STATUS_CHUNKS 表
  「(4)　STATUS_CHUNKS 表の容量見積もり」を参照して求めてください。
- STATUS_SYNONYM_DICTIONARIES 表
  「(5)　STATUS_SYNONYM_DICTIONARIES 表の容量見積もり」を参照して求めてください。

(1)　STATUS_TABLES 表の容量見積もり

STATUS_TABLES 表の容量（*STBLTABLESSIZE*）は，次に示す計算式から求めてください。

なお，adbgetcst コマンドを実行しない場合は 0 を代入してください。

計算式（単位：キロバイト）

```
STBLTABLESSIZE＝
   0.4×(COSTTBL_NUM×(1＋GETCOST_NUM)＋ALTERTBL_NUM)
```

変数の説明

COSTTBL_NUM

コスト情報を収集する表の数

GETCOST_NUM

1 つの表にadbgetcst コマンドを実行する回数

adbreorgsystemdata コマンドを実行してから再実行するまでの間に，adbgetcst コマンドを実行する回数を見積もってください。

adbreorgsystemdata コマンドを実行する間隔については，「11.17.2　システム表を再編成するタイミング」を参照してください。

次の条件で運用する場合を例に説明します。

■運用の条件

- 1 日に 1 回，コスト情報を収集する
- 6 か月に 1 回，システム表（実表）の再編成を行う

■条件を基にした変数 GETCOST_NUM の求め方

```
GETCOST_NUM＝ A×B

  ■変数の説明
    A：1日の間にadbgetcstコマンドを実行する頻度（単位：回数）
       1を代入してください。
       （1日に1回，実行するとします。）
    B：adbreorgsystemdataコマンドを実行する頻度（単位：日）
       186を代入してください。
       （6か月に1回，実行するとします。また，1か月は31日とします。「6×31」で求めます。）
```

上記の条件で運用する場合，変数*GETCOST_NUM* には，186 を代入してください。

ALTERTBL_NUM

ALTER TABLE 文を使用して，表名を変更する回数

(2)　STATUS_COLUMNS 表の容量見積もり

STATUS_COLUMNS 表の容量（*STBLCOLUMNSSIZE*）は，次に示す計算式から求めてください。

なお，adbgetcst コマンドを実行しない場合は 0 を代入してください。

計算式（単位：キロバイト）

```
STBLCOLUMNSSIZE＝
   6.4×（COSTCOL_NUM×（1＋GETCOST_NUM）＋ALTTBLCOL_NUM＋ALTERCOL_NUM）
```

変数の説明

COSTCOL_NUM

コスト情報を収集する表の列数を合計した値

例えば，コスト情報を収集する表A の列数が 10，表B の列数が 20 の場合は，30 となります。

GETCOST_NUM

「(1)　STATUS_TABLES 表の容量見積もり」の変数 *GETCOST_NUM* を参照してください。

ALTTBLCOL_NUM

ALTER TABLE 文を使用して，表名を変更する表の列数を合計した値

例えば，表名を 2 回変更する表C の列数が 10，表名を 1 回変更する表D の列数が 20 の場合は，次に示すとおり 40 となります。

```
10×2＋20＝40
```

ALTERCOL_NUM

ALTER TABLE 文で列の名称を変更する回数

(3)　STATUS_INDEXES 表の容量見積もり

STATUS_INDEXES 表の容量（*STBLINDEXESSIZE*）は，次に示す計算式から求めてください。

なお，adbgetcst コマンドを実行しない場合は 0 を代入してください。

計算式（単位：キロバイト）

```
STBLINDEXESSIZE＝
  0.4×(COSTIDX_NUM×(1＋GETCOST_NUM)＋ALTTBLIDX_NUM)
```

変数の説明

COSTIDX_NUM

コスト情報を収集する表に定義されたインデクス数を合計した値

例えば，コスト情報を収集する表A に定義されたインデクス数が 2，表B に定義されたインデクス数が 1 の場合は，3 となります。

GETCOST_NUM

「(1)　STATUS_TABLES 表の容量見積もり」の変数 *GETCOST_NUM* を参照してください。

ALTTBLIDX_NUM

ALTER TABLE 文を使用して，表名を変更する表に定義されたインデクス数を合計した値

例えば，表名を 2 回変更する表C に定義されたインデクス数が 2，表名を 1 回変更する表D に定義されたインデクス数が 1 の場合は，次に示すとおり 5 となります。

```
2×2＋1＝5
```

(4) STATUS_CHUNKS 表の容量見積もり

STATUS_CHUNKS 表の容量（*STBLCHUNKSSIZE*）は，次に示す計算式から求めてください。

なお，マルチチャンク表を定義しない場合は 0 を代入してください。

計算式（単位：キロバイト）

```
STBLCHUNKSSIZE＝
   (0.4＋CHUNK_COMMENT_SIZE)×(STBLCHUNKS_ROWNUM＋ALTTBLCNK_NUM)
```

変数の説明

CHUNK_COMMENT_SIZE

チャンクのコメントの平均データ長（単位：キロバイト）

STBLCHUNKS_ROWNUM

STATUS_CHUNKS 表の行数

次に示す計算式から求めてください。

計算式

```
STBLCHUNKS_ROWNUM＝TOTAL_CHUNKNUM＋STBLCHUNK_INVALID_ROWNUM
```

TOTAL_CHUNKNUM

運用中の各マルチチャンク表のチャンク数を合計した値

例えば，マルチチャンク表A のチャンク数が 700，マルチチャンク表B のチャンク数が 600 の場合は，1,300 となります。

STBLCHUNK_INVALID_ROWNUM

STATUS_CHUNKS 表に発生する無効な行データの行数

adbreorgsystemdata コマンドを実行してから再実行するまでの間に，STATUS_CHUNKS 表に発生する無効な行データの行数を見積もってください。計算式を次に示します。

計算式

```
STBLCHUNK_INVALID_ROWNUM＝ reorg_period×backgroundimport_table_num×INVALID_ROW_NUM
```

reorg_period

adbreorgsystemdata コマンドを実行する間隔（単位：月）

例えば，6 か月に 1 回，adbreorgsystemdata コマンドを実行する場合は，6 を代入してください。

adbreorgsystemdata コマンドを実行する間隔については，「11.17.2　システム表を再編成するタイミング」を参照してください。

backgroundimport_table_num

バックグラウンドインポートを行うマルチチャンク表の数（単位：個数）

例えば，毎日５つのマルチチャンク表にバックグラウンドインポートを行う場合は，５を代入してください。

INVALID_ROW_NUM

1 つのマルチチャンク表に対して，1 か月の間に発生する無効な行データの数

マルチチャンク表に対して想定される SQL 文とコマンドの実行回数を見積もってください。そして，次に示す表から，発生する無効な行データの行数を求めてください。

表 5-22　SQL 文やコマンドと STATUS_CHUNKS 表に発生する無効な行データの行数の関係

項番	マルチチャンク表に対して実行する SQL 文とコマンド	STATUS_CHUNKS 表に発生する無効な行データの行数※1
1	定義系 SQL の実行 マルチチャンク表を削除する場合が対象です。	各削除対象表のチャンク数の総和
2	PURGE CHUNK 文の実行	削除したチャンク数
3	TRUNCATE TABLE 文の実行	処理対象表のチャンク数
4	adbimport コマンドの実行 作成モードで実行する場合が対象です。	処理対象表のチャンク数
5	adbimport コマンドの実行 バックグラウンドインポートで実行する場合が対象です。	1
6	adbidxrebuild コマンドの実行 adbimport コマンドによるバックグラウンドインポートが中断した場合が対象です。	1
7	adbmergechunk コマンドの実行	マージ対象のチャンク数×2※2
8	adbchgchunkcomment コマンドの実行	1
9	adbchgchunkstatus コマンドの実行	状態を変更するチャンク数
10	adbarchivechunk コマンドの実行	アーカイブするチャンク数
11	adbunarchivechunk コマンドの実行	アーカイブ状態を解除するチャンク数

注※1

左記の列「マルチチャンク表に対して実行する SQL 文とコマンド」を 1 回実行した場合に，発生する無効な行データの行数です。複数回実行する場合は，その分だけ加算する必要があります。

注※2

次の理由でマージ元チャンクを削除できなかった場合は，無効な行データは発生しません。

・実行中のadbmergechunk コマンドが，マージ先チャンクの完成からマージ元チャンクの削除完了までの間に中断した場合

・adbmergechunk コマンドの--purge-chunk オプションにNOWAIT を指定して実行した場合に，マージチャンク対象の表を参照している処理が存在したとき

変数 STBLCHUNK_INVALID_ROWNUM の計算例

変数*STBLCHUNK_INVALID_ROWNUM* の求め方について，次に示す条件で運用する場合を例に説明します。

■運用の条件

- 6 か月に 1 回，システム表（実表）の再編成を行う
- 10 個のマルチチャンク表に対して，バックグラウンドインポートを行う
- 5 分に 1 回（1 日で 288 回），バックグラウンドインポートを行う
- バックグラウンドインポートで作成された 288 個のチャンクを，1 日分のチャンクとしてマージする
- 1 日分のチャンク 31 個を，1 か月分のチャンクとして，月末にマージする（1 か月は 31 日と仮定する）
- 2 年以上経過した 1 か月分のチャンク 1 個を，月末に削除する

■条件を基にした変数 STBLCHUNK_INVALID_ROWNUM の求め方

変数*reorg_period* は，adbreorgsystemdata コマンドを実行する間隔であるため，6 を代入してください。

変数*backgroundimport_table_num* は，バックグラウンドインポートを行うマルチチャンク表の数であるため，10 を代入してください。

変数*INVALID_ROW_NUM* は，1 つのマルチチャンク表に対して，1 か月の間に発生する無効な行データの数であるため，次の計算式から求めます。

変数 INVALID_ROW_NUM の計算式

```
INVALID_ROW_NUM＝ backgroundimport_exec_num＋mergechunk_chunk_num×2＋purgechunk_chunk_num

  ■変数の説明
    backgroundimport_exec_num
      1か月の間にバックグラウンドインポートを行う回数
      8,928を代入してください。
       （1日分の実行回数は288回，1か月は31日とします。「288×31」で求めます。）
    mergechunk_chunk_num
      1か月の間にマージするチャンク数
      8,959を代入してください。
       （1日分としてマージするチャンクは288個，1か月は31日とします。
         また，1か月分として月末にマージするチャンクは31個とします。
         「288×31＋31」で求めます。）
    purgechunk_chunk_num
      1か月で削除するチャンク数
      1を代入してください。
       （2年以上経過した1か月分のチャンク1個を削除するとします。）
```

変数*INVALID_ROW_NUM* には，26,847 を代入してください。

各変数の値を代入して，変数*STBLCHUNK_INVALID_ROWNUM* を求めてください。6 か月の間にSTATUS_CHUNKS 表に発生する無効な行データの数は，1,610,820 であると見積もることができます。

ALTTBLCNK_NUM

ALTER TABLE 文を使用して，表名を変更するマルチチャンク表のチャンク数を合計した値

例えば，表名を 2 回変更するマルチチャンク表C のチャンク数が 700，表名を 1 回変更するマルチチャンク表D のチャンク数が 600 の場合は，次に示すとおり 2,000 となります。

```
700×2＋600＝2,000
```

マルチチャンク表のチャンク数については，「付録 C.9　システム表の検索」の「(3)　表名から表内の全チャンクの情報を調べる場合」を参照してください。SQL 文の検索結果の行数が，マルチチャンク表のチャンク数になります。

(5)　STATUS_SYNONYM_DICTIONARIES 表の容量見積もり

STATUS_SYNONYM_DICTIONARIES 表の容量（*STBLSYNONYMDICSIZE*）は，次に示す計算式から求めてください。

なお，同義語辞書を 1 つも作成しない場合は 0 を代入してください。

計算式（単位：キロバイト）

```
STBLSYNONYMDICSIZE＝
    (0.8＋MAX（BINARY_PATH_SIZE, SYNONYMDIC_COMMENT_SIZE）×2)
    ×（SYNONYMDIC_NUM＋SYNONYMDIC_INVALID_ROWNUM）
```

変数の説明

BINARY_PATH_SIZE

同義語辞書ファイルの格納ディレクトリのパス名のデータ長（単位：キロバイト）

SYNONYMDIC_COMMENT_SIZE

同義語辞書のコメントの平均データ長（単位：キロバイト）

SYNONYMDIC_NUM

作成する同義語辞書の数

SYNONYMDIC_INVALID_ROWNUM

STATUS_SYNONYM_DICTIONARIES 表に発生する無効な行データの行数

adbreorgsystemdata コマンドを実行してから再実行するまでの間に，STATUS_SYNONYM_DICTIONARIES 表に発生する無効な行データの行数を見積もってください。adbreorgsystemdata コマンドを実行する間隔については，「11.17.2　システム表を再編成するタイミング」を参照してください。

次に示す表から，STATUS_SYNONYM_DICTIONARIES 表に発生する無効な行データの行数を求めてください。その際，対象のコマンドを実行する回数も見積もってください。

表 5-23　コマンドと STATUS_SYNONYM_DICTIONARIES 表に発生する無効な行データの行数の関係

項番	実行するコマンド	STATUS_SYNONYM_DICTIONARIES 表に発生する無効な行データの行数※
1	adbsyndict コマンドの実行 同義語辞書を更新する場合が対象です。	更新した同義語辞書の数
2	adbsyndict コマンドの実行 同義語辞書を削除する場合が対象です。	削除した同義語辞書の数

注※

左記の列「実行するコマンド」を 1 回実行した場合に，発生する無効な行データの行数です。複数回実行する場合は，その分だけ加算する必要があります。

6 リソースの設計

この章では，ディスクの設計，および各リソースの見積もりについて説明します。

6.1　ディスクの設計

ここでは，ディスクの設計時に考慮する点について説明します。

6.1.1　ディスク設計時の考慮点

HADB の処理性能を向上させるには，I/O が分散するようにディスク設計することが重要なポイントになります。例えば，データを 1 つのディスクに格納した場合，そのディスクに対して I/O が集中します。ディスクの処理性能を超える数の I/O が発生すると，I/O 待ちによって処理性能が低下します。そのため，データを複数のディスクに分散させて，1 つのディスクに対する I/O を減らすように設計します。

特に，大規模なデータを扱う場合，次の表に示すファイルを格納するディスクを分けると I/O の分散に効果があります。

表 6-1　格納先のディスクを分散すると性能上の効果があるファイル

項番	ファイルまたはディレクトリの種類	ファイルの詳細
1	DB エリアファイル	• マスタディレクトリ用 DB エリアファイル • ディクショナリ用 DB エリアファイル • システム表用 DB エリアファイル • 作業表用 DB エリアファイル • データ用 DB エリアファイル
2	HADB サーバが使用するディレクトリおよびファイル	• サーバディレクトリとサーバディレクトリ下のファイル • DB ディレクトリと DB ディレクトリ下のファイル
3	作業用一時ファイル	次に示すコマンドの実行時に作成される作業用一時ファイル • adbimport コマンド • adbidxrebuild コマンド • adbmergechunk コマンド • adbunarchivechunk コマンド • adbreorgsystemdata コマンド
4	入力データファイル	• adbimport コマンドの実行時に使用する入力データファイル • ADB_CSVREAD 関数の実行時に使用する入力データファイル（CSV ファイル）
5	出力データファイル	adbexport コマンドの実行時に使用する出力データファイル
6	アーカイブディレクトリ	アーカイブファイル
7	同義語辞書ファイルの格納ディレクトリ	同義語辞書ファイル
8	アンロードファイル	adbreorgsystemdata コマンドの実行時に作成されるアンロードファイル

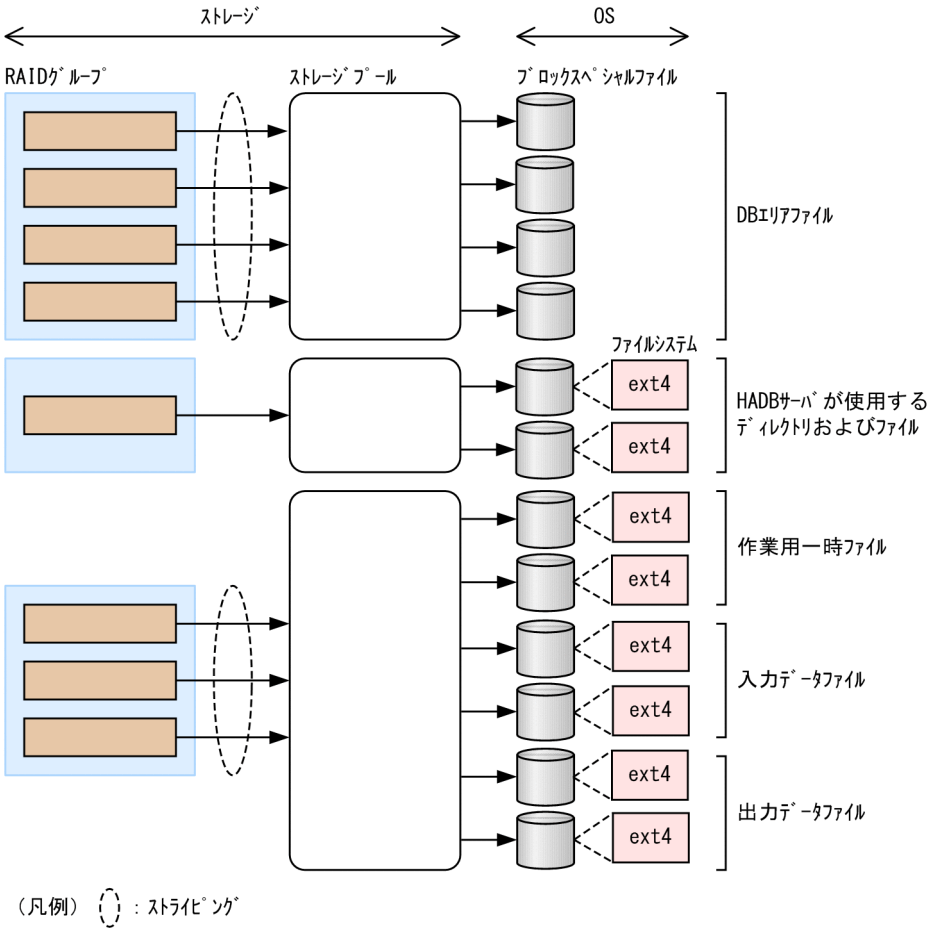
上記の表で説明しているファイルの格納先ディスクを分散する場合，通常は RAID 構成にします。RAID グループ（RAID 構成のディスク群）が複数ある場合は，次のどちらかの構成にすると，より I/O を分散させることができます。

- ストレージの機能によるディスクの仮想化でデータを分散配置する
- OS の機能である LVM を使用してストライピング構成にする

■例 1

ストレージの機能によるディスクの仮想化でデータを分散配置する例を次の図に示します。

図 6-1　ストレージの機能によるディスクの仮想化でデータを分散配置する例



上記の例では，DB エリアファイル用の 4 つの RAID グループを，まとめて 1 つのストレージプール（仮想ディスク）としています。このとき，4 つの RAID グループに分散してデータを記憶する方式（ストライピング）を使用しています。

仮想ディスクで幾つかのロジカルユニットを作成し，DB エリアファイルを構成するブロックスペシャルファイルとして割り当てます。これによって，DB エリアファイル中のデータにアクセスするときは，4 つの RAID グループに対してほぼ均等に I/O が発生します。そのため，RAID グループが 1 つのときに比べてより多くの I/O を発行できます。

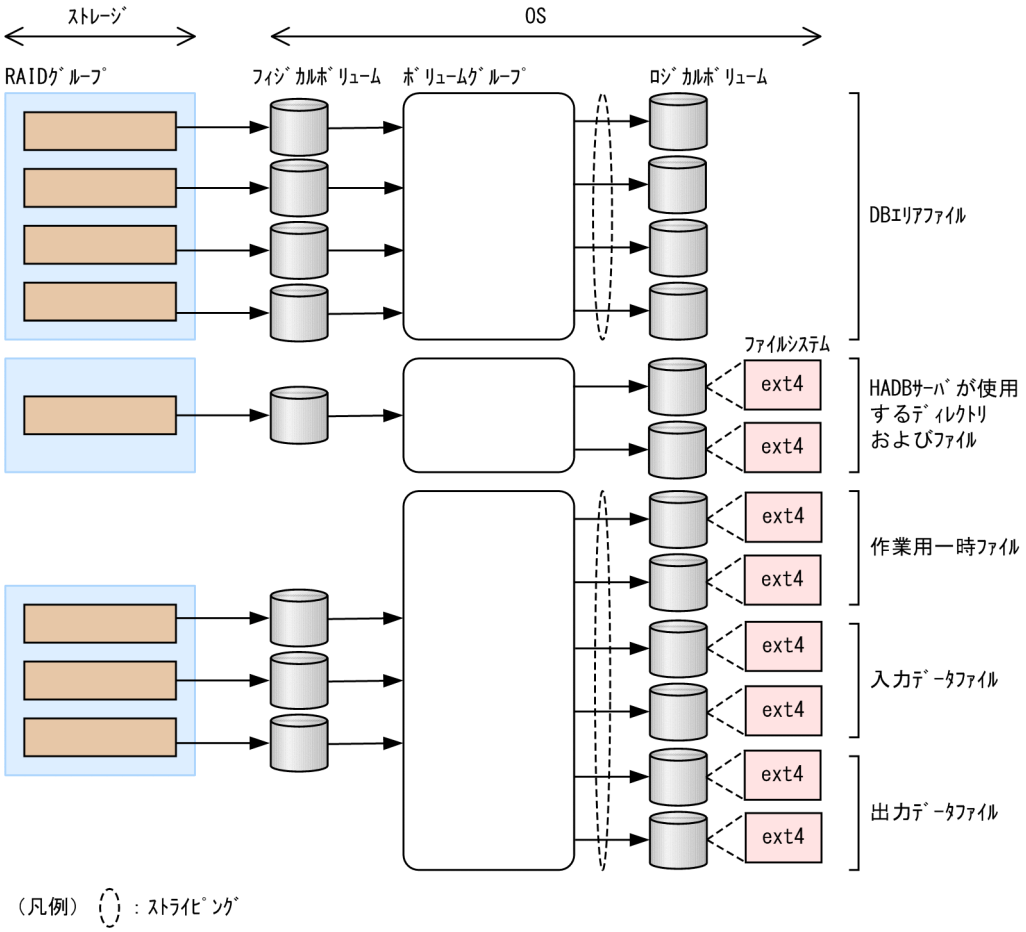
また，上記の例では，DB エリアファイル用のストレージプールと，そのほかのストレージプールを分けていますが，使用できる RAID グループが少ない場合は，ストレージプールを 1 つにして DB エリアファイルの I/O 分散を優先するようにしてください。

ストレージの設定方法については，ストレージ製品のマニュアルを参照してください。

■例 2

OS の機能である LVM を使用してストライピング構成にする例を次の図に示します。

図 6-2　OS の機能である LVM を使用してストライピング構成にする例



上記の例では，DB エリアファイル用の 4 つの RAID グループをボリュームグループに登録しています。また，ボリュームグループからロジカルボリュームをストライピング指定で作成しています。これによって，DB エリアファイル中のデータにアクセスするときは，4 つの RAID グループに対してほぼ均等に I/O が発生します。そのため，RAID グループが 1 つのときに比べてより多くの I/O を発行できます。

また，上記の例では，DB エリアファイル用のボリュームグループと，そのほかのボリュームグループを分けていますが，使用できる RAID グループが少ない場合は，ボリュームグループを 1 つにして DB エリアファイルの I/O 分散を優先するようにしてください。

LVM の設定方法については，各 OS のマニュアルを参照してください。

6.1.2　LVM 設定時の考慮点

ここでは，LVM を設定する際の考慮点について説明します。LVM の設定方法については，OS のマニュアルを参照してください。

- 「8.2.1　インストール前に実施する作業」で OS に設定する HADB 管理者が，ロジカルボリューム（LV）にアクセスできるように，アクセス権限を与えてください。

  なお，LVM で作成したロジカルボリュームの場合，OS を再起動するたびにアクセス権限情報が初期化されることがあります。そのため，OS を再起動した場合は，アクセス権限が初期化されていないかどうかを確認してください。

- 処理性能を向上させるためには，ストライピングに使用するディスク数をできるだけ増やしてください。ディスク数が多いほど，データの格納先が分散されて，1 つのディスクに対する I/O を減らすことができ，その分処理性能が向上します。

- 1 つの DB エリアを複数のロジカルボリューム（DB エリアファイル）で構成すると，ロジカルボリュームへのアクセスが分散されるため，処理性能がより向上します。1 つの DB エリアを構成するロジカルボリュームの個数の目安は，HADB サーバをインストールしたマシンの CPU のコア数の 20%～40%となります。例えば，マシンの CPU のコア数が 40 個である場合，1 つの DB エリアは 8 個～16 個のロジカルボリュームで構成してください。

- LVM で設定するストライプサイズは，4,096 キロバイトを設定することを推奨します。ほかの値を設定する場合は，DB エリアのページサイズ以上，16,384 キロバイト以下で，かつ DB エリアのページサイズの整数倍となるような値を設定してください。

- ストレージの機能によってデータの格納先が十分に分散されている場合は，I/O を分散させる目的のために，LVM を使用してストライピング構成にする必要はありません。

  複数の RAID グループに分散してデータを記憶させるストライピングが，ストレージの機能でできない場合は，LVM でストライピングを設定してください。例えば，ディスクが 60 台ある場合に，1 つの RAID グループを構成するディスク数の上限が 15 台のときは，4 つの RAID グループを作成し，各 RAID グループに対してストライピングを設定してください。

- アクセスの競合による処理性能の低下を防ぐため，1 つのロジカルボリューム（DB エリアファイル）のサイズは，16 テラバイト以下に設定することを推奨します。

- マルチノード機能またはコールドスタンバイ構成を使用する際に，共有ディスクの SCSI リザーブを使用する場合は，SCSI リザーブの指定はストレージのディスク（RAID グループ）単位にしてください。SCSI リザーブの対象となる領域と，それ以外の領域は必ずディスクを分けてください。SCSI リザーブの対象とする領域については，次の個所を参照してください。

  - マルチノード機能の場合：「16.2.4　ストレージ構成」
  - コールドスタンバイ構成の場合：「17.2.4　ストレージ構成」

6.1.3　電源断に対する考慮点

電源断に対応するための考慮点を次に示します。

- ファイルシステムは ext3，ext4 または XFS を使用してください。
- ディスクのキャッシュは，ライトスルー（Write Through）に設定するなどして，電源断時にディスクのキャッシュが失われないようにしてください。

6.2　カーネルパラメタの見積もり

ここでは，カーネルパラメタ（Linux のオペレーティングシステムパラメタ）の見積もりについて説明します。

HADB サーバを動作させるためには，カーネルパラメタを適切に設定する必要があります。カーネルパラメタの値が小さいと，HADB が正常に動作しないおそれがあります。カーネルパラメタの指定値の目安，カーネルパラメタを設定するファイル，および対象のパラメタ名を次の表に示します。

表 6-2　カーネルパラメタの指定値の目安および設定先

項番	カーネルパラメタ	指定値の目安	設定ファイルとパラメタ名
1	shmmax	共有メモリセグメントの大きさの上限をバイト単位で指定します。「6.3.1　共有メモリの所要量の求め方」で求めた最大値以上の値を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 kernel.shmmax
2	shmmni	システム全体の共有メモリセグメント数の上限を指定します。2,000 以上を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 kernel.shmmni
3	shmall	システム全体の共有メモリページ数の上限を指定します。次に示す計算式から求められる値が，SHMMAX に指定した値以上になるように，SHMALL の値を指定してください。 計算式 *共有メモリのページサイズ*×SHMALL	設定ファイル /etc/sysctl.conf パラメタ名 kernel.shmall
4	threads-max	HADB が使用するリアルスレッド数の上限を指定します。 次の条件をすべて満たす値を推奨します。 • 「図 6-3　カーネルパラメタ threads-max の求め方」に示す計算式から求められる値以上の値 「図 6-3　カーネルパラメタ threads-max の求め方」はこの表の後ろに記載しています。 • ほかの業務で使用するプロセス数も考慮した余裕のある値 なお，見積もりの時点でこのカーネルパラメタに指定されている値（OS のデフォルト値）が，上記の条件をすべて満たす場合には，指定値の変更は不要です。	設定ファイル /etc/sysctl.conf パラメタ名 kernel.threads-max
5	core_pattern	コアファイルのパターン名を指定します。 このカーネルパラメタは通常指定する必要はありません。コアファイルの出力が systemd-coredump によって制御されている場合に，コアファイルの出力制御を従来の伝統的な Linux の動作に戻したいときは，このカーネルパラメタにcore を指定します。	設定ファイル /etc/sysctl.conf パラメタ名 kernel.core_pattern

項番	カーネルパラメタ	指定値の目安	設定ファイルとパラメタ名
6	aio-max-nr	「図 6-4　カーネルパラメタ aio-max-nr の求め方」に示す計算式から求められる値以上の値を指定します。「図 6-4　カーネルパラメタ aio-max-nr の求め方」はこの表の後ろに記載しています。	設定ファイル /etc/sysctl.conf パラメタ名 fs.aio-max-nr
7	rmem_default	受信用ウィンドウサイズのデフォルト値を指定します。33,554,432 以上を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 net.core.rmem_default
8	rmem_max	受信用ウィンドウサイズの上限を指定します。33,554,432 以上を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 net.core.rmem_max
9	wmem_default	送信用ウィンドウサイズのデフォルト値を指定します。33,554,432 以上を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 net.core.wmem_default
10	wmem_max	送信用ウィンドウサイズの上限を指定します。33,554,432 以上を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 net.core.wmem_max
11	nr_hugepages	共有メモリにHugePages を適用する場合，HugePages の確保ページ数を指定します。次に示す計算式から求められる値以上の値を指定してください。 $\uparrow \frac{\text{HADBサーバが使用する共有メモリの所要量}}{\text{HugePagesの1ページ当たりのページサイズ}} \uparrow$ HADB サーバが使用する共有メモリの所要量については「6.3.1 共有メモリの所要量の求め方」を参照してください。 HugePages の 1 ページ当たりのページサイズは次に示すコマンドで確認できます。数値の単位に注意してください。 **実行するコマンド** grep Hugepagesize /proc/meminfo **実行結果の例** Hugepagesize: 2048 kB なお，HugePages は，サーバマシンの OS 起動時に，このカーネルパラメタに指定したページ数分のメモリを確保します。HADB サーバをインストールしたサーバマシンに搭載されたメモリサイズ以上となるページ数をこのカーネルパラメタに指定すると，OS が起動できません。	設定ファイル /etc/sysctl.conf パラメタ名 vm.nr_hugepages

項番	カーネルパラメタ	指定値の目安	設定ファイルとパラメタ名
		そのため，このカーネルパラメタには，必ず HADB サーバをインストールしたサーバマシンに搭載されたメモリサイズ以下となるページ数を指定してください。	
12	hugetlb_shm_group	スーパユーザではないユーザで共有メモリにHugePages を適用して HADB サーバを開始する場合，そのユーザのグループ ID を Linux カーネルケーパビリティのhugetlb_shm_group に追加する必要があります。ユーザのグループ ID は OS の id コマンドで取得してください。	設定ファイル /etc/sysctl.conf パラメタ名 vm.hugetlb_shm_group
13	nofile	「図 6-5　カーネルパラメタ nofile の求め方」に示す計算式から求められる値以上の値を指定します。「図 6-5　カーネルパラメタ nofile の求め方」はこの表の後ろに記載しています。	設定ファイル※ /etc/security/limits.conf パラメタ名 • soft nofile • hard nofile
14	file-max	このカーネルパラメタには，nofile に指定した値以上の値を指定してください。	設定ファイル /etc/sysctl.conf パラメタ名 fs.file-max
15	memlock	共有メモリのメモリロックの上限値を設定します。 HADB サーバをインストールしたサーバマシンに搭載されたメモリサイズを指定してください。	設定ファイル※ /etc/security/limits.conf パラメタ名 • soft memlock • hard memlock
16	nproc	HADB が使用するリアルスレッド数の上限を指定します。 次の条件をすべて満たす値を推奨します。 • 「図 6-6　カーネルパラメタ nproc の求め方」に示す計算式から求められる値以上の値 「図 6-6　カーネルパラメタ nproc の求め方」はこの表の後ろに記載しています。 • ほかの業務で使用するプロセス数も考慮した余裕のある値 なお，見積もりの時点でこのカーネルパラメタに指定されている値（OS のデフォルト値）が，上記の条件をすべて満たす場合には，指定値の変更は不要です。	設定ファイル※ /etc/security/limits.conf 次に示すファイルにも同じ値を指定してください。 RHEL 6 の場合 /etc/security/limits.d/90-nproc.conf RHEL 7 の場合 /etc/security/limits.d/20-nproc.conf RHEL 8 の場合 /etc/security/limits.conf 以外に指

項番	カーネルパラメタ	指定値の目安	設定ファイルとパラメタ名
			定値を変更するファイルはありません。 パラメタ名 • soft nproc • hard nproc
17	/sys/block/*デバイス名*/queue/nr_requests	カーネル側の I/O キューの深さを設定します。 • RHEL 6 または RHEL 7 の場合 8192 を指定することを推奨します。 • RHEL 8 の場合 2048 を指定することを推奨します。 上記の値を設定できない場合は，設定可能な最大値を指定してください。 また，コマンド側のキューの深さqueue_depth も変更してください。queue_depth パラメタの設定方法については，OS や HBA などのドキュメントを参照してください。	設定ファイル /sys/block/*デバイス名*/queue/nr_requests
18	/sys/block/*デバイス名*/queue/scheduler	デバイスごとの I/O スケジューラを設定します。 • RHEL 6 または RHEL 7 の場合 deadline を指定することを推奨します。 • RHEL 8 の場合 デバイスがハードディスクの場合は，mq-deadline を指定することを推奨します。 デバイスがフラッシュドライブの場合は，kyber を指定することを推奨します。	設定ファイル /sys/block/*デバイス名*/queue/scheduler
19	/sys/block/*デバイス名*/queue/rq_affinity	フラッシュドライブの場合は2 を指定してください。	設定ファイル /sys/block/*デバイス名*/queue/rq_affinity
20	/sys/block/*デバイス名*/queue/add_random	フラッシュドライブの場合は0 を指定してください。	設定ファイル /sys/block/*デバイス名*/queue/add_random
21	/sys/block/*デバイス名*/queue/rotational	フラッシュドライブの場合は0 を指定してください。	設定ファイル /sys/block/*デバイス名*/queue/rotational

注※

HADB サーバを JP1/AJS3 から自動起動するように設定する場合，カーネルパラメタを/etc/security/limits.conf に設定しても有効となりません。そのため，次のどちらかの方法でカーネルパラメタを設定してください。

- ジョブ実行環境の環境設定パラメーターに，カーネルパラメタを設定する。
- JP1/AJS3 の自動起動スクリプトに，カーネルパラメタを設定する。

詳細については，マニュアル『JP1/AJS3　設計ガイド（業務設計編）』の『UNIX ジョブ使用時の注意事項』にある『JP1/AJS3 サービスを自動起動している場合の注意』を参照してください。

「表 6-2　カーネルパラメタの指定値の目安および設定先」のカーネルパラメタthreads-max の指定値は，次の計算式から求めてください。

図 6-3　カーネルパラメタ threads-max の求め方

計算式

(max_users×2)＋rthd_num＋10＋multi_num※1＋worker_num※2

変数の説明

max_users：
サーバ定義adb_sys_max_usersオペランドの指定値

rthd_num：
サーバ定義adb_sys_rthd_numオペランドの指定値

multi_num：
（サーバ定義adb_sys_multi_node_infoオペランドに指定したホスト名の数－1）
×（サーバ定義adb_sys_max_parallel_exec_numオペランドの指定値×2＋2）
＋5

worker_num：
・プライマリノードまたはセカンダリノードの場合
サーバ定義adb_sys_multi_max_worker_nodesオペランドの指定値
×（サーバ定義adb_sys_max_parallel_exec_numオペランドの指定値×2＋2）
・ワーカーノードの場合
サーバ定義adb_sys_multi_max_worker_nodesオペランドの指定値
×サーバ定義adb_sys_max_parallel_exec_numオペランドの指定値×2＋2

注※1
マルチノード機能を使用する場合に加算してください。

注※2
マルチノード機能でワーカーノードを使用する場合に加算してください。

「表 6-2　カーネルパラメタの指定値の目安および設定先」のカーネルパラメタaio-max-nr の指定値は，次の計算式から求めてください。

図 6-4　カーネルパラメタ aio-max-nr の求め方

計算式

```
{ (max_users＋rthd_num＋10＋multi_num※1＋worker_num※2) × (uthd_num＋2) }
  ＋{ (write_rthd＋read_buff_num) ×20 }
  ＋{ max_users× (wrktbl_clt_blk_num×4) ×sql_rthd_num }
  ＋csvread_aio_num※3＋auditread_aio_num※4
```

変数の説明

max_users：
サーバ定義adb_sys_max_usersｵﾍﾟﾗﾝﾄﾞの指定値

rthd_num：
サーバ定義adb_sys_rthd_numｵﾍﾟﾗﾝﾄﾞの指定値

multi_num：
（サーバ定義adb_sys_multi_node_infoｵﾍﾟﾗﾝﾄﾞに指定したﾎｽﾄ名の数－1）
× （サーバ定義adb_sys_max_parallel_exec_numｵﾍﾟﾗﾝﾄﾞの指定値×2＋2）
＋5

worker_num：
・ﾌﾟﾗｲﾏﾘﾉｰﾄﾞまたはｾｶﾝﾀﾞﾘﾉｰﾄﾞの場合
サーバ定義adb_sys_multi_max_worker_nodesｵﾍﾟﾗﾝﾄﾞの指定値
×（サーバ定義adb_sys_max_parallel_exec_numｵﾍﾟﾗﾝﾄﾞの指定値×2＋2）
・ﾜｰｶｰﾉｰﾄﾞの場合
サーバ定義adb_sys_multi_max_worker_nodesｵﾍﾟﾗﾝﾄﾞの指定値
×サーバ定義adb_sys_max_parallel_exec_numｵﾍﾟﾗﾝﾄﾞの指定値×2＋2

uthd_num：
サーバ定義adb_sys_uthd_numｵﾍﾟﾗﾝﾄﾞの指定値

write_rthd：
次に示す値のうち，最大の値
・ｲﾝﾎﾟｰﾄｵﾌﾟｼｮﾝadb_import_rthd_numの指定値－1
・↓（ｲﾝﾃﾞｸｽ再作成ｵﾌﾟｼｮﾝadb_idxrebuild_rthd_numの指定値－1）÷2↓
・↓（ﾏｰｼﾞﾁｬﾝｸｵﾌﾟｼｮﾝadb_mergechunk_rthd_numの指定値－1）÷2↓
・↓（ｱﾝｱｰｶｲﾌﾞﾁｬﾝｸｵﾌﾟｼｮﾝadb_unarcv_rthd_numの指定値－1）÷2↓

read_buff_num：
2を代入する

wrktbl_clt_blk_num：
次に示す値のうち，最大の値
・サーバ定義adb_dbbuff_wrktbl_clt_blk_numｵﾍﾟﾗﾝﾄﾞの指定値
・ｸﾗｲｱﾝﾄ定義adb_dbbuff_wrktbl_clt_blk_numｵﾍﾟﾗﾝﾄﾞの指定値
・ｴｸｽﾎﾟｰﾄｵﾌﾟｼｮﾝadb_export_wrktbl_blk_numの指定値

sql_rthd_num：
サーバ定義adb_sql_exe_max_rthd_numｵﾍﾟﾗﾝﾄﾞの指定値

csvread_aio_num：
次に示す計算式で求めた値
（max_users＋rthd_num）×2,048

auditread_aio_num：
次に示す計算式で求めた値
（max_users＋rthd_num）×2,048

注※1
ﾏﾙﾁﾉｰﾄﾞ機能を使用する場合に加算してください。

注※2
ﾏﾙﾁﾉｰﾄﾞ機能でﾜｰｶｰﾉｰﾄﾞを使用する場合に加算してください。

注※3
次に示す場合に加算してください。
・ADB_CSVREAD関数を使用する場合
・ｱｰｶｲﾌﾞﾏﾙﾁﾁｬﾝｸ表を使用する場合

注※4
ADB_AUDITREAD関数を使用する場合に加算してください。

「表 6-2　カーネルパラメタの指定値の目安および設定先」のカーネルパラメタnofileの指定値は，次の計算式から求めてください。

図 6-5　カーネルパラメタ nofile の求め方

計算式

```
(max_users＋(multi_num＋3)※1＋worker_num※2＋dbarea_file_num＋50)
＋(max_users＋rthd_num＋10＋multi_num＋worker_num※2)×129
＋export_file_num＋trc_file_num＋optlog_file_num＋csvread_file_num※3
＋arcvchk_file_num※4＋1＋syndict_file_num※5
＋auditread_file_num※6＋convertauditrailfile_file_num※7＋1
```

変数の説明

max_users：
　サーバ定義adb_sys_max_usersオペランドの指定値

multi_num：
　（サーバ定義adb_sys_multi_node_infoオペランドに指定されたホスト名の数－1）
　×（サーバ定義adb_sys_max_parallel_exec_numオペランドの指定値×2＋2）
　＋5

worker_num：
　・プライマリノードまたはセカンダリノードの場合
　サーバ定義adb_sys_multi_max_worker_nodesオペランドの指定値
　×（サーバ定義adb_sys_max_parallel_exec_numオペランドの指定値×2＋2）
　・ワーカーノードの場合
　サーバ定義adb_sys_multi_max_worker_nodesオペランドの指定値
　×サーバ定義adb_sys_max_parallel_exec_numオペランドの指定値×2＋2

dbarea_file_num：
　DBエリアを構成する全DBエリアファイル数

rthd_num：
　サーバ定義adb_sys_rthd_numオペランドの指定値

export_file_num：
　・adbexportコマンドの出力データパスファイルに指定した出力データファイルの数
　・複数のadbexportコマンドを同時に実行する場合は出力データファイルの総数

trc_file_num：
　9を代入する

optlog_file_num：
　3を代入する

csvread_file_num：
　次に示す計算式で求めた値
　（max_users＋rthd_num）×2,048

arcvchk_file_num：
　次に示す計算式で求めた値
　（max_users＋rthd_num）×2,048

syndict_file_num：
　次に示す計算式で求めた値
　max_users＋1

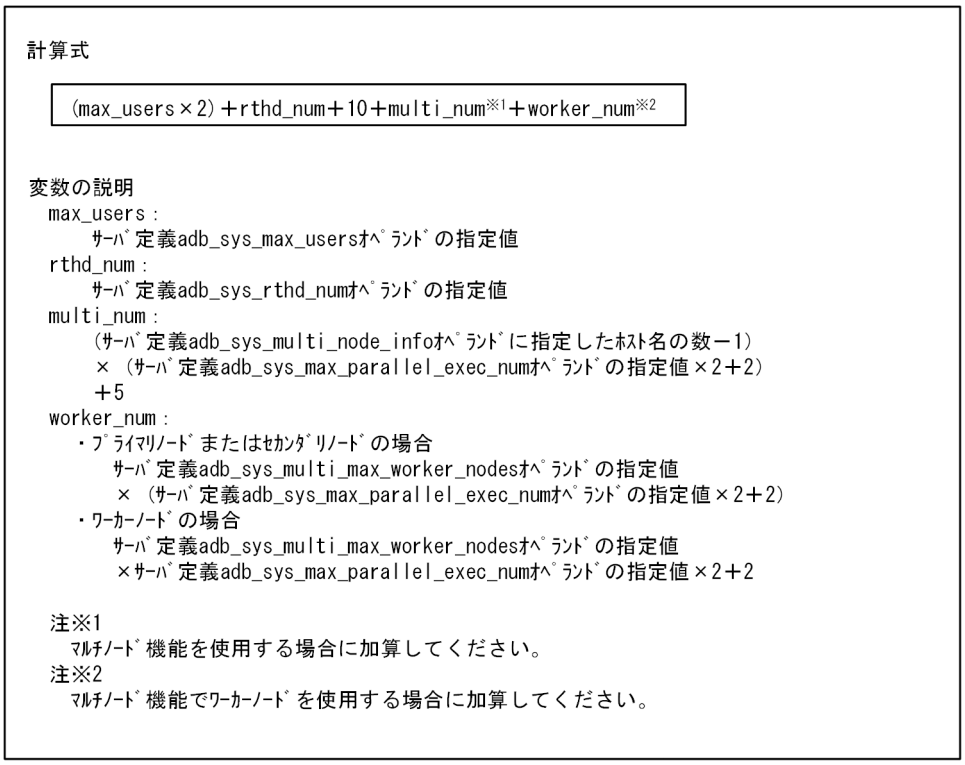
auditread_file_num：
　次に示す計算式で求めた値
　（max_users＋rthd_num）×2,048

convertauditrailfile_file_num：
　3を代入する

注※1：マルチノード機能を使用する場合に加算してください。
注※2：マルチノード機能でワーカーノードを使用する場合に加算してください。
注※3：次に示す場合に加算してください。
　・ADB_CSVREAD関数を使用する場合
　・アーカイブマルチチャンク表を使用する場合
注※4：アーカイブマルチチャンク表を使用する場合に加算してください。
注※5：同義語辞書を使用して同義語検索を行う場合に加算してください。
注※6：ADB_AUDITREAD関数を使用する場合に加算してください。
注※7：adbconvertauditrailfileコマンドを使用する場合に加算してください。

「表 6-2　カーネルパラメタの指定値の目安および設定先」のカーネルパラメタnproc の指定値は，次の計算式から求めてください。

図 6-6　カーネルパラメタ nproc の求め方



6.3 HADB サーバのメモリ所要量の見積もり

ここでは，HADB サーバが使用するメモリ所要量を見積もる方法について説明します。

HADB サーバが使用するメモリの最大サイズは，次に示す計算式で求められます。次に示す計算式で求めた値以上の値を，サーバ定義adb_sys_memory_limit オペランドに指定してください。詳細については，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_memory_limit オペランドを参照してください。

計算式（単位：キロバイト）

```
HADBサーバが使用するメモリの最大サイズ＝
  共有メモリ管理
  ＋グローバルバッファページ
  ＋プロセス共通メモリ
  ＋リアルスレッド固有メモリ※1×（サーバ定義adb_sys_max_usersオペランドの指定値＋10）
  ＋（リアルスレッド固有メモリ※1×（2×マルチノード機能で使用するノード数＋3））※2
  ＋MAX（900,000　，リアルスレッド固有メモリ※1）×サーバ定義adb_sys_rthd_numオペランドの指定値
  ＋リアルスレッド固有メモリ※3
  ＋ヒープメモリ
```

注※1

「6.3.1　共有メモリの所要量の求め方」の「(4)　リアルスレッド固有メモリの所要量の求め方」の次に示す変数を除いた変数が該当します。

- *RTHD_IMPORTSZ*
- *RTHD_IDXRBLDSZ*
- *RTHD_GETCOSTSZ*
- *RTHD_DBSTATUSSZ*
- *RTHD_EXPORTSZ*
- *RTHD_MERCHKSZ*
- *RTHD_ARCCKSZ*
- *RTHD_UNARCCKSZ*
- *RTHD_REORGSZ*
- *RTHD_CLTDEFMNG*
- *RTHD_SYNDICTSZ*
- *RTHD_AUDTRAILSZ*
- *RTHD_CONVERTAUDSZ*

注※2

マルチノード機能を使用する場合に加算してください。

注※3

「6.3.1　共有メモリの所要量の求め方」の「(4)　リアルスレッド固有メモリの所要量の求め方」の次に示す変数が該当します。

- *RTHD_IMPORTSZ*
- *RTHD_IDXRBLDSZ*
- *RTHD_GETCOSTSZ*
- *RTHD_DBSTATUSSZ*
- *RTHD_EXPORTSZ*
- *RTHD_MERCHKSZ*
- *RTHD_ARCCKSZ*
- *RTHD_UNARCCKSZ*
- *RTHD_REORGSZ*
- *RTHD_CLTDEFMNG*
- *RTHD_SYNDICTSZ*
- *RTHD_AUDTRAILSZ*
- *RTHD_CONVERTAUDSZ*

HADB サーバが使用するメモリは，**共有メモリ**および**プロセスメモリ**です。

■共有メモリの種類

共有メモリを求める場合，次に示す 4 種類に分けて計算する必要があります。

- 共有メモリ管理
- グローバルバッファページ
- プロセス共通メモリ
- リアルスレッド固有メモリ

■プロセスメモリの種類

- ヒープメモリ

HADB サーバが各メモリを使用するときを次に示します。

- HADB サーバ開始時
- 通常運用時（HADB クライアントが HADB サーバに接続するとき，および SQL 文を実行するとき）
- `adbinit` コマンド実行時
- `adbimport` コマンド実行時
- `adbidxrebuild` コマンド実行時
- `adbgetcst` コマンド実行時
- `adbdbstatus` コマンド実行時
- `adbexport` コマンド実行時

- adbstat コマンド実行時
- adbmodarea コマンド実行時
- adbmergechunk コマンド実行時
- adbchgchunkcomment コマンド実行時
- adbchgchunkstatus コマンド実行時
- adbarchivechunk コマンド実行時
- adbunarchivechunk コマンド実行時
- adbreorgsystemdata コマンド実行時
- adbclientdefmang コマンド実行時
- adbsyndict コマンド実行時
- adbconvertaudittrailfile コマンド実行時

メモ

サーバ定義adb_sys_memory_limit オペランドを指定しない場合は，プロセス共通メモリの最大サイズを，サーバ定義adb_sys_proc_area_max オペランドに指定してください。また，1 リアルスレッド当たりのリアルスレッド固有メモリの最大サイズを，サーバ定義adb_sys_rthd_area_max オペランドに指定してください。

6.3.1 共有メモリの所要量の求め方

ここでは，HADB サーバが使用する共有メモリの所要量の求め方について説明します。

HADB サーバが使用する共有メモリの所要量は次に示す計算式から求めてください。

計算式

共有メモリ＝
共有メモリ管理＋グローバルバッファページ＋プロセス共通メモリ
＋リアルスレッド固有メモリ※1
×（サーバ定義adb_sys_max_usersオペランドの指定値
＋サーバ定義adb_sys_rthd_numオペランドの指定値＋10）
＋リアルスレッド固有メモリ※2

注※1

「(4) リアルスレッド固有メモリの所要量の求め方」の次に示す変数を除いた変数が該当します。

- *RTHD_IMPORTSZ*
- *RTHD_IDXRBLDSZ*
- *RTHD_GETCOSTSZ*

- *RTHD_DBSTATUSSZ*
- *RTHD_EXPORTSZ*
- *RTHD_MERCHKSZ*
- *RTHD_ARCCKSZ*
- *RTHD_UNARCCKSZ*
- *RTHD_REORGSZ*
- *RTHD_CLTDEFMNG*
- *RTHD_SYNDICTSZ*
- *RTHD_AUDTRAILSZ*
- *RTHD_CONVERTAUDSZ*

注※2

「(4)　リアルスレッド固有メモリの所要量の求め方」の次に示す変数が該当します。

- *RTHD_IMPORTSZ*
- *RTHD_IDXRBLDSZ*
- *RTHD_GETCOSTSZ*
- *RTHD_DBSTATUSSZ*
- *RTHD_EXPORTSZ*
- *RTHD_MERCHKSZ*
- *RTHD_ARCCKSZ*
- *RTHD_UNARCCKSZ*
- *RTHD_REORGSZ*
- *RTHD_CLTDEFMNG*
- *RTHD_SYNDICTSZ*
- *RTHD_AUDTRAILSZ*
- *RTHD_CONVERTAUDSZ*

共有メモリ管理，グローバルバッファページ，プロセス共通メモリ，およびリアルスレッド固有メモリの所要量の求め方を次に示します。

(1)　共有メモリ管理の所要量の求め方

共有メモリ管理は，次に示すときに使用されます。そのため，それぞれで使用される共有メモリ管理の所要量を求めてください。

- HADB サーバ開始時

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(1)　共有メモリ管理の所要量の求め方（HADB サーバ開始時）」を参照してください。

- adbinit コマンド実行時

  「6.3.5　adbinit コマンド実行時のメモリ所要量の求め方」の「(1)　共有メモリ管理の所要量の求め方（adbinit コマンド実行時）」を参照してください。

(2)　グローバルバッファページの所要量の求め方

グローバルバッファページは，HADB サーバ開始時に使用されます。グローバルバッファページの所要量を求める場合，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」を参照してください。

(3)　プロセス共通メモリの所要量の求め方

プロセス共通メモリの所要量は次に示す計算式から求めてください。

計算式（単位：メガバイト）

プロセス共通メモリ（変数PROC_STARTSZの値が4ギガバイトを超えない場合）＝

$$\mathrm{MAX}\left\{4{,}000\ ,\ \left(\left\uparrow\left(\mathrm{PROC_STARTSZ}+\mathrm{PROC_EXECSZ}\right)\div 1{,}024\right\uparrow\times 1.01\right)\right\}$$

プロセス共通メモリ（変数PROC_STARTSZの値が4ギガバイトを超える場合）＝

$$\mathrm{MAX}\left\{4{,}000\ ,\ \left(\left\uparrow\left(\mathrm{PROC_STARTSZ}^{※}+\mathrm{PROC_EXECSZ}\right)\div 1{,}024\right\uparrow\times 1.01\right)+\left(\left\uparrow\mathrm{PROC_STAINF}\div 1{,}024\right\uparrow\times 1.01\right)\right\}$$

注※

変数*PROC_STARTSZ* の値から，変数*PROC_STAINF* の値を除いた値を加算してください。

変数の説明

- *PROC_STARTSZ*

  HADB サーバ開始時に使用するプロセス共通メモリ

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」を参照して求めてください。

- *PROC_EXECSZ*

  通常運用時に使用するプロセス共通メモリ

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」を参照して求めてください。

- *PROC_STAINF*

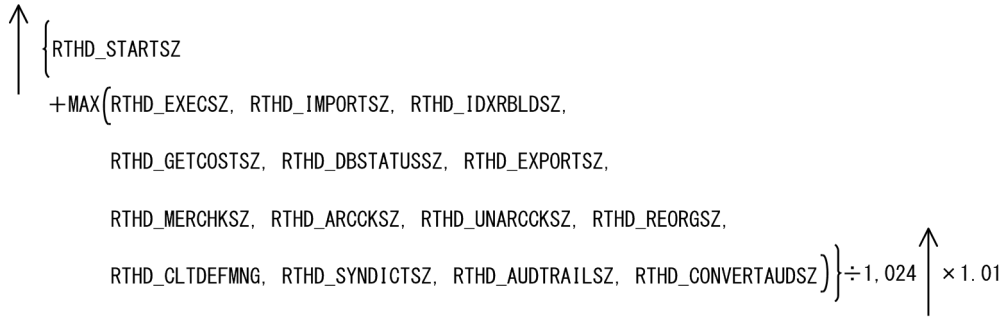
  統計情報の制御情報を格納するプロセス共通メモリ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(p)　変数 PROC_STAINF の求め方」を参照して求めてください。

(4)　リアルスレッド固有メモリの所要量の求め方

リアルスレッド固有メモリの所要量は次に示す計算式から求めてください。

計算式（単位：メガバイト）



変数の説明

- *RTHD_STARTSZ*

  HADB サーバ開始時に使用するリアルスレッド固有メモリ

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(4)　リアルスレッド固有メモリの所要量の求め方（HADB サーバ開始時）」を参照して求めてください。

- *RTHD_EXECSZ*

  通常運用時に使用するリアルスレッド固有メモリ

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」を参照して求めてください。

- *RTHD_IMPORTSZ*

  `adbimport` コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbimport コマンド実行時）」を参照して求めてください。

- *RTHD_IDXRBLDSZ*

  `adbidxrebuild` コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.7　adbidxrebuild コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbidxrebuild コマンド実行時）」を参照して求めてください。

- *RTHD_GETCOSTSZ*

  `adbgetcst` コマンド実行時に使用するリアルスレッド固有メモリ

「6.3.8　adbgetcst コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbgetcst コマンド実行時）」を参照して求めてください。

- *RTHD_DBSTATUSSZ*

  adbdbstatus コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.9　adbdbstatus コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbdbstatus コマンド実行時）」を参照して求めてください。

- *RTHD_EXPORTSZ*

  adbexport コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.10　adbexport コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbexport コマンド実行時）」を参照して求めてください。

- *RTHD_MERCHKSZ*

  adbmergechunk コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.13　adbmergechunk コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbmergechunk コマンド実行時）」を参照して求めてください。

- *RTHD_ARCCKSZ*

  adbarchivechunk コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.16　adbarchivechunk コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbarchivechunk コマンド実行時）」を参照して求めてください。

- *RTHD_UNARCCKSZ*

  adbunarchivechunk コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.17　adbunarchivechunk コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbunarchivechunk コマンド実行時）」を参照して求めてください。

- *RTHD_REORGSZ*

  adbreorgsystemdata コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」を参照して求めてください。

- *RTHD_CLTDEFMNG*

  adbclientdefmang コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.19　adbclientdefmang コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbclientdefmang コマンド実行時）」を参照して求めてください。

- *RTHD_SYNDICTSZ*

  adbsyndict コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.20　adbsyndict コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbsyndict コマンド実行時）」を参照して求めてください。

- *RTHD_AUDTRAILSZ*

adbaudittrail コマンド実行時に使用するリアルスレッド固有メモリ

「6.3.21　adbaudittrail コマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbaudittrail コマンド実行時）」を参照して求めてください。

- *RTHD_CONVERTAUDSZ*

  adbconvertaudittrailfile コマンド実行時に使用するリアルスレッド固有メモリ

  「6.3.22　adbconvertaudittrailfile コマンド実行時のメモリ所要量の求め方」を参照して求めてください。

6.3.2　プロセスメモリの所要量の求め方

プロセスメモリ（ヒープメモリ）の所要量は次に示す計算式から求めてください。

計算式（単位：メガバイト）

```
プロセスメモリ（ヒープメモリ）＝
↑(HEAP_STARTSZ
  ＋HEAP_EXECSZ＋HEAP_IMPORTSZ＋HEAP_IDXRBLDSZ
  ＋HEAP_EXPORTSZ＋HEAP_MERCHKSZ＋HEAP_ARCCKSZ
  ＋HEAP_UNARCCKSZ＋HEAP_REORGSZ＋HEAP_CLTDEFMNG) ÷1,024↑ ×1.01
```

変数の説明

- *HEAP_STARTSZ*

  HADB サーバ開始時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(5)　ヒープメモリの所要量の求め方（HADB サーバ開始時）」を参照して求めてください。

- *HEAP_EXECSZ*

  通常運用時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（通常運用時）」を参照して求めてください。

- *HEAP_IMPORTSZ*

  adbimport コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbimport コマンド実行時）」を参照して求めてください。

- *HEAP_IDXRBLDSZ*

  adbidxrebuild コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

「6.3.7　adbidxrebuild コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbidxrebuild コマンド実行時）」を参照して求めてください。

- *HEAP_EXPORTSZ*

  adbexport コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.10　adbexport コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbexport コマンド実行時）」を参照して求めてください。

- *HEAP_MERCHKSZ*

  adbmergechunk コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.13　adbmergechunk コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbmergechunk コマンド実行時）」を参照して求めてください。

- *HEAP_ARCCKSZ*

  adbarchivechunk コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.16　adbarchivechunk コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbarchivechunk コマンド実行時）」を参照して求めてください。

- *HEAP_UNARCCKSZ*

  adbunarchivechunk コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.17　adbunarchivechunk コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbunarchivechunk コマンド実行時）」を参照して求めてください。

- *HEAP_REORGSZ*

  adbreorgsystemdata コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」を参照して求めてください。

- *HEAP_CLTDEFMNG*

  adbclientdefmang コマンド実行時に使用するプロセスメモリ（ヒープメモリ）

  「6.3.19　adbclientdefmang コマンド実行時のメモリ所要量の求め方」の「(3)　ヒープメモリの所要量の求め方（adbclientdefmang コマンド実行時）」を参照して求めてください。

6.3.3　HADB サーバ開始時のメモリ所要量の求め方

adbstart コマンドで HADB サーバを開始する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- 共有メモリ管理
- グローバルバッファページ
- プロセス共通メモリ

- リアルスレッド固有メモリ

■プロセスメモリ

- ヒープメモリ

それぞれのメモリ所要量の計算式を次に示します。

(1) 共有メモリ管理の所要量の求め方（HADB サーバ開始時）

HADB サーバ開始時の共有メモリ管理（*SHMMAN*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SHMMAN＝
        104,000＋968×(rthd_num＋max_users＋10＋multi_num＋worker_num)
        ＋(2,885×max_users)＋SHM_LSGBUF＋SHM_LSLBUF
```

変数の説明

rthd_num

サーバ定義adb_sys_rthd_num オペランドの指定値

max_users

サーバ定義adb_sys_max_users オペランドの指定値

multi_num

次に示す計算式から求めてください。

計算式

```
multi_num＝
    ｻｰﾊﾞ定義adb_sys_multi_node_infoｵﾍﾟﾗﾝﾄﾞに指定するﾎｽﾄ名の数×2＋3
```

worker_num

サーバ定義adb_sys_multi_max_worker_nodes オペランドの指定値×2

SHM_LSGBUF

adbls -d gbuf コマンドの表示内容が格納される共有メモリ

次に示す値を代入してください。

値（単位：キロバイト）

```
SHM_LSGBUF＝ 337
```

SHM_LSLBUF

adbls -d lbuf コマンドの表示内容が格納される共有メモリ

次に示す値を代入してください。

値（単位：キロバイト）

SHM_LSLBUF＝ 33

(2) グローバルバッファページの所要量の求め方（HADB サーバ開始時）

HADB サーバ開始時のグローバルバッファページ（*SHM_BUFGLOBAL*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$SHM_BUFGLOBAL = \left\lceil \frac{\left\{ MSTPAGES + WRKPAGES + \sum (BUFPAGES + SCANPAGES) \right\}}{1,024} \right\rceil$$

変数の説明

MSTPAGES：マスタディレクトリ用 DB エリアのページ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$MSTPAGES = RTHNUM \times 8,192$$

RTHNUM：「(3) プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a) 変数 SCI の求め方」の変数 *RTHNUM* を参照してください。

WRKPAGES：作業表用 DB エリアのページ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$WRKPAGES = (wrk_page_num + 1) \times wrk_page_size$$

- *wrk_page_num*

  サーバ定義adb_dbbuff_wrktbl_glb_blk_num オペランドの指定値

- *wrk_page_size*

  「表 6-3 DB エリアのページサイズ」の作業表用 DB エリアのページサイズを参照して，バイト単位で求めてください。

Σ（*BUFPAGES* ＋ *SCANPAGES*）：指定したすべてのadbbuff オペランドについて，*BUFPAGES* と *SCANPAGES* を計算した結果の合計値

この合計値には次に示すadbbuff オペランドも含まれます。

- -o オプションを指定したadbbuff オペランド
- -n オプションおよび-o オプションを指定しなかったadbbuff オペランド

それぞれの詳細について説明します。

-o オプションを指定した adbbuff オペランド

-o オプションを指定した場合，adbbuff オペランドの-n オプションに指定しなかった DB エリアはページサイズごとに分類されて自動的にadbbuff オペランドが定義されます。それを踏まえて，*BUFPAGES* を求めてください。例を次に示します。

例：次に示す 4 つの DB エリアを-n オプションで指定しなかった場合

■対象の DB エリア

- ADBDIC，ADBSTBL，ADBUTBL01，ADBUTBL02（ページサイズ：4 キロバイト）
- ADBUTBL03（ページサイズ：32 キロバイト）

■サーバ定義の指定例

```
adbbuff -g SAMPLEBUF -o -p 100
```

この場合，4 つの DB エリアはページサイズごとに分類されて自動的に定義されます。-o オプションを指定したadbbuff オペランドに各オプション（-p，-a）を指定した場合，オプションの指定値も自動的に定義されます。

■自動的に定義された例

```
adbbuff -g SAMPLEBUF#0000004096 -n ADBDIC，ADBSTBL，ADBUTBL01，ADBUTBL02 -p 100
adbbuff -g SAMPLEBUF#0000032768 -n ADBUTBL03 -p 100
```

注　下線部分は自動的に定義されたときに追加されたグローバルバッファ名です。

-n オプションおよび-o オプションを指定しなかった adbbuff オペランド

adbbuff オペランドの-n オプションに指定しなかった DB エリアがあり，-o オプションも指定しなかった場合，-o オプションを指定したときと同じように，DB エリアはページサイズごとに分類されて自動的にadbbuff オペランドが定義されます。それを踏まえて，*BUFPAGES* を求めてください。例を次に示します。

例：次に示す 4 つの DB エリアを-n オプションに指定しないで，-o オプションも指定しなかった場合

■対象の DB エリア

- ADBDIC，ADBSTBL，ADBUTBL01，ADBUTBL02（ページサイズ：4 キロバイト）
- ADBUTBL03（ページサイズ：32 キロバイト）

■サーバ定義の指定例

- 指定なし

この場合，4 つの DB エリアはページサイズごとに分類されて自動的に定義されます。-p オプションを省略した場合に仮定される値が定義されます。

■自動的に定義された例

```
adbbuff -g ##ADBOTHER#0000004096 -n ADBDIC，ADBSTBL，ADBUTBL01，ADBUTBL02
adbbuff -g ##ADBOTHER#0000032768 -n ADBUTBL03
```

注　下線部分は自動的に定義されたグローバルバッファ名です。

BUFPAGES：ページ領域

次に示す計算式から求めてください。

計算式（単位：バイト）

BUFPAGES＝ （*GLBPAGES*＋*OPTTYPES*）×*page_size*

GLBPAGES：

サーバ定義adbbuff オペランドの各オプション（-p，-a）の指定値の合計値

OPTTYPES：

サーバ定義adbbuff オペランドの各オプション（-p，-a）の指定数

すべてのオプションを省略した場合，*OPTTYPES* は 1 としてください。例えば，次に示す指定例の場合，-p および-a の 2 つのオプションが指定されているため，*OPTTYPES* は 2 となります。

指定例

```
adbbuff -g SAMPLEBUF -n ADBUTBL01 -p 10 -a 10
```

page_size：

サーバ定義adbbuff オペランドの-n オプションに指定した DB エリアのページサイズ（単位：バイト）

「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

SCANPAGES：テーブルスキャンページ

次に示す計算式から求めてください。ただし，サーバ定義adbbuff オペランドの-v オプション，および-k オプションを省略した場合，*SCANPAGES* に0 を代入してください。

計算式（単位：バイト）

SCANPAGES＝ *SCANBLOCKS* × 4,194,304

SCANBLOCKS：

テーブルスキャンバッファのブロック数

次に示す計算式から求めてください。

$$SCANBLOCKS = \left\lfloor \frac{adbbuff_opt_v \times 1,048,576 - (rthd_num \times 8,841 + 706)}{\frac{1,612,223,349}{page_size} + 4,194,369} \right\rfloor$$

adbbuff_opt_v：

サーバ定義adbbuff オペランドの-v オプションの*テーブルスキャンバッファに使用するメモリサイズ*の指定値

サーバ定義adbbuff オペランドの-k オプションを指定した場合は次の計算式から求めてください。

adbbuffオペランドの-kオプションの指定値 × 4

page_size：

サーバ定義adbbuff オペランドの-n オプションに指定した DB エリアのページサイズ（単位：バイト）

「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

表 6-3　DB エリアのページサイズ

項番	DB エリアの種類	ページサイズ（単位：バイト）
1	マスタディレクトリ用 DB エリア	4,096
2	ディクショナリ用 DB エリア	
3	システム表用 DB エリア	
4	作業表用 DB エリア	↑*work_page_size*÷32↑×32×1,024 *work_page_size* には次の値を指定してください。 **サーバ定義に adb_dbarea_wrk_page_size オペランドを指定している場合** サーバ定義のadb_dbarea_wrk_page_size オペランドの指定値 **サーバ定義に adb_dbarea_wrk_page_size オペランドを指定していない場合** adbinit コマンドの初期設定オプションのadb_init_wrk_page_size オペランドの指定値
5	データ用 DB エリア	**対象の DB エリアを adbinit コマンドで作成した場合** 次に示す計算式から求めてください。 *初期設定オプションadbinitdbareaに指定した-pオプションの指定値*×1,024 **対象の DB エリアを adbmodarea コマンドで作成した場合** 次に示す計算式から求めてください。 *DBエリア追加・変更オプションadbaddareaに指定した-pオプションの指定値*×1,024

メモ

データベースを構築していれば，次に示すコマンドで DB エリアのページサイズを調べることもできます。

作業表用 DB エリアの場合

- adbls -d gbuf コマンドで，グローバルバッファの情報を取得する
- adbls -d lbuf コマンドで，ローカル作業表用バッファの情報を取得する

データ用 DB エリアの場合

- adbdbstatus -d summary -c dbarea コマンドで，DB エリアのサマリ情報を取得する

- adbls -d gbuf コマンドで，グローバルバッファの情報を取得する

rthd_num：

サーバ定義adb_sys_rthd_num オペランドの指定値

(3) プロセス共通メモリの所要量の求め方（HADB サーバ開始時）

HADB サーバ開始時のプロセス共通メモリ（*PROC_STARTSZ*）の所要量は，次の計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_STARTSZ＝
    SCI＋RPCS＋DICS＋BUFGLOBAL＋DBAREAINF＋DBSYSINF＋RECCTL＋DBUPDINFMNG
  ＋RNGPREREADINF＋RNGPREREADHANDLE＋MODA＋PROC_CNCTSZ＋LSINF＋DBHTRC
  ＋LSTCMNINF＋LSTRTHDINF＋PROC_STAINF＋SYNDICTINF＋AUDINF＋CLMNZ
```

変数の説明

- *SCI*

  システム管理制御情報

  「(a)　変数 SCI の求め方」を参照してください。

- *RPCS*

  通信サービス管理情報

  「(b)　変数 RPCS の求め方」を参照してください。

- *DICS*

  ディクショナリ表，およびシステム表の検索用領域

  「(c)　変数 DICS の求め方」を参照してください。

- *BUFGLOBAL*

  グローバルバッファ

  「(d)　変数 BUFGLOBAL の求め方」を参照してください。

- *DBAREAINF*

  DB エリア情報

  「(e)　変数 DBAREAINF の求め方」を参照してください。

- *DBSYSINF*

  HADB サーバ起動用システム情報

  「(f)　変数 DBSYSINF の求め方」を参照してください。

- *RECCTL*

  回復制御領域

  「(g)　変数 RECCTL の求め方」を参照してください。

- *DBUPDINFMNG*

  DB エリア，表，インデクス，およびチャンク更新情報管理領域

  「(h)　変数 DBUPDINFMNG の求め方」を参照してください。

- *RNGPREREADINF*

  レンジインデクスの先読み

  「(i)　変数 RNGPREREADINF の求め方」を参照してください。

- *RNGPREREADHANDLE*

  レンジインデクス先読み開始処理

  「(j)　変数 RNGPREREADHANDLE の求め方」を参照してください。

- *MODA*

  DB エリア変更制御情報

  「(k)　変数 MODA の求め方」を参照してください。

- *PROC_CNCTSZ*

  データベース接続時

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(a)　変数 PROC_CNCTSZ の求め方」を参照してください。

- *LSINF*

  adbls コマンドの管理情報

  「(l)　変数 LSINF の求め方」を参照してください。

- *DBHTRC*

  バッファ，ディスク I/O，HADB サーバ間通信トレース，および作業表の状態遷移トレースで使用する格納領域

  「(m)　変数 DBHTRC の求め方」を参照してください。

- *LSTCMNINF*

  作業表の共通管理情報

  「(n)　変数 LSTCMNINF の求め方」を参照してください。

- *LSTRTHDINF*

  ローカル作業表の管理情報

  「(o)　変数 LSTRTHDINF の求め方」を参照してください。

- *PROC_STAINF*

  統計情報の制御情報

  「(p)　変数 PROC_STAINF の求め方」を参照してください。

- *SYNDICTINF*

  同義語辞書へのアクセス情報

  「(q)　変数 SYNDICTINF の求め方」を参照してください。

- *AUDINF*

  監査証跡の管理情報

  「(r)　変数 AUDINF の求め方」を参照してください。

- *CLMNZ*

  更新行のカラム化機能を有効にする場合に必要なメモリ

  「(s)　変数 CLMNZ の求め方」を参照してください。

(a)　変数 SCI の求め方

変数*SCI* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SCI＝ ↑（34,584＋CON＋RTH＋UTH＋TRN＋DEF＋MLT※1＋CGR※2＋CMI※3 ）÷1,024↑
```

注※1

マルチノード機能を使用する場合に加算してください。

注※2

クライアントグループ機能を使用する場合に加算してください。

注※3

クライアント定義の集中管理機能を使用する場合に加算してください。

変数の説明

CON：コネクション制御情報

次に示す計算式から求めてください。

計算式（単位：バイト）

```
CON＝ 10,712×max_users
```

max_users

サーバ定義adb_sys_max_users オペランドの指定値

RTH：リアルスレッド制御情報

次に示す計算式から求めてください。

計算式（単位：バイト）

```
RTH＝ RTHNUM×1,600
```

RTHNUM：総リアルスレッド数

計算式

```
RTHNUM＝ 7＋rthd_num＋CTHNUM＋MLTNUM※
```

注※

マルチノード機能を使用する場合に加算してください。

rthd_num

サーバ定義adb_sys_rthd_num オペランドの指定値

CTHNUM：コネクションスレッド数

計算式

```
CTHNUM＝ 3＋max_users
```

max_users

サーバ定義adb_sys_max_users オペランドの指定値

MLTNUM：マルチノード機能専用スレッド数

マルチノード機能を使用する場合に求めてください。

計算式

```
MLTNUM＝（multi_node_num＋worker_node_num）×2＋3
```

multi_node_num

サーバ定義adb_sys_multi_node_info オペランドに指定したホスト名の数

worker_node_num

サーバ定義adb_sys_multi_max_worker_nodes オペランドの指定値

UTH：擬似リアルスレッド制御情報

次に示す計算式から求めてください。

計算式（単位：バイト）

```
UTH＝ UTHNUM×RTHNUM×1,696
```

UTHNUM：総擬似スレッド数

計算式

```
UTHNUM＝ 2＋uthd_num
```

uthd_num：サーバ定義adb_sys_uthd_num オペランドの指定値

RTHNUM

変数*RTH* の変数*RTHNUM* を参照してください。

TRN：トランザクション管理情報

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TRN＝ 96＋（88×max_users）
```

max_users

サーバ定義adb_sys_max_users オペランドの指定値

DEF：定義解析結果情報

次に示す計算式から求めてください。

計算式（単位：バイト）

```
DEF＝ 3,701＋475×dbbuff_defnum
```

dbbuff_defnum：サーバ定義に指定されたadbbuff オペランドの数

MLT：マルチノード制御情報

マルチノード機能を使用する場合に求めてください。

計算式（単位：バイト）

```
MLT＝23,024＋（22,584＋792×RTHNUM
　　＋（RNGPREREADINF×1,024＋BUFDEF＋BUFINT＋200＋40×cltgrp_num）※）
　　×（multi_node_num＋worker_node_num）
```

注※

サーバ定義adb_sys_max_parallel_exec_num オペランドに 1 以上が指定された場合に加算してください。

RTHNUM

変数 *RTH* の変数 *RTHNUM* を参照してください。

RNGPREREADINF

「(i)　変数 RNGPREREADINF の求め方」を参照してください。

BUFDEF

「(d)　変数 BUFGLOBAL の求め方」の変数 *BUFDEF* を参照してください。

BUFINT

「(d)　変数 BUFGLOBAL の求め方」の変数 *BUFINT* を参照してください。

cltgrp_num

サーバ定義adbcltgrp オペランドに指定したクライアントグループの数

multi_node_num

サーバ定義adb_sys_multi_node_info オペランドに指定したホスト名の数

worker_node_num

サーバ定義adb_sys_multi_max_worker_nodes オペランドの指定値

CGR：クライアントグループ制御情報

クライアントグループ機能を使用する場合に求めてください。

計算式（単位：バイト）

```
CGR＝ 160＋104×cltgrp_num
```

cltgrp_num

サーバ定義adbcltgrp オペランドに指定したクライアントグループの数

CMI：クライアント定義の集中管理情報

クライアント定義の集中管理機能を使用する場合に求めてください。

計算式（単位：バイト）

```
CMI = (↑(50,376＋760×(cltmng_num－1))÷32↑×32＋64)＋224×authid_num
```

cltmng_num

クライアント管理定義ファイルに指定された，クライアント管理定義のadbclientmang オペランドの総数

authid_num

クライアント管理定義のadbclientmang オペランドに指定された，認可識別子の総数

(b)　変数 RPCS の求め方

変数*RPCS* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RPCS＝ \uparrow\left\{(1,024\times2)＋(262,656\times(CTHNUM＋MLTNUM＋2))＋(4\times ipv4_num)\right\}\div1,024\uparrow＋512$$

変数の説明

CTHNUM

「(a)　変数 SCI の求め方」の変数*CTHNUM* を参照してください。

MLTNUM

「(a)　変数 SCI の求め方」の変数*MLTNUM* を参照してください。

ipv4_num

自ノードの有効な IPv4 アドレスの数（単位：個）

(c)　変数 DICS の求め方

変数*DICS* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
DICS＝ tbldef_cache_size ＋ 84
```

変数の説明

tbldef_cache_size：サーバ定義adb_sql_tbldef_cache_size オペランドの指定値

(d) 変数 BUFGLOBAL の求め方

変数*BUFGLOBAL* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
BUFGLOBAL＝
  ↑（BUFDEF＋BUFCTL＋BUFINT＋BUFPGE＋BUFTBL＋BUFUPDLISTCTL※）÷1,024↑×1.05
```

注※

マルチノード機能を使用する場合に加算してください。

変数の説明

BUFDEF：adbbuff オペランド解析

次に示す計算式から求めてください。

計算式（単位：バイト）

```
BUFDEF＝ 3,640＋608×BUFOPE_NUM
```

BUFOPE_NUM：サーバ定義adbbuff オペランドの定義数

この合計値には次に示すadbbuff オペランドも含まれます。

- -o オプションを指定したadbbuff オペランド
- -n オプションおよび-o オプションを指定しなかったadbbuff オペランド

それぞれの詳細について説明します。

-o オプションを指定した adbbuff オペランド

-o オプションを指定した場合，adbbuff オペランドの-n オプションに指定しなかった DB エリアはページサイズごとに分類されて自動的にadbbuff オペランドが定義されます。それを踏まえて，*BUFOPE_NUM* を求めてください。例を次に示します。

例：次に示す 4 つの DB エリアを-n オプションで指定しなかった場合

■対象の DB エリア

- ADBDIC，ADBSTBL，ADBUTBL01，ADBUTBL02（ページサイズ：4 キロバイト）
- ADBUTBL03（ページサイズ：32 キロバイト）

■サーバ定義の指定例

```
adbbuff -g SAMPLEBUF -o
```

この場合，4 つの DB エリアはページサイズごとに分類されて自動的に定義されます。そのため，*BUFOPE_NUM* は 2 となります。

■自動的に定義された例

```
adbbuff -g SAMPLEBUF#0000004096 -n ADBDIC, ADBSTBL, ADBUTBL01, ADBUTBL02
adbbuff -g SAMPLEBUF#0000032768 -n ADBUTBL03
```

注　下線部分は自動的に定義されたときに追加されたグローバルバッファ名です。

-n オプションおよび-o オプションを指定しなかった adbbuff オペランド

adbbuff オペランドの-n オプションに指定しなかった DB エリアがあり，-o オプションも指定しなかった場合，-o オプションを指定したときと同じように，DB エリアはページサイズごとに分類されて自動的にadbbuff オペランドが定義されます。それを踏まえて，*BUFOPE_NUM* を求めてください。例を次に示します。

例：DB エリア（ADBDIC，ADBSTBL）を-n オプションに指定しないで，-o オプションも指定しなかった場合

■対象の DB エリア

- ADBDIC，ADBUTBL01，ADBUTBL02（ページサイズ：4 キロバイト）
- ADBUTBL03（ページサイズ：32 キロバイト）

■サーバ定義の指定例

```
adbbuff -g SAMPLEBUF01 -n ADBUTBL01, ADBUTBL02
adbbuff -g SAMPLEBUF02 -n ADBUTBL03
```

この場合，指定されていないADBDIC およびADBSTBL は自動的に定義されます。そのため，*BUFOPE_NUM* は指定した 2 つと合わせて 4 となります。

■自動的に定義された例

```
adbbuff -g ##ADBOTHER#0000004096 -n ADBDIC, ADBSTBL
```

注　下線部分は自動的に定義されたグローバルバッファ名です。

BUFCTL：グローバルバッファ管理領域

次に示す計算式から求めてください。

計算式（単位：バイト）

```
BUFCTL＝
    568×ΣOPTTYPES＋88×DBAREA_NUM＋256×BUFOPT_V_NUM＋18,896＋240×RTHNUM
```

Σ*OPTTYPES*：各adbbuff オペランドに指定したオプション（-p，-a）の指定数の合計値

この合計値には次に示すadbbuff オペランドも含まれます。

- -o オプションを指定したadbbuff オペランド
- -n オプションおよび-o オプションを指定しなかったadbbuff オペランド

詳細については，「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」の変数 Σ（*BUFPAGES* ＋ *SCANPAGES*）を参照してください。

オプションの指定数の求め方について，例を次に示します。

例

adbbuff オペランドを次に示すとおりに定義した場合，ADBDIC およびADBSTBL への自動定義を含め，オプションの指定数は合計５（１＋２＋１＋１）となります。

■サーバ定義の指定例

```
adbbuff -g SAMPLEBUF1 -n ADBUTBL01 -p 100
adbbuff -g SAMPLEBUF2 -n ADBUTBL02 -p 100 -a 100
adbbuff -g SAMPLEBUF3 -n ADBUTBL03※
```

注※　-p オプションが仮定されます。

■自動的に定義された例

```
adbbuff -g ##ADBOTHER#0000004096 -n ADBDIC, ADBSTBL※
```

注　下線部分は自動的に定義されたグローバルバッファ名です。

注※　-p オプションが仮定されます。

DBAREA_NUM：

DB エリアの総数

BUFOPT_V_NUM：

-v オプション，または-k オプションが指定されているadbbuff オペランドの定義数

RTHNUM：

「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

BUFINT：多重初期化用一時領域

次に示す計算式から求めてください。

計算式（単位：バイト）

$$BUFINT = 480 + 80 \times \Sigma BUFWRK_NUM + 8 \times rthd_num + INITQUE$$

rthd_num：

サーバ定義adb_sys_rthd_num オペランドの指定値

INITQUE：初期化処理待ちキュー

次に示す計算式から求めてください。

計算式（単位：バイト）

$$INITQUE = 80 \times RTHNUM \times RTHNUM + 632 \times RTHNUM + 960 + 64 \times \Sigma BUFWRK_NUM$$

RTHNUM：「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

Σ*BUFWRK_NUM*：adbbuff オペランドごとに計算した変数*BUFWRK_NUM* の合計値

この合計値には次に示すadbbuff オペランドも含まれます。

- -o オプションを指定したadbbuff オペランド
- -n オプションおよび-o オプションを指定しなかったadbbuff オペランド

詳細については，「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」の変数 Σ（*BUFPAGES* ＋ *SCANPAGES*）を参照してください。

BUFWRK_NUM：次に示す計算式から求めてください。

計算式（単位：個）

$$BUFWRK_NUM = 1 + OPTTYPES + \left\lceil \frac{adbbuff_opt_p}{500,000} \right\rceil + \left\lceil \frac{adbbuff_opt_a}{500,000} \right\rceil + BUFOPT_V + SCAN_INIT$$

OPTTYPES：

「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」の変数 *OPTTYPES* を参照してください。

adbbuff_opt_p：

adbbuff オペランドの-p オプションの指定値

adbbuff_opt_a：

adbbuff オペランドの-a オプションの指定値

BUFOPT_V：adbbuff オペランドの-v オプション，または-k オプションの指定有無

-v オプション，または-k オプションが指定されている場合は1，それ以外の場合は0 を代入してください。

SCAN_INIT：テーブルスキャンバッファのマルチ初期化用領域

adbbuff オペランドの-v オプション，または-k オプションが指定されている場合は，サーバ定義 adb_sys_rthd_num オペランドの指定値を代入してください。-v オプション，および-k オプションが指定されていない場合は0 を代入してください。

BUFPGE：ページ管理領域

次に示す計算式から求めてください。

計算式（単位：バイト）

$$BUFPGE = \Sigma TYPEBUF + MSTBUF + WRKBUF$$

Σ*TYPEBUF*：各adbbuff オペランドに指定したオプション（-p，-a）ごとに求めた変数 *TYPEBUF* の合計値

この合計値には次に示すadbbuff オペランドも含まれます。

- -o オプションを指定したadbbuff オペランド
- -n オプションおよび-o オプションを指定しなかったadbbuff オペランド

詳細については，「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」の変数 Σ（*BUFPAGES* ＋ *SCANPAGES*）を参照してください。

TYPEBUF：次に示す計算式から求めてください。

計算式（単位：バイト）

$$TYPEBUF = GCLBUF\ (page_num) + BUFMEM\ (page_num)$$

page_num：

サーバ定義adbbuff オペランドに指定した各オプション（-p，-a）のどれかの指定値

GCLBUF：次に示す計算式から求めてください。

計算式（単位：バイト）

GCLBUF（X） ＝ 192×*RTHNUM*＋192＋*X*×（176＋*BUFLOG*）

RTHNUM：

「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

X：

対象となるページ数

BUFLOG：ログ領域

次に示す計算式から求めてください。

計算式（単位：バイト）

BUFLOG＝ ↓（（（16＋（（*RTHNUM*-6）÷8＋1））＋15）÷16）↓×16

BUFMEM：次に示す計算式から求めてください。

計算式（単位：バイト）

BUFMEM（X） ＝
480×*RTHNUM*×*RTHNUM*＋（584＋24×*UTHNUM*）×*RTHNUM*
＋（352＋16×*RTHNUM*）×*HASHTBL_NUM*＋80×*X*＋24×*UTHNUM*＋368

RTHNUM：

「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

UTHNUM：

「(a)　変数 SCI の求め方」の変数*UTHNUM* を参照してください。

X：

対象となるページ数

HASHTBL_NUM：次に示す計算式から求めてください。

計算式

HASHTBL_NUM＝ *X*÷4

MSTBUF：次に示す計算式から求めてください。

計算式（単位：バイト）

MSTBUF＝ *RTHNUM*×2

RTHNUM：

「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

WRKBUF：

サーバ定義adb_dbbuff_wrktbl_glb_blk_num オペランドの指定値

BUFTBL：テーブルスキャンバッファ

次に示す計算式から求めてください。

計算式（単位：バイト）

BUFTBL ＝ Σ*SCANBUF*

Σ*SCANBUF*：各adbbuff オペランドに指定したテーブルスキャン用一括読み込みバッファの数

adbbuff オペランドに指定した-v オプション，または-k オプションごとに計算した変数*SCANBUF*の合計値になります。

SCANBUF：次に示す計算式から求めてください。

計算式（単位：バイト）

$$SCANBUF = SCANBLOCKS \times \left(64 + \frac{4,194,304}{page_size} \times 384\right) + rthd_num \times 8,896 + 320$$

なお，-v オプション，および-k オプションを指定しなかったadbbuff オペランドについては，*SCANBUF* を 0 としてください。

- *SCANBLOCKS*

  テーブルスキャンバッファのブロック数

  「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」の変数*SCANBLOCKS* を参照して求めてください。

- *page_size*

  -v オプション，または-k オプションを指定したadbbuff オペランドの-n オプションに指定したDB エリアのページサイズ（単位：バイト）

  「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3 DB エリアのページサイズ」を参照して求めてください。

- *rthd_num*

  サーバ定義adb_sys_rthd_num オペランドの指定値

BUFUPDLISTCTL

ノード間更新リスト管理領域

次に示す値を代入してください。

値（単位：バイト）

BUFUPDLISTCTL＝ 576

(e)　変数 DBAREAINF の求め方

変数*DBAREAINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$
\begin{aligned}
\mathrm{DBAREAINF} = {} & 10{,}068 + 64 \times \mathrm{dbarea_num} + 11 \times \mathrm{dbarea_file_num} \\
& + (\mathrm{table_num} + \mathrm{index_num}) \times 1{,}876 \\
& + \left\{ \left\lceil \frac{32 + \left(16 + \mathrm{dbarea_file_num} \times \left\lceil \frac{\mathrm{SEGSIZE}}{16} \right\rceil \times 8 \right) \times 600}{1{,}024} \right\rceil \right\} \\
& + (\mathrm{max_users} + \mathrm{rthd_num}) \times 11 + 3 \times \mathrm{dbarea_file_num} \times \mathrm{column_table_num}
\end{aligned}
$$

変数の説明

dbarea_num

すべての DB エリア数

dbarea_file_num

すべての DB エリアファイル数

table_num

ユーザが定義した実表の総数

column_table_num

ユーザが定義したカラムストア表の実表の総数

index_num

ユーザが定義したインデクスの総数

SEGSIZE

DB エリアファイルを定義した DB エリアのセグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

SEGSIZE＝ 1セグメントの容量（単位：バイト）÷*page_size*

- *page_size*

  DB エリアファイルを定義した DB エリアのページサイズ（単位：バイト）

  「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3 DB エリアのページサイズ」を参照して求めてください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

rthd_num

サーバ定義adb_sys_rthd_num オペランドの指定値

(f)　変数 DBSYSINF の求め方

変数*DBSYSINF* には，次に示す値を代入してください。

値（単位：キロバイト）

DBSYSINF＝ 1,308

(g) 変数 RECCTL の求め方

変数*RECCTL* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

RECCTL＝ 2,177＋（65＋64×log_usrbuf_num）×（log_usrfile_num）＋max_users

変数の説明

log_usrbuf_num：サーバ定義adb_log_usrbuf_num オペランドの指定値

log_usrfile_num：サーバ定義adb_log_usrfile_num オペランドの指定値

max_users：サーバ定義adb_sys_max_users オペランドの指定値

(h) 変数 DBUPDINFMNG の求め方

変数*DBUPDINFMNG* には，次に示す値を代入してください。

なお，マルチノード機能を使用するかどうかで，値が異なります。

値（単位：キロバイト）

■マルチノード機能を使用しない場合
DBUPDINFMNG＝ 89

■マルチノード機能を使用する場合
DBUPDINFMNG＝ 369

(i) 変数 RNGPREREADINF の求め方

変数*RNGPREREADINF* は，次に示す計算式から求めてください。サーバ定義adb_sql_rngidx_preread オペランドに指定した項目によって計算式が異なります。

なお，adb_sql_rngidx_preread オペランドにNO を指定した場合は，変数*RNGPREREADINF* は 0 になります。

■adb_sql_rngidx_preread オペランドにALL を指定した場合

計算式（単位：キロバイト）

$$RNGPREREADINF = \left\lceil \frac{\{484 + (uthd_num + 2) \times 8\}}{1,024} \right\rceil$$

変数の説明

uthd_num：サーバ定義adb_sys_uthd_num オペランドの指定値

■adb_sql_rngidx_preread オペランドにレンジインデクス名を指定した場合

計算式（単位：キロバイト）

$$RNGPREREADINF = \left\uparrow \frac{\left\{484 + 224 \times preread_rng_num + (uthd_num + 2) \times 8\right\}}{1,024} \right\uparrow$$

変数の説明

preread_rng_num：adb_sql_rngidx_preread オペランドに指定したレンジインデクス名の数

uthd_num：サーバ定義adb_sys_uthd_num オペランドの指定値

(j)　変数 RNGPREREADHANDLE の求め方

変数*RNGPREREADHANDLE* は，次に示す値を代入してください。

値（単位：キロバイト）

RNGPREREADHANDLE＝ 1

(k)　変数 MODA の求め方

変数*MODA* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

MODA＝ 9,216＋↑*wrk_page_size*÷1,024↑

変数の説明

wrk_page_size

作業表用 DB エリアのページサイズ（単位：バイト）

「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」を参照して求めてください。

(l)　変数 LSINF の求め方

変数*LSINF* は，次に示す値を代入してください。

値（単位：キロバイト）

LSINF＝ 1

(m)　変数 DBHTRC の求め方

変数*DBHTRC* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
DBHTRC＝4,570×RTHNUM
```

変数の説明

RTHNUM

「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

(n)　変数 LSTCMNINF の求め方

変数*LSTCMNINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
LSTCMNINF ＝ ↑（4,128＋8,200×max_users）÷1,024↑
```

変数の説明

max_users

サーバ定義adb_sys_max_users オペランドの指定値

(o)　変数 LSTRTHDINF の求め方

変数*LSTRTHDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
LSTRTHDINF ＝ ↑（8,200×RTHNUM）÷1,024↑
```

変数の説明

RTHNUM

「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

(p)　変数 PROC_STAINF の求め方

変数*PROC_STAINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_STAINF＝ ↑max_users×（443,472×128＋16）÷1,024↑
```

変数の説明

max_users

サーバ定義adb_sys_max_users オペランドの指定値

(q) 変数 SYNDICTINF の求め方

変数*SYNDICTINF* は，次に示す値を代入してください。

値（単位：キロバイト）

```
SYNDICTINF＝ 76,951
```

(r) 変数 AUDINF の求め方

変数*AUDINF* は，次に示す計算式から求めてください。監査証跡機能が有効な場合に加算してください。監査証跡機能を無効にすると，変数*AUDINF* で求めた値は解放されます。

計算式（単位：キロバイト）

```
AUDINF＝ 1＋PROC_AUDINFSZ
```

変数の説明

PROC_AUDINFSZ：監査証跡の情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_AUDINFSZ＝ 2＋（↑sql_text_size÷1,024↑）※1
              ＋（↑PARAM_INFO_SIZE÷1,024↑）※2
              ＋OBJECT_INFO_SIZE※3
              ＋system_info_size
```

注※1

SQL 文の実行時に加算してください。

注※2

？パラメタを含む SQL 文の実行時に加算してください。

注※3

SQL 文，またはコマンドの実行時に加算してください。

sql_text_size

SQL テキスト長（単位：バイト）

PARAM_INFO_SIZE

パラメタ格納領域（単位：バイト）

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」にある変数*PARAM_INFO_SIZE*を参照してください。

OBJECT_INFO_SIZE

オブジェクト情報（単位：キロバイト）

次に示す計算式から求めてください。

```
OBJECT_INFO_SIZE＝ 1×obj_num×user_num
```

obj_num

SQL 文，またはコマンドに指定したオブジェクト数

GRANT 文やREVOKE 文にALL TABLES を指定した場合は，ALL TABLES の対象となる表数となります。

user_num

GRANT 文やREVOKE 文に指定した権限受領者数

権限受領者を指定しない SQL 文やコマンドの場合は，1 になります。

system_info_size

システム情報（単位：キロバイト）

HADB サーバ開始時のメモリ所要量を求める場合は，次の計算式で求めてください。

```
system_info_size = ↑サーバ定義ファイルのファイルサイズ※÷1,024↑
```

注※

サーバ定義ファイルのファイルサイズの単位はバイトです。

HADB サーバ開始時以外のメモリ所要量を求める場合は，次のどちらかで求めてください。

- adbclientdefmang コマンド実行時

  ```
  system_info_size = 2×↑クライアント管理定義ファイルのファイルサイズ※÷1,024↑
  ```

  注※

  クライアント管理定義ファイルのファイルサイズの単位はバイトです。

- adbclientdefmang コマンド実行時以外

  ```
  system_info_size = 0
  ```

(s)　変数 CLMNZ の求め方

変数*CLMNZ* は，次に示す値を代入してください。

値（単位：キロバイト）

```
CLMNZ ＝ 5,120＋1
```

(4)　リアルスレッド固有メモリの所要量の求め方（HADB サーバ開始時）

HADB サーバ開始時のリアルスレッド固有メモリ（*RTHD_STARTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_STARTSZ＝ RTI＋DICL＋DICL2※1＋RTHD_STAINF＋MLTMNG※2＋TRCMON＋AUDTHDINF※3
```

注※1

HADB サーバのバージョンアップを実施するときに加算してください。

注※2

マルチノード機能を使用するときに加算してください。

注※3

監査証跡機能が有効なときに加算してください。

変数の説明

RTI

リアルスレッド管理情報

次に示す計算式で求めてください。

計算式（単位：キロバイト）

```
RTI＝　↑（AIO＋DBI）÷1,024↑
```

AIO

非同期 I/O 管理情報

次に示す計算式で求めてください。

計算式（単位：バイト）

AIO＝ 32×*UTHNUM*

UTHNUM

「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a)　変数 SCI の求め方」の変数 *UTHNUM* を参照してください。

DBI

次に示す計算式で求めてください。

計算式（単位：バイト）

DBI＝ 336×*UTHNUM*

DICL

次に示す値を代入してください。

値（単位：キロバイト）

```
DICL＝ 46
```

DICL2

次に示す値を代入してください。

値（単位：キロバイト）

```
DICL2＝ 7,790
```

RTHD_STAINF

統計情報の制御情報

次に示す計算式で求めてください。

計算式（単位：キロバイト）

```
RTHD_STAINF＝
    (394＋tables×6) × (2＋max_sql_concurrent_exec_num)
```

tables

1 トランザクション内で実行する SQL 文に出現する表の延べ数の合計

max_sql_concurrent_exec_num

1 トランザクション内で同時に実行する SQL 文の数の最大値

MLTMNG

マルチノード機能管理情報

次に示す計算式で求めてください。

計算式（単位：キロバイト）

```
MLTMNG＝
   ↑ {2,120＋↑ (1,216×authid_num) ÷4,096↑ ×4,096
      ＋ (8,536＋760×UTHNUM) × (multi_node_num＋worker_node_num)
      } ÷1,024↑
```

authid_num

クライアント管理定義のadbclientmang オペランドに指定した認可識別子の総数

multi_node_num

サーバ定義adb_sys_multi_node_info オペランドに指定したホスト名の数

worker_node_num

サーバ定義adb_sys_multi_max_worker_nodes オペランドの指定値

TRCMON

SQL トレース機能のモニタアウトスレッド管理情報

次に示す計算式で求めてください。

計算式（単位：キロバイト）

```
TRCMON ＝ 98,323＋(RTHD_EXPSQLTRCSZ×3)※
```

注※

SQL トレースバッファ拡張時に確保されるメモリです。

RTHD_EXPSQLTRCSZ

SQL トレースバッファの拡張時に使用するリアルスレッド固有メモリ

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(i)　変数 RTHD_EXPSQLTRCSZ の求め方」を参照して求めてください。

AUDTHDINF

監査証跡の管理情報

次に示す計算式で求めてください。監査証跡機能を無効にすると，変数*AUDTHDINF* で求めた値は解放されます。

計算式（単位：キロバイト）

```
AUDTHDINF＝↑（272,631,408＋CTHNUM×260,000）÷1,024↑
```

CTHNUM

「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a)　変数 SCI の求め方」の変数*CTHNUM* を参照してください。

(5)　ヒープメモリの所要量の求め方（HADB サーバ開始時）

HADB サーバ開始時のヒープメモリ（*HEAP_STARTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_STARTSZ = 161 × UTHNUM × RTHNUM
```

変数の説明

UTHNUM

「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a)　変数 SCI の求め方」の変数*UTHNUM* を参照してください。

RTHNUM

「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a)　変数 SCI の求め方」の変数*RTHNUM* を参照してください。

6.3.4　通常運用時のメモリ所要量の求め方

HADB サーバ開始後，通常運用時（HADB クライアントが HADB サーバに接続するとき，および SQL 文を実行するとき）に，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_EXECSZ*）
- リアルスレッド固有メモリ（*RTHD_EXECSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_EXECSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（通常運用時）

通常運用時のプロセス共通メモリ（*PROC_EXECSZ*）の所要量は，次の計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_EXECSZ＝
    PROC_CNCTSZ＋PROC_TOTALSQLSSZ＋PROC_EXECSQLSZ
    ＋PROC_BUFWORK_CTL＋PROC_DEFSQLSZ＋PROC_IMPORTSZ
    ＋PROC_IDXRBLDSZ＋PROC_GETCOSTSZ＋PROC_DBSTATUSSZ
    ＋PROC_MODASZ＋PROC_EXPORTSZ＋PROC_MERGCSZ
    ＋PROC_UPDSZ＋PROC_CHGCCMSZ＋PROC_CHGCSTSZ
    ＋PROC_UPDLISTSZ※＋PROC_ARCCKSZ＋PROC_UNARCCKSZ
    ＋PROC_REORGSZ＋PROC_SYNDICTSZ＋PROC_AUDTRAILSZ
```

注※

マルチノード機能を使用する場合に加算してください。

変数の説明

- *PROC_CNCTSZ*
  データベース接続時に使用するプロセス共通メモリ
  「(a)　変数 PROC_CNCTSZ の求め方」を参照してください。
- *PROC_TOTALSQLSSZ*
  SQL 前処理時に使用するプロセス共通メモリ
  「(b)　変数 PROC_TOTALSQLSSZ の求め方」を参照してください。
- *PROC_EXECSQLSZ*
  SQL 実行時に使用するプロセス共通メモリ
  「(c)　変数 PROC_EXECSQLSZ の求め方」を参照してください。
- *PROC_BUFWORK_CTL*

作業表使用時に使用するプロセス共通メモリ

「(d)　変数 PROC_BUFWORK_CTL の求め方」を参照してください。

- *PROC_DEFSQLSZ*

  定義系 SQL 実行時に使用するプロセス共通メモリ

  「(e)　変数 PROC_DEFSQLSZ の求め方」を参照してください。

- *PROC_IMPORTSZ*

  adbimport コマンド実行時に使用するプロセス共通メモリ

  「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」を参照して求めてください。

- *PROC_IDXRBLDSZ*

  adbidxrebuild コマンド実行時に使用するプロセス共通メモリ

  「6.3.7　adbidxrebuild コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbidxrebuild コマンド実行時）」を参照して求めてください。

- *PROC_GETCOSTSZ*

  adbgetcst コマンド実行時に使用するプロセス共通メモリ

  「6.3.8　adbgetcst コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbgetcst コマンド実行時）」を参照して求めてください。

- *PROC_DBSTATUSSZ*

  adbdbstatus コマンド実行時に使用するプロセス共通メモリ

  「6.3.9　adbdbstatus コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbdbstatus コマンド実行時）」を参照して求めてください。

- *PROC_MODASZ*

  adbmodarea コマンド実行時に使用するプロセス共通メモリ

  「6.3.12　adbmodarea コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbmodarea コマンド実行時）」を参照してください。

- *PROC_EXPORTSZ*

  adbexport コマンド実行時に使用するプロセス共通メモリ

  「6.3.10　adbexport コマンド実行時のメモリ所要量の求め方」を参照して求めてください。

- *PROC_MERGCSZ*

  adbmergechunk コマンド実行時に使用するプロセス共通メモリ

  「6.3.13　adbmergechunk コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbmergechunk コマンド実行時）」を参照して求めてください。

- *PROC_UPDSZ*

  データベース接続後の定義系 SQL の初回実行時，および更新系 SQL の初回実行時に使用するプロセス共通メモリ

  「(f)　変数 PROC_UPDSZ の求め方」を参照して求めてください。

- *PROC_CHGCCMSZ*

  adbchgchunkcomment コマンド実行時に使用するプロセス共通メモリ

  「6.3.14　adbchgchunkcomment コマンド実行時のメモリ所要量の求め方」を参照して求めてください。

- *PROC_CHGCSTSZ*

  adbchgchunkstatus コマンド実行時に使用するプロセス共通メモリ

  「6.3.15　adbchgchunkstatus コマンド実行時のメモリ所要量の求め方」を参照して求めてください。

- *PROC_UPDLISTSZ*

  マルチノード機能適用時に使用するプロセス共通メモリ

  「(g)　変数 PROC_UPDLISTSZ の求め方」を参照してください。

- *PROC_ARCCKSZ*

  adbarchivechunk コマンド実行時に使用するプロセス共通メモリ

  「6.3.16　adbarchivechunk コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbarchivechunk コマンド実行時）」を参照してください。

- *PROC_UNARCCKSZ*

  adbunarchivechunk コマンド実行時に使用するプロセス共通メモリ

  「6.3.17　adbunarchivechunk コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbunarchivechunk コマンド実行時）」を参照してください。

- *PROC_REORGSZ*

  adbreorgsystemdata コマンド実行時に使用するプロセス共通メモリ

  「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」を参照してください。

- *PROC_SYNDICTSZ*

  adbsyndict コマンド実行時に使用するプロセス共通メモリ

  「6.3.20　adbsyndict コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbsyndict コマンド実行時）」を参照してください。

- *PROC_AUDTRAILSZ*

  adbaudittrail コマンド実行時に使用するプロセス共通メモリ

  「6.3.21　adbaudittrail コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbaudittrail コマンド実行時）」を参照してください。

(a)　変数 PROC_CNCTSZ の求め方

変数 *PROC_CNCTSZ* は，データベース接続時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_CNCTSZ＝
    BUFCNCT＋SQLWQUE＋SQLACTB＋RTHDLOG＋2
```

変数の説明

BUFCNCT：コネクションバッファ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
BUFCNCT＝
        ↑（56＋8,216×rthd_num）×max_users÷1,024↑
```

rthd_num

次の計算式から求めてください。

```
サーバ定義adb_sys_rthd_numオペランドの指定値 ＋ 1
```

ただし，サーバ定義に-v オプション，または-k オプションを指定したadbbuff オペランドがない場合は，変数*rthd_num* は0 としてください。

max_users

サーバ定義adb_sys_max_users オペランドの指定値

SQLWQUE：SQL 文実行待ちキュー

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SQLWQUE＝ ↑8×max_users÷1,024↑
```

max_users：サーバ定義adb_sys_max_users オペランドの指定値

SQLACTB：SQL 文テーブルアクセス情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SQLACTB＝
    ↑56×max_table_num÷1,024↑
```

max_table_num：対象のコネクションで実行する各 SQL 文でアクセスする表数の最大値

RTHDLOG

処理リアルスレッド用ログ管理領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHDLOG＝
    ↑(48＋8×rthd_num)÷1,024↑
```

rthd_num

サーバ定義adb_sys_rthd_num オペランドの指定値

(b) 変数 PROC_TOTALSQLSSZ の求め方

変数*PROC_TOTALSQLSSZ* は，SQL の前処理時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_TOTALSQLSSZ＝ *MAXSQLWRK*＋*TOTALSQLSCT*＋*PREDICSZ*＋*PROC_AUDINFSZ*

変数の説明

MAXSQLWRK

1 トランザクション中に，SQL の前処理で使用される作業領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

MAXSQLWRK＝ （*max_sql*（*SQLPSZ*）＋20）×*max_users*

max_sql（）

SQLPSZ の計算結果から最大値を代入します。

max_users

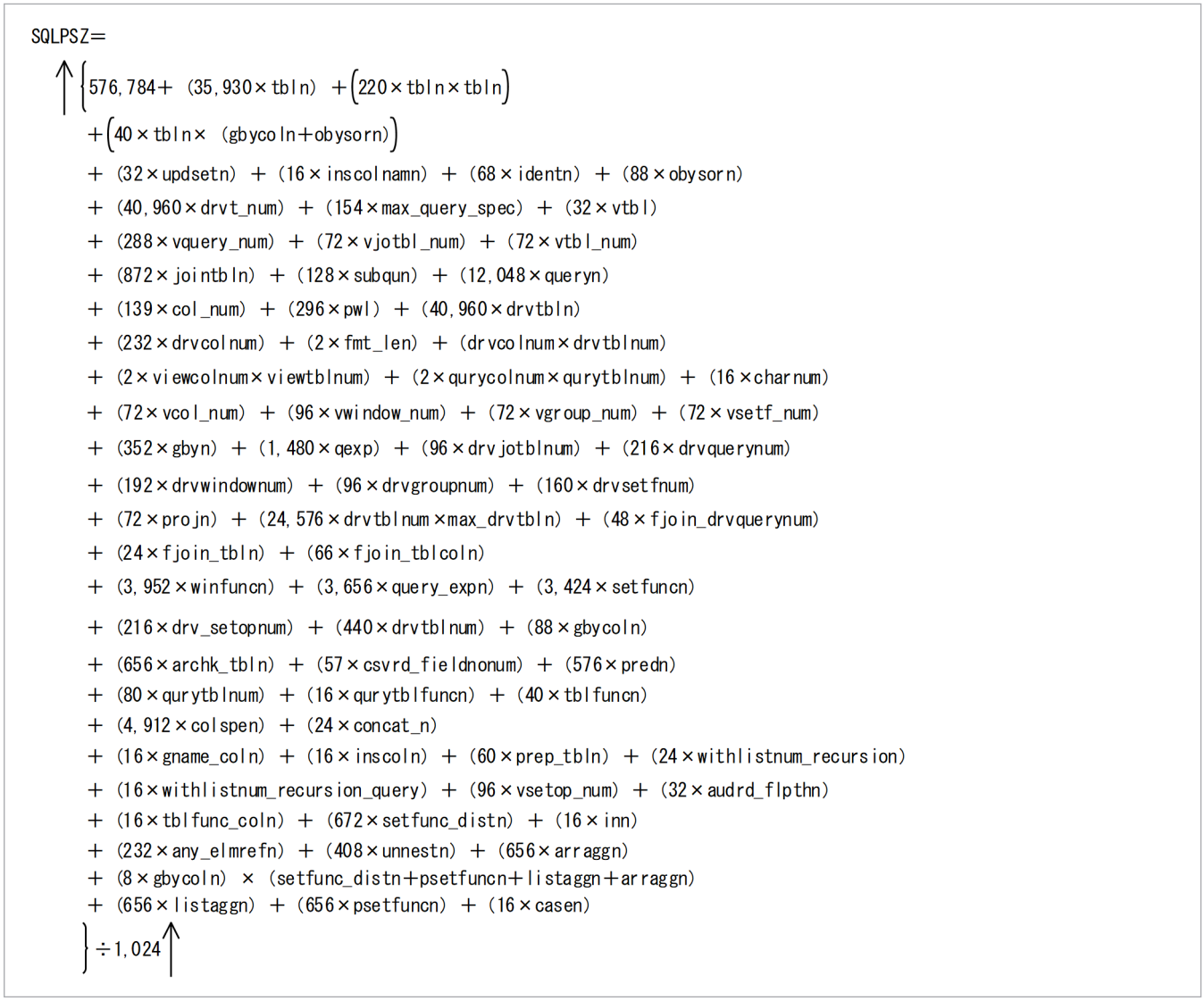
サーバ定義adb_sys_max_users オペランドの指定値

SQLPSZ

SQL 文の前処理で使用される作業領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）



tbln

FROM 句に指定した表数

gbycoln

GROUP BY 句に指定した列の数

obysorn

ORDER BY 句に指定したソートキーの数

updsetn

UPDATE 文のSET 句の数

inscolnamn

INSERT 文の挿入対象列の数

identn

識別子の数

drvt_num

導出表の個数

max_query_spec

1SQL 文に指定できる問合せ指定の最大数

vtbl

1SQL 文中のビュー表の数

vquery_num

ビュー定義に指定した問合せ指定の数

vjotbl_num

ビュー定義に指定した結合表の数

vtbl_num

ビュー定義に指定した実表の数

jointbln

結合表の数

subqun

副問合せの数

queryn

問合せ指定の数

col_num

選択式の数

pwl

WITH 句の個数

drvtbln

FROM 句に指定した導出表の数

drvcolnum

導出表の列数

fmt_len

スカラ関数CONVERT の書式サイズ（単位：バイト）

drvtblnum

導出表の数

viewcolnum

ビュー表の列数

viewtblnum

ビュー表の数

qurycolnum

問合せ名の列数

qurytblnum

問合せ名の数

charnum

1SQL 文のバイト数

vcol_num

ビュー定義に指定した列の数

vwindow_num

ビュー定義に指定したウィンドウ関数の数

vgroup_num

ビュー定義に指定したGROUP BY 句の数

vsetf_num

ビュー定義に指定した集合関数の数

gbyn

GROUP BY 句の数

qexp

集合演算の数

drvjotblnum

導出表に指定した結合表の数

drvquerynum

導出表に指定した問合せの数

drvwindownum

導出表に指定したウィンドウ関数の数

drvgroupnum

導出表に指定したGROUP BY 句の数

drvsetfnum

導出表に指定した集合関数の数

projn

値式の数

max_drvtbln

FROM 句に指定できる表数の最大値

fjoin_drvquerynum

FULL OUTER JOIN 中に含まれる導出表に指定した問合せの数

fjoin_tbln

FULL OUTER JOIN 中の表参照の数

fjoin_tblcoln

FULL OUTER JOIN 中の表参照の列数

winfuncn

ウィンドウ関数の数

query_expn

問合せ式の数

setfuncn

集合関数の数（AVG の場合は＋２とします）

drv_setopnum

導出表に指定した集合演算の数

archk_tbln

アーカイブマルチチャンク表の数

csvrd_fieldnonum

ADB_CSVREAD 関数に指定されたフィールドデータ番号の数

predn

述語の数

qurytblfuncn

問合せ名に指定した表関数導出表の数

tblfuncn

表関数導出表の数

colspen

列指定の数

concat_n

連結演算の数

gname_coln

グループ化列の列名の数

inscoln

挿入列の数

prep_tbln

前処理表の数

withlistnum_recursion

再帰的問合せのWITH リスト要素の数

withlistnum_recursion_query

再帰的問合せ中の問合せ指定の数

vsetop_num

ビュー定義に指定した集合演算の数

audrd_flpthn

ADB_AUDITREAD 関数に指定した監査証跡ファイルのパス名の数

tblfunc_coln

表関数導出表の列数

setfunc_distn

DISTINCT 集合関数の数

inn

IN 述語の数

any_elmrefn

ANY を指定した配列要素参照の数

unnestn

集まり導出表の数

arraggn

ARRAY_AGG 集合関数の数

psetfuncn

逆分布関数の数

listaggn

LISTAGG 集合関数の数

casen

CASE 式の数

TOTALSQLSCT

1 トランザクション中で作成される，SQL 文の前処理の前処理結果管理領域の合計

なお，JDBC ドライバから SQL 文を実行する場合は，1 コネクション内で同時に作成される SQL 文の前処理の前処理結果管理領域の合計となります。

次に示す計算式から求めてください。

計算式（単位：キロバイト）

TOTALSQLSCT ＝ *sum_sql*（*SQLPSSZ*＋*SQLSSZ*＋*APATHVIEW*）×*max_users*

max_users

サーバ定義adb_sys_max_users オペランドの指定値

sum_sql（）

変数 *SQLSSZ* の計算結果から最大値を代入します。

なお，JDBC ドライバから SQL 文を実行する場合は，1 コネクション内で実行する各 SQL 文について求めた値のうち，最大値を代入します。

SQLPSSZ

SQL 文の前処理時に取得される前処理結果制御領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

SQLPSSZ＝ ↑184÷1,024↑

SQLSSZ

SQL 文の前処理で使用される前処理結果管理領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

SQLSSZ＝

↑{33,154＋（768※×tblcoln）＋（2,224×tblidxn）＋（8,350×tbln）
＋（1,256×col_num）＋（8×insvaln）＋（280×inscoln）＋（173×updcoln）
＋（2,368×colspen）＋（1,656×gbycoln）＋（104×operan）＋（16×liken）＋（128×inn）
＋（188×scafuncn）＋（7,784×setfuncn）＋（224×notn）＋（928×orn）
＋（24×param_num）＋（16×sregn）＋（176×rown）＋（64×jointbln）
＋（704×inscolnamn）＋（2,032×subqun）＋（616×ord_num）＋（6,810×queryn）
＋（256×logicn）＋（920×predn）＋（128×base_num）＋（244×sel_num）
＋（72×pwg）＋（240×scl）＋（8×sgc）＋（8×ssk）＋（16×srw）
＋（944×exp_num）＋（2×view_size）＋（sum_view_object（view_size）×2.5）
＋（1,432×qexp_coln）＋（96×gname_coln）＋（24×default_coln）
＋（16×act_num）＋（16×decode_revn）
＋（2×fmt_len）＋（168×obysorn）＋（16×pw29）
＋（24×projn）＋（2,168×fulljoin_num）＋（920×fj_join_num）
＋（208×gbyn）＋（384×qexp）＋（3,728×query_expn）＋（30,888×winfuncn）
＋（576×winfuncn×winfuncn）＋（576×col_num×winfuncn）＋（576×gbycoln×winfuncn）
＋（576×setfuncn×winfuncn）＋（24×withlistnum）
＋（16×likeregn）＋（688×tblfuncn）＋（112×multisetn）＋（40×rowvaluecon）
＋（32×tablevaluecon）＋（（8×rowvaluecon＋264）×rowvaluecon_coln）
＋（160×gexp_coln）＋（8×fucall_paramn）＋（64×csvrd_n）
＋（5×csvrd_fieldnonum）＋（48×multiset_projn）＋（2,928×archk_tbln）
＋（5×archk_coln）＋（320×drvtblnum）＋（16×comp_dec_num）＋（104×contains_n）
＋（13×def_char_datesn）＋（402×drvcolnum）＋（256×tblfunc_coln）
＋（16×pw_when）＋（regex_num）＋（8×scalfunc_paramn）＋（176×setfuncn0）
＋（48×winf_part_projn）＋（1,661×audrd_n）＋（16×ltdecode_revn）
＋（64×randcursornum）＋（72×randrownum）＋（2,096×listaggn）
＋（96×elmrefn）＋（352×unnestn）＋（304×unnest_coln）
＋（2,048×arraggn）＋（2,048×setfunc_distn）＋（2,048×psetfuncn）
＋（120×randnum）＋（64×rand_normalnum）
}÷1,024↑

注※

コスト情報を収集した表の場合は，32,776 となります。

tblcoln

処理対象表の列数

tblidxn

処理対象表に定義したインデクス数

tbln

FROM 句に指定した表の数

col_num

選択式の数

insvaln

挿入値の数

inscoln

挿入列の数

updcoln

更新列の数

colspen

列指定の数

gbycoln

GROUP BY 句に指定したグループ化列の数

operan

四則演算の数

liken

LIKE 述語の数

inn

IN 述語の数

scafuncn

スカラ関数の数

setfuncn

集合関数の数（AVG の場合は＋ 2 とします）

notn

NOT の数

orn

OR の数

param_num

？パラメタの数

sregn

日時情報取得関数およびユーザ情報取得関数の総数

rown

ROW の数

jointbln

結合表の数

inscolnamn

INSERT 文の挿入対象列の数

subqun

副問合せの数

ord_num

ORDER BY 句に指定した列の数

queryn

問合せ指定の数

logicn

AND，OR およびNOT の論理演算の数

predn

述語の数

base_num

ビュー定義に指定した基表の数

sel_num

ビュー定義に指定した問合せ指定の選択式の数

pwg

ラベル付き間隔の数

scl

列数（列指定の数と，*指定の展開後の列指定の数の総和）

sgc

グループ化列数

ssk

ORDER BY 句に指定したソートキーの数

srw

ROW の対象表に含まれるDECIMAL 型，NUMERIC 型，TIMESTAMP 型の数

exp_num

値式の総数

sum_view_object（）

括弧内のものについて，1SQL 文内で指定したビュー表のビューオブジェクトサイズの合計値（単位：バイト）

view_size

ビューオブジェクトのサイズ（ディクショナリ表（実表）のSQL_VIEWS 表のVIEW_OBJECT_SIZE 列の値）（単位：バイト）

qexp_coln

集合演算中の問合せ式の列数

gname_coln

グループ化列の列名の数

default_coln

DEFAULT 句を指定した列数

act_num

CASE 式のTHEN とELSE の総数

decode_revn

スカラ関数DECODE の返却値の数

fmt_len

スカラ関数CONVERT の書式サイズ（単位：バイト）

obysorn

ORDER BY 句に指定したソート項目の数

pw29

WHEN の個数

projn

値式の数

fulljoin_num

SQL 文に指定されたFULL OUTER JOIN の数

fj_join_num

FULL OUTER JOIN に指定された結合表の数

gbyn

GROUP BY 句の数

qexp

集合演算の数

query_expn

問合せ式の数

winfuncn

ウィンドウ関数の数

withlistnum

WITH リスト要素の数

likeregn

LIKE_REGEX 述語の数

listaggn

LISTAGG 集合関数の数

tblfuncn

表関数導出表の数

multisetn

マルチ集合値式の数

rowvaluecon

行値構成子の数

tablevaluecon

表値構成子の数

rowvaluecon_coln

行値構成子の列数

gexp_coln

グループ値式の数

fucall_paramn

関数呼び出しの引数の数

csvrd_n

ADB_CSVREAD 関数の数

csvrd_fieldnonum

ADB_CSVREAD 関数に指定されたフィールドデータ番号の数

multiset_projn

マルチ集合値式のマルチ集合要素の数

archk_tbln

アーカイブマルチチャンク表の数

archk_coln

アーカイブマルチチャンク表の導出列の数

drvtblnum

導出表の数

comp_dec_num

比較演算項にDECIMAL 型またはNUMERIC 型のデータと，INTEGER 型またはSMALLINT 型のデータを指定している演算数

contains_n

スカラ関数CONTAINS の数

def_char_datesn

日時データの既定の文字列表現の数

drvcolnum

導出表の列数

tblfunc_coln

表関数導出表の列数

pw_when

CASE 式に指定したWHEN の個数

regex_num

LIKE_REGEX 述語の指定数

scalfunc_paramn

スカラ関数の引数の数

setfuncn0

集合関数の数

winf_part_projn

ウィンドウ分割句中の値式の数

audrd_n

ADB_AUDITREAD 関数の数

ltdecode_revn

スカラ関数LTDECODE の返却値の数

randcursornum

スカラ関数RANDOMCURSOR の数

randrownum

スカラ関数RANDOMROW の数

elmrefn

配列要素参照の数

unnestn

集まり導出表の数

unnest_coln

集まり導出表の列数

arraggn

ARRAY_AGG 集合関数の数

setfunc_distn

DISTINCT 集合関数の数

psetfuncn

逆分布関数の数

randnum

スカラ関数RANDOM の数

rand_normalnum

スカラ関数RANDOM_NORMAL の数

APATHVIEW

アクセスパスを表示する際に作成されるアクセスパス情報管理領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$APATHVIEW = \uparrow((9{,}728 + 1{,}024 \times scl) \times tbln + 2{,}200) \div 4{,}096 \uparrow \times 5$$

tbln

FROM 句に指定した表の数

scl

列数（列指定の数と，*指定の展開後の列指定の数の総和）

PREDICSZ

次に示す値を代入してください。

値（単位：キロバイト）

```
PREDICSZ= 19,320
```

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照してください。

(c)　変数 PROC_EXECSQLSZ の求め方

変数 *PROC_EXECSQLSZ* は，SQL 実行時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_EXECSQLSZ＝ TRANSNAP＋RTHDUPDINF＋DBUPDINF※
```

注※

PURGE CHUNK 文およびTRUNCATE TABLE 文を実行する場合に加算してください。

変数の説明

TRANSNAP

トランザクション情報スナップショット

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
TRANSNAP ＝ ↑（112＋（88×max_users））÷1,024↑
```

max_users

サーバ定義adb_sys_max_users オペランドの指定値

RTHDUPDINF

処理リアルスレッド更新情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHDUPDINF＝
    ↑（648＋（168×rthd_num））÷1,024↑
```

rthd_num

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値

なお，クライアント定義adb_sql_exe_max_rthd_num オペランドを指定した場合は，クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値を代入してください。ただし，クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値が，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値よりも大きい場合は，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値を代入してください。

詳細については，「7.2.2　性能に関するオペランド（set 形式）」のadb_sql_exe_max_rthd_num オペランドを参照してください。

DBUPDINF

DB エリア，表，インデクス，およびチャンク更新情報

次に示す計算式から求めてください。なお，マルチノード機能を使用するかどうかで，計算式が異なります。

計算式：(単位：キロバイト)

```
■マルチノード機能を使用しない場合
  DBUPDINF＝ ↑((40＋↑(archive_file_num÷1,024)↑×122,888)÷1,024)↑

■マルチノード機能を使用する場合
  DBUPDINF＝ 129＋↑((40＋↑(archive_file_num÷1,024)↑×122,888)÷1,024)↑
```

archive_file_num

アーカイブ状態のチャンクに対応しているアーカイブファイル数

adbdbstatus コマンドで「アーカイブ状態のチャンクのサマリ情報」を出力して，「Archive_file_num」を確認してください。「Archive_file_num」については，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』の『アーカイブ状態のチャンクのサマリ情報で出力される項目』の『アーカイブ状態のチャンクのサマリ情報で出力される項目の一覧』を参照してください。

(d)　変数 PROC_BUFWORK_CTL の求め方

変数 *PROC_BUFWORK_CTL* は，作業表使用時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_BUFWORK_CTL＝ max_users＋rthd_num
```

変数の説明

max_users：サーバ定義adb_sys_max_users オペランドの指定値

rthd_num：サーバ定義adb_sys_rthd_num オペランドの指定値

(e)　変数 PROC_DEFSQLSZ の求め方

変数 *PROC_DEFSQLSZ* は，定義系 SQL 実行時に加算します。次に示す値を代入してください。

なお，マルチノード機能を使用するかどうかで，代入する値が異なります。

マルチノード機能を使用しない場合に代入する値（単位：キロバイト）

```
PROC_DEFSQLSZ＝ 2,261
```

マルチノード機能を使用する場合に代入する値（単位：キロバイト）

```
PROC_DEFSQLSZ＝ 7,208
```

注

実表を定義した場合，または実表にインデクスを定義した場合は，プロセス共通メモリ（DB エリア情報）の所要量が増加します。「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時)」の「(e)　変数 DBAREAINF の求め方」を参照して，変数 *DBAREAINF* を再度求めてください。

(f)　変数 PROC_UPDSZ の求め方

変数 *PROC_UPDSZ* は，データベース接続後の定義系 SQL の初回実行時，および更新系 SQL の初回実行時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_UPDSZ ＝ *BUFUPD* ×*upd_users*

変数の説明

upd_users

更新ページ管理用のプロセス共通メモリを取得するコネクション（AP およびコマンド）の同時実行数

BUFUPD

更新ページ管理用のプロセス共通メモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

BUFUPD ＝
↑（56＋*BUFHIST* ＋*BUFFLU* ＋*BUFBLK*）÷1,024↑

更新ページ管理用のプロセス共通メモリは，次に示すタイミングで確保されます。

表 6-4　メモリが確保されるタイミング

項番	実行する種別		確保の有無
1	• AP 実行時 • adbsql コマンド実行時	• 定義系 SQL を実行したコネクション • 更新系 SQL を実行したコネクション	○※1
2		検索系 SQL だけを実行したコネクション	×※2，※3
3	adbinit コマンド実行時		○
4	adbimport コマンド実行時		○
5	adbidxrebuild コマンド実行時		○
6	adbgetcst コマンド実行時		○※4
7	adbdbstatus コマンド実行時		×
8	adbstat コマンド実行時		×
9	adbmodarea コマンド実行時		○
10	adbexport コマンド実行時		×※2
11	adbmergechunk コマンド実行時		○
12	adbchgchunkcomment コマンド実行時		○※4
13	adbmodbuff コマンド実行時		×
14	adbarchivechunk コマンド実行時		○

項番	実行する種別	確保の有無
15	adbunarchivechunk コマンド実行時	○

（凡例）

○：更新ページ管理用のプロセス共通メモリを確保します。

×：更新ページ管理用のプロセス共通メモリを確保しません。

注※1

前処理でエラーになった場合は，更新ページ管理用のプロセス共通メモリを確保しません。

注※2

作業表への書き込みのためには，更新ページ管理用のプロセス共通メモリを確保しません。

注※3

一度でも定義系 SQL，または更新系 SQL を実行したコネクションでは，コネクションバッファを確保したままとなります。

注※4

システム表を更新するため，更新ページ管理用のプロセス共通メモリを確保します。

BUFHIST

次に示す計算式から求めてください。

計算式（単位：バイト）

BUFHIST＝
116×*RTHNUM*＋44,052＋*BUFMEM*（*RTHNUM*＋257）＋*BUFMEM*（*RTHNUM*＋4,097）

BUFMEM

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFMEM* を参照してください。

BUFFLU

次に示す計算式から求めてください。

計算式（単位：バイト）

BUFFLU＝
8×*uthd_num*＋96×*RTHNUM*＋393,312

uthd_num

サーバ定義adb_sys_uthd_num オペランドの指定値

BUFBLK

次に示す計算式から求めてください。

計算式（単位：バイト）

```
BUFBLK＝
    33,734,656＋1,024×BUFLOG
```

BUFLOG

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFLOG* を参照してください。

(g)　変数 PROC_UPDLISTSZ の求め方

変数*PROC_UPDLISTSZ* は，マルチノード機能適用時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_UPDLISTSZ＝ 147×max_users
```

変数の説明

max_users

サーバ定義adb_sys_max_users オペランドの指定値

(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）

通常運用時のリアルスレッド固有メモリ（*RTHD_EXECSZ*）の所要量は，次の計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_EXECSZ ＝ RTHD_CNCTSZ＋RTHD_DEFSQLSZ＋RTHD_EXESQLSZ＋RTHD_ROLLBKSZ
             ＋RTHD_DBEXTSZ＋RTHD_WORKBUF＋RTHD_EXESQLDICSZ
             ＋RTHD_COMMUSZ＋RTHD_MODASZ＋RTHD_EXPSQLTRCSZ
             ＋RTHD_AUDINFSZ＋RTHD_COLUMNIZESZ
```

変数の説明

- *RTHD_CNCTSZ*

  データベース接続時に使用するリアルスレッド固有メモリ

  「(a)　変数 RTHD_CNCTSZ の求め方」を参照して求めてください。

- *RTHD_DEFSQLSZ*

  定義系 SQL の実行時に使用するリアルスレッド固有メモリ

  「(b)　変数 RTHD_DEFSQLSZ の求め方」を参照して求めてください。

- *RTHD_EXESQLSZ*

  操作系 SQL の実行時に使用するリアルスレッド固有メモリ

  「(c)　変数 RTHD_EXESQLSZ の求め方」を参照して求めてください。

- *RTHD_ROLLBKSZ*
  ロールバック処理時，またはHADBサーバの再開始時に使用するリアルスレッド固有メモリ
  「(d)　変数RTHD_ROLLBKSZの求め方」を参照して求めてください。
- *RTHD_DBEXTSZ*
  DBエリア拡張時に使用するリアルスレッド固有メモリ
  「(e)　変数RTHD_DBEXTSZの求め方」を参照して求めてください。
- *RTHD_ WORKBUF*
  作業表使用時に使用するリアルスレッド固有メモリ
  「(f)　変数RTHD_WORKBUFの求め方」を参照して求めてください。
- *RTHD_EXESQLDICSZ*
  操作系SQLの前処理時に使用するリアルスレッド固有メモリ
  「(g)　変数RTHD_EXESQLDICSZの求め方」を参照して求めてください。
- *RTHD_COMMUSZ*
  通信処理時に使用するリアルスレッド固有メモリ
  「(h)　変数RTHD_COMMUSZの求め方」を参照して求めてください。
- *RTHD_MODASZ*
  adbmodareaコマンド実行時に使用するリアルスレッド固有メモリ
  「6.3.12　adbmodareaコマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbmodareaコマンド実行時）」を参照して求めてください。
- *RTHD_EXPSQLTRCSZ*
  SQLトレースバッファの拡張時に使用するリアルスレッド固有メモリ
  「(i)　変数RTHD_EXPSQLTRCSZの求め方」を参照して求めてください。
- *RTHD_AUDINFSZ*
  監査証跡機能の有効時に使用するリアルスレッド固有メモリ
  「(j)　変数RTHD_AUDINFSZの求め方」を参照して求めてください。
- *RTHD_COLUMNIZESZ*
  更新行のカラム化機能の有効時に使用するリアルスレッド固有メモリ
  「(k)　変数RTHD_COLUMNIZESZの求め方」を参照して求めてください。

(a)　変数RTHD_CNCTSZの求め方

変数*RTHD_CNCTSZ*は，データベース接続時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_CNCTSZ＝ TRANSNAP＋SQLTRC
```

変数の説明

TRANSNAP

「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(c)　変数 PROC_EXECSQLSZ の求め方」の変数 *TRANSNAP* を参照してください。

SQLTRC

SQL トレース機能の管理情報

65,536 キロバイトを代入してください。

(b)　変数 RTHD_DEFSQLSZ の求め方

変数 *RTHD_DEFSQLSZ* は，定義系 SQL 実行時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_DEFSQLSZ＝
      17＋SZ_ALTER_TABLE＋SZ_ALTER_VIEW＋SZ_CREATE_INDEX＋SZ_CREATE_TABLE
      ＋SZ_CREATE_VIEW＋SZ_DROP_INDEX＋SZ_DROP_TABLE
      ＋SZ_DROP_VIEW＋SZ_GRANT＋SZ_REVOKE
```

変数の説明

SZ_ALTER_TABLE

実表の定義を変更するときに，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_ALTER_TABLE＝ 377
```

SZ_ALTER_VIEW

ビュー表の定義を変更するときに，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_ALTER_VIEW＝ 266
```

SZ_CREATE_INDEX

インデクスを定義するときに，次に示す値を代入してください。

また，CREATE TABLE 文で一意性制約定義，またはチャンクアーカイブ指定を指定するときにも，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_CREATE_INDEX＝ 42
```

SZ_CREATE_TABLE

実表を定義するときに，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_CREATE_TABLE＝ 671
```

SZ_CREATE_VIEW

ビュー表を定義するときに，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_CREATE_VIEW= 217
```

SZ_DROP_INDEX

インデクスを削除するときに，次に示す値を代入してください。

また，DROP USER 文，DROP SCHEMA 文，またはDROP TABLE 文を実行した場合に，処理の延長でインデクスが削除されるときにも，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_DROP_INDEX= 39
```

SZ_DROP_TABLE

実表を削除するときに，次に示す値を代入してください。

また，DROP USER 文またはDROP SCHEMA 文を実行した場合に，処理の延長で実表が削除されるときにも，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_DROP_TABLE= 467
```

SZ_DROP_VIEW

ビュー表を削除するときに，次に示す値を代入してください。

また，DROP USER 文，DROP SCHEMA 文またはDROP TABLE 文を実行した場合に，処理の延長でビュー表が削除されるときにも，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_DROP_VIEW= 191
```

SZ_GRANT

権限を付与するときに，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_GRANT= 33,178
```

SZ_REVOKE

権限を取り消すときに，次に示す値を代入してください。

また，DROP USER 文，DROP SCHEMA 文またはDROP TABLE 文を実行した場合に，処理の延長で権限が取り消されるときにも，次に示す値を代入してください。

値（単位：キロバイト）

```
SZ_REVOKE= 33,410
```

(c)　変数 RTHD_EXESQLSZ の求め方

変数 *RTHD_EXESQLSZ* は，操作系 SQL 実行時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_EXESQLSZ＝
  MAX（max_sql(SQL_CNCT_SIZE)×max_sql_concurrent_exec_num , max_sql(PROC_THD_SIZE) , SQL_PRL_COMM_SIZE)
  ＋BUF_DLYSZ＋ARC_DIR_PATH※
```

注※

アーカイブマルチチャンク表に対してPURGE CHUNK 文およびTRUNCATE TABLE 文を実行する場合に加算してください。

変数の説明

max_sql（）：

実行する各 SQL について括弧内の変数の計算結果を求めて，それらの最大値を代入します。

max_sql_concurrent_exec_num：

1 トランザクション内で同時に実行する SQL の数の最大値

SQL_CNCT_SIZE：操作系 SQL の管理領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

SQL_CNCT_SIZE＝ *STATEMENT_SIZE*＋*CNCT_THD_SIZE*＋0.3

STATEMENT_SIZE：文ハンドル管理情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STATEMENT_SIZE＝
  ↑（2,848＋PREPARE_INFO_SIZE＋PARAM_INFO_SIZE＋APATHVIEW＋PARALLEL_INFO_SIZE※）÷1,024↑
```

注※

SQL パラレル実行機能を使用する場合に加算してください。

PREPARE_INFO_SIZE：SQL 文の前処理情報

次に示す計算式から求めてください。

計算式（単位：バイト）

PREPARE_INFO_SIZE＝ *col_num*×214＋*param_num*×80＋*idx_num*×512

col_num：SQL に指定した選択式の数

param_num：SQL に指定した？パラメタの数

idx_num：SQL で使用するインデクス数

PARAM_INFO_SIZE：パラメタ格納領域

次に示す計算式から求めてください。

計算式（単位：バイト）

$$PARAM_INFO_SIZE = \sum_{i=1}^{param_num} param_size(i)$$

param_num：SQL に指定した？パラメタの数

param_size（i）：各？パラメタの最大データ長

APATHVIEW：アクセスパスを表示する際に作成されるアクセスパス情報管理領域

「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(b)　変数 PROC_TOTALSQLSSZ の求め方」にある変数 *APATHVIEW* を参照してください。

PARALLEL_INFO_SIZE：SQL パラレル実行機能情報

SQL パラレル実行機能を使用する場合に，次に示す計算式から求めてください。

計算式（単位：バイト）

$$PARALLEL_INFO_SIZE = 2{,}212 + sql_length + 128 \times (MAX_NODE_NUM - 1) + 48 \times MAX_NODE_NUM \times (sql_rthd_num + 1)$$

sql_length：SQL 文長

sql_rthd_num：最大 SQL 処理リアルスレッド数（クライアント定義の`adb_sql_exe_max_rthd_num` オペランドの指定値）

MAX_NODE_NUM：SQL 文を処理するノード数の最大値

次に示す計算式から求めてください。

$$MAX_NODE_NUM = \mathrm{MIN}(db_area_file_num\ ,\ multi_node_num + worker_node_num)$$

db_area_file_num：処理対象表が格納されている DB エリアファイルの数

multi_node_num：サーバ定義`adb_sys_multi_node_info` オペランドに指定されたホスト名の数

worker_node_num：サーバ定義`adb_sys_multi_max_worker_nodes` オペランドの指定値

CNCT_THD_SIZE：SQL 実行管理領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

CNCT_THD_SIZE＝

$$\left\lceil \left\{ 3,920+752\times tbl_num+2,976\times sql_rthd_num+200\times setop_num+query_num \right.\right.$$

$$\times\left(1,632+688\times sql_rthd_num+16\times(sql_rthd_num\times sql_rthd_num)+160\times window_num\right)$$

$$+tbl_num\times sql_rthd_num\times(96+192\times sql_rthd_num)+112\times drvtbl_num$$

$$+(72+16\times sql_rthd_num)\times tblfunc_num$$

$$+RESULT_WORK_SIZE+OPE_WORK_SIZE+\sum(HASHGRP_SIZE)$$

$$+HASHTBL_CNCT_SIZE+(SCAN_WORK_SIZE+88)\times5\times tbl_num$$

$$+IN_PRD_SIZE+LIKE_PRD_SIZE+REG_SIZE+\sum(WRK_TABLE_OPE_SIZE)$$

$$+\sum(LMTWRK_SIZE)+COR_QUERY_SIZE+TXT_WORK_SIZE+REGEXP_WORK_SIZE$$

$$+MAX(CSVREAD_WORK_SIZE)$$

$$+384\times tvc_num+8\times query_num\times MIN(1,\ MAX(tvc_num,\ in_query_num))$$

$$+256\times in_query_num+RECURSIVE_QUERY_CNCT_SIZE+RANDOMCURSOR_WORK_SIZE$$

$$+RANDOM_NUM_WORK_SIZE+RNGIDX_INFO_SIZE+SORTGRP_CNCT_SIZE$$

$$\left.\left.+RT_AGGR_CNCT_WORK_SIZE+PRL_CNCT_SIZE\right\}\div1,024\right\rceil$$

重要

■ビュー表または問合せ名を指定する場合の留意事項

SQL 文中にビュー表または問合せ名を指定する場合は，それぞれに対応する内部的な導出表を適用したものとして見積もりを行ってください。ただし，再帰的問合せ内にある，対象となる再帰的問合せを参照している再帰的問合せ名については，その再帰的問合せの結果が格納される作業表と仮定して見積もりを行ってください。

（例）

・ビュー表を定義する SQL 文

```
CREATE VIEW "V1" AS SELECT * FROM "T1","T2" WHERE "T1"."C1"="T2"."C2"
```

・実行する SQL 文

```
SELECT * FROM "V1","T3"
          WHERE "V1"."C1"="T3"."C3"
```

・見積もり対象とする SQL 文

```
SELECT * FROM (SELECT * FROM "T1","T2" WHERE "T1"."C1"="T2"."C2"),"T3"
          WHERE "V1"."C1"="T3"."C3"
```

■アーカイブマルチチャンク表を指定する場合の留意事項

SQL 文中にアーカイブマルチチャンク表を指定する場合は，導出表に等価変換されたものとして見積もりを行ってください。アーカイブマルチチャンク表の等価変換については，マニュアル『HADB　AP 開発ガイド』の『アーカイブマルチチャンク表を検索する SQL 文の等価変換』を参照してください。

tbl_num：指定した表数

FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数を加算してください。

sql_rthd_num：

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値

setop_num：SQL 文中に指定した集合演算子の数

FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数を加算してください。

query_num：問合せの数

SQL 文中に指定したFROM 句の数から求めてください。

また，FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数×2 で求めた値を加算してください。

window_num：SQL 文中に指定したウィンドウ関数の数

1 つの問合せ指定中に，引数の異なるDISTINCT 集合関数を複数指定している場合は，その指定数－1 で求めた値も加算してください。

drvtbl_num：SQL 文中に指定した導出表の数

FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数を加算してください。

tblfunc_num：

表関数導出表の個数

Σ（*HASHGRP_SIZE*）：

SQL 文に指定した問合せ単位に *HASHGRP_SIZE* を計算した合計値（単位：バイト）

Σ（*WRK_TABLE_OPE_SIZE*）：

作成される作業表単位に *WRK_TABLE_OPE_SIZE* を計算した合計値（単位：バイト）

次に示すことについては，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』を参照してください。

- SQL 文の実行時に作業表が作成されるかどうか
- 作成される作業表の個数および構成列

Σ（*LMTWRK_SIZE*）：

作成される作業表単位に *LMTWRK_SIZE* を計算した合計値（単位：バイト）

次に示すことについては，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』を参照してください。

- LIMIT 句を指定した SQL 文が作業表を作成するかどうか
- 作成される作業表の個数

なお，LIMIT 句で集合演算の結果の取得行数の最大値を設定する場合は，集合演算を構成する問合せ指定ごとに，LIMIT 句の指定値が適用されます。

MAX（CSVREAD_WORK_SIZE）：

ADB_CSVREAD 関数を指定した表関数導出表ごとに変数 *CSVREAD_WORK_SIZE* をバイト単位で求めてください。その中で，最も大きい値を代入してください。

tvc_num：

SQL 文中に指定した表値構成子の数

in_query_num：

次に示す条件を満たす述語の数の合計値

- IN 述語の場合

  右側に表副問合せを指定し，左側の値式に指定した列が検索に使用する B-tree インデクスの先頭構成列

- 限定述語の場合

  ＝ANY または＝SOME で左側の値式に指定した列が検索に使用する B-tree インデクスの先頭構成列

RESULT_WORK_SIZE：SQL 実行結果格納領域

次に示す計算式から求めてください。データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

```
RESULT_WORK_SIZE＝
    ↑（40＋sum_col_size＋row_size
    ＋2×（sum_grp_size＋sum_window_spec_size）
    ＋3×（sum_set_func_size＋sum_dist_col_size＋sum_window_func_size）
    ＋8×（all_col_num＋row_num＋2×（grp_col_num＋window_spec_col_num）
    ＋3×（set_func_num＋dist_col_num＋window_func_num））
    ＋sum_inpred_size＋sum_reg_size
    ＋8×group_query_num＋12×list_num
    ＋16×join_num＋supquery_num
    ＋sum_setop_size×3＋full_join_drvc_size×3
    ＋（func_filepath_size＋4）×（csvread_num＋auditread_num））÷16↑×16
```

- *sum_col_size*

  指定したすべての列のデータ長の合計値

- *row_size*

  ROW でデータを取得する表の行長の合計値

- *sum_grp_size*

  グループ化列のデータ長の合計値

- *sum_window_spec_size*

  ウィンドウ指定に指定した列のデータ長の合計値

- *sum_set_func_size*
  集合関数の結果のデータ長の合計値
- *sum_dist_col_size*
  DISTINCT 集合関数の引数に指定した列のデータ長の合計値
- *sum_window_func_size*
  ウィンドウ関数の結果のデータ長の合計値
- *all_col_num*
  指定したすべての列の数
- *row_num*
  指定したROW の数
- *grp_col_num*
  GROUP BY 句に指定したグループ化列の数
- *window_spec_col_num*
  ウィンドウ指定に指定した列の数
- *set_func_num*
  集合関数の数
- *dist_col_num*
  DISTINCT 集合関数の引数に指定した列の数
- *window_func_num*
  ウィンドウ関数の数
- *sum_inpred_size*
  指定したすべてのIN 述語の結果作成用領域（29 ＋ 8×*IN 述語の IN の右側に指定した値式の数*）の合計値。ただし，IN 述語を指定していない場合は 0
- *sum_reg_size*
  指定したすべての日時情報取得関数およびユーザ情報取得関数のデータ長の合計値
- *group_query_num*
  GROUP BY 句を指定した問合せ指定の数
- *list_num*
  作業表の個数
  次の個数を合計したものを代入してください。
  ・ORDER BY 句を指定した場合は＋ 1
  ・GROUP BY 句を指定した問合せ指定の数
  ・SELECT DISTINCT を指定した問合せ指定の数
  ・DISTINCT 集合関数の指定数
  ・逆分布関数の指定数

・ウィンドウ関数の指定数
・FROM 句に複数の表参照を指定した問合せ指定の数
・導出表の数
・ビュー表の指定数
・WITH 句の数
・結合表の指定数（FULL OUTER JOIN の場合は 2 とカウントする）
・副問合せの指定数
・集合演算の指定数＋ 1
・LISTAGG 集合関数でALL（省略値）を指定した数
・LISTAGG 集合関数でDISTINCT を指定した数×2
・ARRAY_AGG 集合関数の指定数

- *join_num*

  結合表の数

  FULL OUTER JOIN の場合は，1 つ当たり 2 とカウントしてください。

- *supquery_num*

  副問合せの数

- *sum_setop_size*

  各集合演算について，集合演算の結果の選択式のデータ長を合計してください。そして，それらを合計した値を代入してください。

- *full_join_drvc_size*

  FULL OUTER JOIN を指定した各問合せ指定について，選択式のデータ長を求めてください。そして，それらを合計した値を代入してください。

- *func_filepath_size*

  ADB_CSVREAD 関数およびADB_AUDITREAD 関数に指定するファイルパス名の最大長

- *csvread_num*

  ADB_CSVREAD 関数の数

- *auditread_num*

  ADB_AUDITREAD 関数の数

OPE_WORK_SIZE：演算管理領域

次に示す計算式から求めてください。ただし，四則演算，論理演算，CASE 式，スカラ関数およびANY を指定した配列要素参照を指定していない場合は 0 になります。データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

```
OPE_WORK_SIZE＝
  {↓(571＋32×arith_num＋24×concat_num＋24×logical_num＋case_size＋sclfunc_size
      ＋80×array_any_num＋↑(array_any_id_num×array_any_max_cardinality÷4)↑) ÷512↓
  }×512＋40
```

- *arith_num*

  各値式中の四則演算子（＋，-，＊または／）の数の最大値

- *concat_num*

  各値式中の連結演算子（＋または||）およびスカラ関数CONCAT の数の合計値の中で最大の値

- *logical_num*

  探索条件中の論理演算子の数の最大値

- *case_size*

  CASE 式の結果長の合計値（単位：バイト）

- *sclfunc_size*

  スカラ関数の結果長の合計値（単位：バイト）

- *array_any_num*

  ANY を指定した配列要素参照の数

- *array_any_id_num*

  ANY を指定した配列要素参照で，異なる識別番号を指定した配列要素参照の数（同じ識別番号を指定した配列要素参照は全部で 1 と数える）と，識別番号を省略した配列要素参照の数の合計

- *array_any_max_cardinality*

  ANY を指定した配列要素参照に指定した配列データの最大要素数

  該当する配列データが複数ある場合は，その中で最も大きい最大要素数で計算してください。

HASHGRP_SIZE：ハッシュグループ化作業領域

次に示す計算式から求めてください。ただし，サーバ定義のadb_sql_exe_hashgrp_area_size オペランドに0 を指定している場合，または SQL にGROUP BY 句と集合関数が指定されていない場合は 0 になります。データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

```
HASHGRP_SIZE＝
  160＋24×（grp_col_num＋set_func_num）＋sum_grp_size
  ＋sum_set_func_size＋hashgrp_byte
  ＋↓（hashgrp_byte×4-MIN（8,192×128,
    ↓（hashgrp_byte×10）÷25↓））
  ÷（↑（16＋↑（sum_grp_size）÷4↑×4
  ＋sum_set_func_size）÷8↑×8）↓
```

- *grp_col_num*

  GROUP BY 句に指定したグループ化列の数

- *set_func_num*
  集合関数の数
- *sum_grp_size*
  グループ化列のデータ長の合計値
- *sum_set_func_size*
  集合関数の結果のデータ長の合計値
- *hashgrp_byte*
  ハッシュグループ化領域サイズの指定値
  次に示す計算式から求めてください。

  サーバ定義 adb_sql_exe_hashgrp_area_size オペランドの指定値×1,024

HASHTBL_CNCT_SIZE：ハッシュテーブルを使用した検索時の管理領域

SQL 文中に，次に示すどれかを含む場合は求めてください。すべて含まない場合は，0 になります。また，最大 SQL 処理リアルスレッド数，またはハッシュテーブル領域サイズが 0 の場合も，0 になります。

- 複数の表を＝で結合する問合せ指定
- 副問合せ
- GROUP BY 句
- DISTINCT 集合関数
- SELECT DISTINCT
- UNION またはUNION DISTINCT

計算式（単位：バイト）

$$\begin{aligned}&\mathrm{HASHTBL_CNCT_SIZE}=\\&\quad 256+160\times \mathrm{sql_rthd_num}+\left\{43,744+\left\lceil\frac{\mathrm{hashtbl_row_num}}{2}\right\rceil+64\times \mathrm{hashtbl_col_num}\right.\\&\quad \left.+(2,320+392\times \mathrm{bucket_num}+24\times \mathrm{sql_rthd_num})\times \mathrm{sql_rthd_num}\right\}\\&\quad \times \mathrm{hashtbl_num}+\mathrm{HASHFLT_CNCT_SIZE}+\mathrm{HASHRNG_CNCT_SIZE}+\mathrm{GRSETS_CNCT_SIZE}\\&\quad +\mathrm{HASHLOCAL_CNCT_SIZE}+\mathrm{DHJOIN_CNCT_SIZE}\end{aligned}$$

- *sql_rthd_num*
  サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値
- *hashtbl_row_num*
  ハッシュテーブルの最大行長
  「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。
- *hashtbl_col_num*
  SQL 文中の指定内容に従って求めた値のうち，最大値を代入

■複数の表を＝で結合する問合せ指定

ハッシュテーブル列数の最大のものになります。問合せ指定ごとのハッシュテーブルの列数を求める場合，＝で結合している各表に対して選択式および探索条件中に指定したすべての列の数を求めてください。そのあとで求めた表ごとの値のうち，最小値を除いた合計値を求めてください。

■副問合せ

副問合せごとに，次に示す値の合計を求めてください。その中で最大の値が副問合せの値になります。

・副問合せの選択式に指定した列数

・副問合せに指定した集合関数の数

・探索条件中の外への参照列を含む述語に指定した列の数

■GROUP BY 句，および DISTINCT を指定した集合関数

GROUP BY 句，およびDISTINCT 集合関数を指定した問合せごとに次に示す値の合計を求めてください。その中で最大の値がGROUP BY 句およびDISTINCT を指定した集合関数の値になります。

・GROUP BY 句に指定したグループ化列の数

・集合関数の数

■SELECT DISTINCT

SELECT DISTINCT を指定した問合せ指定ごとに，次に示す値を求めてください。その中で最大の値がSELECT DISTINCT の値になります。

・選択式に指定した列数

■UNION または UNION DISTINCT

UNION またはUNION DISTINCT を指定した問合せ式本体ごとに，次に示す値を求めてください。その中で最大の値が，UNION またはUNION DISTINCT の値になります。

・問合せ式本体の結果導出される表の列の列数

- *bucket_num*

  ハッシュテーブルを使用した検索時に作成される作業表の数

  「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

- *hashtbl_num*

  ハッシュテーブル数

  「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

HASHFLT_CNCT_SIZE：ハッシュフィルタ管理領域

次のどちらかの指定が SQL 文中にある場合に求めてください。どちらの指定もない場合は，0 になります。また，ハッシュフィルタ領域サイズ（サーバ定義のadb_sql_exe_hashflt_area_size オペランドの指定値）が 0 の場合も，0 になります。

- 複数の表を＝で結合する問合せ指定
- 副問合せ

計算式（単位：バイト）

HASHFLT_CNCT_SIZE ＝ 256＋256×*hashflt_num*

- *hashflt_num*

  SQL 文中のハッシュフィルタの総数

  次のすべての値の合計値を求めてください。

  ・複数の表を＝で結合する問合せ指定の，＝指定の結合条件の数

  ・外への参照列を含まない副問合せの数

  ・外への参照列を含む副問合せの，外への参照列を含む＝条件の数

HASHRNG_CNCT_SIZE：ハッシュ検索でのレンジインデクスの条件評価の管理領域

次のどちらかの指定が SQL 文中にある場合に求めてください。どちらの指定もない場合は，0 になります。

- 複数の表を＝で結合する問合せ指定
- 副問合せ

計算式（単位：バイト）

HASHRNG_CNCT_SIZE ＝ 64＋136×*hash_retrieval_rngidx_num*＋56×*hash_cond_rngidx_num*

hash_retrieval_rngidx_num

SQL 文中のハッシュ検索のうち，レンジインデクスを使用するハッシュ検索の総数

次のすべての値の合計値を求めてください。この合計数が 0 の場合，*HASHRNG_CNCT_SIZE* は 0 になります。

- 複数の表を＝で結合する問合せ指定のうち，レンジインデクスの構成列を＝に指定している各問合せ指定について，次の計算式で求めた値

  ＝で結合する表の総数 － ＝で結合する問合せ指定の総数

- レンジインデクスの構成列を左側に指定している限定述語の数
- レンジインデクスの構成列を左側に指定し，右側に表副問合せを指定している**IN** 述語の数
- 上記 2 つの副問合せ以外で，次のすべての条件を満たす副問合せの数

  ・探索条件中に＝で外への参照列を指定している

  ・指定した外への参照列がレンジインデクスの構成列である

hash_cond_rngidx_num

SQL 文中のハッシュ検索のうち，レンジインデクスを使用する条件の総数

次のすべての値の合計値を求めてください。

- 複数の表を＝で結合する問合せ指定のうち，レンジインデクスの構成列を指定している＝結合条件の数
- レンジインデクスの構成列を左側に指定している限定述語の数
- レンジインデクスの構成列を左側に指定し，右側に表副問合せを指定している**IN** 述語の数
- 上記 2 つの副問合せ以外で，次のすべての条件を満たす＝条件の数

・探索条件中に＝で外への参照列を指定している

・指定した外への参照列がレンジインデクスの構成列である

GRSETS_CNCT_SIZE：DISTINCT 集合関数の作業領域

SQL 文中に，引数の異なるDISTINCT 集合関数を複数指定した問合せ指定を含む場合に求めてください。引数の異なるDISTINCT 集合関数を複数指定した問合せ指定を含まない場合は，0 になります。

計算式（単位：バイト）

```
GRSETS_CNCT_SIZE ＝
  (1,640＋8,328×dset_func_num＋sql_rthd_num×（664＋72×dset_func_num）
   ）×dset_query_num
```

dset_func_num

引数の異なるDISTINCT 集合関数の数

sql_rthd_num

最大 SQL 処理リアルスレッド数（サーバ定義のadb_sql_exe_max_rthd_num オペランドの指定値）

dset_query_num

引数の異なるDISTINCT 集合関数を指定した問合せ指定の数

HASHLOCAL_CNCT_SIZE：部分集計管理領域

次に示すどれかの指定が SQL 文中にある場合に求めてください。指定がない場合は，0 になります。

- GROUP BY 句
- DISTINCT 集合関数
- SELECT DISTINCT
- UNION またはUNION DISTINCT

計算式（単位：バイト）

```
HASHLOCAL_CNCT_SIZE＝
          128＋8×hashtbl_num＋192×hashtbl_aggr_num＋64×hashtbl_aggr_col_num
```

hashtbl_num

ハッシュテーブル数

「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

hashtbl_aggr_num

SQL 処理リアルスレッドごとの部分集計を適用するハッシュ検索の総数

次に示す数の合計を求めてください。

- GROUP BY 句の指定数
- DISTINCT 集合関数を指定した問合せ指定の数
- SELECT DISTINCT の指定数
- UNION，またはUNION DISTINCT の指定数

hashtbl_aggr_col_num

SQL 処理リアルスレッドごとの部分集計を適用するハッシュ検索のうち，集計に関係する列の総数
次に示す数の合計を求めてください。

- GROUP BY 句の指定を含む場合
  グループ化列の数，および集合関数の数の合計数
- DISTINCT 集合関数を指定した問合せ指定を含む場合
  グループ化列の数，引数の異なるDISTINCT 集合関数の数，およびDISTINCT 集合関数ではない集合関数の数の合計数
- SELECT DISTINCT の指定を含む場合
  選択式に指定した列数
- UNION，またはUNION DISTINCT を指定した問合せ式本体を含む場合
  問合せ式本体によって導出される表の列の数

DHJOIN_CNCT_SIZE：ハッシュジョイン分割実行管理領域

SQL 文中に，次のすべての条件に該当する問合せ指定を含む場合に求めてください。条件に該当する問合せ指定を含まない場合は，0 になります。

1. コンマ結合，INNER JOIN，またはCROSS JOIN を使用して，複数の表を＝で結合する問合せ指定である
2. 1.の条件の表に，マルチチャンク表が 2 つ以上含まれている
3. 2.の条件のマルチチャンク表を結合する＝の 1 つに，レンジインデクスを定義しているマルチチャンク表の列が指定されている

計算式（単位：バイト）

DHJOIN_CNCT_SIZE＝
128＋*dhjoin_tbl_num*×64＋（*dhjoin_tbl_num*－*dhjoin_query_num*）×832

dhjoin_tbl_num

コンマ結合，INNER JOIN，またはCROSS JOIN を使用して，＝で結合する表の数

dhjoin_query_num

コンマ結合，INNER JOIN，またはCROSS JOIN を使用して，＝で結合する表が指定されている問合せ指定の数

IN_PRD_SIZE：IN 述語，限定述語実行管理領域

次に示す計算式から求めてください。ただし，IN 述語，限定述語を指定していない場合は 0 になります。なお，変数*in_pred_num* が 0 の場合も*IN_PRD_SIZE* は 0 になります。

計算式（単位：バイト）

IN_PRD_SIZE＝ 48＋*in_pred_num*×（96＋*sql_rthd_num*×48）

sql_rthd_num：サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値

in_pred_num：次に示す条件を満たす述語の数の合計値

- IN 述語の場合

  左側の値式に指定した列が検索に使用する B-tree インデクスの先頭構成列

- 限定述語の場合

  ＝ANY または＝SOME で左側の値式に指定した列が検索に使用する B-tree インデクスの先頭構成列

LIKE_PRD_SIZE：LIKE 述語実行管理領域

次に示す計算式から求めてください。ただし，LIKE 述語を指定していない場合は 0 になります。また，すべてのLIKE 述語の一致値に指定した列が検索に使用する B-tree インデクスの構成列に含まれない場合は 0 になります。データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

LIKE_PRD_SIZE＝ 16＋*like_param_num*×（64＋*like_param_size*×2）

- *like_param_num*

  パターン文字列に？パラメタを指定したLIKE 述語の数。ただし，一致値に指定した列が検索に使用する B-tree インデクスの構成列に含まれないLIKE 述語は除きます。

- *like_param_size*

  パターン文字列に指定した？パラメタが仮定するデータの長さの最大値。ただし，一致値に指定した列が検索に使用する B-tree インデクスの構成列に含まれないLIKE 述語の分は除きます。

REG_SIZE：日時情報取得関数およびユーザ情報取得関数の管理領域

次に示す計算式から求めてください。ただし，日時情報取得関数およびユーザ情報取得関数を指定していない場合は 0 になります。データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

REG_SIZE＝ 24×*date_spec*＋32×*time_spec*＋32×*timestmp_spec*＋120×*user_spec*＋24

- *date_spec*

  CURRENT_DATE を指定した場合は 1，指定していない場合は 0

- *time_spec*

  CURRENT_TIME を指定した場合は 1，指定していない場合は 0

- *timestmp_spec*

  CURRENT_TIMESTAMP を指定した場合は 1，指定していない場合は 0

- *user_spec*

  CURRENT_USER を指定した場合は 1，指定していない場合は 0

WRK_TABLE_OPE_SIZE

作業表の処理領域

次に示す計算式から求めてください。ただし，作業表が作成されない場合は 0 になります。

計算式（単位：バイト）

$$WRK_TABLE_OPE_SIZE = 16 \times arrray_col_num + 176 \times \left(\frac{clt_wrk_page_num}{2} + 1 \right)$$

- *arrray_col_num*

  配列型の列の列数

  作業表の構成列に配列型の列が存在しない場合は 0 を代入してください。

- *clt_wrk_page_num*

  クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値（ただし，adbexport コマンド実行時はエクスポートオプションadb_export_wrktbl_blk_num の指定値）

LMTWRK_SIZE

作業表の管理領域

ただし，LIMIT 句を指定していない場合，または作業表が作成されない場合は 0 になります。

計算式（単位：バイト）

$$LMTWRK_SIZE = \uparrow (96 \times \uparrow (offset_num + limit_num) \div \downarrow (wrk_page_size - 64) \div ROWSZ \downarrow \uparrow + (4 + ROWSZ) \times (offset_num + limit_num)) \div 1{,}024 \uparrow$$

- *offset_num*

  LIMIT 句のオフセット行数の指定値

- *limit_num*

  LIMIT 句のリミット行数の指定値

- *wrk_page_size*

  作業表用 DB エリアのページサイズ（単位：バイト）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」を参照してください。

- *ROWSZ*

  作業表の行長（単位：バイト）

  作業表の行長については，「5.9.2　作業表を格納するために必要な基本行用ページ数の求め方」の変数*ROWSZ* の計算式を基に求めてください。ただし，変数*col_size*（*i*）を求める場合で，かつデータ型が可変長の列のときは，*d*（実際のデータ長）に*n*（定義長）を代入してください。

COR_QUERY_SIZE

外への参照列を含む副問合せの作業領域

16,777,216 バイトを代入してください。ただし，外への参照列を含む副問合せを指定していない場合は 0 を代入してください。

TXT_WORK_SIZE

テキストインデクスの作業領域

次に示す計算式から求めてください。ただし，テキストインデクスを定義していない場合，またはテキストインデクスを使用しない場合は 0 を代入してください。

計算式（単位：バイト）

```
TXT_WORK_SIZE＝ 28×scan_str_num
```

- *scan_str_num*

  テキストインデクスを使用するLIKE 述語のパターン文字列に指定した文字列の文字数

CSVREAD_WORK_SIZE

ADB_CSVREAD 関数の検索作業領域

次に示す計算式から求めてください。ただし，ADB_CSVREAD 関数を使用しない場合は0 を代入してください。

計算式（単位：バイト）

$$\text{CSVREAD_WORK_SIZE} = 269,280 + \text{decomp_areasize} + 32 \times \text{col_num} + \sum_{i=1}^{\text{col_num}} \text{temp_coldata_areasize}(i)$$

decomp_areasize

伸長処理領域サイズ（単位：バイト）

入力データファイルの伸長処理に使用する領域サイズです。ADB_CSVREAD 関数の圧縮形式オプションCOMPRESSION_FORMAT の指定値によって，代入する値が異なります。

伸長処理領域サイズについては，次に示す表から求めてください。

表 6-5　伸長処理領域サイズの一覧

項番	圧縮形式オプション COMPRESSION_FORMAT の指定値	伸長処理領域サイズ（単位：バイト）
1	GZIP	112
2	NONE	40

col_num

入力データファイル中から取り出すフィールドデータの数

temp_coldata_areasize（i）

一時格納領域サイズ（単位：バイト）

入力データファイル中から取り出したｉ番目のフィールドデータを一時的に格納する領域サイズです。*i* は表関数導出表の列名リストの順番に対応します。

一時格納領域サイズについては，次に示す表から求めてください。

表 6-6　各データ型の一時格納領域サイズの一覧

項番	分類	データ型	一時格納領域サイズ（単位：バイト）
1	数データ	INTEGER	24

項番	分類	データ型		一時格納領域サイズ（単位：バイト）
2		SMALLINT		16
3		DECIMAL		48
4		NUMERIC		
5		DOUBLE PRECISION		512
6		FLOAT		
7	文字データ	CHAR		((2 +*定義長*×2 + 7) ÷8) ×8
8		VARCHAR		((2 +*定義長*×2 + 7) ÷8) ×8
9	日時データ	DATE		16
10		TIME		24
11		TIMESTAMP		40
12	バイナリデータ	BINARY	入力データファイル中の記述形式が 16 進の場合	((2 +*定義長*×2 + 7) ÷8) ×8
13			入力データファイル中の記述形式が 2 進の場合	((2 +*定義長*×8 + 7) ÷8) ×8
14		VARBINARY	入力データファイル中の記述形式が 16 進の場合	((2 +*定義長*×2 + 7) ÷8) ×8
15			入力データファイル中の記述形式が 2 進の場合	((2 +*定義長*×8 + 7) ÷8) ×8

REGEXP_WORK_SIZE

正規表現の評価用作業領域（単位：バイト）

LIKE_REGEX 述語を指定していない場合は，0 を代入してください。

変数 *REGEXP_WORK_SIZE* は，LIKE_REGEX 述語のパターン文字列に指定された正規表現内に，0 文字の繰り返しが存在するかどうかで計算式が異なります。

0 文字の繰り返しとは，メタ文字や繰り返し因子の組み合わせによって，該当する文字が 0 文字になる可能性がある記述を指します。例えば，「*」，「?」，「{0,}」，および「{0,*x*}」（*x*：1 以上の整数）をネストした記述です。

0 文字の繰り返しになる記述例

```
(a*)+, (c{0,2})*, (e?){3}
```

変数 *REGEXP_WORK_SIZE* を求める計算式を，それぞれ次に示します。

計算式（正規表現内に 0 文字の繰り返しが存在しない場合）

```
REGEXP_WORK_SIZE＝ (push_instr_num＋scan_str_num＋2) ×14
```

計算式（正規表現内に 0 文字の繰り返しが存在する場合）

```
REGEXP_WORK_SIZE＝ （push_instr_num×ostrrep_instr_num＋1）×（scan_str_num＋1）×14
```

重要

正規表現内に 0 文字の繰り返しが存在する場合，大量のメモリを使用するおそれがあります。ただし，0 文字となる可能性がある場合でも，量指定子や繰り返し因子をネストして記述しなければ，大量のメモリを使用しません。

push_instr_num

次に示す計算式で求めてください。

```
push_instr_num＝ A＋B＋C

  変数の説明
    A：パターン文字列に指定された正規表現内の「*」，「+」，「?」および「|」の合計値
    B：正規文字集合の数の合計値
    C：繰り返し因子に指定された繰り返しの数
```

変数 C（繰り返し因子に指定された繰り返しの数）の求め方を，次の表に示します。

表 6-7　繰り返し因子に指定された繰り返しの数の求め方

項番	繰り返し因子に指定された内容	変数 C に代入する値	例
1	上限値と下限値の両方が指定されている	「上限値－下限値」で求めた値	「{10,20}」と指定されている場合は，10 を代入します。
2	次に示す条件を満たしている • 下限値だけが指定されている • 下限値の後ろにコンマが存在しない	下限値	「{10}」と指定されている場合は，10 を代入します。
3	次に示す条件を満たしている • 下限値だけが指定されている • 下限値の後ろにコンマが存在する	「256－下限値」で求めた値	「{10,}」と指定されている場合は，246 を代入します。

scan_str_num

パターン文字列に指定した文字列の文字数

ostrrep_instr_num

0 文字の繰り返しの合計値

正規表現の記述例と，変数 *REGEXP_WORK_SIZE* の計算式の例を，次に示します。

正規表現の記述例 1

```
KFAA11[0-9]{3}-E
```

計算式の例（単位：バイト）

```
REGEXP_WORK_SIZE＝ （4＋16＋2)×14 ＝ 308

  説明
    ・push_instr_numに，4を代入する。
    ・scan_str_numに，16を代入する。
```

正規表現の記述例 2

```
KFAA6000[2-9]-E.*uid=12345.*
```

計算式の例（単位：バイト）

```
REGEXP_WORK_SIZE＝ （4＋28＋2）×14 ＝ 476

  説明
    ・push_instr_numに，3を代入する。
    ・scan_str_numに，28を代入する。
```

RECURSIVE_QUERY_CNCT_SIZE

再帰的問合せ管理領域（単位：バイト）

SQL 文中に再帰的問合せを含まない場合は，0 としてください。

```
RECURSIVE_QUERY_CNCT_SIZE ＝
  (512＋72×sql_rthd_num)×recursive_query_num
  ＋(recurmb_query_num＋recurmb_setop_num＋recurmb_list_num＋recurmb_subquery_num)
  ×32＋(648＋16×sql_rthd_num)×recurmb_hashtbl_num
```

sql_rthd_num

最大 SQL 処理リアルスレッド数（サーバ定義のadb_sql_exe_max_rthd_num オペランドの指定値）

recursive_query_num

SQL 文中に指定した再帰的問合せの数

recurmb_query_num

再帰的メンバ内に指定したFROM 句の数

recurmb_setop_num

再帰的メンバ内に指定した集合演算子の数

recurmb_list_num

再帰的メンバ内に指定した次の数の合計値

- ウィンドウ関数の指定数
- FROM 句に複数の表参照を指定した問合せの指定数
- 結合表の指定数
- 副問合せの指定数

recurmb_subquery_num

再帰的メンバ内に指定した副問合せの数

recurmb_hashtbl_num

再帰的メンバ内に指定したハッシュテーブル数

詳細については，マニュアル『HADB　AP 開発ガイド』の『SQL を実行した場合に作成される作業表について』を参照してください。

RANDOMCURSOR_WORK_SIZE

スカラ関数RANDOMCURSOR 用の作業領域

SQL 文中にスカラ関数RANDOMCURSOR を含む場合に，次に示す値を代入してください。ただし，SQL 文中にスカラ関数RANDOMCURSOR を含まない場合は 0 を代入してください。

値（単位：バイト）

RANDOMCURSOR_WORK_SIZE＝ 3,176

RANDOM_NUM_WORK_SIZE

擬似乱数生成用の作業領域（単位：バイト）

```
RANDOM_NUM_WORK_SIZE ＝ RANDOM_NUM_UNIFORM_WORK_SIZE
                     ＋ RANDOM_NUM_NORMAL_WORK_SIZE
                     ＋ RANDOMROW_WORK_SIZE
```

RANDOM_NUM_UNIFORM_WORK_SIZE

一様分布に従う擬似乱数生成用の作業領域

SQL 文中にスカラ関数RANDOM を含む場合に，次に示す値を代入してください。ただし，SQL 文中にスカラ関数RANDOM を含まない場合は 0 を代入してください。

値（単位：バイト）

RANDOM_NUM_UNIFORM_WORK_SIZE＝ 3,176

RANDOM_NUM_NORMAL_WORK_SIZE

正規分布に従う擬似乱数生成用の作業領域

SQL 文中にスカラ関数RANDOM_NORMAL を含む場合に，次に示す値を代入してください。ただし，SQL 文中にスカラ関数RANDOM_NORMAL を含まない場合は 0 を代入してください。

値（単位：バイト）

RANDOM_NUM_NORMAL_WORK_SIZE＝ 6,288

RANDOMROW_WORK_SIZE

スカラ関数RANDOMROW 用の作業領域

SQL 文中にスカラ関数RANDOMROW を含む場合に，次に示す値を代入してください。ただし，SQL 文中にスカラ関数RANDOMROW を含まない場合は 0 を代入してください。

値（単位：バイト）

RANDOMROW_WORK_SIZE＝ 3,176

SORTGRP_CNCT_SIZE

ソートグループ化の作業領域

次のどちらかの集合関数を SQL 文中に指定する場合に求めてください。どちらの指定もない場合は，0 を代入してください。

- LISTAGG 集合関数
- ARRAY_AGG 集合関数

計算式（単位：バイト）

SORTGRP_CNCT_SIZE＝
352×Σ（*sortgrp_wrk_num*）×*sql_rthd_num*
＋*LISTAGG_CNCT_SIZE*＋*ARRAY_AGG_CNCT_SIZE*

Σ（*sortgrp_wrk_num*）

SQL 文中にLISTAGG 集合関数，またはARRAY_AGG 集合関数を指定した問合せ指定ごとに変数 *sortgrp_wrk_num* を求めてください。そして，それぞれ求めた値を合計してください。

sortgrp_wrk_num

ソートグループ化の作業領域の個数

次の数を合計したものを代入してください。

- ALL（省略値）を指定したLISTAGG 集合関数の指定数
- DISTINCT を指定したLISTAGG 集合関数の指定数×2
- ARRAY_AGG 集合関数の指定数
- DISTINCT 集合関数の指定数
- 逆分布関数の指定数

sql_rthd_num

最大 SQL 処理リアルスレッド数

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値です。指定値が 0 の場合は 1 を代入してください。

LISTAGG_CNCT_SIZE

LISTAGG 集合関数の作業領域

SQL 文中に，LISTAGG 集合関数を含む場合に求めてください。LISTAGG 集合関数を含まない場合は，0 を代入してください。

計算式（単位：バイト）

LISTAGG_CNCT_SIZE＝128×*listagg_num*×*sql_rthd_num*

listagg_num

SQL 文中に指定したLISTAGG 集合関数の数

sql_rthd_num

最大 SQL 処理リアルスレッド数

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値です。指定値が 0 の場合は 1 を代入してください。

ARRAY_AGG_CNCT_SIZE

ARRAY_AGG 集合関数の作業領域

ARRAY_AGG 集合関数を SQL 文中に指定する場合に求めてください。ARRAY_AGG 集合関数を指定しない場合は，0 を代入してください。

計算式（単位：バイト）

```
ARRAY_AGG_CNCT_SIZE＝
    Σ {array_agg_num×（（array_agg_num＋1）×sql_rthd_num×104＋RESULT_WORK_SIZE）}
```

ARRAY_AGG_CNCT_SIZE を算出するには，ARRAY_AGG 集合関数を指定した問合せ指定ごとに，次の値を求めてください。そして，それぞれ求めた値を合計してください。

```
array_agg_num×（（array_agg_num＋1）×sql_rthd_num×104＋RESULT_WORK_SIZE）
```

array_agg_num

問合せ指定に指定したARRAY_AGG 集合関数の数

sql_rthd_num

最大 SQL 処理リアルスレッド数

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値です。指定値が 0 の場合は 1 を代入してください。

RT_AGGR_CNCT_WORK_SIZE

リアルスレッド単位の集計作業領域

SQL 文中に，次に示すどれかの要素を含む場合に求めてください。含まない場合は，0 を代入してください。

- LISTAGG 集合関数
- ARRAY_AGG 集合関数
- 逆分布関数
- ウィンドウ関数

データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

```
RT_AGGR_CNCT_WORK_SIZE＝
        (400＋328×sql_rthd_num）×rt_aggr_num
        ＋Σ（sum_grp_size）×sql_rthd_num
```

sql_rthd_num

最大 SQL 処理リアルスレッド数

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値です。指定値が 0 の場合は 1 を代入してください。

rt_aggr_num

該当する要素の数

Σ（*sum_grp_size*）

SQL 文中に該当する要素を指定した問合せ指定ごとに変数*sum_grp_size* を求めてください。そして，それぞれ求めた値を合計してください。

sum_grp_size

SQL 文中に含む要素によって，値が異なります。どちらの要素も含む場合は，それぞれ求めた値を合計してください。

- LISTAGG 集合関数，ARRAY_AGG 集合関数または逆分布関数の場合

  グループ化列のデータ長の合計

- ウィンドウ関数の場合

  ウィンドウ分割句に指定された列のデータ長の合計

PRL_CNCT_SIZE

SQL パラレル実行機能用管理領域

SQL パラレル実行機能が適用される場合に求めてください。SQL パラレル実行機能が適用されない場合は，0 を代入してください。

計算式（単位：バイト）

```
PRL_CNCT_SIZE＝
    1,438＋8×(1＋order_by_num＋group_by_num×7)
    ＋128×prl_scan_num＋202×node_num
    ＋PRL_NDMRG_CNCT_SIZE＋PRL_HASH_CNCT_SIZE
```

order_by_num

ORDER BY 句を指定する場合は 1 を代入してください。ORDER BY 句を指定しない場合は 0 を代入してください。

group_by_num

GROUP BY 句の指定数を代入してください。GROUP BY 句を指定しない場合は 0 を代入してください。

prl_scan_num

1 を代入してください。

node_num

マルチノード構成のノード数

PRL_NDMRG_CNCT_SIZE

ノード間マージの管理領域

SQL パラレル実行機能が適用される場合に求めてください。SQL パラレル実行機能が適用されない場合は，0 を代入してください。

計算式（単位：バイト）

```
PRL_NDMRG_CNCT_SIZE＝
    （128＋PRL_MAIN_NDMRG_CNCT_SIZE＋PRL_NDMRG_SORT_SIZE）×prl_scan_num
```

prl_scan_num

1 を代入してください。

PRL_MAIN_NDMRG_CNCT_SIZE

SQL メインノードのノード間マージの管理領域

SQL パラレル実行機能が適用される場合に求めてください。SQL パラレル実行機能が適用されない場合は，0 を代入してください。

計算式（単位：バイト）

```
PRL_MAIN_NDMRG_CNCT_SIZE＝
    8,448＋（8,754＋RESULT_WORK_SIZE＋MAX（131,072 , max_def_row_size）×2）
          ×node_num
```

max_def_row_size

選択式の結果のデータ長の合計値を問合せ指定ごとに求めてください。そして，求めた値の最大値を代入してください。

node_num

マルチノード構成のノード数

PRL_NDMRG_SORT_SIZE

ノード間マージの結果をソートする際の管理領域

SQL パラレル実行機能が適用され，かつ SQL 文中にORDER BY 句を含む場合に求めてください。SQL パラレル実行機能が適用されない場合，またはORDER BY 句を含まない場合は，0 を代入してください。

計算式（単位：バイト）

```
PRL_NDMRG_SORT_SIZE＝64＋16×node_num
```

node_num

マルチノード構成のノード数

PRL_HASH_CNCT_SIZE

ノード間共有するハッシュテーブルを使用した検索時の管理領域

SQL パラレル実行機能が適用され，かつ SQL 文中にGROUP BY 句を含む場合に求めてください。SQL パラレル実行機能が適用されない場合，またはGROUP BY 句を含まない場合は，0 を代入してください。また，最大 SQL 処理リアルスレッド数，またはハッシュテーブル領域サイズが 0 の場合も，0 を代入してください。

計算式（単位：バイト）

```
PRL_HASH_CNCT_SIZE＝
    4,175＋8×hashtbl_num＋（12,757＋（8,875＋72×sql_rthd_num）×node_num）
          ×prl_hashtbl_num
```

hashtbl_num

ハッシュテーブル数

「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

sql_rthd_num

最大 SQL 処理リアルスレッド数（サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値）

node_num

マルチノード構成のノード数

prl_hashtbl_num

GROUP BY 句の数

PROC_THD_SIZE：SQL 処理リアルスレッド管理領域

次に示す計算式から求めてください。なお，変数*RESULT_WORK_SIZE*，変数*OPE_WORK_SIZE*，変数*HASHGRP_SIZE*，変数*COR_QUERY_SIZE*，変数*REGEXP_WORK_SIZE*，および変数*RANDOM_NUM_WORK_SIZE* については，上記にある各計算式から求めた値を代入してください。

計算式（単位：キロバイト）

```
PROC_THD_SIZE＝
↑{3,584＋tbl_num
×(400,400＋672×uthd_num＋256×tbl_num＋idx_col_num×(26,624＋2,048×idx_size))
＋40×sql_rthd_num＋544×uthd_num＋RESULT_WORK_SIZE
×(1,024×tbl_num＋(tbl_num＋2×setop_num)×16×MAX(128, uthd_num)
×(1＋cor_query_num×uthd_num)＋4,096×(drvtbl_num＋in_query_num)＋256×tblfunc_num＋11)
＋OPE_WORK_SIZE×(8×uthd_num)
＋Σ_[i=cor_query](HASHGRP_SIZE(i))×uthd_num×sql_rthd_num
＋Σ_[j=not_cor_query](HASHGRP_SIZE(j))
＋HASHTBL_PROC_SIZE＋(SCAN_WORK_SIZE＋88)×uthd_num×5×tbl_num
＋(3,232＋sql_rthd_num×(2,208＋tbl_num×(96＋192×sql_rthd_num))
＋max_cor_query(inprd_cor_query_num)×(96＋sql_rthd_num×48)
＋200×max_cor_query(setop_cor_query_num)＋160×max_cor_query(window_cor_query_num)
＋16×max_cor_query(limit_cor_query_num)＋208×sql_rthd_num)
×cor_query_num＋(786,432＋163,840×sql_rthd_num)×(drvtbl_num＋in_query_num))
＋∑(drvtbl_row_len×MIN(drvtbl_row_num, 1,044,480))
＋∑(in_query_drvc_len×MIN(in_query_drvc_num, 1,044,480))
＋120×8×uthd_num×leftouter_num ＋∑(WRK_TABLE_OPE_SIZE)
＋∑(LMTWRK_SIZE) ＋COR_QUERY_SIZE
＋TXT_WORK_SIZE＋(536＋sql_rthd_num×40)×tblfunc_num＋REGEXP_WORK_SIZE×uthd_num
＋∑(tblfunc_filename_len＋5)
＋∑((tblfunc_row_len＋4)×MIN(tblfunc_row_num , 65,025)) ＋∑(CSVREAD_WORK_SIZE)
＋∑(AUDITREAD_WORK_SIZE)＋SCAN_INFO_SIZE
＋(8×query_num×MIN(1, MAX(tvc_num, in_query_num))＋320×tvc_num＋256×in_query_num)
×uthd_num＋RANDOM_NUM_WORK_SIZE＋SORTGRP_PROC_SIZE＋RT_AGGR_PROC_WORK_SIZE
＋PRL_PROC_SIZE}÷1,024↑
```

重要

- ビュー表または問合せ名を指定する場合の留意事項

  SQL 文中にビュー表または問合せ名を指定する場合は，それぞれに対応する内部的な導出表を適用したものとして見積もりを行ってください。ただし，再帰的問合せ内にある，対象となる再帰的問合せを参照している再帰的問合せ名については，その再帰的問合せの結果が格納される作業表と仮定して見積もりを行ってください。

- アーカイブマルチチャンク表を指定する場合の留意事項

  SQL 文中にアーカイブマルチチャンク表を指定する場合は，導出表に等価変換されたものとして見積もりを行ってください。アーカイブマルチチャンク表の等価変換については，マニュアル『HADB　AP 開発ガイド』の『アーカイブマルチチャンク表を検索する SQL 文の等価変換』を参照してください。

tbl_num：指定した表数

FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数を加算してください。

sql_rthd_num：

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値

idx_col_num：

検索に使用する B-tree インデクスおよびテキストインデクスの構成列数

idx_size：

使用する各 B-tree インデクスおよびテキストインデクスの構成列の定義長の合計値

uthd_num：

サーバ定義adb_sys_uthd_num オペランドの指定値

Σ_[i =cor_query](HASHGRP_SIZE(i))：

SQL 文中の外への参照列を含む副問合せごとに *HASHGRP_SIZE* を計算した合計値（単位：バイト）

Σ_[j =not_cor_query](HASHGRP_SIZE(j))：

SQL 文中の外への参照列を含む副問合せではない問合せ指定単位に *HASHGRP_SIZE* を計算した合計値（単位：バイト）

HASHTBL_PROC_SIZE：ハッシュテーブルを使用した検索時のリアルスレッド管理領域

SQL 文中に，次に示すどれかを含む場合は求めてください。すべて含まない場合は，0 になります。また，最大 SQL 処理リアルスレッド数，またはハッシュテーブル領域サイズが 0 の場合も，0 となります。

- 複数の表を＝で結合する問合せ指定
- 副問合せ
- GROUP BY 句
- DISTINCT 集合関数
- SELECT DISTINCT

- UNION またはUNION DISTINCT

計算式（単位：バイト）

$$HASHTBL_PROC_SIZE = 4,160 + \left\lceil \frac{hashtbl_byte}{sql_rthd_num} \right\rceil + 128 \times uthd_num + (RESULT_WORK_SIZE + 144) \times clt_wrk_page_num + (3 \times RESULT_WORK_SIZE + 17,720 + 32 \times sql_rthd_num) \times hashtbl_num + HASHFLT_PROC_SIZE + HASHRNG_PROC_SIZE + HASHLOCAL_PROC_SIZE + DHJOIN_PROC_SIZE$$

- *hashtbl_byte*

  次に示す計算式で求めた値

  サーバ定義adb_sql_exe_hashtbl_area_sizeオペランドの指定値×1,024×1,024

- *sql_rthd_num*

  クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値

- *uthd_num*

  サーバ定義adb_sys_uthd_num オペランドの指定値

- *clt_wrk_page_num*

  クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値（ただし，adbexport コマンド実行時はエクスポートオプションadb_export_wrktbl_blk_num の指定値）

- *hashtbl_num*

  ハッシュテーブル数

  「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

HASHFLT_PROC_SIZE：SQL 処理リアルスレッド用ハッシュフィルタ管理領域

次のどちらかの指定が SQL 文中にある場合に求めてください。どちらの指定もない場合は，0 になります。また，ハッシュフィルタ領域サイズ（サーバ定義adb_sql_exe_hashflt_area_size オペランドの指定値）が 0 の場合も，0 になります。

- 複数の表を＝で結合する問合せ指定
- 副問合せ

計算式（単位：バイト）

$$HASHFLT_PROC_SIZE = \left\lceil hashflt_byte \div sql_rthd_num \right\rceil + 136 + 192 \times hashflt_num$$

- *hashflt_byte*

  次に示す計算式で求めた値

  サーバ定義adb_sql_exe_hashflt_area_sizeオペランドの指定値×1,024×1,024

- *sql_rthd_num*

  クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値

- *hashflt_num*

  SQL 文中のハッシュフィルタの総数

  次のすべての値の合計値を求めてください。

  ・複数の表を＝で結合する問合せ指定の，＝指定の結合条件の数

  ・外への参照列を含まない副問合せの数

  ・外への参照列を含む副問合せの，外への参照列を含む＝条件の数

HASHRNG_PROC_SIZE：ハッシュ検索での SQL 処理リアルスレッド用のレンジインデクスの条件評価の管理領域

次のどちらかの指定が SQL 文中にある場合に求めてください。どちらの指定もない場合は，0 になります。

- 複数の表を＝で結合する問合せ指定
- 副問合せ

計算式（単位：バイト）

HASHRNG_PROC_SIZE ＝ 64＋72×*hash_retrieval_rngidx_num*＋64×*hash_cond_rngidx_num*
＋Σ（↑（*hash_cond_rngidx_len*）÷16↑×32）

hash_retrieval_rngidx_num

SQL 文中のハッシュ検索のうち，レンジインデクスを使用するハッシュ検索の総数

次のすべての値の合計値を求めてください。この合計数が 0 の場合，*HASHRNG_PROC_SIZE* は 0 になります。

- 複数の表を＝で結合する問合せ指定のうち，レンジインデクスの構成列を＝に指定している各問合せ指定について，次の計算式で求めた値

  ＝で結合する表の総数 － ＝で結合する問合せ指定の総数

- レンジインデクスの構成列を左側に指定している限定述語の数
- レンジインデクスの構成列を左側に指定し，右側に表副問合せを指定している IN 述語の数
- 上記 2 つの副問合せ以外で，次のすべての条件を満たす副問合せの数

  ・探索条件中に＝で外への参照列を指定している

  ・指定した外への参照列がレンジインデクスの構成列である

hash_cond_rngidx_num

SQL 文中のハッシュ検索のうち，レンジインデクスを使用する条件の総数

次のすべての値の合計値を求めてください。

- 複数の表を＝で結合する問合せ指定のうち，レンジインデクスの構成列を指定している＝結合条件の数
- レンジインデクスの構成列を左側に指定している限定述語の数
- レンジインデクスの構成列を左側に指定し，右側に表副問合せを指定している IN 述語の数
- 上記 2 つの副問合せ以外で，次のすべての条件を満たす＝条件の数

・探索条件中に＝で外への参照列を指定している

・指定した外への参照列がレンジインデクスの構成列である

Σ（↑（*hash_cond_rngidx_len*）÷16↑×32）

SQL 文中のハッシュ検索のうち，レンジインデクスを使用する条件ごとに値を算出し，その値を合計してください。

hash_cond_rngidx_len

ハッシュ検索でレンジインデクスを使用する条件ごとに次の値を求めてください。

- 複数の表を＝で結合する問合せ指定のうち，レンジインデクスの構成列を指定している＝結合条件について，その＝結合条件の左右に指定した値式の結果のデータ長の，どちらか大きい方の値
- レンジインデクスの構成列を左側に指定している限定述語で，次のどちらか大きい方の値

  ・右側に指定した表副問合せの選択式の結果のデータ長の合計値

  ・左側に指定したレンジインデクスの構成列の列長
- レンジインデクスの構成列を左側に指定し，右側に表副問合せを指定しているIN 述語で，次のどちらか大きい方の値

  ・右側に指定した表副問合せの選択式の結果のデータ長の合計値

  ・左側に指定したレンジインデクスの構成列の列長
- 副問合せの探索条件中に指定した＝条件の左右の値式の結果のデータ長のうち，どちらか大きい方の値（上記 2 つの副問合せに該当しない副問合せで，次の 2 つの条件を満たす副問合せであること）

  ・副問合せの探索条件中に，＝で外への参照列を指定している

  ・指定した外への参照列がレンジインデクスの構成列である

HASHLOCAL_PROC_SIZE：SQL 処理リアルスレッド用の部分集計管理領域

次に示すどれかの指定が SQL 文中にある場合に求めてください。指定がない場合は，0 になります。

- GROUP BY 句
- DISTINCT 集合関数
- SELECT DISTINCT
- UNION またはUNION DISTINCT

計算式（単位：バイト）

```
HASHLOCAL_PROC_SIZE＝
        64＋8×hashtbl_num＋2,097,280×hashtbl_aggr_num
```

hashtbl_num

ハッシュテーブル数

「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

hashtbl_aggr_num

SQL 処理リアルスレッドごとの部分集計を適用するハッシュ検索の総数

次に示す数の合計を求めてください。

- GROUP BY 句の指定数
- DISTINCT 集合関数を指定した問合せ指定の数
- SELECT DISTINCT の指定数
- UNION，またはUNION DISTINCT の指定数

DHJOIN_PROC_SIZE：ハッシュジョイン分割実行のリアルスレッド管理領域

SQL 文中に，次のすべての条件に該当する問合せ指定を含む場合に求めてください。条件に該当する問合せ指定を含まない場合は，0 になります。

1. コンマ結合，INNER JOIN，またはCROSS JOIN を使用して，複数の表を＝で結合する問合せ指定である
2. 1.の条件の表に，マルチチャンク表が 2 つ以上含まれている
3. 2.の条件のマルチチャンク表を結合する＝の 1 つに，レンジインデクスを定義しているマルチチャンク表の列が指定されている

計算式（単位：バイト）

```
DHJOIN_PROC_SIZE＝
        64＋320×（dhjoin_tbl_num－dhjoin_query_num）
```

dhjoin_tbl_num

コンマ結合，INNER JOIN，またはCROSS JOIN を使用して，＝で結合する表の数

dhjoin_query_num

コンマ結合，INNER JOIN，またはCROSS JOIN を使用して，＝で結合する表が指定されている問合せ指定の数

setop_num：SQL 文中に指定した集合演算子の数

FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数を加算してください。

drvtbl_num：SQL 文中に指定した導出表の数

FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数を加算してください。

in_query_num：次に示す条件を満たすIN 述語，限定述語の数の合計値

- IN 述語の場合

  右側に表副問合せを指定し，左側の値式に指定した列が検索に使用する B-tree インデクスの先頭構成列

- 限定述語の場合

  ＝ANY または＝SOME で左側の値式に指定した列が検索に使用する B-tree インデクスの先頭構成列

inprd_cor_query_num：

外への参照列を含む副問合せ中に指定されており，かつ変数*in_query_num* に示した条件を満たすIN 述語，限定述語の数の合計値

max_cor_query（X）：

SQL 文中の外への参照列を含む副問合せごとに，その副問合せ中に指定した*X* の最大値を代入します。

setop_cor_query_num：

外への参照列を含む副問合せ中に指定されている集合演算子の数

window_cor_query_num：外への参照列を含む副問合せ中に指定されているウィンドウ関数の数

外への参照列を含む副問合せ中で，1 つの問合せ指定中に，引数の異なるDISTINCT 集合関数を複数指定している場合は，その指定数－1 で求めた値も加算してください。

limit_cor_query_num：

外への参照列を含む副問合せ中に指定されているLIMIT 句の数

cor_query_num：

外への参照列を含む副問合せ，およびその副問合せ中に含まれる副問合せの合計値

Σ（*drvtbl_row_len*×MIN（*drvtbl_row_num*，1,044,480））：

導出表を指定する場合に，導出表ごとに（*drvtbl_row_len*×MIN（*drvtbl_row_num*，1,044,480））を求めてください。そして，求めた値を合計してください。

- *drvtbl_row_len*

  導出表の選択式のデータ長を合計した値を代入してください。

  データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

- *drvtbl_row_num*

  導出表の行数を代入してください。行数がわからない場合は，1,044,480 を代入してください。

Σ（*in_query_drvc_len*×MIN（*in_query_drvc_num*，1,044,480））：

変数*in_query_num* の対象となる述語ごとに（*in_query_drvc_len*×MIN（*in_query_drvc_num*，1,044,480））を求めてください。そして，求めた値を合計してください。

- *in_query_drvc_len*

  変数*in_query_num* の対象となる述語の副問合せの選択式のデータ長を代入してください。

  データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

- *in_query_drvc_num*

  変数*in_query_num* の対象となる述語の副問合せの行数を代入してください。行数がわからない場合は，1,044,480 を代入してください。

leftouter_num：

LEFT OUTER JOIN の指定数

Σ（*WRK_TABLE_OPE_SIZE*）：

作成される作業表単位に *WRK_TABLE_OPE_SIZE* を計算した合計値（単位：バイト）

次に示すことについては，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』を参照してください。

- SQL 文の実行時に作業表が作成されるかどうか
- 作成される作業表の個数および構成列

Σ（*LMTWRK_SIZE*）：

作成される作業表単位に *LMTWRK_SIZE* を計算した合計値（単位：バイト）

次に示すことについては，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』を参照してください。

- LIMIT 句を指定した SQL 文が作業表を作成するかどうか
- 作成される作業表の個数

なお，LIMIT 句で集合演算の結果の取得行数の最大値を設定する場合は，集合演算を構成する問合せ指定ごとに，LIMIT 句の指定値が適用されます。

tblfunc_num

表関数導出表の個数

Σ（*tblfunc_filename_len* ＋ 5）

表関数導出表ごとに（*tblfunc_filename_len* ＋ 5）を求めてください。そして，求めた値を合計してください。

tblfunc_filename_len

アクセスするファイルのファイル名（フルパス）の長さ

Σ（(*tblfunc_row_len* ＋ 4）×MIN（*tblfunc_row_num*，65,025)）

表関数導出表ごとに（(*tblfunc_row_len* ＋ 4）×MIN（*tblfunc_row_num*，65,025)）を求めてください。そして，求めた値を合計してください。

tblfunc_row_len

表関数導出表の列のデータ長の合計

データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

tblfunc_row_num

表関数導出表の行数

行数がわからない場合は，65,025 を代入してください。

Σ（*CSVREAD_WORK_SIZE*）

ADB_CSVREAD 関数を指定した表関数導出表ごとに変数 *CSVREAD_WORK_SIZE* をバイト単位で求めてください。そして，求めた値を合計してください。

Σ（*AUDITREAD_WORK_SIZE*）

ADB_AUDITREAD 関数を指定した表関数導出表ごとに変数*AUDITREAD_WORK_SIZE* をバイト単位で求めてください。そして，求めた値を合計してください。

query_num

問合せの数

SQL 文中に指定したFROM 句の数から求めてください。

また，FULL OUTER JOIN を指定した場合は，FULL OUTER JOIN の数×2 で求めた値を加算してください。

tvc_num

SQL 文中に指定した表値構成子の数

SCAN_WORK_SIZE：検索作業領域

次に示す計算式から求めてください。

計算式（単位：バイト）

SCAN_WORK_SIZE ＝ *MAX（SCB_SIZE）* ＋24

MAX（SCB_SIZE）：検索内部処理用の作業領域の最大値

検索する表ごとに変数*SCB_SIZE* を求めてください。その中で，最も大きい値を代入してください。

SCB_SIZE：検索内部処理用の作業領域

次に示す計算式から求めてください。

計算式（単位：バイト）

```
SCB_SIZE＝
    472＋MAX（256 , ↑KEYSZ÷8↑×8）＋64×idx_col_num
    ＋↑（KEYSZ×2＋10×logical_num）÷8↑×8＋56×（scan_str_num＋1）
    ＋64＋（txt_nest_num＋txt_contains_str）×8＋CS_WORK_SIZE
```

KEYSZ：検索に使用する B-tree インデクスのキー長

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を参照してください。

idx_col_num

検索に使用する B-tree インデクスの構成列数

logical_num

探索条件中の論理演算子の数の最大値

scan_str_num

次に示すどちらかの文字数を代入してください。

- テキストインデクスを使用するLIKE 述語のパターン文字列に指定した文字列の文字数
- スカラ関数CONTAINS に指定した検索文字列の文字数

  なお，スカラ関数CONTAINS に，ワード検索指定のWORDCONTEXT_PREFIX を指定して，単語単位の前方一致検索を行う場合は，「検索文字列に指定した単語の数×32」で求めた値を代入してください。

パターン文字列が 1 文字の場合，または，パターン文字列が 1,000 文字を超える場合は，1,000 を代入してください。

txt_nest_num

次に示すネスト数を合計した値※

- テキストインデクスを使用するLIKE 述語のOR 条件のネスト数
- スカラ関数CONTAINS のOR 条件のネスト数

txt_contains_str

スカラ関数CONTAINS に指定した検索文字列の文字数※

注※

変数 *txt_nest_num*，および変数 *txt_contains_str* に代入した値の合計値の上限は 1,001 です。合計値が 1,001 を超える場合は，1,001 を代入してください。

CS_WORK_SIZE：カラムストア表検索用の作業領域

次に示す計算式から求めてください。カラムストア表を検索しない場合は，0 になります。

計算式（単位：バイト）

```
CS_WORK_SIZE=
    2,576＋208×（MAX(col_num＋unnest_col_num , 3)＋in_projcol_num)
    ＋29×tbl_col_num×MIN（in_projcol_num , 1）＋↑tbl_col_num÷8↑×2
    ＋8×tbl_col_num＋27×SEGSIZE＋(page_size－56)×↑16,384÷page_size↑
    ＋65,568×pred_num＋branch_col_size＋Σ(DECOMPRESS_WORK_SIZE)
    ＋↑10×grp_col_num÷8↑×8＋Σ(↑grp_col_size÷8↑×8)
    ＋GROUPING_IN_SGMT_WORK_SIZE＋Σ(unnest_col_size)
```

col_num

実表の選択式の数

unnest_col_num

集まり導出表に指定する列指定の数

in_projcol_num

IN 述語の右側の値式に指定した列名の数

IN 述語を指定していない場合，またはIN 述語の右側の値式に列名を指定しない場合は，0 を代入してください。

tbl_col_num

表の列数

SEGSIZE

DB エリアファイルを定義した DB エリアのセグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

SEGSIZE＝ 1セグメントの容量（単位：バイト）÷*page_size*

- *page_size*

  表が定義されている DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」を参照してください。

pred_num

述語の数

探索条件を指定しない場合は，1 を代入してください。

branch_col_size

データ型が文字データ，またはバイナリデータの列の最大定義長（単位：バイト）

選択式，または探索条件に指定する列に定義長が 128 バイト以上の列が含まれる場合に，その定義長の最大値を代入してください。

Σ（*DECOMPRESS_WORK_SIZE*）

選択式，および探索条件に指定する列ごとに，変数 *DECOMPRESS_WORK_SIZE* を求めてください。そして，求めた値を合計してください。

grp_col_num

GROUP BY 句に指定した列の数

Σ（↑*grp_col_size*÷8↑×8）

GROUP BY 句に指定した列ごとに（↑*grp_col_size*÷8↑×8）を算出し，それぞれ算出した値を合計してください。

grp_col_size

GROUP BY 句に指定した各列のデータ長

データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

GROUPING_IN_SGMT_WORK_SIZE

セグメント内のグループ化作業領域

次に示す計算式から求めてください。なお，GROUP BY 句を指定していない場合は 0 を代入してください。

計算式（単位：バイト）

$$\begin{aligned}\text{GROUPING_IN_SGMT_WORK_SIZE} = {} & (\text{grp_col_num} + \text{set_func_num}) \times 258 \\ & + \sum_{i=1}^{\text{grp_col_num}} (\text{grouping_col_data_size}(i) \times 126 + 9) \\ & + \sum_{i=1}^{\text{set_func_num}} ((\text{grouping_set_func_size}(i) + 22) \times 126) + 263{,}305\end{aligned}$$

grp_col_num

GROUP BY 句に指定した列の数

set_func_num

集合関数の数

grouping_col_data_size(i)

各列のデータ長

データ長については，次に示す表から求めてください。

表 6-8　grouping_col_data_size に代入する値

項番	分類	データ型		データ長（単位：バイト）
1	数データ	INTEGER		8
2		SMALLINT		
3		DECIMAL NUMERIC	1≦*精度*≦4	8
4			5≦*精度*≦8	
5			9≦*精度*≦16	
6			17≦*精度*≦38	16
7		DOUBLE PRECISION		8
8		FLOAT		
9	文字データ	CHAR		*定義長*＋7
10		VARCHAR	1≦*定義長*≦32,000	*定義長*＋9
11			32,001≦*定義長*≦64,000	*定義長*＋11
12	日時データ	DATE		8
13		TIME(*p*)		3＋↑*p*÷2↑＋7
14		TIMESTAMP(*p*)		7＋↑*p*÷2↑＋7
15	バイナリデータ	BINARY		*定義長*＋7
16		VARBINARY		*定義長*＋9

（凡例）

p：小数秒の桁数（0，3，6，9 または 12。省略した場合は 0 を仮定）

grouping_set_func_size(i)

各集合関数の結果のデータ長

集合関数の結果のデータ長は，集合関数の結果のデータ型を基に「表 6-8　grouping_col_data_size に代入する値」を参照して求めてください。

Σ（*unnest_col_size*）

集まり導出表に指定する列指定ごとに変数 *unnest_col_size* を算出し，それぞれ算出した値を合計してください。

unnest_col_size

集まり導出表に指定する各列指定の列のデータ長

データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

DECOMPRESS_WORK_SIZE

伸長作業領域

次に示す計算式から求めてください。

計算式（単位：バイト）

$$DECOMPRESS_WORK_SIZE = \left\uparrow \frac{col_data_size(i)+1}{8} \right\uparrow \times 16 + page_size + 8,344$$

col_data_size(i)

各列のデータ長

データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

page_size

表を格納する DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」を参照してください。

TXT_WORK_SIZE

テキストインデクスの作業領域

次の計算式から求めてください。なお，テキストインデクスを定義していない場合，またはテキストインデクスを使用しない場合は 0 を代入してください。

計算式（単位：バイト）

$$TXT_WORK_SIZE = 28 \times scan_str_num + (txt_nest_num \times 5 + 2) \times 40 + 32,768$$

AUDITREAD_WORK_SIZE

ADB_AUDITREAD 関数の検索作業領域（単位：バイト）

次に示す計算式から求めてください。ただし，ADB_AUDITREAD 関数を使用しない場合は 0 を代入してください。

$$AUDITREAD_WORK_SIZE = (AUDREADSZ + AUDBLOCKSZ) \times 1,024$$

AUDREADSZ

監査証跡ファイルの管理領域

次に示す値を代入してください。ただし，ADB_AUDITREAD 関数を使用しない場合は 0 を代入してください。

値（単位：キロバイト）

$$AUDREADSZ = 1$$

AUDBLOCKSZ

監査証跡ファイルの作業領域

次に示す値を代入してください。ただし，ADB_AUDITREAD 関数を使用しない場合は 0 を代入してください。

値（単位：キロバイト）

AUDBLOCKSZ＝262,144

SCAN_INFO_SIZE：表検索用管理領域（単位：バイト）

```
SCAN_INFO_SIZE = Σ SCAN_INFO_DATA_SIZE + Σ SCAN_INFO_HASH_SIZE
```

Σ SCAN_INFO_DATA_SIZE

検索する表ごとに変数*SCAN_INFO_DATA_SIZE* を算出し，それぞれ算出した値を合計してください。

Σ SCAN_INFO_HASH_SIZE

ハッシュジョインから導出される結果が，ネストループジョインの最も外側の表に対応する場合，そのハッシュジョインごとに変数*SCAN_INFO_HASH_SIZE* を算出してください。また，副問合せのハッシュ実行から導出される結果が，ネストループジョインの最も外側に対応する場合，その副問合せのハッシュ実行ごとに変数*SCAN_INFO_HASH_SIZE* を算出してください。それぞれ算出した値を合計してください。

SCAN_INFO_DATA_SIZE：実表および結合表検索時の表検索用管理領域（単位：バイト）

```
SCAN_INFO_DATA_SIZE = tbl_access_num × (TBL_ACCESS_INFO_SIZE + OUTERJOIN_INFO_SIZE)
```

tbl_access_num：表の検索回数

次の条件のうち該当する値を代入してください。

- 結合していない表の場合：1
- ネストループジョインで結合している表の場合
  - ・ネストループジョインの最も外側の表の場合：1
  - ・ネストループジョインの内表の場合：その内表に対応する外表の行数
- ハッシュジョインで結合している表の場合
  - ・ハッシュジョインの外表の場合：1
  - ・ハッシュジョインの内表の場合：1

なお，表が外への参照列を含む副問合せの中に指定されていて，かつその副問合せの処理方式がネストループ作業表実行またはネストループ行値実行の場合，上記の値に「*uthd_num*×*sql_rthd_num*」を掛けた値を表の検索回数としてください。

uthd_num

サーバ定義のadb_sys_uthd_num オペランドの指定値

sql_rthd_num

サーバ定義のadb_sql_exe_max_rthd_num オペランドの指定値

＜表の検索回数の計算例＞

```
SELECT * FROM "T1" A,"T2" B,"T3" C
          WHERE A."C1"=B."C1" AND B."C2"=C."C2"
```

上記のSELECT 文を実行する際，表 T1，T2，T3 をこの順番で外側からネストループジョインする場合，表の検索回数は次のようになります。

- 表 T1 の検索回数：1
- 表 T2 の検索回数：表 T1 を検索して得られる結果の行数
- 表 T3 の検索回数：表 T1，表 T2 を検索して得られる結果の行数

TBL_ACCESS_INFO_SIZE：表検索情報領域

次の条件のうち該当する値を代入してください。

- 検索対象の表の検索方式がテーブルスキャンの場合：*TBL_SCI_MEM*
- 検索対象の表の検索方式が B-tree インデクスを使用したインデクススキャンまたはキースキャンの場合：*IDX_SCI_MEM*
- 検索対象の表の検索方式がテキストインデクスを使用したインデクススキャンの場合：*IDX_SCI_TXT_MEM*
- 検索対象の表の検索方式が作業表の検索の場合：*WRKTBL_SCI_MEM*

TBL_SCI_MEM：テーブルスキャン時の表検索情報領域（単位：バイト）

$$TBL_SCI_MEM = (336 + RESULT_WORK_SIZE) \times (dbarea_file_num + chunk_num + SGTBL - 2)$$

dbarea_file_num

検索対象表が格納されている DB エリアの DB エリアファイル数

chunk_num

検索対象表のチャンク数（検索対象表がシングルチャンク表の場合は 1）

SGTBL

検索対象表のセグメント数

SGTBL を求める計算式については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2) 変数の説明」にある次の個所を参照してください。なお，検索対象表の種類によって参照先が異なります。

- シングルチャンク表かつローストア表の場合
  「(a)　変数 SGROWTBL の求め方（シングルチャンク表の場合）」
- シングルチャンク表かつカラムストア表の場合
  「(b)　変数 SGCOLUMNTBL の求め方（シングルチャンク表の場合）」
- マルチチャンク表かつローストア表の場合
  「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」
- マルチチャンク表かつカラムストア表の場合
  「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」

IDX_SCI_MEM：B-tree インデクスを使用した検索時の表検索情報領域（単位：バイト）

IDX_SCI_MEM＝IDX_CHUNK_SCI_MEM ＋ IDX_SCAN_IDX_MEM

IDX_CHUNK_SCI_MEM

B-tree インデクスを使用した検索時のチャンクに関する表検索情報領域（単位：バイト）

IDX_CHUNK_SCI_MEM＝368＋*RESULT_WORK_SIZE*

ただし，次に示すどちらかの条件を満たす場合は，*IDX_CHUNK_SCI_MEM* ＝ 0 になります。

- 検索対象表が，チャンクが 1 個のマルチチャンク表の場合
- 検索対象表がシングルチャンク表の場合

IDX_SCAN_IDX_MEM

B-tree インデクスを使用した検索時の全チャンクの表検索情報領域（単位：バイト）

$$IDX_SCAN_IDX_MEM = \sum_{s=1}^{chunk_num} IDX_SCAN_IDX_MEM(s)$$

ただし，検索対象のチャンク数が 0 個の場合は，*IDX_SCAN_IDX_MEM* ＝ 0 になります。

chunk_num

検索対象となるチャンク数

IDX_SCAN_IDX_MEM(s)

s 番目のチャンクの B-tree インデクスに関する表検索情報領域（単位：バイト）

・s番目のチャンクにサーチ条件を満たす行※が2行以上ある場合

$$IDX_SCAN_IDX_MEM(s) = PIDX_MEM \times \left\{ r + \sum_{k=1}^{n} PIDX(k) \right\} + PDUP_MEM \times PDUP$$

・s番目のチャンクにサーチ条件を満たす行※が1行以下しかない場合

$$IDX_SCAN_IDX_MEM(s) = 0$$

注※　DELETE文による行の削除や，UPDATE文による行の更新によって無効となった行データも含めてサーチ条件を満たす行数を計算してください。

r

サーチ条件を満たす行のキー値が 2 種類以上の場合は 2，1 種類の場合は 0 になります。

n

PIDX(k) ＝ 1 となる最小の k になります。

PIDX(k)

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』に従って求めてください。

なお，*PIDX(k)* を求める計算式に使用されている変数 *key_num*，*dup_key_num*，および *key_dup* については，サーチ条件を満たすキーについて，キーの種類の個数やキーの重複数の平均値を求め

てください。キーの種類やキーの重複数を計算する際，DELETE 文による行の削除や，UPDATE 文による行の更新によって無効となった行データも含めて計算してください。また，INSERT 文による行の追加，またはUPDATE 文による行の変更を繰り返した場合や，キーの重複数やキー長にばらつきがある場合は，求めた値を 1.2～1.5 倍した値を用いてください。

PDUP

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の『計算式 2（PDUP の求め方)』に従って求めてください。

なお，*PDUP* を求める計算式に使用されている変数 *dup_key_num* と *dup_key_dup* については，サーチ条件を満たすキーについて，キーの種類の個数やキーの重複数の平均値を求めてください。キーの種類やキーの重複数を計算する際，DELETE 文による行の削除や，UPDATE 文による行の更新によって無効となった行データも含めて計算してください。また，INSERT 文による行の追加，または UPDATE 文による行の変更を繰り返した場合や，キーの重複数やキー長にばらつきがある場合は，求めた値を 1.2～1.5 倍した値を用いてください。

PIDX_MEM

B-tree インデクスの上位またはリーフページに関する表検索情報領域（単位：バイト）

$$PIDX_MEM = 432 + RESULT_WORK_SIZE + 24 \times (idx_col_num - 1) + 2 \times idx_col_num + 2 \times \sum_{i=1}^{idx_col_num} key_size(i)$$

idx_col_num

B-tree インデクスのインデクス構成列数

key_size(i)

i 番目のインデクス構成列のキー長

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を参照して，*key_size(i)* を計算してください。

なお，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」は，「データの定義長」と読み替えて計算してください。

PDUP_MEM

B-tree インデクスのロー ID ディレクトリページ，またはロー ID リストページに関する表検索情報領域（単位：バイト）

$$PDUP_MEM = 368 + RESULT_WORK_SIZE + 2 \times idx_col_num + \sum_{i=1}^{idx_col_num} key_size(i)$$

idx_col_num

B-tree インデクスのインデクス構成列数

key_size(i)

i 番目のインデクス構成列のキー長

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を参照して，*key_size(i)* を計算してください。

なお，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」は，「データの定義長」と読み替えて計算してください。

IDX_SCI_TXT_MEM

テキストインデクスを使用した検索時の表検索情報領域（単位：バイト）

```
IDX_SCI_TXT_MEM ＝ IDX_CHUNK_TXT_MEM ＋ IDX_SCAN_TXT_MEM
```

IDX_CHUNK_TXT_MEM

テキストインデクスを使用した検索時のチャンクに関する表検索情報領域（単位：バイト）

```
IDX_CHUNK_TXT_MEM ＝ 328＋8×MIN(TXT_SCAN_NODE_NUM ，  1,001)
                    ＋8×MIN(scan_str_num ，  1,001)＋RESULT_WORK_SIZE
```

ただし，次に示すどちらかの条件を満たす場合は，*IDX_CHUNK_TXT_MEM* ＝ 0 になります。

- 検索対象表が，チャンクが 1 個のマルチチャンク表の場合
- 検索対象表がシングルチャンク表の場合

scan_str_num

テキストインデクスを使用するLIKE 述語，LIKE_REGEX 述語，スカラ関数CONTAINS に指定した検索文字列の文字数の合計（1,001 を超える場合は 1,001 とする）

TXT_SCAN_NODE_NUM

テキストインデクス内部の検索に使用するノード数

```
TXT_SCAN_NODE_NUM ＝ scan_nest_num ＋ sortcode_str_num ＋ TXT_SYNONYM_NUM
```

scan_nest_num

テキストインデクスを使用するLIKE 述語，LIKE_REGEX 述語，スカラ関数CONTAINS の OR 条件のネスト数

sortcode_str_num

スカラ関数CONTAINS で表記ゆれ補正検索を行う場合の検索文字列の文字数

TXT_SYNONYM_NUM

スカラ関数CONTAINS で同義語検索を使用する場合の同義語グループの同義語数の平均値

```
TXT_SYNONYM_NUM ＝ ↑all_synonym_num ÷ synonym_group_num↑
```

all_synonym_num

スカラ関数CONTAINS で同義語検索を行う場合に指定した同義語辞書に登録されている同義語数すべての同義語グループの同義語の合計数です。同義語検索を行わない場合は，0 になります。

synonym_group_num

スカラ関数CONTAINS で同義語検索を行う場合に指定した同義語辞書に登録されている同義語グループ数

IDX_SCAN_TXT_MEM

テキストインデクスを使用した検索時の全チャンクの表検索情報領域（単位：バイト）

$$IDX_SCAN_TXT_MEM = \sum_{s=1}^{chunk_num} IDX_SCAN_TXT_MEM(s)$$

ただし，検索対象のチャンク数が 0 個の場合は，*IDX_SCAN_TXT_MEM* ＝ 0 になります。

chunk_num

検索対象のチャンク数

IDX_SCAN_TXT_MEM(s)

s 番目のチャンクのテキストインデクスに関する表検索情報領域（単位：バイト）

$$IDX_SCAN_TXT_MEM(s) = 16 \times \left\{ \begin{array}{l} 328 + 8 \times MIN(TXT_SCAN_NODE_NUM\ ,\ \ 1{,}001) \\ +56 \times MIN(scan_str_num\ ,\ \ 1{,}001) + RESULT_WORK_SIZE \end{array} \right\}$$

WRKTBL_SCI_MEM

作業表検索時の表検索情報領域（単位：バイト）

$$WRKTBL_SCI_MEM = (304 + RESULT_WORK_SIZE) \times WRKTBL_PGNUM$$

WRKTBL_PGNUM

検索対象の作業表のページ数

「5.9.2　作業表を格納するために必要な基本行用ページ数の求め方」を参照して，*WP(i)* を計算してください。

SCAN_INFO_HASH_SIZE

ハッシュテーブルを使用した検索時の表検索用管理領域（単位：バイト）

$$\begin{aligned} SCAN_INFO_HASH_SIZE = \\ & hashtbl_row_len \times MIN(hashtbl_row_num\ ,\ \ 1{,}044{,}480) \\ & + (RESULT_WORK_SIZE + 200 + 40 \times sql_rthd_num) \times \uparrow MIN(hashtbl_row_num\ ,\ \ 1{,}044{,}480) \div 255 \uparrow \end{aligned}$$

hashtbl_row_len

ハッシュテーブル行長

「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照して，ハッシュテーブル行長を計算してください。

hashtbl_row_num

ハッシュジョインまたは副問合せのハッシュ実行から導出される結果行数

わからない場合は，1,044,480 としてください。

sql_rthd_num

最大 SQL 処理リアルスレッド数（サーバ定義のadb_sql_exe_max_rthd_num オペランドの指定値）

OUTERJOIN_INFO_SIZE

外結合管理領域（単位：バイト）

OUTERJOIN_INFO_SIZE ＝ 120＋16×*sql_rthd_num* ＋ *RESULT_WORK_SIZE*

ただし，次の条件をすべてを満たす場合は，*OUTERJOIN_INFO_SIZE* ＝ 0 になります。

- 検索対象の表が外結合の左側に指定されていない場合
- 検索対象の表が外結合の右側に指定されていない場合

sql_rthd_num

最大 SQL 処理リアルスレッド数（サーバ定義のadb_sql_exe_max_rthd_num オペランドの指定値）

SORTGRP_PROC_SIZE

SQL 処理リアルスレッド用のソートグループ化の作業領域

次のどちらかの集合関数を SQL 文中に指定する場合に求めてください。どちらの指定もない場合は，0 を代入してください。

- LISTAGG 集合関数
- ARRAY_AGG 集合関数

計算式（単位：バイト）

SORTGRP_PROC_SIZE＝
352×Σ（*sortgrp_cor_query_wrk_num*）×*uthd_num*
＋*LISTAGG_PROC_SIZE*＋*ARRAY_AGG_PROC_SIZE*

Σ（*sortgrp_cor_query_wrk_num*）

外への参照列を含む副問合せ中に含まれるLISTAGG 集合関数，またはARRAY_AGG 集合関数を指定した問合せ指定ごとに，変数*sortgrp_cor_query_wrk_num* を求めてください。そして，それぞれ求めた値を合計してください。

sortgrp_cor_query_wrk_num

外への参照列を含む副問合せ中に含まれるソートグループ化の作業領域の個数

次の数を合計したものを代入してください。

- ALL（省略値）を指定したLISTAGG 集合関数の指定数
- DISTINCT を指定したLISTAGG 集合関数の指定数×2
- ARRAY_AGG 集合関数の指定数
- DISTINCT 集合関数の指定数
- 逆分布関数の指定数

uthd_num

1 リアルスレッドで生成できる擬似スレッド数（サーバ定義adb_sys_uthd_num オペランドの指定値）

LISTAGG_PROC_SIZE

SQL 処理リアルスレッド用のLISTAGG 集合関数の作業領域

SQL 文中に，LISTAGG 集合関数を含む場合に求めてください。LISTAGG 集合関数を含まない場合は，0 になります。

計算式（単位：バイト）

LISTAGG_PROC_SIZE＝128×*listagg_cor_query_num*×*uthd_num*

listagg_cor_query_num

外への参照列を含む副問合せ中に指定されているLISTAGG 集合関数の数

uthd_num

1 リアルスレッドで生成できる擬似スレッド数（サーバ定義adb_sys_uthd_num オペランドの指定値）

ARRAY_AGG_PROC_SIZE

SQL 処理リアルスレッド用のARRAY_AGG 集合関数の作業領域

ARRAY_AGG 集合関数を SQL 文中に指定する場合に求めてください。ARRAY_AGG 集合関数を指定しない場合は，0 を代入してください。

計算式（単位：バイト）

ARRAY_AGG_PROC_SIZE＝
Σ {*array_agg_cor_query_num*×（（*array_agg_cor_query_num*＋1）×104＋*RESULT_WORK_SIZE*）} ×*uthd_num*
＋Σ（*array_agg_not_cor_query_num*）×*RESULT_WORK_SIZE*

Σ {*array_agg_cor_query_num*×（(*array_agg_cor_query_num* ＋ 1）×104 ＋ *RESULT_WORK_SIZE*)}

外への参照列を含む副問合せ中の，ARRAY_AGG 集合関数を指定した問合せ指定ごとに，次の値を求めてください。そして，それぞれ求めた値を合計してください。

array_agg_cor_query_num×（（*array_agg_cor_query_num*＋1）×104＋*RESULT_WORK_SIZE*）

array_agg_cor_query_num

外への参照列を含む副問合せ中の問合せ指定に指定されているARRAY_AGG 集合関数の数

uthd_num

1 リアルスレッドで生成できる擬似スレッド数（サーバ定義adb_sys_uthd_num オペランドの指定値）

Σ（*array_agg_not_cor_query_num*）

外への参照列を含む副問合せ中には指定されていない問合せ指定ごとに，変数 *array_agg_not_cor_query_num* を求めてください。そして，それぞれ求めた値を合計してください。

array_agg_not_cor_query_num

外への参照列を含む副問合せ中には指定されていない問合せ指定に指定されているARRAY_AGG 集合関数の数

RT_AGGR_PROC_WORK_SIZE

SQL 処理リアルスレッド用のリアルスレッド単位の集計作業領域

SQL 文中に，次に示すどれかの要素を含む場合に求めてください。含まない場合は，0 を代入してください。

- LISTAGG 集合関数
- ARRAY_AGG 集合関数
- 逆分布関数
- ウィンドウ関数

データ長の計算は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

計算式（単位：バイト）

```
RT_AGGR_PROC_WORK_SIZE＝
        （64×RESULT_WORK_SIZE×sql_rthd_num）×rt_aggr_num
```

sql_rthd_num

最大 SQL 処理リアルスレッド数

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値です。指定値が 0 の場合は 1 を代入してください。

rt_aggr_num

該当する要素の数

PRL_PROC_SIZE

SQL パラレル実行機能のリアルスレッド管理領域

SQL パラレル実行機能が適用される場合に求めてください。SQL パラレル実行機能が適用されない場合は，0 を代入してください。

値（単位：バイト）

```
PRL_PROC_SIZE＝512＋PRL_HASH_PROC_SIZE
```

PRL_HASH_PROC_SIZE

ノード間共有するハッシュテーブルのリアルスレッド管理領域

SQL パラレル実行機能が適用され，かつ SQL 文中にGROUP BY 句を含む場合に求めてください。SQL パラレル実行機能が適用されない場合，またはGROUP BY 句を含まない場合は，0 を代入してください。また，最大 SQL 処理リアルスレッド数，またはハッシュテーブル領域サイズが 0 の場合も，0 を代入してください。

計算式（単位：バイト）

```
PRL_HASH_PROC_SIZE＝
   4,175＋8×hashtbl_num＋（8,512＋（13,632＋MAX（131,072 , hashtbl_row_len）×80）
        ×node_num）×prl_hashtbl_num
```

hashtbl_num

ハッシュテーブル数

「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

hashtbl_row_len

ハッシュテーブル行長

「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

node_num

マルチノード構成のノード数

prl_hashtbl_num

GROUP BY 句の数

SQL_PRL_COMM_SIZE

操作系 SQL の通信管理領域

次の値を代入してください。

値（単位：バイト）

- SQL パラレル実行機能が適用される場合

  SQL_PRL_COMM_SIZE＝8,256

- SQL パラレル実行機能が適用されない場合

  SQL_PRL_COMM_SIZE＝0

BUF_DLYSZ

更新ページの通知領域

次の値を代入してください。

値（単位：キロバイト）

```
■マルチノード機能を使用している場合で，INSERT文，UPDATE文およびDELETE文を実行するとき
  BUF_DLYSZ＝ 41

■上記以外の場合
  BUF_DLYSZ＝ 0
```

ARC_DIR_PATH

削除対象のアーカイブディレクトリのパス名を管理する領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
ARC_DIR_PATH＝ ↑archivedir_path_num×513÷1,024↑
```

archivedir_path_num

TRUNCATE TABLE 文を実行する場合は，1 を代入してください。

PURGE CHUNK 文を実行する場合は，削除対象のチャンク数を代入してください。

RNGIDX_INFO_SIZE

チャンクのレンジインデクス情報領域

次に示す計算式から求めてください。

計算式（単位：バイト）

$$RNGIDX_INFO_SIZE=$$

$$32+8\times rngidx_num$$

$$+\sum_{i=1}^{rngidx_num}(16+8\times(chunk_num(i)-1)+RNGIDX_INFO_ENTRY_SIZE(i)\times chunk_num(i))$$

rngidx_num

SQL 文の実行時に，チャンクのスキップに使用されるレンジインデクスのうち，B-tree インデクスを使用して検索する表の検索に使用されるレンジインデクスの合計数

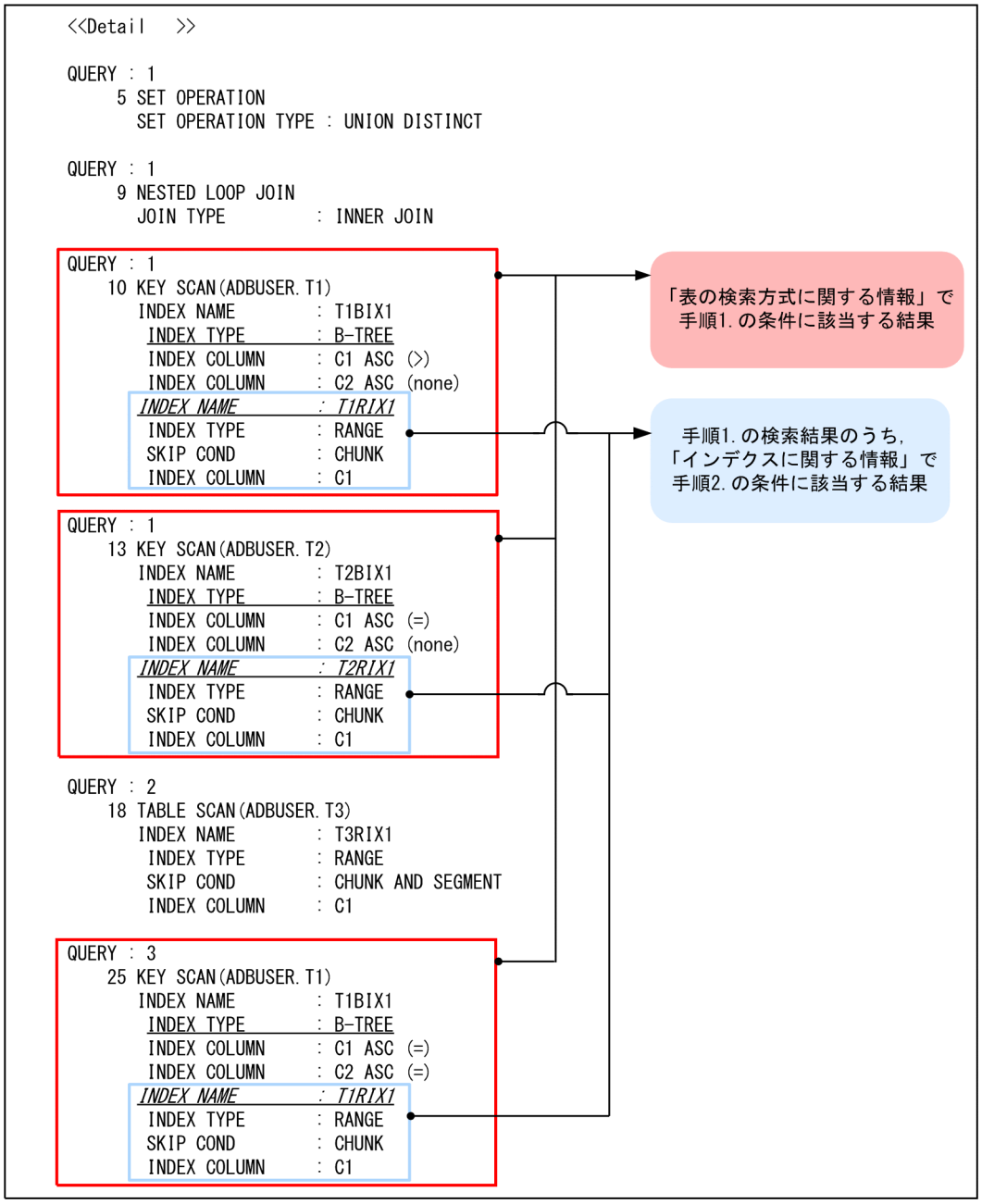
同じレンジインデクスで SQL 文が複数回実行される場合は 1 として合計数を求めてください。合計数が 0 の場合，*RNGIDX_INFO_SIZE* は 0 としてください。

なお，この値はアクセスパス情報の表示結果から，次の手順で求めることができます。

1. アクセスパス情報の表示結果の詳細表示の，『表の検索方式に関する情報』と『インデクスに関する情報』を確認してください。次のどちらかの条件に該当するものをすべて探してください。
   - 表の検索方式がINDEX SCAN であり，『インデクスに関する情報』の 1 つ目のインデクス情報の INDEX TYPE がB-TREE（B-tree インデクス）である
   - 表の検索方式がKEY SCAN である
2. 手順 1.の条件に該当し，かつ『インデクスに関する情報』が次の両方の条件に該当するものをすべて探してください。
   - INDEX TYPE がRANGE（レンジインデクス）である
   - SKIP COND がCHUNKである
3. 手順 2.に該当する『インデクスに関する情報』にあるINDEX NAME のうち，同じ名前のものは重複排除して合計数を求めてください。求めた値が *rngidx_num* となります。

   次に示すアクセスパス情報の表示結果の例の場合，INDEX NAME は，「T1RIX1」（2 つ）と，「T2RIX1」（1 つ）の 2 種類です。そのため，*rngidx_num* は 2 となります。

■アクセスパス情報の表示結果の例



アクセスパスについては，マニュアル『HADB　AP 開発ガイド』の『アクセスパスの見方（SQL 文の実行計画の見方)』を参照してください。

chunk_num(i)

i 番目のレンジインデクスが定義されている表に作成されたチャンク数

アーカイブチャンクの数は除いてください。シングルチャンク表の場合は 1 となります。

RNGIDX_INFO_ENTRY_SIZE(i)

レンジインデクス情報エントリ長

次に示す計算式から求めてください。

計算式（単位：バイト）

$$RNGIDX_INFO_ENTRY_SIZE(i) = 16 + \left(\frac{2 \times col_size(i) - 1}{8} + 1 \right) \times 8$$

col_size(i)

レンジインデクスを定義した列のデータ長（単位：バイト）

各列のデータ長は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

データ長については，次に示す表から求めてください。

表 6-9　各データ型のデータ長の一覧

項番	分類	データ型		データ長（単位：バイト）
1	数データ	INTEGER		8
2		SMALLINT		4
3		DECIMAL NUMERIC	1≦*精度*≦4	2
4			5≦*精度*≦8	4
5			9≦*精度*≦16	8
6			17≦*精度*≦38	16
7		DOUBLE PRECISION		8
8		FLOAT		
9	文字データ	CHAR		*定義長*
10		VARCHAR	1≦*定義長*≦32,000	*定義長*＋2
11			32,001≦*定義長*≦64,000	*定義長*＋4
12	日時データ	DATE		4
13		TIME(p)		3＋↑p÷2↑
14		TIMESTAMP(p)		7＋↑p÷2↑
15	バイナリデータ	BINARY		*定義長*
16		VARBINARY		*定義長*＋2
17	配列データ	ARRAY		（*要素データ型のデータ長*×n）＋2

（凡例）

p：小数秒の桁数（0，3，6，9 または 12。省略した場合は 0 を仮定）

n：配列データの最大要素数（2～30,000 の整数）

注

CURRENT_USER はデータ型VARCHAR，定義長 100 として計算してください。

(d)　変数 RTHD_ROLLBKSZ の求め方

変数*RTHD_ROLLBKSZ* は，ロールバック処理時，または HADB サーバの再開始時に加算します。次に示す値を代入してください。

値（単位：キロバイト）

```
RTHD_ROLLBKSZ＝ 67
```

(e)　変数 RTHD_DBEXTSZ の求め方

変数*RTHD_DBEXTSZ* は，DB エリア拡張時に加算します。次に示す値を代入してください。

値（単位：キロバイト）

```
RTHD_DBEXTSZ＝ 32,768
```

(f)　変数 RTHD_WORKBUF の求め方

変数*RTHD_WORKBUF* は，作業表使用時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_WORKBUF＝
    ↑（clt_wrk_page_num×（wrk_page_size＋304）＋8
      ×（clt_wrk_page_num＋WIO_NUM）＋wrk_page_size）÷1,024↑×1.05
```

変数の説明

- *clt_wrk_page_num*

  クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値（ただし，adbexport コマンド実行時はエクスポートオプションadb_export_wrktbl_blk_num の指定値）

- *wrk_page_size*

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」の「表 6-3　DB エリアのページサイズ」の作業表用 DB エリアのページサイズを参照して，バイト単位で求めてください。

- *WIO_NUM*

  クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値（ただし，adbexport コマンド実行時はエクスポートオプションadb_export_wrktbl_blk_num の指定値）と変数*UTHNUM* のうち，小さい方の値を代入してください。変数*UTHNUM* については，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a)　変数 SCI の求め方」を参照してください。

(g)　変数 RTHD_EXESQLDICSZ の求め方

変数*RTHD_EXESQLDICSZ* は，操作系 SQL の前処理時に加算します。次に示す計算式から求めてください。

なお，対象表がビュー表の場合は，計算式で求めた値に 39 キロバイトを加算してください。

計算式（単位：キロバイト）

```
RTHD_EXESQLDICSZ＝ 104＋↑（（LIKE_REG_PRD_SIZE＋CONTAINS_PRD_SIZE）÷1,024）↑
```

変数の説明

LIKE_REG_PRD_SIZE

LIKE_REGEX 述語の実行管理領域（単位：バイト）

次に示す計算式から求めてください。

なお，LIKE_REGEX 述語を指定していない場合は，変数*LIKE_REG_PRD_SIZE* に 0 を代入してください。

```
LIKE_REG_PRD_SIZE＝
  ↑（scan_str_num÷50）↑×2,400
  ＋MAX（（5＋normal_factor_size＋paren_nest_num)×8　,
         (1＋vertical_line_num＋quantifier_nest_num＋repetition_factor_sum)×16)
  ＋BYTE_CODE_SIZE
```

scan_str_num

パターン文字列に指定した文字列の文字数

normal_factor_size

パターン文字列に指定された正規表現内の正規因子の数

paren_nest_num

パターン文字列に指定された正規表現内の丸括弧のネスト数

vertical_line_num

パターン文字列に指定された正規表現内の垂直棒の数

quantifier_nest_num

パターン文字列に指定された正規表現内の量指定子のネストの数

repetition_factor_sum

パターン文字列に指定された正規表現内の繰り返し因子に指定された上限値の合計値

繰り返し因子に指定された上限値が存在しない場合は，繰り返し因子に指定された下限値の値を代入してください。

正規表現の記述例

```
((a{2}){5}){8,10}
```

計算式（単位：個数）

```
repetition_factor_sum＝ 2＋5＋10 ＝ 17
```

正規表現の記述例と，変数*LIKE_REG_PRD_SIZE* の計算式の例を，次に示します。

正規表現の記述例 1

```
KFAA11[0-9]{3}-E
```

計算式の例（単位：バイト）

```
LIKE_REG_PRD_SIZE＝
    ↑(16÷50)↑×2,400＋MAX（（5＋9＋0）×8，（1＋0＋0＋3）×16)＋170 ＝ 2,682

  説明
    下線部分は，変数BYTE_CODE_SIZEの計算式で求める。
```

正規表現の記述例 2

```
KFAA6000[2-9]-E.*uid=12345.*
```

計算式の例（単位：バイト）

```
LIKE_REG_PRD_SIZE＝
    ↑（28÷50）↑×2,400＋MAX（（5＋22＋0）×8，（1＋0＋0＋0）×16）＋165 ＝  2,781

  説明
    下線部分は，変数BYTE_CODE_SIZEの計算式で求める。
```

BYTE_CODE_SIZE

LIKE_REGEX 述語の実行管理領域内のパターン文字列に指定された正規表現を評価用に変換したバイトコードを格納する領域（単位：バイト）

次に示す計算式から求めてください。

```
BYTE_CODE_SIZE＝
  char_specifier_num×5＋quantifier_num×18
  ＋char_class_num×16＋normal_char_set_num×16
  ＋char_list_num×34＋repetition_factor_sum×9＋1
```

char_specifier_num

パターン文字列に指定された正規表現内の文字指定子の数の合計値

文字指定子に繰り返し因子が指定されている場合は，その繰り返し数を加算してください。また，「(a{10}){10}」のように繰り返し因子がネストしている場合は，繰り返し数を乗算してください。そして，乗算した結果を，変数 *char_specifier_num* に加算してください。

繰り返し因子に指定された繰り返しの数については，「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-7　繰り返し因子に指定された繰り返しの数の求め方」を参照してください。

quantifier_num

パターン文字列に指定された正規表現内の繰り返し因子を除く量指定子の数

char_class_num

パターン文字列に指定された正規表現内の文字クラスの数の合計値

文字クラスに繰り返し因子が指定されている場合は，その繰り返し数を加算してください。また，「(¥d{10}){10}」のように繰り返し因子がネストしている場合は，繰り返し数を乗算してください。そして，乗算した結果を，変数 *char_class_num* に加算してください。

繰り返し因子に指定された繰り返しの数については，「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-7　繰り返し因子に指定された繰り返しの数の求め方」を参照してください。

normal_char_set_num

パターン文字列に指定された正規表現内の正規文字集合識別子指定の数の合計値

正規文字集合識別子指定に繰り返し因子が指定されている場合は，その繰り返し数を加算してください。また，「([[:alpha:]]{10}){10}」のように繰り返し因子がネストしている場合は，繰り返し数を乗算してください。そして，乗算した結果を，変数*normal_char_set_num* に加算してください。

繰り返し因子に指定された繰り返しの数については，「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-7　繰り返し因子に指定された繰り返しの数の求め方」を参照してください。

> **メモ**
>
> 「([[:alpha:]]{10}){10}」の場合は，正規文字集合も繰り返しているため，変数*char_list_num* にも 100 を加算する必要があります。

char_list_num

パターン文字列に指定された正規表現内の正規文字集合の数の合計値

正規文字集合に繰り返し因子が指定されている場合は，その繰り返し数を加算してください。また，「([abc]{10}){10}」のように繰り返し因子がネストしている場合は，繰り返し数を乗算してください。そして，乗算した結果を，変数*char_list_num* に加算してください。

繰り返し因子に指定された繰り返しの数については，「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-7　繰り返し因子に指定された繰り返しの数の求め方」を参照してください。

repetition_factor_sum

パターン文字列に指定された正規表現内の繰り返し因子に指定された上限値の合計値

繰り返し因子に指定された上限値が存在しない場合は，繰り返し因子に指定された下限値の値を代入してください。

正規表現の記述例

```
((a{2}){5}){8,10}
```

計算式（単位：個数）

```
repetition_factor_sum= 2+5+10 ＝ 17
```

正規表現の記述例と，変数*BYTE_CODE_SIZE* の計算式の例を，次に示します。

正規表現の記述例 1

```
KFAA11[0-9]{3}-E
```

計算式の例（単位：バイト）

```
BYTE_CODE_SIZE= 8×5+1×34×3+3×9+1 = 170

  説明
    ・char_specifier_numに，8を代入する。
    ・char_list_numに，3を代入する（正規文字集合が1つで，かつ繰り返し数が3であるため）。
    ・repetition_factor_sumに，3を代入する。
```

正規表現の記述例 2

```
KFAA6000[2-9]-E.*uid=12345.*
```

計算式の例（単位：バイト）

```
BYTE_CODE_SIZE= 19×5+2×18+34×1 = 165

  説明
    ・char_specifier_numに，19を代入する。
    ・quantifier_numに，2を代入する。
    ・char_list_numに，1を代入する。
```

CONTAINS_PRD_SIZE

スカラ関数CONTAINS の実行管理領域（単位：バイト）

次に示す計算式から求めてください。

なお，スカラ関数CONTAINS を指定していない場合は，変数 *CONTAINS_PRD_SIZE* に 0 を代入してください。

```
CONTAINS_PRD_SIZE＝scan_str_num×8＋SYN_LIST_SIZE
```

scan_str_num

パターン文字列に指定した文字列の文字数

SYN_LIST_SIZE

同義語の格納領域（単位：バイト）

次に示す計算式から求めてください。

なお，スカラ関数CONTAINS で同義語検索指定を指定していない場合は，変数 *SYN_LIST_SIZE* に 0 を代入してください。

```
SYN_LIST_SIZE＝syn_str_num×syn_group_size×8
```

syn_str_num

同義語の平均文字数

syn_group_size

1 同義語グループ中に指定されている同義語数の平均

(h) 変数 RTHD_COMMUSZ の求め方

変数 *RTHD_COMMUSZ* は，通信処理時に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_COMMUSZ＝ *RCVBUF*＋*SNDBUF*

変数の説明

RCVBUF：受信バッファ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RCVBUF＝ ↑（352＋*CMSG*）÷1,024↑

CMSG：HADB クライアントからの送信データサイズ

マニュアル『HADB　AP 開発ガイド』の『HADB クライアントのメモリ所要量の見積もり』の『HADB クライアントと HADB サーバの通信時に使用するメモリ所要量』にある変数 *CMSG* を参照してください。

SNDBUF：送信バッファ

初期確保時の送信バッファは次に示す値を代入してください。

値（単位：キロバイト）

SNDBUF＝ 4

初期確保した送信バッファを超える送信データが作成されると，HADB サーバは次に示す計算式で求められた値を再確保します。

計算式（単位：キロバイト）

SNDBUF＝ ↑*SMSG*÷4,096↑×4

SMSG：HADB サーバの送信データサイズ

送信データサイズは処理の内容によって異なります。各処理で確保される送信データサイズを次の表に示します。

表 6-10　各処理で確保される送信データサイズの一覧

項番	処理内容	送信データサイズの計算式（単位：バイト）
1	SQL 文を実行した場合	SMSG＝ srv_base_info＋srv_execute＋ARRAY_RESULT
2	SQL 文の実行がNOROW の場合	SMSG＝ srv_base_info＋srv_execute＋ARRAY_RESULT＋srv_norow
3	FETCH の場合	SMSG＝ srv_base_info＋srv_fetch＋FETCH_DATA
4	FETCH がNOROW の場合（一括転送で 1 行以上 FETCH できた場合）	SMSG＝ srv_base_info＋srv_fetch＋srv_norow

項番	処理内容	送信データサイズの計算式（単位：バイト）
5	SQL 文の前処理をした場合	SMSG＝ srv_base_info＋srv_prepare＋PREPARE_INFO_SIZE＋APATHVIEW
6	adbdbstatus コマンドを実行した場合	SMSG＝ srv_base_info＋srv_status_info

（凡例）

srv_base_info：送信データの基本情報

152 バイトを代入してください。

srv_execute：実行固有情報

24 バイトを代入してください。

srv_fetch：FETCH 固有情報

16 バイトを代入してください。

srv_norow：NOROW 固有情報

520 バイトを代入してください。

srv_prepare：前処理固有情報

56 バイトを代入してください。

srv_status_info：データベース状態解析固有情報

1,048,576 バイトを代入してください。

ARRAY_RESULT：一括更新結果

次に示す計算式から求めてください。

計算式（単位：バイト）

ARRAY_RESULT＝ （*array_num*-1）×8

array_num：一括更新した？パラメタの組の数

AP の実装方法によって異なります。

- CLI 関数を使用した AP の場合

  a_rdb_SQLBindArrayParams()の引数ArrayCount に指定した数になります。
  a_rdb_SQLBindParams()を使用してパラメタ結合した場合は，1 で固定されます。

- JDBC ドライバを使用した AP の場合

  addBatch メソッドで登録したパラメタリストの数になります。executeBatch メソッドまたは executeLargeBatch メソッドを使用しない場合は，1 で固定されます。

FETCH_DATA：一括転送行データ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$FETCH_DATA = fetch_size \times \sum_{i=1}^{col_num} col_size(i)$$

col_num

SQL 文に指定した選択式の数

col_size（i）

各選択式のデータ長

fetch_size：FETCH 処理時の一括送信行数

クライアント定義adb_clt_fetch_size オペランドの指定値を代入してください。クライアント定義adb_clt_fetch_size オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

PREPARE_INFO_SIZE：SQL 文の前処理情報

「(c)　変数 RTHD_EXESQLSZ の求め方」の変数*PREPARE_INFO_SIZE* を参照してください。

APATHVIEW：アクセスパスを表示する際に作成されるアクセスパス情報管理領域

「(1)　プロセス共通メモリの所要量の求め方（通常運用時)」の「(b)　変数 PROC_TOTALSQLSSZ の求め方」にある変数*APATHVIEW* を参照してください。

(i)　変数 RTHD_EXPSQLTRCSZ の求め方

変数RTHD_EXPSQLTRCSZ は，SQL トレースバッファの拡張時に加算します（変数RTHD_EXPSQLTRCSZ の値が32,768 キロバイトを超えると，SQL トレースバッファが拡張されます)。次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_EXPSQLTRCSZ = \uparrow (PARAM_INFO_SIZE \times array_num) \div 1,024 \uparrow$$

変数の説明

PARAM_INFO_SIZE

「(c)　変数 RTHD_EXESQLSZ の求め方」の変数*PARAM_INFO_SIZE* を参照してください。

array_num

一括更新する際に指定した？パラメタセットの数

(j)　変数 RTHD_AUDINFSZ の求め方

変数*RTHD_AUDINFSZ* は，監査証跡機能が有効な場合に加算します。次に示す値を代入してください。

値（単位：キロバイト）

$$RTHD_AUDINFSZ = 3$$

(k)　変数 RTHD_COLUMNIZESZ の求め方

変数 *RTHD_COLUMNIZESZ* は，更新行のカラム化機能が有効な場合に加算します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_COLUMNIZESZ = TABLELIST + \max_{1 \leqq i \leqq table_num} (TABLEINFO(i) + RELOCATE_DAT(i) + RTHD_DATACOMPRESS(i) \times 2)$$

変数の説明

TABLELIST：表名リスト情報

次に示す値を代入してください。

値（単位：キロバイト）

$$TABLELIST = 833$$

table_num：更新行のカラム化機能の対象となるカラムストア表の総数

更新行のカラム化機能の対象となるカラムストア表については，「11.18.2　準備作業」の「(1)　機能の適用対象となるカラムストア表を確認する」を参照してください。

TABLEINFO (i)：更新行のカラム化機能の対象となる i 番目のカラムストア表の情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$TABLEINFO(i) = \uparrow (232 \times column_num + 1,092 \times index_num + 2,504) \div 1,024 \uparrow + \max_{1 \leqq k \leqq chunk_num} SEGMENTLIST(k)$$

column_num

更新行のカラム化機能の対象となる i 番目のカラムストア表の列数

index_num

更新行のカラム化機能の対象となる i 番目のカラムストア表に定義されたレンジインデクス数

chunk_num

更新行のカラム化機能の対象となる i 番目のカラムストア表のチャンク数

シングルチャンク表の場合は 1 としてください。

SEGMENTLIST (k)：k 番目のチャンクのセグメントリスト情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$SEGMENTLIST(k) = \uparrow 24 \times segment_num \div 1,024 \uparrow$$

segment_num

k 番目のチャンクのローデータ用セグメント数

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある次の変数を参照して求めてください。

■マルチチャンク表の場合

「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」の変数 *ROWDATASEGNUM（i, k）*

■シングルチャンク表の場合

「(b)　変数 SGCOLUMNTBL の求め方（シングルチャンク表の場合）」の変数 *ROWDATASEGNUM（i）*

RELOCATE_DAT（i）：更新行のカラム化機能の対象となる i 番目のカラムストア表の再配置用作業領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RELOCATE_DAT(i)＝
  ↑（max_rowsz×2＋1,768＋16×col_num
    ＋↓（96＋216※＋544＋（652＋page_size）×col_num
          ＋（80＋page_size）×SEGSIZE＋7）÷8↓×8
    ＋432＋128×col_num
    ＋↓（2×col_num＋7）÷8↓×8
    ＋（1,736＋16×col_num）×col_num＋max_rowsz
  ）÷1,024↑
＋16,384＋1,024×col_num
```

注※

更新行のカラム化機能の対象となる i 番目のカラムストア表にレンジインデクスが定義されているときに加算してください。

max_rowsz

最大行長（単位：バイト）

最大行長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の行長 *ROWSZ* の計算式を基に求めてください。

page_size

更新行のカラム化機能の対象となる i 番目のカラムストア表が格納されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」を参照して求めてください。

col_num

更新行のカラム化機能の対象となる i 番目のカラムストア表の列数

SEGSIZE

更新行のカラム化機能の対象となるi番目のカラムストア表が格納されているデータ用DBエリアのセグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

SEGSIZE＝1セグメントの容量（単位：バイト）÷*page_size*

RTHD_DATACOMPRESS(i)：更新行のカラム化機能の対象となるi番目のカラムストア表のデータ圧縮用作業領域

「6.3.6　adbimportコマンド実行時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（adbimportコマンド実行時）」の「(d)　変数RTHD_DATALOADの求め方」の変数*RTHD_DATACOMPRESS*を参照して求めてください。

(3)　ヒープメモリの所要量の求め方（通常運用時）

通常運用時のヒープメモリ（*HEAP_EXECSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

HEAP_EXECSZ = *max_sql*(*UTHD_WORKSZ*)×*rthd_num*

変数の説明

rthd_num

サーバ定義adb_sys_rthd_numオペランドの指定値

max_sql( )

実行する各SQL文について，括弧内の変数の計算結果を求めてください。その中で最も大きい値を代入してください。

UTHD_WORKSZ：擬似スレッド用作業領域

次に示す計算式から求めてください。なお，次の条件に該当する場合は，0を代入してください。

- 実行するSQL文がSELECT文でない

計算式（単位：キロバイト）

UTHD_WORKSZ＝　↑*QE*×*uthd_num*÷2↑

uthd_num

サーバ定義adb_sys_uthd_numオペランドの指定値

QE

問合せ式の問合せ式本体に対して，変数*QB*の規則を適用して求められた値

QB

問合せ式本体に集合演算子を含むかどうかによって，求め方が異なります。

■問合せ式本体に集合演算子を含まない場合

問合せ式本体の問合せ指定に対して，変数 *QS* の規則を適用して求められた値

■問合せ式本体に集合演算子を含む場合

次の計算式から求めてください。

```
QB＝ MAX(ST1, ... , STi)＋setop_num
```

ST1, … , STi

集合演算項の各問合せ指定に対して，変数 *QS* の規則を適用して求められた値

setop_num

問合せ式本体に指定された集合演算の数

QS

問合せ式本体に副問合せを含むかどうかによって，求め方が異なります。

■問合せ指定に副問合せを含まない場合

次の計算式から求めてください。

```
QS＝ scan_num＋1
```

■問合せ指定に副問合せを含む場合

次の計算式から求めてください。

```
QS＝ MAX(SQ1, ... , SQi)＋scan_num＋1
```

SQ1, … , SQi

問合せ指定に指定された各副問合せに対して，変数 *QB* の規則を適用して求められた値

scan_num

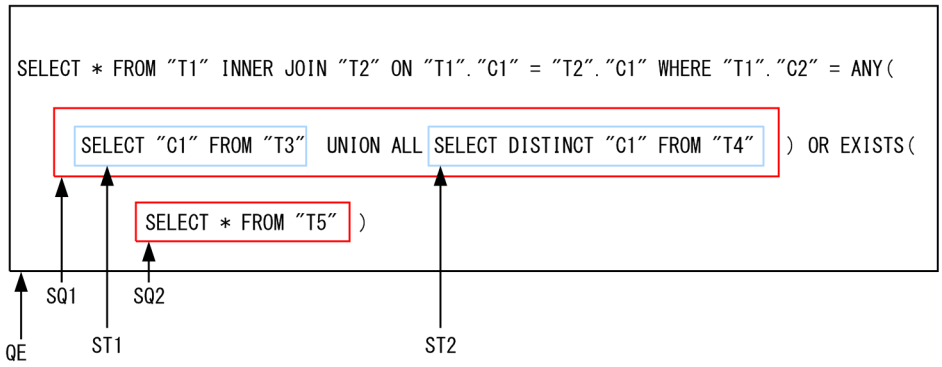
問合せ指定に指定された次の個数を合計した値

- GROUP BY 句
- SELECT DISTINCT
- DISTINCT 集合関数
- 逆分布関数
- LISTAGG 集合関数
- ARRAY_AGG 集合関数
- ウィンドウ関数
- 導出表の数
- 結合表の数（FULL OUTER JOIN の場合は，1 つ当たり 3 とカウントしてください）
- *結合表でない表参照の数*－1
- *アーカイブマルチチャンク表の数*×3

変数 *QE* の求め方の例を次に示します。

■変数 *QE* の求め方の例

■実行するSQL文



■計算式

QEは次の計算式から求めてください。QBは下記の1.～4.から求めてください。

QE＝ QB

1. QBは次の計算式から求めてください。
（副問合せ2つ，scan_num＝1）

QB＝ MAX(SQ1 ,SQ2)＋1＋1

2. SQ1は次の計算式から求めてください。
（集合演算項2つ，setop_num＝1）

SQ1＝ MAX(ST1 ,ST2)＋1

ST1＝ 1（副問合せなし，scan_num＝0）
ST2＝ 2（副問合せなし，scan_num＝1）

上記の計算結果から，SQ1の値は「3」になります。

SQ1＝ MAX(1 ,2)＋1 ＝ 3

3. SQ2は次の計算式から求めてください。
（副問合せなし，scan_num＝0）

SQ2＝ 1

4. 1.～3の計算式から，QBの値は「5」になります。

QB＝ MAX(3 ,1)＋2 ＝ 5

上記の計算結果から，QEは「5」になります。

ビュー表または問合せ名を指定した場合の注意事項を次に示します。

■ビュー表または問合せ名を指定した場合の注意事項

実行するSQL文中にビュー表または問合せ名を指定した場合は，それぞれに対応する内部的な導出表を適用したものとして見積もってください。例を次に示します。

例：ビュー定義

```
CREATE VIEW ”V1” AS SELECT * FROM ”T1”,”T2” WHERE ”T1”.”C1”=”T2”.”C2”
```

例：実行するSQL文

```
SELECT * FROM ”V1”,”T3” WHERE ”V1”.”C1”=”T3”.”C3”
```

上記のようなビュー定義と実行するSQL文の場合，見積もり対象となるSQL文は，次のとおりです。

見積もり対象となる SQL 文

```
SELECT * FROM (SELECT * FROM "T1","T2" WHERE "T1"."C1"="T2"."C2"),"T3"
         WHERE "V1"."C1"="T3"."C3"
```

6.3.5　adbinit コマンド実行時のメモリ所要量の求め方

adbinit コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- 共有メモリ管理（*INIT_CMN*）

■プロセスメモリ

- ヒープメモリ（*INIT_MEM*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　共有メモリ管理の所要量の求め方（adbinit コマンド実行時）

adbinit コマンド実行時の共有メモリ管理（*INIT_CMN*）の所要量は，変数 *SHMMAN* と同じ値になります。変数 *SHMMAN* については，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(1)　共有メモリ管理の所要量の求め方（HADB サーバ開始時）」を参照してください。

(2)　ヒープメモリの所要量の求め方（adbinit コマンド実行時）

adbinit コマンド実行時のヒープメモリ（*INIT_MEM*）の所要量は，次に示す計算式から求めてください。

計算式（単位：メガバイト）

INIT_MEM＝ ↑（16,384＋*DBAINF*＋*IOCTL*）÷1,024↑＋557

変数の説明

DBAINF：DB エリア情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

DBAINF＝ 1,024×*dbarea_defnum*

dbarea_defnum：初期設定オプション adbinitdbarea の指定数

IOCTL：DB エリアファイル出力制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

IOCTL ＝ 8,201×*multi_max*

multi_max：初期設定オプションadb_init_multi_max の指定値

6.3.6 adbimport コマンド実行時のメモリ所要量の求め方

adbimport コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_IMPORTSZ*）
- リアルスレッド固有メモリ（*RTHD_IMPORTSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_IMPORTSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1) プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）

adbimport コマンド実行時のプロセス共通メモリ（*PROC_IMPORTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IMPORTSZ＝
  Σ（PROC_IMPT＋PROC_IDX※1＋PROC_IDXBUILD※1
    ＋PROC_RNGIDX※2＋PROC_INFILE※3＋PROC_DBUPDINF＋PROC_RTHDUPDINF）※4
```

変数の説明

- *PROC_IMPT*
  データインポート処理で使用するプロセス共通メモリ
  「(a) 変数 PROC_IMPT の求め方」を参照して求めてください。
- *PROC_IDX*
  B-tree インデクスを作成するときに使用するプロセス共通メモリ
  「(b) 変数 PROC_IDX の求め方」を参照して求めてください。
- *PROC_IDXBUILD*
  B-tree インデクスを作成するときに使用するプロセス共通メモリ
  「(c) 変数 PROC_IDXBUILD の求め方」を参照して求めてください。
- *PROC_RNGIDX*

レンジインデクスを作成するときに使用するプロセス共通メモリ
「(d)　変数 PROC_RNGIDX の求め方」を参照して求めてください。

- *PROC_INFILE*
  入力データファイル数が 1,024 個を超える場合に使用するプロセス共通メモリ
  「(e)　変数 PROC_INFILE の求め方」を参照して求めてください。
- *PROC_DBUPDINF*
  DB エリア，表，インデクス，およびチャンク更新情報の格納で必要なプロセス共通メモリ
  「(f)　変数 PROC_DBUPDINF の求め方」を参照して求めてください。
- *PROC_RTHDUPDINF*
  処理リアルスレッド更新情報
  「(g)　変数 PROC_RTHDUPDINF の求め方」を参照して求めてください。

注※1

B-tree インデクスを定義した表にデータインポートする場合に加算します。

注※2

レンジインデクスを定義した表にデータインポートする場合に加算します。

注※3

入力データパスファイル名に指定した入力データファイル数が 1,024 個を超える場合に加算します。

注※4

複数のadbimport コマンドを同時に実行する場合は，各adbimport コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

(a)　変数 PROC_IMPT の求め方

変数 *PROC_IMPT* は，データインポート処理で使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IMPT＝ 3＋135＋DIC＋IOA＋LOD＋IMPORTBUF_CTL
            ＋STS＋PAGEALLOC＋PROC_AUDINFSZ※1＋PROC_IMPT_COLUMN※2
```

注※1

監査証跡機能が有効な場合に加算してください。

注※2

カラムストア表にデータインポートする場合に加算してください。

変数の説明

DIC：インポート対象表の定義情報を取得するためのメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
DIC＝ ↑（1,012＋（512＋32,008※1）×col_num＋max_rowsz＋1,156×idx_num＋15,864※2）÷1,024↑
```

注※1

コスト情報を収集したことがある表の場合に加算してください。

コスト情報を収集したことがある表かどうかを確認する方法については，「付録 C.9　システム表の検索」の「(1)　コスト情報を収集した全実表の表名と収集日時を調べる場合」を参照してください。

注※2

処理対象表がアーカイブマルチチャンク表の場合に加算してください。

col_num

インポート対象表の列数

max_rowsz

最大行長（単位：バイト）

最大行長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の行長*ROWSZ* の計算式を基に求めてください。

idx_num

インポート対象表に定義されたインデクス数

IOA：表に格納するデータイメージを作成するためのメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$IOA＝ ↑（max_rowsz＋436＋448×col_num＋124×idx_num＋4×dba_num）÷1,024↑×load_rthd$$

- *max_rowsz*

  最大行長（単位：バイト）

  最大行長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の行長*ROWSZ* の計算式を基に求めてください。

- *col_num*

  インポート対象表の列数

- *idx_num*

  インポート対象表に定義されたインデクス数

- *dba_num*

  インポート対象表のデータ用 DB エリアの DB エリアファイル数

- *load_rthd*

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

LOD：データの格納処理で必要となるメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
LOD=
  ↑（（（read_size×read_buff_num＋decomp_buff_size※1＋1）×1,024×load_rthd）
  ＋（（5,200＋1,024,000※2＋input_edit_area）×load_rthd）
  ＋（511×input_file_num））÷1,024↑
```

注※1

入力データファイルに圧縮ファイル（GZIP 形式）を指定したときに加算してください。

注※2

列構成情報ファイルを指定したときに加算してください。

read_size

インポートオプションadb_import_read_size の指定値

read_buff_num

3 を代入してください。

decomp_buff_size

インポートオプションadb_import_decompress_buff_size の指定値

load_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

input_file_num

1,024 を代入してください。

input_edit_area

入力データの編集領域長

次に示す計算式から求めてください。

計算式

```
input_edit_area=
    MIN（ MAX(↑input_recsize÷1,024↑×1,024 , 32,768) , 536,870,912)
```

input_recsize

入力レコード長

- 固定長データ形式の場合

  インポートオプションadb_import_input_record_size の指定値を代入してください。

- CSV 形式の場合

  次に示す計算式から求めてください。

  計算式

```
input_recsize＝ Σ（DATASIZE）＋フィールドデータ数＋1
```

フィールドデータ数

列構成情報ファイルに指定した各列のフィールドデータ番号，または指定を省略した際の仮定値を求めてください。その中で最大の値を代入してください。

Σ(*DATASIZE*)

処理対象表にある各列のデータ型のデータを文字形式で表した場合の最大長の合計

各データ型のデータを文字形式で表したときの最大長については，次に示す表から求めてください。

表 6-11　各データ型のデータを文字形式で表した場合の最大長の一覧

項番	分類	データ型	文字形式のデータ最大長
1	数データ	INTEGER	23
2		SMALLINT	13
3		DECIMAL(*m*,*n*)	*m* ＋ 5
4		NUMERIC(*m*,*n*)	
5		DOUBLE PRECISION	511
6		FLOAT	
7	文字データ	CHAR(*n*)	*n*×2 ＋ 2
8		VARCHAR(*n*)	*n*×2 ＋ 2
9	日時データ	DATE	12
10		TIME(*p*)	8 ＋*p* ＋ 3
11		TIMESTAMP(*p*)	19 ＋*p* ＋ 3
12	バイナリデータ	BINARY(*n*)	*n*×8 ＋ 2
13		VARBINARY(*n*)	*n*×8 ＋ 2
14	配列データ	ARRAY	*要素データ型の文字形式のデータ最大長*×*最大要素数*

（凡例）

m，*n*，*p*：マニュアル『HADB　SQL リファレンス』の『データ型の種類』を参照してください。

IMPORTBUF_CTL：データインポートおよびインデクス作成用バッファを制御するために使用するメモリ

次に示す値を代入してください。

値（単位：キロバイト）

IMPORTBUF_CTL＝ ↑（248＋*BUFBLK*）÷1,024↑×*load_rthd*

- *BUFBLK*

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(f)　変数 PROC_UPDSZ の求め方」の変数*BUFBLK* を参照してください。

- *load_rthd*

  次に示す計算式から求めてください。

  インポートオプションadb_import_rthd_numの指定値－1

STS：ステータス情報を退避するためのメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STS=
  ↑（256＋（1,185×idx_num）＋（2,067×（b_tree_idx_num＋range_idx_num）×load_rthd
    ＋2,067×text_idx_num×16）＋1,025）÷1,024↑
```

idx_num

インポート対象表に定義されたインデクス数

b_tree_idx_num

インポート対象表に定義された B-tree インデクスの数

range_idx_num

インポート対象表に定義されたレンジインデクスの数

load_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

text_idx_num

インポート対象表に定義されたテキストインデクスの数

PAGEALLOC：データインポート用ページ割り当て制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

PAGEALLOC＝ *import_rthd*×310

- *import_rthd*

  次に示す計算式から求めてください。

  インポートオプションadb_import_rthd_numの指定値－1

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照してください。

PROC_IMPT_COLUMN：カラムストア表へのデータインポートで使用するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IMPT_COLUMN＝
          ↑（（16,128＋8×col_num）×rthd_num
         ＋512×（infile_num＋workfile_num））÷1,024↑
```

col_num

インポート対象表の列数

rthd_num

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

infile_num

入力データファイルの数

workfile_num

作業用一時ファイルの格納先ディレクトリの数

(b)　変数 PROC_IDX の求め方

変数*PROC_IDX* は，インデクスを作成する場合に使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDX＝
   ↑（idxf_buff_size×1,024＋（13,151＋（33,928＋（13×KEYSZ））
     ×idx_num×load_rthd）×cmd_d_opt
     ＋（152＋rd_buff_size×load_rthd×2×1,024＋2×ld_buff_size×1,024
     ＋（KEYSZ＋CTRL）×text_idx_num）×dividx_rthd）÷1,024↑
```

変数の説明

idxf_buff_size

1,024 を代入してください。

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

idx_num

処理対象表に定義された B-tree インデクスの数

load_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

cmd_d_opt

adbimport コマンドのオプションの指定有無，およびインポート対象表の状態によって，変数*cmd_d_opt* の値は異なります。次に示す表に従って，変数*cmd_d_opt* の値を求めてください。

項番	adbimport コマンドのオプションの指定有無，およびインポート対象表の状態	cmd_d_opt の値
1	• adbimport コマンドに-d オプションを指定し，-b オプションを指定しないで実行するとき • 行を格納するセグメントが割り当てられていない状態のインポート対象表に対して，adbimport コマンドに-d オプションと-b オプションを指定しないで実行するとき • マルチチャンク表に対して，adbimport コマンドに-b オプションを指定し，-d オプションを指定しないで実行するとき	1
2	行を格納するセグメントが割り当てられている状態のインポート対象表に対して，adbimport コマンドに-d オプションと-b オプションを指定しないで実行するとき	3

rd_buff_size

インポートオプションadb_import_dividx_rd_buff_size の指定値

ld_buff_size

インポートオプションadb_import_dividx_wt_buff_size の指定値

CTRL

- インデクスのインデクス構成列が固定長の場合
  10 バイトを代入してください。
- インデクスのインデクス構成列に可変長を含む場合
  12 バイトを代入してください。

text_idx_num

処理対象表に定義されたテキストインデクスの数

dividx_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

(c) 変数 PROC_IDXBUILD の求め方

変数*PROC_IDXBUILD* は，B-tree インデクスを作成する場合に使用するプロセス共通メモリの所要量を求めるときに使用します。

PROC_IDXBUILD の値は，adbimport コマンドの-d オプションと-b オプションの指定有無，およびインポート対象表の状態によって，次に示すどちらかになります。

PROC_IDXBUILD の値（単位：キロバイト）

PROC_IDXBUILD＝ *PROC_IDXCREATE*，または*PROC_IDXCREATE*＋*PROC_IDXMAINT*

■*PROC_IDXBUILD* の値が *PROC_IDXCREATE* になる場合

次に示すどれかに該当するときは，*PROC_IDXCREATE* になります。

- adbimport コマンドに-d オプションを指定し，-b オプションを指定しないで実行するとき
- 行を格納するセグメントが割り当てられていない状態のインポート対象表に対して，adbimport コマンドに-d オプションと-b オプションを指定しないで実行するとき
- マルチチャンク表に対して，adbimport コマンドに-b オプションを指定し，-d オプションを指定しないで実行するとき

■*PROC_IDXBUILD* の値が *PROC_IDXCREATE* ＋ *PROC_IDXMAINT* になる場合

次に該当するときは，*PROC_IDXCREATE* ＋ *PROC_IDXMAINT* になります。

- 行を格納するセグメントが割り当てられている状態のインポート対象表に対して，adbimport コマンドに-d オプションと-b オプションを指定しないで実行するとき

PROC_IDXCREATE，および*PROC_IDXMAINT* は次に示す計算式から求めてください。

変数の説明

PROC_IDXCREATE：次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXCREATE = \max_{1 \leqq k \leqq idx_num} PROC_IDXCREATE_MEM(k)$$

idx_num

インポート対象表に定義された B-tree インデクスの数

PROC_IDXCREATE_MEM（k）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_IDXCREATE_MEM（k） ＝
↑（456＋256×*idx_col_num*＋（24＋144×↑*idx_lv*÷16↑）
×（*dividx_rthd*＋1）＋（*KEYSZ*＋（*idx_col_num*＋1）×2＋12）
×2＋*page_size*×2＋268）÷1,024↑

idx_col_num

B-tree インデクスのインデクス構成列の数

idx_lv：B-tree インデクスの段数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』の漸化式を計算して，*PIDX（n）* ＝ 1 となったときの *n* の値を代入してください。

ただし，*PIDX（1）* ＝ 1 の場合は，B-tree インデクスの段数は 2 としてください。

dividx_rthd

次に示す計算式から求めてください。

インポートオプション adb_import_rthd_num の指定値 －1

KEYSZ：B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

PROC_IDXMAINT：次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXMAINT = \max_{1 \leqq k \leqq idx_num} PROC_IDXMAINT_MEM(k)$$

idx_num

インポート対象表に定義された B-tree インデクスの数

PROC_IDXMAINT_MEM（k）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$\begin{aligned}
&PROC_IDXMAINT_MEM\ (k)\ = \\
&\quad \uparrow\ (16 + (8 + KEYSZ + idx_col_num + 1) \times (dividx_rthd - 1) \\
&\quad + 528 + 144 \times idx_col_num \\
&\quad + \uparrow\ (KEYSZ + (idx_col_num + 1) \times 2 + 12) \div 8 \uparrow \times 8 \\
&\quad + (176 + 48 \times (idx_lv - 1) + 32 \times (idx_lv + 1) \\
&\quad + page_size \times 2 \\
&\quad + \uparrow\ (KEYSZ + (idx_col_num + 1) \times 2 + 3{,}068) \div 8 \uparrow \times 8 \\
&\quad + \uparrow\ (KEYSZ + (idx_col_num + 1) \times 2 + 4) \div 8 \uparrow \times 16 \\
&\quad + \uparrow\ (KEYSZ + (idx_col_num + 1) \times 2 + 12) \div 8 \uparrow \times 8) \\
&\quad \times dividx_rthd) \div 1{,}024 \uparrow
\end{aligned}$$

KEYSZ：B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

idx_col_num

B-tree インデクスのインデクス構成列の数

dividx_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

idx_lv：B-tree インデクスの段数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』の漸化式を計算して，*PIDX（n）* ＝ 1 となったときの *n* の値を代入してください。

ただし，*PIDX（1）* ＝ 1 の場合は，B-tree インデクスの段数は 2 としてください。

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

(d)　変数 PROC_RNGIDX の求め方

変数 *PROC_RNGIDX* は，レンジインデクスを作成する場合に使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_RNGIDX＝　↑64×*rngidx_num*÷1,024↑

変数の説明

rngidx_num：インポート対象表に定義されたレンジインデクスの数

(e)　変数 PROC_INFILE の求め方

変数 *PROC_INFILE* は，入力データパスファイル名に指定した入力データファイル数が 1,024 個を超える場合に加算する必要があります。次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_INFILE＝　↑（（16＋511）×（*INFMAX*-1,024））÷1,024↑

変数の説明

INFMAX：入力データパスファイル名に指定した入力データファイル数

(f) 変数 PROC_DBUPDINF の求め方

変数 *PROC_DBUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
      ↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×131,136）÷1,024）↑
```

変数の説明

DBUPDINF_ENT_NUM

DB エリア，表，インデクス，およびチャンク更新情報のエントリ数

次に示す計算式から求めてください。なお，マルチノード機能を使用するかどうかで，計算式が異なります。

マルチノード機能を使用しない場合の計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝  1＋idx_num＋rngidx_num＋txtidx_num
```

マルチノード機能を使用する場合の計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝
  （1＋idx_num＋rngidx_num＋txtidx_num）×2＋idx_num×6＋rngidx_num×5＋txtidx_num×2＋48
```

idx_num

処理対象表に定義している B-tree インデクス数

rngidx_num

処理対象表に定義しているレンジインデクス数

txtidx_num

処理対象表に定義しているテキストインデクス数

(g) 変数 PROC_RTHDUPDINF の求め方

変数 *PROC_RTHDUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_RTHDUPDINF＝ ↑（648＋（168×rthd_num））÷1,024↑
```

変数の説明

rthd_num

処理リアルスレッド数

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

(2) リアルスレッド固有メモリの所要量の求め方（adbimport コマンド実行時）

adbimport コマンド実行時のリアルスレッド固有メモリ（*RTHD_IMPORTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IMPORTSZ＝
        IMPORTBUF＋SORTIOBUF※1＋SORTBUF※1
        ＋RTHD_DATALOAD＋RTHD_IDXREC※1＋RTHD_IDXBUILD※1
        ＋RTHD_RNGIDX※2＋RTHD_TXTIDX※3
```

変数の説明

- *IMPORTBUF*

  データインポート処理時およびインデクス作成時に使用するバッファ

  「(a)　変数 IMPORTBUF の求め方」を参照して求めてください。

- *SORTIOBUF*

  ソート処理 I/O バッファ

  「(b)　変数 SORTIOBUF の求め方」を参照して求めてください。

- *SORTBUF*

  インデクス作成用ソートバッファ

  「(c)　変数 SORTBUF の求め方」を参照して求めてください。

- *RTHD_DATALOAD*

  データインポート処理時に使用するバッファ

  「(d)　変数 RTHD_DATALOAD の求め方」を参照して求めてください。

- *RTHD_IDXREC*

  B-tree インデクス作成時に使用するバッファ

  「(e)　変数 RTHD_IDXREC の求め方」を参照して求めてください。

- *RTHD_IDXBUILD*

  B-tree インデクス作成時に使用するバッファ

  「(f)　変数 RTHD_IDXBUILD の求め方」を参照して求めてください。

- *RTHD_RNGIDX*

レンジインデクス作成時に使用するバッファ

「(g)　変数 RTHD_RNGIDX の求め方」を参照して求めてください。

- *RTHD_TXTIDX*

  テキストインデクス作成時に使用するバッファ

  「(h)　変数 RTHD_TXTIDX の求め方」を参照して求めてください。

注※1

B-tree インデクスを定義した表にデータインポートする場合に加算します。

注※2

レンジインデクスを定義した表にデータインポートする場合に加算します。

注※3

テキストインデクスを定義した表にデータインポートする場合に加算します。

(a)　変数 IMPORTBUF の求め方

変数 *IMPORTBUF* は，データインポート処理時，およびインデクスの作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IMPORTBUF＝
      ↑（IMP_SQBLK＋IMP_SQIO＋IMP_SQPGE＋IMP_SQHS）÷1,024↑
```

変数の説明

IMP_SQBLK：次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMP_SQBLK＝
   imp_blknum×（112＋↑IMP_BLKSZ÷imp_pagesize÷64↑×24）
```

imp_blknum：

インポートオプションadb_import_buff_blk_num の指定値

IMP_BLKSZ：次に示す計算式から求めてください。

```
IMP_BLKSZ＝ 4,096×1,024
```

imp_pagesize：

次の DB エリアのページサイズのうち，最小の値（単位：バイト）

- 処理対象表が格納されている DB エリア
- 処理対象表に定義されている B-tree インデクスまたはテキストインデクスが格納されている DB エリア

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

IMP_SQIO：次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_SQIO＝ 568×*imp_blknum*

imp_blknum：

インポートオプションadb_import_buff_blk_num の指定値

IMP_SQPGE：次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_SQPGE＝
（176＋*imp_pagesize*＋*BUFLOG*）×（*IMP_BLKSZ*÷*imp_pagesize*）
×*imp_blknum*

imp_pagesize：

次の DB エリアのページサイズのうち，最小の値（単位：バイト）

- 処理対象表が格納されている DB エリア
- 処理対象表に定義されている B-tree インデクスまたはテキストインデクスが格納されている DB エリア

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

BUFLOG：

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFLOG* を参照してください。

IMP_BLKSZ：次に示す計算式から求めてください。

IMP_BLKSZ＝ 4,096×1,024

imp_blknum：

インポートオプションadb_import_buff_blk_num の指定値

IMP_SQHS：次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_SQHS＝ （8×*imp_blknum*）＋（40×*imp_blknum*）

imp_blknum：

インポートオプションadb_import_buff_blk_num の指定値

(b) 変数 SORTIOBUF の求め方

変数*SORTIOBUF* は，ソート処理時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

SORTIOBUF ＝ *sort_io_buff_size*

変数の説明

sort_io_buff_size

2,048 を代入してください。

(c) 変数 SORTBUF の求め方

変数*SORTBUF* は，ソート処理時のインデクス作成で使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

SORTBUF ＝ 48＋*sort_buff_size* ×1,024

変数の説明

sort_buff_size

インポートオプションadb_import_sort_buff_size の指定値※

注※

ソート処理では，作業用ワークファイルが作成されます。このファイルの容量を最小にする変数*sort_buff_size* の計算式を次に示します。変数*sort_buff_size* は，次に示す計算式を満たすようにしてください。ただし，メモリに余裕がない場合，またはほとんどのデータがインデクスキー順に並んでいる場合は，必要以上に大きな値を指定しないでください。

計算式

$$\text{sort_buff_size} \geqq \left\lceil \left(\frac{\text{REC_SIZE}+7}{2} + \sqrt{(\text{BLK}+8) \times \text{row_num} \times \text{KEY_INF} + \frac{\text{REC_SIZE}+7}{4}} + \text{COL_INF0} \right) \div 1{,}024 \right\rceil$$

変数の説明

REC_SIZE：次に示す計算式から求めてください。

計算式

REC_SIZE ＝ （*KEYSZ*＋*SYS1*）＋*SYS2*

KEYSZ：B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

SYS1：次に示す値を仮定してください。

- インポート対象表に定義されたインデクスのインデクス構成列が固定長の場合：10
- インポート対象表に定義されたインデクスのインデクス構成列に可変長を含む場合：12

SYS2：次に示す値を仮定してください。

- インポート対象表に定義されたインデクスが可変長の複数列インデクスの場合：（*インデクス構成列数*×4）
- そのほかのインデクスの場合：0

BLK：次に示す計算式から求めてください。

計算式

BLK＝ *REC_SIZE*＋（*KEYSZ*＋8）＋56

row_num：

adbimport コマンドで表に格納する行数

KEY_INF：次に示す計算式から求めてください。

計算式

KEY_INF＝ *REC_SIZE*＋（*KEYSZ*＋8）＋28

COL_INFO：次に示す計算式から求めてください。

計算式

COL_INFO＝ 2,112＋（*MULTI_KEY_INFO*×32）＋（*KEYSZ*＋8）

MULTI_KEY_INFO：次に示す値を仮定してください。

- インポート対象表に定義された B-tree インデクスが可変長の複数列インデクスの場合：（*インデクス構成列数*×2）＋ 2
- そのほかの場合：5

(d)　変数 RTHD_DATALOAD の求め方

変数 *RTHD_DATALOAD* は，データインポート処理時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

なお，インポート対象表がローストア表の場合と，カラムストア表の場合で計算式が異なります。

■ローストア表の場合

計算式（単位：キロバイト）

```
RTHD_DATALOAD＝
 ↑(↓(80＋192※1＋8×(col_num＋var_col_num)＋32×var_col_num※2＋7)÷8↓×8＋513※3)÷1,024↑
```

注※1

インポート対象表にレンジインデクスが定義されている場合に加算してください。

注※2

インポート対象表が非 FIX 表の場合に加算してください。

注※3

アーカイブマルチチャンク表に対して，-d オプションを指定してadbimport コマンドを実行する場合に加算してください。

変数の説明

col_num

インポート対象表の列数（単位：個）

var_col_num

インポート対象表にあるVARCHAR 型，およびVARBINARY 型の列数（単位：個）

■カラムストア表の場合

計算式（単位：キロバイト）

$$\mathrm{RTHD_DATALOAD} = \left\lceil \frac{\left(\left\lfloor \frac{80 + 192^{※} + 544 + (612 + page_size \times 2) \times col_num + (80 + page_size) \times SEGSIZE + 7}{8} \right\rfloor \times 8 \right) \times 6}{1,024} \right\rceil + IMP_COLUMN + RTHD_DATACOMPRESS$$

注※

インポート対象表にレンジインデクスが定義されている場合に加算してください。

変数の説明

page_size

インポート対象表を格納する DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」を参照してください。

col_num

インポート対象表の列数（単位：個）

SEGSIZE

DB エリアファイルを定義した DB エリアのセグメントサイズ

次に示す計算式から求めてください。

計算式（単位：ページ数）

SEGSIZE＝ *1セグメントの容量*（単位：バイト）÷*page_size*

IMP_COLUMN

カラムストア表へのデータインポート用作業領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IMP_COLUMN＝
    1,024×col_num
    ＋↑（（376＋16×col_num）×col_num＋1,824＋input_recsize＋8×col_num
    ＋MAX(input_recsize ，  2,097,152)＋IMP_ERRDATA※1＋IMP_ARRAY※2)
    ÷1,024↑
```

注※1

インポートオプションadb_import_errdata_file_name を指定し，かつ CSV 形式のデータをインポートする場合に加算してください。

注※2

配列型の列を定義している表にデータをインポートする場合に加算してください。

col_num

インポート対象表の列数（単位：個）

input_recsize

入力レコード長

「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*LOD* を参照してください。

IMP_ERRDATA

エラーレコード出力用バッファ

次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_ERRDATA＝*input_recsize*×1.2

input_recsize

入力レコード長

「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*LOD* を参照してください。

IMP_ARRAY

配列型の列にデータをインポートする際のデータインポート用作業領域

次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_ARRAY＝56×*arr_col_num*＋160×*col_num*

arr_col_num

インポート対象表に定義されている配列型の列数（単位：個）

col_num

インポート対象表の列数（単位：個）

RTHD_DATACOMPRESS

データ圧縮用の作業領域

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$\mathrm{RTHD_DATACOMPRESS} = \left\lceil \frac{\sum_{i=1}^{col_num_in_table} col_compress_work_size(i)}{1,024} \right\rceil + \left\lceil \frac{\sum_{j=1}^{arr_col_num} page_size + 4,076,000 + data_len(j) + 16 \times n(j) + 104}{1,024} \right\rceil$$

col_num_in_table

インポート対象表の構成列数（単位：個）

col_compress_work_size（i）

各列を圧縮するのに必要な作業領域（単位：バイト）

CREATE TABLE 文でインポート対象表を定義する際に，各列に指定した列データの圧縮方式によって，代入する値が異なります。次の表に示す列データの圧縮方式に対応する値を代入してください。列データの圧縮方式の指定については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE TABLE（表の定義)』を参照してください。

表 6-12　各列を圧縮するのに必要な作業領域（変数 col_compress_work_size に代入する値）

項番	列データの圧縮方式	各列を圧縮するのに必要な作業領域（変数 *col_compress_work_size* に代入する値）（単位：バイト）
1	NONE	65,552
2	RUNLENGTH	65,552
3	DICTIONARY	23,270,960
4	DELTA	65,552
5	DELTA_RUNLENGTH	65,552

項番	列データの圧縮方式	各列を圧縮するのに必要な作業領域（変数 *col_compress_work_size* に代入する値）（単位：バイト）
6	AUTO	23,270,960

arr_col_num

インポート対象表に定義されている配列型の列数（単位：個）

page_size

インポート対象表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」を参照してください。

data_len（j）

j 番目の配列型の列のデータ長（単位：バイト）

配列型の列のデータ長については，「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」にある「表 6-9　各データ型のデータ長の一覧」の「配列データ」を参照して求めてください。

n（j）

j 番目の配列型の列の最大要素数（単位：個）

(e)　変数 RTHD_IDXREC の求め方

変数 *RTHD_IDXREC* は，B-tree インデクス作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXREC＝
    ↑（386＋328×（idx_num-1）＋32×max_idx_col_num
    ＋（64＋buf_size×1,024）×2×idx_num）÷1,024↑
```

変数の説明

- *idx_num*

  インポート対象表に定義された B-tree インデクスの数

- *max_idx_col_num*

  インポート対象表に定義された B-tree インデクスのインデクス構成列数の最大値

- *buf_size*

  1,024 を代入してください。

(f)　変数 RTHD_IDXBUILD の求め方

変数*RTHD_IDXBUILD* は，B-tree インデクス作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。変数*RTHD_IDXBUILD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXBUILD = \max_{1 \leqq k \leqq idx_num} RTHD_IDXBUILD_MEM(k)$$

変数の説明

idx_num

インポート対象表に定義された B-tree インデクスの数

RTHD_IDXBUILD_MEM（*k*）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXBUILD_MEM(k) = \uparrow (61,474 + \uparrow MAX(255, KEYSZ) \div 2 \uparrow \times 2 + 3,070 + \uparrow KEYSZ + CTRL \div 8 \uparrow \times 8 + MAX(255, KEYSZ) + 14 + \downarrow 8 + page_size \times 95 \div 100 \downarrow \times 2) \div 1,024 \uparrow$$

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL

- インデクス構成列のキー長がすべて固定長の場合
  10 バイトを代入してください。
- インデクス構成列のキー長に可変長を含む場合
  12 バイトを代入してください。

(g)　変数 RTHD_RNGIDX の求め方

変数*RTHD_RNGIDX* は，レンジインデクス作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_RNGIDX＝ ↑（16＋128×*rngidx_num*）×*load_rthd*÷1,024↑

変数の説明

rngidx_num

インポート対象表に定義されたレンジインデクスの数

load_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

(h) 変数 RTHD_TXTIDX の求め方

変数*RTHD_TXTIDX* は，テキストインデクス作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_TXTIDX = \max_{1 \leqq k \leqq idx_num} RTHD_TXTIDXBUILD_MEM(k)$$

変数の説明

idx_num

処理対象表に定義されたテキストインデクスの数

RTHD_TXTIDXBUILD_MEM(*k*)

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_TXTIDXBUILD_MEM(k)＝
buff_blk_size×buff_blk_num＋3,072＋↑(idx_div_num×24)÷1,024↑
＋txt_sort_buff_size×1,024

buff_blk_size

4,096 を代入してください。

buff_blk_num

インポートオプションadb_import_buff_blk_num の指定値

idx_div_num

k 番目のテキストインデクスのインデクス構成列定義長

txt_sort_buff_size

インポートオプションadb_import_txt_buff_size の指定値

(3) ヒープメモリの所要量の求め方（adbimport コマンド実行時）

adbimport コマンド実行時のヒープメモリ（*HEAP_IMPORTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_IMPORTSZ＝ ∑(HEAP_ZLIB＋HEAP_SRTHMEM)※
```

注※

複数のadbimport コマンドを同時に実行する場合は，各adbimport コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

変数の説明

HEAP_ZLIB

「(a) 変数 HEAP_ZLIB の求め方」を参照してください。

HEAP_SRTHMEM

「(b) 変数 HEAP_SRTHMEM の求め方」を参照してください。

(a) 変数 HEAP_ZLIB の求め方

変数*HEAP_ZLIB* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_ZLIB＝ 42×load_rthd
```

変数の説明

load_rthd

次に示す計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値－1

(b) 変数 HEAP_SRTHMEM の求め方

変数*HEAP_SRTHMEM* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_SRTHMEM＝ 505×sort_rthd
```

変数の説明

sort_rthd

次に示す計算式から求めてください。

```
インポートオプションadb_import_rthd_numの指定値－1
```

6.3.7　adbidxrebuild コマンド実行時のメモリ所要量の求め方

adbidxrebuild コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_IDXRBLDSZ*）
- リアルスレッド固有メモリ（*RTHD_IDXRBLDSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_IDXRBLDSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（adbidxrebuild コマンド実行時）

adbidxrebuild コマンド実行時のプロセス共通メモリ（*PROC_IDXRBLDSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDXRBLDSZ＝
    Σ（PROC_MNG＋PROC_IDX※1＋PROC_IDXREBUILD※1＋PROC_RNGIDX※2
      ＋PROC_DBUPDINF※3＋PROC_RTHDUPDINF）※4
```

変数の説明

- *PROC_MNG*
  インデクス再作成の管理で使用するプロセス共通メモリ
  「(a)　変数 PROC_MNG の求め方」を参照して求めてください。
- *PROC_IDX*
  インデクス一括作成で使用するプロセス共通メモリ
  「(b)　変数 PROC_IDX の求め方」を参照して求めてください。
- *PROC_IDXREBUILD*

B-tree インデクスの再作成で使用するプロセス共通メモリ

「(c)　変数 PROC_IDXREBUILD の求め方」を参照して求めてください。

- *PROC_RNGIDX*

  レンジインデクスの一括更新で使用するプロセス共通メモリ

  「(d)　変数 PROC_RNGIDX の求め方」を参照して求めてください。

- *PROC_DBUPDINF*

  DB エリア，表，インデクス，およびチャンク更新情報の格納で必要なプロセス共通メモリ

  「(e)　変数 PROC_DBUPDINF の求め方」を参照して求めてください。

- *PROC_RTHDUPDINF*

  処理リアルスレッドの更新情報の管理で必要なプロセス共通メモリ

  「(f)　変数 PROC_RTHDUPDINF の求め方」を参照して求めてください。

注※1

処理対象インデクスに B-tree インデクスが含まれる場合に加算してください。

注※2

処理対象インデクスにレンジインデクスが含まれる場合に加算してください。

注※3

マルチノード機能を使用する場合に加算してください。

注※4

複数のadbidxrebuild コマンドを同時に実行する場合は，各adbidxrebuild コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

(a)　変数 PROC_MNG の求め方

変数 *PROC_MNG* は，インデクス再作成の管理で使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_MNG＝
　3＋*DIC*＋*IOA*＋*SCAN*＋*IDXRBLDBUF_CTL*＋*STS*＋*PAGEALLOC*＋*PROC_AUDINFSZ*※

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

DIC：処理対象表の定義情報を取得するメモリ

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数 *DIC* を参照してください。

IOA：データベースアクセスで使用するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

IOA＝ 1,056＋（*scan_buff_size*＋*offset_area*×2）×*scan_rthd*

- *scan_buff_size*

  次に示す計算式から求めてください。

  インデクス再作成オプション adb_idxrebuild_scan_buff_size の指定値×1,024

- *offset_area*

  次に示す計算式から求めてください。

  （8×*処理対象インデクスの構成列数*×（*scan_buff_size*×1,024
  ÷（*処理対象インデクスの構成列長の合計値*＋8））÷1,024

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値*－1）÷2↓

SCAN：インデクスレコードファイルの出力処理で必要となるメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

SCAN＝
↑（*write_size*×*scan_rthd*×*b-tree_index_num*）↑

- *write_size*

  インデクス再作成オプションadb_idxrebuild_dvix_wtbuff_sizeの指定値

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値*－1）÷2↓

- *b-tree_index_num*

  処理対象の B-tree インデクス数

IDXRBLDBUF_CTL：インデクス再作成用バッファ制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

IDXRBLDBUF_CTL＝
↑（248＋*BUFBLK*）÷1,024↑×*scan_rthd*

- *BUFBLK*

「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(f)　変数 PROC_UPDSZ の求め方」の変数*BUFBLK* を参照してください。

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値*－1）÷2↓

STS：ステータス情報を退避するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STS=
  ↑（256＋（1,185×idx_num）＋（2,067×（b_tree_idx_num＋range_idx_num）×scan_rthd
    ＋2,067×text_idx_num×16）＋1,025）÷1,024↑＋imp_sts
```

idx_num

処理対象のインデクス数

b_tree_idx_num

処理対象表に定義された B-tree インデクスの数

range_idx_num

処理対象表に定義されたレンジインデクスの数

scan_rthd

次に示す計算式から求めてください。

↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値*－1）÷2↓

text_idx_num

処理対象表に定義されたテキストインデクスの数

imp_sts

adbimport コマンドのステータス情報を退避するメモリ

adbimport コマンドが中断したあとで，--create-temp-file オプションを指定してadbidxrebuild コマンドを実行する場合に，「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」にある変数*STS* で求めた値を加算してください。

PAGEALLOC：インデクス再作成用ページ割り当て制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

PAGEALLOC＝ *dividx_rthd*×310

- *dividx_rthd*

  次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照してください。

(b)　変数 PROC_IDX の求め方

変数*PROC_IDX* は，インデクス一括作成で使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDX＝
  ↑（（13,151＋（33,928＋（13×KEYSZ））×scan_rthd
  ＋（152＋rd_buff_size×scan_rthd×2×1,024＋2×ld_buff_size×1,024
  ＋（KEYSZ＋CTRL）×text_idx_num）×dividx_rthd）×（b_tree_idx_num＋text_idx_num））÷1,024↑
```

変数の説明

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

scan_rthd

次に示す計算式から求めてください。

↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値*－1）÷2↓

rd_buff_size

インデクス再作成オプションadb_idxrebuild_dvix_rdbuff_size の指定値

ld_buff_size

インデクス再作成オプションadb_idxrebuild_dvix_wtbuff_size の指定値

dividx_rthd

次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

CTRL

- インデクスのインデクス構成列がすべて固定長の場合
  10 バイトを代入してください。
- インデクスのインデクス構成列に可変長を含む場合
  12 バイトを代入してください。

b_tree_idx_num

処理対象の B-tree インデクス数

text_idx_num

処理対象表に定義されたテキストインデクスの数

(c)　変数 PROC_IDXREBUILD の求め方

変数 *PROC_IDXREBUILD* は，B-tree インデクスの再作成で使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXREBUILD = \max_{1 \leqq k \leqq idx_num} PROC_IDXREBUILD_MEM(k)$$

idx_num

再作成対象の B-tree インデクス数

PROC_IDXREBUILD_MEM（k）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXREBUILD_MEM(k) = \uparrow (184 + 256 \times idx_col_num + (18 + 256 \times \uparrow idx_lv \div 16 \uparrow) \times dividx_rthd + KEYSZ + (idx_col_num + 1) \times 2 + 4) \div 1{,}024 \uparrow$$

- *idx_col_num*

  B-tree インデクスのインデクス構成列数

- *idx_lv*

  B-tree インデクスの段数

  「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』の漸化式を計算して，*PIDX（n）* ＝ 1 となったときの *n* の値を代入してください。

  ただし，*PIDX（1）* ＝ 1 の場合は，B-tree インデクスの段数は 2 としてください。

- *dividx_rthd*

  次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

- *KEYSZ*：B-tree インデクスのキー長（単位：バイト）

  「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

  なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

(d)　変数 PROC_RNGIDX の求め方

変数*PROC_RNGIDX* は，レンジインデクスの一括更新で使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_RNGIDX＝　↑64×*rngidx_num*÷1,024↑

変数の説明

rngidx_num：再作成対象のレンジインデクス数

(e)　変数 PROC_DBUPDINF の求め方

変数*PROC_DBUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
    ↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024)↑×131,136）÷1,024）↑
```

変数の説明

DBUPDINF_ENT_NUM

DB エリア，表，インデクス，およびチャンク更新情報のエントリ数

次に示す計算式から求めてください。

計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝
    （idx_num＋（idx_num×2＋rngidx_num＋txtidx_num×2）×chunk_num）
   ＋idx_num×3＋（rngidx_num×3）＋15
```

idx_num

再作成対象の B-tree インデクス数

rngidx_num

再作成対象のレンジインデクス数

txtidx_num

再作成対象のテキストインデクス数

chunk_num

再作成対象のインデクスを定義している表のチャンク数

(f)　変数 PROC_RTHDUPDINF の求め方

変数 *PROC_RTHDUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_RTHDUPDINF＝
      ↑（648＋（168×rthd_num））÷1,024↑
```

変数の説明

rthd_num

処理リアルスレッド数

次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

(2)　リアルスレッド固有メモリの所要量の求め方（adbidxrebuild コマンド実行時）

adbidxrebuild コマンド実行時のリアルスレッド固有メモリ（*RTHD_IDXRBLDSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXRBLDSZ＝
    IDXRBLDBUF※1＋RTHD_SCAN＋SORTIOBUF※2＋SORTBUF※2＋RTHD_IDXREC※2
    ＋RTHD_IDXREBUILD※2＋RTHD_RNGIDX※3＋RTHD_TXTIDX※4
```

変数の説明

- *IDXRBLDBUF*

  インデクス再作成で使用するバッファ

  「(a)　変数 IDXRBLDBUF の求め方」を参照して求めてください。

- *RTHD_SCAN*

  インデクスレコードファイル作成時の表検索に使用するリアルスレッド固有メモリ

  「(b)　変数 RTHD_SCAN の求め方」を参照して求めてください。

- *SORTIOBUF*

  ソート処理 I/O バッファ

「(c)　変数 SORTIOBUF の求め方」を参照して求めてください。

- *SORTBUF*

  インデクス作成用ソートバッファ

  「(d)　変数 SORTBUF の求め方」を参照して求めてください。

- *RTHD_IDXREC*

  インデクス再作成時に使用するリアルスレッド固有メモリ

  「(e)　変数 RTHD_IDXREC の求め方」を参照して求めてください。

- *RTHD_IDXREBUILD*

  B-tree インデクス再作成時に使用するリアルスレッド固有メモリ

  「(f)　変数 RTHD_IDXREBUILD の求め方」を参照して求めてください。

- *RTHD_RNGIDX*

  レンジインデクス一括更新時に使用するリアルスレッド固有メモリ

  「(g)　変数 RTHD_RNGIDX の求め方」を参照して求めてください。

- *RTHD_TXTIDX*

  テキストインデクス一括更新時に使用するリアルスレッド固有メモリ

  「(h)　変数 RTHD_TXTIDX の求め方」を参照して求めてください。

注※1

処理対象インデクスがレンジインデクスだけの場合は，0 として計算してください。

注※2

B-tree インデクスを対象にした場合に加算してください。

注※3

処理対象インデクスにレンジインデクスが含まれる場合に加算してください。

注※4

処理対象インデクスにテキストインデクスが含まれる場合に加算してください。

(a)　変数 IDXRBLDBUF の求め方

変数 *IDXRBLDBUF* は，インデクス再作成で使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IDXRBLDBUF＝
  ↑（RBLD_SQBLK＋RBLD_SQIO＋RBLD_SQPGE＋RBLD_SQHS）÷1,024↑
```

変数の説明

RBLD_SQBLK：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQBLK＝
  rbld_blknum×（112＋↑RBLD_BLKSZ÷rbld_pagesize÷64↑×24）
```

- *rbld_blknum*

  インデクス再作成オプションadb_idxrebuild_buff_blk_num の指定値

- *RBLD_BLKSZ*

  4,194,304 を代入してください。

- *rbld_pagesize*

  インデクス一括作成の対象である B-tree インデクスまたはテキストインデクスが格納されている DB エリアのページサイズのうち，最小の値（単位：バイト）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

RBLD_SQIO：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQIO＝ 568×rbld_blknum
```

- *rbld_blknum*

  インデクス再作成オプションadb_idxrebuild_buff_blk_num の指定値

RBLD_SQPGE：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQPGE＝
  （176＋rbld_pagesize＋BUFLOG）×（RBLD_BLKSZ÷rbld_pagesize）
  ×rbld_blknum
```

- *rbld_pagesize*

  インデクス一括作成の対象である B-tree インデクスまたはテキストインデクスが格納されている DB エリアのページサイズのうち，最小の値（単位：バイト）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

- *BUFLOG*

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時)」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFLOG* を参照してください。

- *RBLD_BLKSZ*

  4,194,304 を代入してください。

- *rbld_blknum*

  インデクス再作成オプションadb_idxrebuild_buff_blk_num の指定値

RBLD_SQHS：次に示す計算式から求めてください。

計算式（単位：バイト）

RBLD_SQHS＝ 8×*rbld_blknum*＋40×*rbld_blknum*

- *rbld_blknum*

  インデクス再作成オプションadb_idxrebuild_buff_blk_num の指定値

(b) 変数 RTHD_SCAN の求め方

変数*RTHD_SCAN* は，インデクスレコードファイル作成時の表検索に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_SCAN ＝ MAX (*RTHD_EXESQLSZ* , *RTHD_EXESQLDICSZ*)

注

次に示す SQL 文を実行したものと仮定して，変数*RTHD_EXESQLSZ* および変数*RTHD_EXESQLDICSZ* の値をそれぞれ求めてください。

SELECT *選択式*※ FROM *”対象表名”*

注※

CHAR(16)の列と，処理対象インデクスの全インデクス構成列について重複を排除した列を指定したものとして求めてください。

変数の説明

RTHD_EXESQLSZ

「6.3.4 通常運用時のメモリ所要量の求め方」の「(2) リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c) 変数 RTHD_EXESQLSZ の求め方」を参照してください。

RTHD_EXESQLDICSZ

「6.3.4 通常運用時のメモリ所要量の求め方」の「(2) リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(g) 変数 RTHD_EXESQLDICSZ の求め方」を参照してください。

(c) 変数 SORTIOBUF の求め方

変数*SORTIOBUF* は，ソート処理で使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

SORTIOBUF＝ *sort_io_buff_size*×*sort_rthd*

変数の説明

- *sort_io_buff_size*

  16 を代入してください。

- *sort_rthd*

  次に示す計算式から求めてください。

  インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

(d) 変数 SORTBUF の求め方

変数*SORTBUF* は，インデクス再作成時のソート処理で使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

SORTBUF＝ （48＋*sort_buff_size*×1,024）×*sort_rthd*

変数の説明

sort_rthd

次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

sort_buff_size

インデクス再作成オプションadb_idxrebuild_sort_buff_size の指定値※

注※

ソート処理では，作業用ワークファイルが作成されます。このファイルの容量を最小にする変数*sort_buff_size* の計算式を次に示します。変数*sort_buff_size* は，次に示す計算式を満たすようにしてください。ただし，メモリに余裕がない場合，またはほとんどのデータがインデクスキー順に並んでいる場合は，必要以上に大きな値を指定しないでください。

計算式

$$\text{sort_buff_size} \geqq \uparrow\left(\frac{\text{REC_SIZE}+7}{2}+\sqrt{(\text{BLK}+8)\times\text{row_num}\times\text{KEY_INF}+\frac{\text{REC_SIZE}+7}{4}}+\text{COL_INFO}\right)\div 1{,}024\uparrow$$

変数の説明

REC_SIZE：次に示す計算式から求めてください。

計算式

REC_SIZE＝ （*KEYSZ*＋*SYS1*）＋*SYS2*

KEYSZ：処理対象表に定義された B-tree インデクスのインデクスキー長

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

SYS1：次に示す値を仮定してください。

- 処理対象表に定義されたインデクスのインデクス構成列が固定長の場合：10
- 処理対象表に定義されたインデクスのインデクス構成列に可変長を含む場合：12

SYS2：次に示す値を仮定してください。

- 処理対象表に定義されたインデクスが可変長の複数列インデクスの場合：(*インデクス構成列数*×4)
- そのほかのインデクスの場合：0

BLK：次に示す計算式から求めてください。

計算式

BLK＝ *REC_SIZE*＋（*KEYSZ*＋8）＋56

row_num：

処理対象表に格納されている行数

KEY_INF：次に示す計算式から求めてください。

計算式

KEY_INF＝ *REC_SIZE*＋（*KEYSZ*＋8）＋28

COL_INFO：次に示す計算式から求めてください。

計算式

COL_INFO＝ 2,112＋（*MULTI_KEY_INFO*×32）＋（*KEYSZ*＋8）

MULTI_KEY_INFO：次に示す値を仮定してください。

- 処理対象表に定義された B-tree インデクスが可変長の複数列インデクスの場合

  （*インデクス構成列数*×2）＋2

- そのほかの場合

  5

(e) 変数 RTHD_IDXREC の求め方

変数*RTHD_IDXREC* は，インデクス再作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXREC＝
    ↑（386＋328×（idx_num-1）＋32×max_idx_col_num
      ＋（64＋buf_size×1,024）×2×idx_num）÷1,024↑
```

変数の説明

- *idx_num*

  再作成対象の B-tree インデクス数

- *max_idx_col_num*

  再作成対象の B-tree インデクスのインデクス構成列数の最大値

- *buf_size*

  1,024 を代入してください。

(f) 変数 RTHD_IDXREBUILD の求め方

変数*RTHD_IDXREBUILD* は，B-tree インデクス再作成時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXREBUILD = \max_{1 \leqq k \leqq idx_num} RTHD_IDXREBUILD_MEM(k)$$

変数の説明

idx_num

再作成対象の B-tree インデクス数

RTHD_IDXREBUILD_MEM（*k*）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXREBUILD_MEM（k）＝
  ↑（61,458＋↑MAX（255，KEYSZ）÷2↑×2＋3,070
   ＋↑KEYSZ＋CTRL÷8↑×8＋MAX（255，KEYSZ）＋14
   ＋↓8＋page_size×95÷100↓×2）÷1,024↑
```

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL

- インデクス構成列のキー長がすべて固定長の場合：10 バイト
- インデクス構成列のキー長に可変長を含む場合：12 バイト

(g)　変数 RTHD_RNGIDX の求め方

変数 *RTHD_RNGIDX* は，レンジインデクス一括更新時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_RNGIDX＝　↑（16＋128×*rngidx_num*）×*scan_rthd*÷1,024↑

変数の説明

- *rngidx_num*

  再作成対象のレンジインデクス数

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*インデクス再作成オプション adb_idxrebuild_rthd_num の指定値*－1）÷2↓

(h)　変数 RTHD_TXTIDX の求め方

変数 *RTHD_TXTIDX* は，テキストインデクス一括更新時に使用するリアルスレッド固有メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_TXTIDX = \max_{1 \leqq k \leqq idx_num} RTHD_TXTIDXBUILD_MEM(k)$$

変数の説明

idx_num

処理対象表に定義されたテキストインデクスの数

RTHD_TXTIDXBUILD_MEM(*k*)

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_TXTIDXBUILD_MEM(k)＝
      buff_blk_size×buff_blk_num＋3,072＋↑(idx_div_num×24)÷1,024↑
      ＋txt_sort_buff_size×1,024
```

buff_blk_size

4,096 を代入してください。

buff_blk_num

インデクス再作成オプションadb_idxrebuild_buff_blk_num の指定値

idx_div_num

k 番目のテキストインデクスのインデクス構成列定義長

txt_sort_buff_size

インデクス再作成オプションadb_idxrebuild_txt_buff_size の指定値

(3) ヒープメモリの所要量の求め方（adbidxrebuild コマンド実行時）

adbidxrebuild コマンド実行時のヒープメモリ（*HEAP_IDXRBLDSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

HEAP_IDXRBLDSZ＝ Σ（505×*sort_rthd*）※

変数の説明

sort_rthd

次に示す計算式から求めてください。

インデクス再作成オプションadb_idxrebuild_rthd_numの指定値－1

注※

複数のadbidxrebuild コマンドを同時に実行する場合は，各adbidxrebuild コマンドが使用するヒープメモリの所要量を求めてください。そして，それぞれで求めたヒープメモリの所要量を合計してください。

6.3.8 adbgetcst コマンド実行時のメモリ所要量の求め方

adbgetcst コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_GETCOSTSZ*）
- リアルスレッド固有メモリ（*RTHD_GETCOSTSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（adbgetcst コマンド実行時）

adbgetcst コマンド実行時のプロセス共通メモリ（*PROC_GETCOSTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_GETCOSTSZ＝ Σ（*PROC_MNG*＋*PROC_CST*）※

変数の説明

- *PROC_MNG*
  コスト情報収集時の管理で使用するプロセス共通メモリ
  「(a)　変数 PROC_MNG の求め方」を参照して求めてください。
- *PROC_CST*
  コスト情報収集時にバッファとして使用するプロセス共通メモリ
  「(b)　変数 PROC_CST の求め方」を参照して求めてください。

注※

複数のadbgetcst コマンドを同時に実行する場合は，各adbgetcst コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

(a)　変数 PROC_MNG の求め方

変数*PROC_MNG* は，コスト情報収集時の管理で使用するプロセス共通メモリの所要量を求めるときに使用します。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_MNG＝
    3＋DIC＋IOA＋TBL＋DBSTA＋PROC_AUDINFSZ※
```

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

DIC

処理対象表の定義情報を取得するメモリ

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照して求めてください。

IOA

データベースアクセスで使用するメモリ

計算式（単位：キロバイト）

```
IOA＝ 32＋1,024
```

TBL

処理対象表の管理で使用するメモリ

計算式（単位：キロバイト）

```
TBL＝ 64×tbl_num
```

- *tbl_num*

  処理対象表の数

DBSTA

データベース状態解析で使用するメモリ

計算式（単位：キロバイト）

$$DBSTA＝ \uparrow \frac{(256,088＋88×処理対象表に作成されたチャンク数)}{1,024} \uparrow$$

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

(b)　変数 PROC_CST の求め方

変数*PROC_CST* は，コスト情報収集時にバッファとして使用するプロセス共通メモリの所要量を求めるときに使用します。

adbgetcst コマンドの-t オプションを省略して実行した場合は，-u オプションに指定した認可識別子のスキーマ内のすべての表について，変数*PROC_CST* を求めてください。その中で最も大きい値が，変数*PROC_CST* となります。

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_CST＝ DBHTBLINF＋DICTBLINF
```

変数の説明

DBHTBLINF

処理対象表のコスト情報を収集するときに使用するメモリ

計算式（単位：キロバイト）

$$DBHTBLINF = \left\lceil \sum_{i=1}^{col_num} \left(5,394,528 + coldata(i) \times 4 + collen(i) \times 60,400\right) \times sql_rthd_num \div 1,024 \right\rceil$$

col_num

処理対象表の列数

coldata（i）

各列のデータ長

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-9　各データ型のデータ長の一覧」を参照して求めてください。

collen（i）

各列の定義長

sql_rthd_num

次に示す計算式から求めてください。

コスト情報収集オプションadb_getcst_rthd_numの指定値－1

DICTBLINF

処理対象表のコスト情報をシステム表に格納するときに使用するメモリ

計算式（単位：キロバイト）

$$DICTBLINF = MAX\left\{64, \left\lceil \frac{(128 + 128 \times col_num + 128 \times (b_tree_idx_num + text_idx_num) + row_size \times 2)}{1,024} \right\rceil + 1\right\}$$

col_num

処理対象表の列数

b_tree_idx_num

処理対象表に定義された B-tree インデクスの数

text_idx_num

処理対象表に定義されたテキストインデクスの数

row_size

処理対象表の行長

計算式

$$row_size = \sum_{i=1}^{col_num} \left(\left\uparrow \frac{(coldata(i)+7)}{8} \right\uparrow \times 8 \right)$$

- *coldata（i）*：各列のデータ長

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-9　各データ型のデータ長の一覧」を参照して求めてください。

(2)　リアルスレッド固有メモリの所要量の求め方（adbgetcst コマンド実行時）

adbgetcst コマンド実行時のリアルスレッド固有メモリ（*RTHD_GETCOSTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_GETCOSTSZ ＝ *COSTSQL*

変数の説明

COSTSQL

コスト情報収集時に実行される検索系 SQL で使用するメモリ

adbgetcst コマンドを実行すると，処理対象表のコスト情報を収集するために，次に示す SQL 文が実行されます。

```
SELECT COUNT(*) FROM "対象表名"
SELECT MIN("列名1"),MAX("列名1") FROM "対象表名"
    :
SELECT MIN("列名n"),MAX("列名n") FROM "対象表名"
SELECT * FROM "対象表名"
```

注　列名 1～列名 *n* は，対象表に定義されている列名です。

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」および「(g)　変数 RTHD_EXESQLDICSZ の求め方」を参照して，検索系 SQL の実行時に使用されるメモリの所要量を，処理対象表ごとに求めてください。その中で，最も大きい値を代入してください。

6.3.9　adbdbstatus コマンド実行時のメモリ所要量の求め方

adbdbstatus コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_DBSTATUSSZ*）
- リアルスレッド固有メモリ（*RTHD_DBSTATUSSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（adbdbstatus コマンド実行時）

adbdbstatus コマンド実行時のプロセス共通メモリ（*PROC_DBSTATUSSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_DBSTATUSSZ＝ Σ（*PROC_MNG*）※

注※

複数のadbdbstatus コマンドを同時に実行する場合は，各adbdbstatus コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

変数の説明

PROC_MNG

データベース状態解析管理で使用するプロセス共通メモリ

計算式（単位：キロバイト）

PROC_MNG＝ 1,353＋*DICINF*＋*DBINF*＋*ARCINF*※1＋*PROC_AUDINFSZ*※2

注※1

adbdbstatus コマンドの-c オプションに，archivechunk を指定した場合に加算してください。

注※2

監査証跡機能が有効な場合に加算してください。

DICINF

定義情報を管理するメモリ

計算式（単位：キロバイト）

```
DICINF＝ MAX(64 , A)

  A：次の計算式で求めた値
  ↑(274,880＋120×dbarea_num＋224×table_num＋224×arctable_num＋232×index_num
    ＋232×arctable_num×3)÷1,024↑
```

- *dbarea_num*
  データ用 DB エリアの数
- *table_num*
  ユーザが定義した実表の数
- *arctable_num*
  ユーザが定義した実表のうち，アーカイブマルチチャンク表の数
- *index_num*
  ユーザが定義したインデクスの数

DBINF

データベース情報を管理するメモリ

計算式（単位：キロバイト）

$$DBINF = \uparrow \frac{(264,600 + 56 \times file_num + 88 \times chunk_num + 24 \times uthd_num)}{1,024} \uparrow$$

- *file_num*
  解析対象の DB エリア，および解析対象の実表やインデクスを格納する DB エリアのうち，最も多い DB エリアファイルの数
- *chunk_num*
  解析対象の実表やインデクスに作成されたチャンクのうち，最も多いチャンクの数
  「付録 C.9　システム表の検索」の「(10)　作成されたチャンクの数を調べる場合」を参照して求めてください。
- *uthd_num*
  サーバ定義adb_sys_uthd_num オペランドの指定値
  adbdbstatus コマンドの-d reorginfo オプションを指定しない場合は，0 としてください。

ARCINF

アーカイブマルチチャンク表を管理するメモリ

計算式（単位：キロバイト）

$$ARCINF = \uparrow (1,012 + 580 \times col_num + max_rowsz + 860 \times idx_num + 5,200) \div 1,024 \uparrow$$

col_num

処理対象表の列数

max_rowsz

最大行長（単位：バイト）

最大行長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の行長*ROWSZ* の計算式を基に求めてください。

idx_num

処理対象表に定義されたインデクス数

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

(2)　リアルスレッド固有メモリの所要量の求め方（adbdbstatus コマンド実行時）

adbdbstatus コマンド実行時のリアルスレッド固有メモリ（*RTHD_DBSTATUSSZ*）の所要量は，次に示す値となります。

値（単位：キロバイト）

RTHD_DBSTATUSSZ＝ 3

6.3.10　adbexport コマンド実行時のメモリ所要量の求め方

adbexport コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_EXPORTSZ*）
- リアルスレッド固有メモリ（*RTHD_EXPORTSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_EXPORTSZ*）

(1)　プロセス共通メモリの所要量の求め方（adbexport コマンド実行時）

adbexport コマンド実行時に使用するプロセス共通メモリ（*PROC_EXPORTSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_EXPORTSZ＝ ∑ (PROC_MNG＋PROC_OUTFILE※1)※2

注※1

出力データパスファイル名に指定した出力データファイル数が 1,024 個を超える場合に加算します。

注※2

複数のadbexport コマンドを同時に実行する場合は，各adbexport コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

変数の説明

PROC_MNG

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_MNG＝ 3＋DIC※1＋IOA＋EXPF＋PROC_EXPSQLS＋PROC_AUDINFSZ※2
```

注※1

adbexport コマンドに-n オプションを指定した場合に加算してください。

注※2

監査証跡機能が有効な場合に加算してください。

DIC

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照して求めてください。

IOA

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IOA＝ sql_size＋（scan_buff_size＋offset_area×2）×scan_rthd×2
```

sql_size

次に示す計算式から求めてください。

計算式

sql_size＝　（*SQL記述ファイルのサイズ*）×2＋1＋1,024

scan_buff_size

次に示す計算式から求めてください。

計算式

エクスポートオプションadb_export_scan_buff_sizeの指定値×1,024

offset_area

次に示す計算式から求めてください。

計算式

（8×*検索結果の列数*×（*scan_buff_size*×1,024÷*検索結果のデータ長*））÷1,024

scan_rthd

次に示す計算式から求めてください。

計算式

↓（*エクスポートオプションadb_export_rthd_numの指定値*－1）÷2↓

EXPF

次に示す計算式から求めてください。

計算式（単位：キロバイト）

EXPF＝ 552＋（512＋512※）×exp_rthd

注※

adbexport コマンドの--compress GZIP オプションを指定したときに加算してください。

exp_rthd

次に示す計算式から求めてください。

計算式

↓（*エクスポートオプションadb_export_rthd_numの指定値*－1）÷2↓

PROC_EXPSQLS

adbexport コマンドを実行すると，処理対象表からデータをエクスポートするために，次に示す SQL 文が 1 回実行されます。

-n オプションを指定した場合

SELECT * FROM *対象表名*;

-q オプションを指定した場合

SQL記述ファイルに記述したSELECT文

「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(c)　変数 PROC_EXECSQLSZ の求め方」を参照して，検索系 SQL の実行時に使用されるメモリの所要量を，処理対象表ごとに求めてください。その中で，最も大きい値を代入してください。

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

PROC_OUTFILE

出力データパスファイル名に指定した出力データファイル数が 1,024 個を超える場合に加算する必要があります。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_OUTFILE＝ ↑（（120＋512）×（OUTFMAX－1,024））÷1,024↑
```

OUTFMAX

出力データパスファイル名に指定した出力データファイル数

(2) リアルスレッド固有メモリの所要量の求め方（adbexport コマンド実行時）

adbexport コマンド実行時に使用するリアルスレッド固有メモリ（*RTHD_EXPORTSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_EXPORTSZ＝ EXPORTSQL
```

変数の説明

EXPORTSQL

adbexport コマンドを実行すると，処理対象表からデータをエクスポートするために，次に示す SQL 文が 1 回実行されます。

-n オプションを指定した場合

```
SELECT ＊ FROM 対象表名;
```

-q オプションを指定した場合

```
SQL記述ファイルに記述したSELECT文
```

「6.3.4 通常運用時のメモリ所要量の求め方」の「(2) リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c) 変数 RTHD_EXESQLSZ の求め方」および「(g) 変数 RTHD_EXESQLDICSZ の求め方」を参照して，検索系 SQL の実行時に使用されるメモリの所要量を，処理対象表ごとに求めてください。その中で，最も大きい値を代入してください。

(3) ヒープメモリの所要量の求め方（adbexport コマンド実行時）

adbexport コマンド実行時に使用するヒープメモリ（*HEAP_EXPORTSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_EXPORTSZ＝ ∑(HEAP_ZLIB)※
```

注※

複数のadbexport コマンドを同時に実行する場合は，各adbexport コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

変数の説明

HEAP_ZLIB

--compress GZIP オプションを指定した場合に，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_ZLIB＝ 262×file_num
```

file_num

出力データファイル数

6.3.11　adbstat コマンド実行時のメモリ所要量の求め方

adbstat コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。

■プロセスメモリ

- ヒープメモリ（*STATMEMSZ*）

adbstat コマンド実行時に使用するヒープメモリ（*STATMEMSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STATMEMSZ＝ 1,273×exec_num
```

変数の説明

exec_num：同時に実行するadbstat コマンドの数

6.3.12　adbmodarea コマンド実行時のメモリ所要量の求め方

adbmodarea コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_MODASZ*）
- リアルスレッド固有メモリ（*RTHD_MODASZ*）

■プロセスメモリ

- ヒープメモリ（*MODA_MEM*）

(1)　プロセス共通メモリの所要量の求め方（adbmodarea コマンド実行時）

adbmodarea コマンド実行時のプロセス共通メモリ（*PROC_MODASZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_MODASZ＝ *MODA_ROWS*＋*MODA_CTL*＋*MODA_DBH*＋*PROC_AUDINFSZ*※＋8

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

- *MODA_ROWS*

  「(a)　変数 MODA_ROWS の求め方」を参照して求めてください。

- *MODA_CTL*

  「(b)　変数 MODA_CTL の求め方」を参照して求めてください。

- *MODA_DBH*

  「(c)　変数 MODA_DBH の求め方」を参照して求めてください。

- *PROC_AUDINFSZ*

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ*を参照して求めてください。

(a)　変数 MODA_ROWS の求め方

変数*MODA_ROWS* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

MODA_ROWS＝　↑（256×*row_cnt*）÷1,024↑

変数の説明

row_cnt

ディクショナリ表SQL_DBAREAS の行数

(b)　変数 MODA_CTL の求め方

変数*MODA_CTL* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
MODA_CTL＝
    36,864＋MODA_REQ＋↑（（8×row_cnt）
    ＋（64×（bdev_num_new＋bdev_num_alloc））
    ＋（8×bdev_num_area）＋（256×bdev_num_file））÷1,024↑
```

変数の説明

MODA_REQ

「(2)　リアルスレッド固有メモリの所要量の求め方（adbmodarea コマンド実行時)」の変数 *MODA_REQ* を参照してください。

row_cnt

ディクショナリ表SQL_DBAREAS の行数

bdev_num_new

新規に割り当てるブロックスペシャルファイル数

bdev_num_alloc

割り当て済みのブロックスペシャルファイル数

bdev_num_area

ブロックスペシャルファイルを割り当てた DB エリア数

bdev_num_file

ブロックスペシャルファイルを割り当てた DB エリアの DB エリアファイル数の合計値

(c)　変数 MODA_DBH の求め方

変数 *MODA_DBH* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
■DBエリアを追加する場合
  MODA_DBH＝ ∑(239＋(add_dbarea_file_num－1)×143＋add_dbarea_file_num×329)※

■DBエリアを拡張する場合（DBエリアファイルを追加する場合）
  MODA_DBH＝ ∑(115＋(add_dbarea_file_num_all－1)×107＋add_dbarea_file_num×365)※
```

注※

複数の DB エリアを追加する場合，または複数の DB エリアを拡張する場合は，DB エリアごとに求めてください。そして，求めた値を合計してください。

変数の説明

add_dbarea_file_num

adbmodarea コマンドによる DB エリアの追加，および DB エリアの拡張によって増加する DB エリアファイルの数

add_dbarea_file_num_all

既存の DB エリアファイルの数と，DB エリアの拡張によって増加する DB エリアファイルの数を合計した値

(2) リアルスレッド固有メモリの所要量の求め方（adbmodarea コマンド実行時）

adbmodarea コマンド実行時のリアルスレッド固有メモリ（*RTHD_MODASZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_MODASZ＝ 20,480＋MODA_REQ
```

変数の説明

MODA_REQ

次に示す計算式から求めてください。

```
MODA_REQ＝ 3＋↑（256×bdev_num_new）÷1,024↑
```

bdev_num_new

新規に割り当てるブロックスペシャルファイル数

(3) ヒープメモリの所要量の求め方（adbmodarea コマンド実行時）

adbmodarea コマンド実行時のヒープメモリ（*MODA_MEM*）の所要量は，次に示す計算式から求めてください。

計算式（単位：メガバイト）

$$MODA_MEM = \left\uparrow \frac{\{17 + bdev_num_new\}}{1,024} \right\uparrow$$

変数の説明

bdev_num_new

新規に割り当てるブロックスペシャルファイル数

重要

adbmodarea コマンドで，DB エリアの追加および DB エリアの削除を実行した場合は，次に示すメモリ所要量が変化します。そのため，再見積もりしてください。

- グローバルバッファページの所要量（変数 SHM_BUFGLOBAL）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」を参照してください。

- プロセス共通メモリの所要量（変数 BUFGLOBAL）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」を参照してください。

6.3.13　adbmergechunk コマンド実行時のメモリ所要量の求め方

adbmergechunk コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_MERGCSZ*）
- リアルスレッド固有メモリ（*RTHD_MERCHKSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_MERCHKSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（adbmergechunk コマンド実行時）

adbmergechunk コマンド実行時のプロセス共通メモリ（*PROC_MERGCSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_MERGCSZ＝
   Σ（PROC_MNG＋PROC_IDX※1＋PROC_IDXREBUILD※1＋PROC_RNGIDX※2
      ＋PROC_CHUNK_INFO＋PROC_DBUPDINF＋PROC_RTHDUPDINF）※3
```

変数の説明

- *PROC_MNG*

  「(a)　変数 PROC_MNG の求め方」を参照して求めてください。

- *PROC_IDX*

  「(b)　変数 PROC_IDX の求め方」を参照して求めてください。

- *PROC_IDXREBUILD*

  「(c)　変数 PROC_IDXREBUILD の求め方」を参照して求めてください。

- *PROC_RNGIDX*

  「(d)　変数 PROC_RNGIDX の求め方」を参照して求めてください。

- *PROC_CHUNK_INFO*

  「(e)　変数 PROC_CHUNK_INFO の求め方」を参照して求めてください。

- *PROC_DBUPDINF*

  「(f)　変数 PROC_DBUPDINF の求め方」を参照して求めてください。

- *PROC_RTHDUPDINF*

  「(g)　変数 PROC_RTHDUPDINF の求め方」を参照して求めてください。

注※1

処理対象インデクスに B-tree インデクスが含まれる場合に加算してください。

注※2

処理対象インデクスにレンジインデクスが含まれる場合に加算してください。

注※3

複数のadbmergechunk コマンドを同時に実行する場合は，各adbmergechunk コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

(a)　変数 PROC_MNG の求め方

変数*PROC_MNG* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_MNG＝
5＋*DIC*＋*IOA*＋*SCAN*＋*IDXRBLDBUF_CTL*＋*STS*＋*PAGEALLOC*＋*CHUNK*＋*PROC_AUDINFSZ*※

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

DIC：処理対象表の定義情報を取得するメモリ

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照してください。

IOA：データベースアクセスで使用するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

IOA＝ *sql_size*＋（*scan_buff_size*＋*offset_area*×2）×*scan_rthd*

- *sql_size*

  次に示す計算式から求めてください。

  32＋1,024

- *scan_buff_size*

  次に示す計算式から求めてください。

  マージチャンクオプション adb_mergechunk_scan_buff_size の指定値×1,024

- *offset_area*

  次に示す計算式から求めてください。

  （8×*処理対象インデクスの構成列数*×（*scan_buff_size*×1,024
  ÷（*処理対象インデクスの構成列長の合計値*＋8））÷1,024

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

SCAN：インデクスレコードファイルの出力処理で必要となるメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

SCAN＝
↑（*write_size*×*scan_rthd*×*b-tree_index_num*）↑

- *write_size*

  マージチャンクオプションadb_mergechunk_dvix_wtbuff_size の指定値

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

- *b-tree_index_num*

  処理対象の B-tree インデクス数

IDXRBLDBUF_CTL：マージチャンク用バッファ制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

IDXRBLDBUF_CTL＝
↑（248＋*BUFBLK*）÷1,024↑×*scan_rthd*

- *BUFBLK*

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(f)　変数 PROC_UPDSZ の求め方」の変数*BUFBLK* を参照してください。

- *scan_rthd*

  次に示す計算式から求めてください。

  ↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

STS：ステータス情報を退避するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STS＝
  ↑（256＋（1,185×idx_num）
  ＋（2,067×（b_tree_idx_num＋range_idx_num）×scan_rthd＋2,067×text_idx_num×16））÷1,024↑
```

idx_num

処理対象のインデクス数

b_tree_idx_num

処理対象表に定義された B-tree インデクスの数

range_idx_num

処理対象表に定義されたレンジインデクスの数

scan_rthd

次に示す計算式から求めてください。

↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

text_idx_num

処理対象表に定義されたテキストインデクスの数

PAGEALLOC：マージチャンク用ページ割り当て制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

PAGEALLOC＝ *dividx_rthd*×310

- *dividx_rthd*

  次に示す計算式から求めてください。

  マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

CHUNK：チャンク管理ブロック

次に示す計算式から求めてください。

計算式

CHUNK＝ ↑（8×（*chunk_num*×2）＋30,000）÷1,024↑

- *chunk_num*

  処理対象表に作成されているチャンク数

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照してください。

(b)　変数 PROC_IDX の求め方

変数*PROC_IDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDX＝
  ↑（（13,151＋（33,928＋（13×KEYSZ））×scan_rthd
    ＋（152＋rd_buff_size×scan_rthd×2×1,024＋2×ld_buff_size×1,024
    ＋（KEYSZ＋CTRL）×text_idx_num）×dividx_rthd）
    ×（b_tree_idx_num＋text_idx_num））÷1,024↑
```

変数の説明

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

scan_rthd

次に示す計算式から求めてください。

↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

rd_buff_size

マージチャンクオプションadb_mergechunk_dvix_rdbuff_size の指定値

ld_buff_size

マージチャンクオプションadb_mergechunk_dvix_wtbuff_size の指定値

CTRL

- インデクスのインデクス構成列が固定長の場合

10 バイトを代入してください。

- インデクスのインデクス構成列に可変長を含む場合

  12 バイトを代入してください。

text_idx_num

処理対象表に定義されたテキストインデクス数

dividx_rthd

次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

b_tree_idx_num

処理対象の B-tree インデクス数

(c) 変数 PROC_IDXREBUILD の求め方

変数 *PROC_IDXREBUILD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXREBUILD = \max_{1 \leqq k \leqq idx_num} PROC_IDXREBUILD_MEM(k)$$

idx_num

再作成対象の B-tree インデクス数

PROC_IDXREBUILD_MEM（k）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXREBUILD_MEM(k) = \uparrow (184 + 256 \times idx_col_num + (18 + 256 \times \uparrow idx_lv \div 16 \uparrow) \times dividx_rthd + KEYSZ + (idx_col_num + 1) \times 2 + 4) \div 1{,}024 \uparrow$$

- *idx_col_num*

  B-tree インデクスのインデクス構成列数

- *idx_lv*

  B-tree インデクスの段数

  「5.8.3 B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2) 上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の，『計算式 1（PIDX（k）の求め方)』の漸化式を計算して，*PIDX（n）* ＝ 1 となったときの *n* の値を代入してください。

  ただし，*PIDX（1）* ＝ 1 の場合は，B-tree インデクスの段数は 2 としてください。

- *dividx_rthd*

  次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

- *KEYSZ*：B-tree インデクスのキー長（単位：バイト）

  「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

  なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

(d)　変数 PROC_RNGIDX の求め方

変数*PROC_RNGIDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_RNGIDX＝　↑64×*rngidx_num*÷1,024↑

変数の説明

rngidx_num：再作成対象のレンジインデクス数

(e)　変数 PROC_CHUNK_INFO の求め方

変数*PROC_CHUNK_INFO* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_CHUNK_INFO＝　↑（1＋*index_num*）×120×（*chunk_num*＋1）÷1,024↑

変数の説明

index_num：マージチャンク対象表に対して定義したインデクスの数

chunk_num：マージチャンク対象表のチャンク数の最大値

なお，adbmergechunk コマンドを実行すると，変数*PROC_CHUNK_INFO* で求められるサイズ分，プロセス共通メモリが確保されます。adbmergechunk コマンドが完了すると，確保されたプロセス共通メモリは解放されます。

しかし，次の理由で**削除仕掛中のチャンク**が残っている場合，確保されたプロセス共通メモリは解放されません。削除仕掛中のチャンクをすべて削除すると，解放されます。

- 実行中のadbmergechunk コマンドが，マージ先チャンクの完成からマージ元チャンクの削除完了までの間に中断した場合

  adbcancel コマンドを実行したとき，HADB サーバでメモリ不足が発生したときなどが該当します。

- adbmergechunk コマンドの--purge-chunk オプションにNOWAIT を指定して実行した場合に，マージチャンク対象の表を参照している処理が存在したとき

(f)　変数 PROC_DBUPDINF の求め方

変数*PROC_DBUPDINF* は，次に示す計算式から求めてください。なお，マルチノード機能を使用するかどうかで，計算式が異なります。

マルチノード機能を使用しない場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
  ↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×131,136）÷1,024）↑
```

マルチノード機能を使用する場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
  ↑（（80＋（↑（DBUPDINF_ENT_NUM÷1,024）↑＋1）×131,136）÷1,024）↑
```

変数の説明

DBUPDINF_ENT_NUM

DB エリア，表，インデクス，およびチャンク更新情報のエントリ数

次に示す計算式から求めてください。なお，マルチノード機能を使用するかどうかで，計算式が異なります。

マルチノード機能を使用しない場合の計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝　1＋1＋index_num
```

マルチノード機能を使用する場合の計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝　3＋（1＋index_num）×chunk_num＋（index_num×6）
```

index_num

マージチャンク対象表に定義しているインデクス数

chunk_num

マージするチャンク数

(g)　変数 PROC_RTHDUPDINF の求め方

変数*PROC_RTHDUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_RTHDUPDINF＝
    ↑（648＋（168×rthd_num））÷1,024↑
```

変数の説明

rthd_num

次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

(2) リアルスレッド固有メモリの所要量の求め方（adbmergechunk コマンド実行時）

adbmergechunk コマンド実行時のリアルスレッド固有メモリ（*RTHD_MERCHKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_MERCHKSZ＝
    IDXRBLDBUF＋RTHD_SCAN＋SORTIOBUF※1＋SORTBUF※1＋RTHD_IDXREC※1
    ＋RTHD_IDXREBUILD※1＋RTHD_RNGIDX※2＋RTHD_TXTIDX※3
```

変数の説明

- *IDXRBLDBUF*

  「(a)　変数 IDXRBLDBUF の求め方」を参照して求めてください。
- *RTHD_SCAN*

  「(b)　変数 RTHD_SCAN の求め方」を参照して求めてください。
- *SORTIOBUF*

  「(c)　変数 SORTIOBUF の求め方」を参照して求めてください。
- *SORTBUF*

  「(d)　変数 SORTBUF の求め方」を参照して求めてください。
- *RTHD_IDXREC*

  「(e)　変数 RTHD_IDXREC の求め方」を参照して求めてください。
- *RTHD_IDXREBUILD*

  「(f)　変数 RTHD_IDXREBUILD の求め方」を参照して求めてください。
- *RTHD_RNGIDX*

  「(g)　変数 RTHD_RNGIDX の求め方」を参照して求めてください。
- *RTHD_TXTIDX*

  「(h)　変数 RTHD_TXTIDX の求め方」を参照して求めてください。

注※1

B-tree インデクスを対象にした場合に加算してください。

注※2

処理対象インデクスにレンジインデクスが含まれる場合に加算してください。

注※3

処理対象インデクスにテキストインデクスが含まれる場合に加算してください。

(a)　変数 IDXRBLDBUF の求め方

変数 *IDXRBLDBUF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IDXRBLDBUF＝
  ↑（RBLD_SQBLK＋RBLD_SQIO＋RBLD_SQPGE＋RBLD_SQHS）÷1,024↑
```

変数の説明

RBLD_SQBLK：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQBLK＝
  rbld_blknum×（112＋↑RBLD_BLKSZ÷rbld_pagesize÷64↑×24）
```

- *rbld_blknum*

  マージチャンクオプションadb_mergechunk_buff_blk_num の指定値

- *RBLD_BLKSZ*

  次に示す計算式から求めてください。

  ```
  4,096×1,024
  ```

- *rbld_pagesize*

  処理対象表に定義されている B-tree インデクスまたはテキストインデクスが格納されている DB エリアのページサイズのうち，最小の値（単位：バイト）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

RBLD_SQIO：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQIO＝ 568×rbld_blknum
```

- *rbld_blknum*

  マージチャンクオプションadb_mergechunk_buff_blk_num の指定値

RBLD_SQPGE：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQPGE＝
   （176＋rbld_pagesize＋BUFLOG）×（RBLD_BLKSZ÷rbld_pagesize）×rbld_blknum
```

- *rbld_pagesize*

  処理対象表に定義されている B-tree インデクスまたはテキストインデクスが格納されている DB エリアのページサイズのうち，最小の値（単位：バイト）

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

- *BUFLOG*

  「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時)」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFLOG* を参照してください。

- *RBLD_BLKSZ*

  次に示す計算式から求めてください。

  ```
  4,096×1,024
  ```

- *rbld_blknum*

  マージチャンクオプションadb_mergechunk_buff_blk_num の指定値

RBLD_SQHS：次に示す計算式から求めてください。

計算式（単位：バイト）

```
RBLD_SQHS＝ 8×rbld_blknum＋40×rbld_blknum
```

- *rbld_blknum*

  マージチャンクオプションadb_mergechunk_buff_blk_num の指定値

(b)　変数 RTHD_SCAN の求め方

変数*RTHD_SCAN* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_SCAN ＝ MAX ( RTHD_EXESQLSZ , RTHD_EXESQLDICSZ )
```

注

次に示す SQL 文を実行したものと仮定して，変数*RTHD_EXESQLSZ* および変数*RTHD_EXESQLDICSZ* の値をそれぞれ求めてください。

```
SELECT 選択式※ FROM "対象表名"
```

注※

CHAR(16)の列と，処理対象表に定義したインデクスの全インデクス構成列について重複を排除した列を指定したものとして求めてください。

変数の説明

RTHD_EXESQLSZ

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」を参照してください。

RTHD_EXESQLDICSZ

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(g)　変数 RTHD_EXESQLDICSZ の求め方」を参照してください。

(c)　変数 SORTIOBUF の求め方

変数*SORTIOBUF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

SORTIOBUF ＝ *sort_io_buff_size* ×*sort_rthd*

変数の説明

- *sort_io_buff_size*
  16 を代入してください。
- *sort_rthd*
  次に示す計算式から求めてください。
  マージチャンクオプションadb_mergechunk_rthd_numの指定値 －1

(d)　変数 SORTBUF の求め方

変数*SORTBUF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

SORTBUF ＝　(48＋*sort_buff_size* ×1,024) ×*sort_rthd*

変数の説明

sort_rthd

次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値 －1

sort_buff_size

マージチャンクオプションadb_mergechunk_sort_buff_size の指定値※

注※

ソート処理では，作業用ワークファイルが作成されます。このファイルの容量を最小にする変数*sort_buff_size* の計算式を次に示します。変数*sort_buff_size* は，次に示す計算式を満たすようにしてください。ただし，メモリに余裕がない場合，またはほとんどのデータがインデクスキー順に並んでいる場合は，必要以上に大きな値を指定しないでください。

計算式

$$\mathrm{sort_buff_size} \geqq \uparrow\left(\frac{\mathrm{REC_SIZE}+7}{2}+\sqrt{(\mathrm{BLK}+8)\times\mathrm{row_num}\times\mathrm{KEY_INF}+\frac{\mathrm{REC_SIZE}+7}{4}}+\mathrm{COL_INFO}\right)\div 1{,}024\uparrow$$

変数の説明

REC_SIZE：次に示す計算式から求めてください。

計算式

$$REC_SIZE = (KEYSZ + SYS1) + SYS2$$

KEYSZ：処理対象表に定義された B-tree インデクスのキー長

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

SYS1：次に示す値を仮定してください。

- 処理対象表に定義されたインデクスのインデクス構成列がすべて固定長の場合：10
- 処理対象表に定義されたインデクスのインデクス構成列に可変長を含む場合：12

SYS2：次に示す値を仮定してください。

- 処理対象表に定義されたインデクスが可変長の複数列インデクスの場合

  次に示す計算式から求めてください。

  $$(インデクス構成列数 \times 4)$$

- そのほかのインデクスの場合

  $$0$$

BLK：次に示す計算式から求めてください。

計算式

$$BLK = REC_SIZE + (KEYSZ + 8) + 56$$

row_num：

処理対象表に格納されている行数

KEY_INF：次に示す計算式から求めてください。

計算式

$$KEY_INF = REC_SIZE + (KEYSZ + 8) + 28$$

COL_INFO：次に示す計算式から求めてください。

計算式

COL_INFO＝ 2,112＋（*MULTI_KEY_INFO*×32）＋（*KEYSZ*＋8）

MULTI_KEY_INFO：次に示す値を仮定してください。

- 処理対象表に定義された B-tree インデクスが可変長の複数列インデクスの場合
  次に示す計算式から求めてください。

  （*インデクス構成列数*×2）＋2

- そのほかの B-tree インデクスの場合

  5

(e) 変数 RTHD_IDXREC の求め方

変数*RTHD_IDXREC* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_IDXREC＝
↑（386＋328×（*idx_num*-1）＋32×*max_idx_col_num*
＋（64＋*buf_size*×1,024）×2×*idx_num*）÷1,024↑

変数の説明

- *idx_num*
  再作成対象の B-tree インデクス数
- *max_idx_col_num*
  再作成対象の B-tree インデクスのインデクス構成列数の最大値
- *buf_size*
  1,024 を代入してください。

(f) 変数 RTHD_IDXREBUILD の求め方

変数*RTHD_IDXREBUILD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXREBUILD = \max_{1 \leqq k \leqq idx_num} RTHD_IDXREBUILD_MEM(k)$$

変数の説明

idx_num
再作成対象の B-tree インデクス数

RTHD_IDXREBUILD_MEM（*k*）

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXREBUILD_MEM（k）＝
  ↑（61,458＋↑MAX（255，KEYSZ）÷2↑×2＋3,070
   ＋↑KEYSZ＋CTRL÷8↑×8＋MAX（255，KEYSZ）＋14
   ＋↓8＋page_size×95÷100↓×2）÷1,024↑
```

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL

- インデクス構成列のキー長がすべて固定長の場合

  10 バイトを代入してください。

- インデクス構成列のキー長に可変長を含む場合

  12 バイトを代入してください。

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

(g)　変数 RTHD_RNGIDX の求め方

変数 *RTHD_RNGIDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_RNGIDX＝ ↑（16＋128×rngidx_num）×scan_rthd÷1,024↑
```

変数の説明

- *rngidx_num*

  再作成対象のレンジインデクス数

- *scan_rthd*

  次に示す計算式から求めてください。

  ```
  ↓（マージチャンクオプションadb_mergechunk_rthd_numの指定値－1）÷2↓
  ```

(h) 変数 RTHD_TXTIDX の求め方

変数*RTHD_TXTIDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_TXTIDX = \max_{1 \leqq k \leqq idx_num} RTHD_TXTIDXBUILD_MEM(k)$$

変数の説明

idx_num

処理対象表に定義されたテキストインデクスの数

RTHD_TXTIDXBUILD_MEM(*k*)

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_TXTIDXBUILD_MEM(k) = buff_blk_size \times buff_blk_num + 3,072 + \uparrow (idx_div_num \times 24) \div 1,024 \uparrow + txt_sort_buff_size \times 1,024$$

buff_blk_size

4,096 を代入してください。

buff_blk_num

マージチャンクオプションadb_mergechunk_buff_blk_num の指定値

idx_div_num

k 番目のテキストインデクスのインデクス構成列定義長

txt_sort_buff_size

マージチャンクオプションadb_mergechunk_txt_buff_size の指定値

(3) ヒープメモリの所要量の求め方（adbmergechunk コマンド実行時）

adbmergechunk コマンド実行時のヒープメモリ（*HEAP_MERCHKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$HEAP_MERCHKSZ = \Sigma\ (505 \times sort_rthd)\ ※$$

注※

複数のadbmergechunk コマンドを同時に実行する場合は，各adbmergechunk コマンドが使用するヒープメモリの所要量を求めてください。そして，それぞれで求めたヒープメモリの所要量を合計してください。

変数の説明

sort_rthd

次に示す計算式から求めてください。

マージチャンクオプションadb_mergechunk_rthd_numの指定値－1

6.3.14　adbchgchunkcomment コマンド実行時のメモリ所要量の求め方

adbchgchunkcomment コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。

■共有メモリ

- プロセス共通メモリ（*PROC_CHGCCMSZ*）

adbchgchunkcomment コマンド実行時に使用するプロセス共通メモリ（*PROC_CHGCCMSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_CHGCCMSZ＝ Σ（5＋*DIC*＋*PROC_AUDINFSZ*※1）※2

注※1

監査証跡機能が有効な場合に加算してください。

注※2

複数のadbchgchunkcomment コマンドを同時に実行する場合は，各adbchgchunkcomment コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

変数の説明

DIC

処理対象表の定義情報を取得するメモリ

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照して求めてください。

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

6.3.15　adbchgchunkstatus コマンド実行時のメモリ所要量の求め方

adbchgchunkstatus コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。

■共有メモリ

- プロセス共通メモリ（*PROC_CHGCSTSZ*）

adbchgchunkstatus コマンド実行時に使用するプロセス共通メモリ（*PROC_CHGCSTSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_CHGCSTSZ＝ PROC_MCS＋PROC_DBUPDINF※
```

注※

マルチノード機能を使用する場合に加算してください。

変数の説明

PROC_MCS

チャンクの状態変更の管理で使用するプロセス共通メモリ

計算式（単位：キロバイト）

```
PROC_MCS＝　5＋DIC＋↑（30×chunk_num）÷1,024↑＋PROC_AUDINFSZ※
```

注※

監査証跡機能が有効な場合に加算してください。

DIC

処理対象表の定義情報を取得するメモリ

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照して求めてください。

chunk_num

処理対象表に作成されたチャンク数

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

PROC_DBUPDINF

DB エリア，表，インデクス，およびチャンク更新情報の格納で必要なプロセス共通メモリ

値（単位：キロバイト）

```
PROC_DBUPDINF＝ 129
```

6.3.16　adbarchivechunk コマンド実行時のメモリ所要量の求め方

adbarchivechunk コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_ARCCKSZ*）
- リアルスレッド固有メモリ（*RTHD_ARCCKSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_ARCCKSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（adbarchivechunk コマンド実行時）

adbarchivechunk コマンド実行時のプロセス共通メモリ（*PROC_ARCCKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_ARCCKSZ＝
  ∑(PROC_ARCCON＋(PROC_MAC＋PROC_OUTFILE＋PROC_DBUPDINF)※1)※2
```

変数の説明

PROC_ARCCON

「(a)　変数 PROC_ARCCON の求め方」を参照して求めてください。

PROC_MAC

「(b)　変数 PROC_MAC の求め方」を参照して求めてください。

PROC_OUTFILE

「(c)　変数 PROC_OUTFILE の求め方」を参照して求めてください。

PROC_DBUPDINF

「(d)　変数 PROC_DBUPDINF の求め方」を参照して求めてください。

注※1

複数のチャンクを処理対象にしてadbarchivechunk コマンドを実行する場合は，チャンク単位にメモリ所要量を求めてください。そして，求めた値の最大値を加算してください。

注※2

複数のadbarchivechunk コマンドを同時に実行する場合は，各adbarchivechunk コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

(a)　変数 PROC_ARCCON の求め方

変数*PROC_ARCCON* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_ARCCON＝ ↑（250,000＋800）÷1,024↑＋PROC_AUDINFSZ※
```

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

(b)　変数 PROC_MAC の求め方

変数*PROC_MAC* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_MAC＝ 3＋DIC＋IOA＋EXPF＋PROC_EXPSQLS＋SGMTGRP
```

変数の説明

DIC

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照してください。

IOA

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IOA＝ sql_size＋（scan_buff_size＋offset_area×2）×scan_rthd×2
```

sql_size

次に示す計算式から求めてください。

計算式

```
sql_size＝32＋1,024
```

scan_buff_size

次に示す計算式から求めてください。

計算式

```
scan_buff_size＝ アーカイブチャンクオプションadb_arcv_scan_buff_sizeの指定値×1,024
```

offset_area

次に示す計算式から求めてください。

計算式

```
offset_area＝ （8×処理対象表の列数×(scan_buff_size×1,024÷処理対象表のデータ長)）÷1,024
```

scan_rthd

次に示す計算式から求めてください。

```
↓（アーカイブチャンクオプションadb_arcv_rthd_numの指定値－1）÷2↓
```

EXPF

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
EXPF＝ 552＋（512＋512）×exp_rthd
```

exp_rthd

次に示す計算式から求めてください。

```
↓（アーカイブチャンクオプションadb_arcv_rthd_numの指定値－1）÷2↓
```

PROC_EXPSQLS

adbarchivechunk コマンドを実行すると，チャンク内のデータをアーカイブするために，次に示す SQL 文が 1 回実行されます。

```
SELECT * FROM 対象表名;
```

「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(c)　変数 PROC_EXECSQLSZ の求め方」を参照して，検索系 SQL の実行時に使用されるメモリの所要量を，処理対象表ごとに求めてください。その中で，最も大きい値を代入してください。

SGMTGRP

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SGMTGRP＝ SGMTIDLIST＋SGMTGRPLIST＋BINTREE
```

SGMTIDLIST

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SGMTIDLIST＝ ↑(max_sgmtnum_in_chunks×16)÷1,024↑
```

max_sgmtnum_in_chunks

アーカイブ対象のチャンクの中で，最も大きな表のセグメント数

なお，インデクスのセグメント数は除きます。

チャンクごとの表のセグメント数は，次に示すどちらかの方法で確認してください。

- adbdbstatus コマンドに-d used オプションと-c table オプションを指定して，「DB エリア，表とインデクスの使用量情報」を出力してください。そして，出力項目の「Used_segments」の値を基に，チャンクごとの表のセグメント数を求めてください。
- アーカイブ対象のチャンクごとに，次に示す計算式で変数 *chunk_sgmtnum* の値を求めてください。求めた値の中から，最も大きい変数 *chunk_sgmtnum* の値を代入してください。

  計算式

$$chunk_sgmtnum = \left\uparrow \frac{CHBP}{SEGSIZE} \right\uparrow + \left\uparrow \frac{CHVP}{SEGSIZE} \right\uparrow$$

chunk_sgmtnum

チャンク内に格納された表のセグメント数

なお，インデクスのセグメント数は除きます。

CHBP

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合)」にある「*CHBP（i, k)*」を参照してください。

CHVP

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合)」にある「*CHVP（i, k)*」を参照してください。

SEGSIZE

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「*SEGSIZE*」を参照してください。

page_size

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」にある「表 6-3　DB エリアのページサイズ」を参照してください。

SGMTGRPLIST

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SGMTGRPLIST＝ ↑(SGMTGRPNUM×312)÷1,024↑
```

SGMTGRPNUM

変数*SGMTGRPNUM*は，変数*max_sgmtnum_in_chunks*の値で異なります。変数*max_sgmtnum_in_chunks*と変数*SGMTGRPNUM*の関係を，次の表に示します。

表 6-13　変数 max_sgmtnum_in_chunks と変数 SGMTGRPNUM の関係

項番	変数 max_sgmtnum_in_chunks の値	変数 SGMTGRPNUM の値（単位：個数）
1	max_sgmtnum_in_chunks ＜ 16	*SGMTGRPNUM*には，1 を代入してください。
2	16≦max_sgmtnum_in_chunks かつ max_sgmtnum_in_chunks ＜ 32	*SGMTGRPNUM*には，4 を代入してください。
3	32≦max_sgmtnum_in_chunks かつ max_sgmtnum_in_chunks ＜ 64	*SGMTGRPNUM*には，8 を代入してください。
4	64≦max_sgmtnum_in_chunks かつ max_sgmtnum_in_chunks ＜ 256	*SGMTGRPNUM*には，16 を代入してください。
5	256≦max_sgmtnum_in_chunks	*SGMTGRPNUM*は，次に示す計算式で求めてください。 SGMTGRPNUM＝ MAX(↓max_sgmtnum_in_chunks÷1,073,741,824 ÷1セグメントの容量（単位：バイト）↓ , 32)

BINTREE

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
BINTREE＝ ↑(SGMTGRPNUM×240)÷1,024↑
```

変数*SGMTGRPNUM*の計算式については，「表 6-13　変数 max_sgmtnum_in_chunks と変数 SGMTGRPNUM の関係」を参照してください。

(c)　変数 PROC_OUTFILE の求め方

変数*PROC_OUTFILE*は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_OUTFILE＝ ↑((120＋512)×(OUTFMAX－1,024))÷1,024↑
```

変数の説明

OUTFMAX

次に示す計算式から求めてください。

↓（*アーカイブチャンクオプションadb_arcv_rthd_numの指定値*－1）÷2↓

(d)　変数 PROC_DBUPDINF の求め方

変数 *PROC_DBUPDINF* は，次に示す計算式から求めてください。なお，マルチノード機能を使用するかどうかで，計算式が異なります。

マルチノード機能を使用しない場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
      ↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×122,888）÷1,024）↑
```

マルチノード機能を使用する場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
  129＋↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×122,888）÷1,024）↑
```

変数の説明

DBUPDINF_ENT_NUM

次に示す計算式から求めてください。

計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝ SGMTGRPNUM×scan_rthd_num
```

SGMTGRPNUM

「(b)　変数 PROC_MAC の求め方」の「表 6-13　変数 max_sgmtnum_in_chunks と変数 SGMTGRPNUM の関係」を参照してください。

scan_rthd_num

次に示す計算式から求めてください。

↓（*アーカイブチャンクオプションadb_arcv_rthd_numの指定値*－1）÷2↓

(2)　リアルスレッド固有メモリの所要量の求め方（adbarchivechunk コマンド実行時）

adbarchivechunk コマンド実行時のリアルスレッド固有メモリ（*RTHD_ARCCKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_ARCCKSZ＝ RTHD_ARCSQL＋RTHD_ARCDIRPATH
```

変数の説明

RTHD_ARCSQL

「(a)　変数 RTHD_ARCSQL の求め方」を参照して求めてください。

RTHD_ARCDIRPATH

「(b)　変数 RTHD_ARCDIRPATH の求め方」を参照して求めてください。

(a)　変数 RTHD_ARCSQL の求め方

変数*RTHD_ARCSQL* の求め方を次に示します。

adbarchivechunk コマンドを実行すると，チャンク内のデータをアーカイブするために，次に示す SQL 文が 1 回実行されます。

```
SELECT * FROM 対象表名;
```

「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」および「(g)　変数 RTHD_EXESQLDICSZ の求め方」を参照して，検索系 SQL の実行時に使用されるメモリの所要量を，処理対象表ごとに求めてください。その中で，最も大きい値を代入してください。

(b)　変数 RTHD_ARCDIRPATH の求め方

変数*RTHD_ARCDIRPATH* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_ARCDIRPATH＝ ↑SGMTGRPNUM×scan_rthd_num×640↑÷1,024
```

変数の説明

SGMTGRPNUM

「(1)　プロセス共通メモリの所要量の求め方（adbarchivechunk コマンド実行時）」の「(b)　変数 PROC_MAC の求め方」の「表 6-13　変数 max_sgmtnum_in_chunks と変数 SGMTGRPNUM の関係」を参照してください。

scan_rthd_num

次に示す計算式から求めてください。

```
↓（アーカイブチャンクオプションadb_arcv_rthd_numの指定値－1）÷2↓
```

(3)　ヒープメモリの所要量の求め方（adbarchivechunk コマンド実行時）

adbarchivechunk コマンド実行時のヒープメモリ（*HEAP_ARCCKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_ARCCKSZ＝ ∑(HEAP_ZLIB)※
```

注※

複数のadbarchivechunk コマンドを同時に実行する場合は，各adbarchivechunk コマンドが使用するヒープメモリの所要量を求めてください。そして，それぞれで求めたヒープメモリの所要量を合計してください。

変数の説明

HEAP_ZLIB

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_ZLIB＝ 262×archive_file_num
```

archive_file_num

アーカイブ状態のチャンクに対応しているアーカイブファイル数

複数のチャンクに対してadbarchivechunk コマンドを実行する場合は，各チャンクの中で，チャンクに対応するアーカイブファイル数が，最も多いアーカイブファイル数を代入してください。

adbdbstatus コマンドで「アーカイブ状態のチャンクのサマリ情報」を出力して，「Archive_file_num」を確認してください。「Archive_file_num」については，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』の『アーカイブ状態のチャンクのサマリ情報で出力される項目』の『アーカイブ状態のチャンクのサマリ情報で出力される項目の一覧』を参照してください。

6.3.17　adbunarchivechunk コマンド実行時のメモリ所要量の求め方

adbunarchivechunk コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_UNARCSZ*）
- リアルスレッド固有メモリ（*RTHD_UNARCCKSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_UNARCCKSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1) プロセス共通メモリの所要量の求め方（adbunarchivechunk コマンド実行時）

adbunarchivechunk コマンド実行時のプロセス共通メモリ（*PROC_UNARCSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_UNARCSZ＝
  ∑(PROC_UNARCCON＋（PROC_UMAC＋PROC_IDX※1＋PROC_IDXBUILD※1
    ＋PROC_RNGIDX※2＋PROC_INFILE※3＋PROC_DBUPDINF＋PROC_RTHDUPDINF）※4）※5
```

注※1

B-tree インデクスを定義した表に，adbunarchivechunk コマンドを実行する場合に加算します。

注※2

レンジインデクスを定義した表に，adbunarchivechunk コマンドを実行する場合に加算します。

注※3

チャンクに対応するアーカイブファイル数が，1,024 個を超える場合に加算します。

注※4

複数のチャンクに対してadbunarchivechunk コマンドを実行する場合は，チャンク単位にメモリ所要量を求めてください。そして，求めた値の最大値を加算してください。

注※5

複数のadbunarchivechunk コマンドを同時に実行する場合は，各adbunarchivechunk コマンドが使用するメモリ所要量を求めてください。そして，それぞれで求めたメモリ所要量を合計してください。

変数の説明

PROC_UNARCCON

「(a)　変数 PROC_UNARCCON の求め方」を参照して求めてください。

PROC_UMAC

「(b)　変数 PROC_UMAC の求め方」を参照して求めてください。

PROC_IDX

「(c)　変数 PROC_IDX の求め方」を参照して求めてください。

PROC_IDXBUILD

「(d)　変数 PROC_IDXBUILD の求め方」を参照して求めてください。

PROC_RNGIDX

「(e)　変数 PROC_RNGIDX の求め方」を参照して求めてください。

PROC_INFILE

「(f)　変数 PROC_INFILE の求め方」を参照して求めてください。

PROC_DBUPDINF

「(g)　変数 PROC_DBUPDINF の求め方」を参照して求めてください。

PROC_RTHDUPDINF

「(h)　変数 PROC_RTHDUPDINF の求め方」を参照して求めてください。

(a)　変数 PROC_UNARCCON の求め方

変数*PROC_UNARCCON* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_UNARCCON＝ ↑(250,000＋800)÷1,024↑＋PROC_AUDINFSZ※
```

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

(b)　変数 PROC_UMAC の求め方

変数*PROC_UMAC* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_UMAC＝ 3＋135＋DIC＋IOA＋LOD＋IMPORTBUF_CTL＋STS＋PAGEALLOC
```

変数の説明

DIC

「6.3.6　adbimport コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbimport コマンド実行時）」の「(a)　変数 PROC_IMPT の求め方」の変数*DIC* を参照してください。

IOA

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IOA＝ ↑（max_rowsz＋436＋448×col_num＋124×idx_num＋4×dba_num）÷1,024↑×load_rthd
```

max_rowsz

最大行長（単位：バイト）

最大行長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の行長*ROWSZ* の計算式を基に求めてください。

col_num

処理対象表の列数

idx_num

処理対象表に定義されたインデクス数

dba_num

処理対象表を格納する DB エリアの DB エリアファイル数

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

LOD

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
LOD＝
  ↑（（（read_size×read_buff_num＋decomp_buff_size＋1）×1,024×load_rthd）
  ＋（（5,200＋1,024,000※＋input_edit_area）×load_rthd）
  ＋（511×input_file_num））÷1,024↑
```

注※

チャンクをアーカイブ状態にしたあとに，列を追加した場合は，追加した列に対する情報を加算してください。

read_size

アンアーカイブチャンクオプションadb_unarcv_read_size の指定値

read_buff_num

3 を代入してください。

decomp_buff_size

アンアーカイブチャンクオプションadb_unarcv_decompress_buff_size の指定値

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

input_file_num

1,024 を代入してください。

input_edit_area

次に示す計算式から求めてください。

計算式

```
input_edit_area＝
    MIN ( MAX(↑input_recsize÷1,024↑×1,024 , 32,768) , 536,870,912)
```

input_recsize

次に示す計算式から求めてください。

計算式

```
input_recsize＝ Σ(DATASIZE)＋表定義列数＋1
```

Σ（*DATASIZE*）

処理対象表にある各列のデータ型のデータを文字形式で表した場合の最大長の合計

各データ型のデータを文字形式で表した場合の最大長については，次に示す表から求めてください。

表 6-14　各データ型のデータを文字形式で表した場合の最大長の一覧

項番	分類	データ型	文字形式のデータ最大長
1	数データ	INTEGER	23
2		SMALLINT	13
3		DECIMAL(*m*,*n*)	*m* ＋ 5
4		NUMERIC(*m*,*n*)	
5		DOUBLE PRECISION	511
6		FLOAT	
7	文字データ	CHAR(*n*)	*n*×2 ＋ 2
8		VARCHAR(*n*)	*n*×2 ＋ 2
9	日時データ	DATE	12
10		TIME(*p*)	8 ＋*p* ＋ 3
11		TIMESTAMP(*p*)	19 ＋*p* ＋ 3
12	バイナリデータ	BINARY(*n*)	*n*×8 ＋ 2
13		VARBINARY(*n*)	*n*×8 ＋ 2

（凡例）

m，*n*，*p*：マニュアル『HADB　SQL リファレンス』の『データ型の種類』を参照してください。

IMPORTBUF_CTL

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IMPORTBUF_CTL＝ ↑(248＋BUFBLK)÷1,024↑×load_rthd
```

BUFBLK

「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時)」の「(f)　変数 PROC_UPDSZ の求め方」の変数*BUFBLK* を参照してください。

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

STS

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STS＝
  ↑（256＋（1,185×idx_num）＋（2,067×（b_tree_idx_num＋range_idx_num）×load_rthd
    ＋2,067×text_idx_num×16）＋1,025）÷1,024↑
```

idx_num

処理対象表に定義されたインデクスの数

b_tree_idx_num

処理対象表に定義された B-tree インデクスの数

range_idx_num

処理対象表に定義されたレンジインデクスの数

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

text_idx_num

処理対象表に定義されたテキストインデクスの数

PAGEALLOC

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PAGEALLOC ＝ unarcv_rthd × 310
```

unarcv_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

(c)　変数 PROC_IDX の求め方

変数 *PROC_IDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDX＝
  ↑（idxf_buff_size×1,024＋（13,151＋（33,928＋（13×KEYSZ））
    ×idx_num×load_rthd）×cmd_d_opt＋（152＋rd_buff_size×load_rthd
    ×2×1,024＋2×ld_buff_size×1,024＋（KEYSZ＋CTRL）×text_idx_num）
    ×dividx_rthd）÷1,024↑
```

変数の説明

idxf_buff_size

1,024 を代入してください。

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

idx_num

処理対象表に定義された B-tree インデクスの数

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

cmd_d_opt

1 を代入してください。

rd_buff_size

アンアーカイブチャンクオプションadb_unarcv_dividx_rd_buff_size の指定値

ld_buff_size

アンアーカイブチャンクオプションadb_unarcv_dividx_wt_buff_size の指定値

CTRL

- インデクスのインデクス構成列が固定長の場合

  10 バイトを代入してください。

- インデクスのインデクス構成列に可変長を含む場合

12 バイトを代入してください。

text_idx_num

処理対象表に定義されたテキストインデクスの数

dividx_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

(d)　変数 PROC_IDXBUILD の求め方

変数*PROC_IDXBUILD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDXBUILD＝ PROC_IDXCREATE
```

変数の説明

PROC_IDXCREATE：次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXCREATE = \max_{1 \leqq k \leqq idx_num} PROC_IDXCREATE_MEM(k)$$

idx_num

処理対象表に定義された B-tree インデクスの数

PROC_IDXCREATE_MEM(k)

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDXCREATE_MEM(k)＝
  ↑（456＋256×idx_col_num＋（24＋144×↑idx_lv÷16↑）
   ×（dividx_rthd＋1）＋（KEYSZ＋（idx_col_num＋1）×2＋12）
   ×2＋page_size×2＋268）÷1,024↑
```

idx_col_num

B-tree インデクスのインデクス構成列の数

idx_lv

B-tree インデクスの段数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』の漸化式を計算して，*PIDX（n）* ＝ 1 となったときの *n* の値を代入してください。

ただし，*PIDX（1）* ＝ 1 の場合は，B-tree インデクスの段数は 2 としてください。

dividx_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

(e)　変数 PROC_RNGIDX の求め方

変数 *PROC_RNGIDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_RNGIDX＝ ↑64×rngidx_num÷1,024↑
```

変数の説明

rngidx_num

処理対象表に定義されたレンジインデクス数

(f)　変数 PROC_INFILE の求め方

変数 *PROC_INFILE* は，1 つのチャンクに対応したアーカイブファイルが 1,024 個を超える場合に加算する必要があります。次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_INFILE＝ ↑((16＋511)×(INFMAX－1,024))÷1,024↑
```

変数の説明

INFMAX

チャンクに対応したアーカイブファイル数

(g) 変数 PROC_DBUPDINF の求め方

変数*PROC_DBUPDINF* は，次に示す計算式から求めてください。なお，マルチノード機能を使用するかどうかで，計算式が異なります。

マルチノード機能を使用しない場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
  ↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×122,888）÷1,024）↑
```

マルチノード機能を使用する場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
  129＋↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×122,888）÷1,024）↑
```

変数の説明

DBUPDINF_ENT_NUM

次に示す計算式から求めてください。

計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝ archive_file_num
```

archive_file_num

アーカイブ状態のチャンクに対応しているアーカイブファイル数

adbdbstatus コマンドで「アーカイブ状態のチャンクのサマリ情報」を出力して，「Archive_file_num」を確認してください。「Archive_file_num」については，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』の『アーカイブ状態のチャンクのサマリ情報で出力される項目』の『アーカイブ状態のチャンクのサマリ情報で出力される項目の一覧』を参照してください。

(h) 変数 PROC_RTHDUPDINF の求め方

変数*PROC_RTHDUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_RTHDUPDINF＝ ↑（648＋（168×rthd_num））÷1,024↑
```

変数の説明

rthd_num

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

(2) リアルスレッド固有メモリの所要量の求め方（adbunarchivechunk コマンド実行時）

adbunarchivechunk コマンド実行時のリアルスレッド固有メモリ（*RTHD_UNARCCKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_UNARCCKSZ＝
        IMPORTBUF＋SORTIOBUF※1＋SORTBUF※1
        ＋RTHD_DATALOAD＋RTHD_IDXREC※1＋RTHD_IDXBUILD※1
        ＋RTHD_RNGIDX※2＋RTHD_TXTIDX※3＋RTHD_ARCDIRPATH
```

注※1

B-tree インデクスを定義した表に，adbunarchivechunk コマンドを実行する場合に加算します。

注※2

レンジインデクスを定義した表に，adbunarchivechunk コマンドを実行する場合に加算します。

注※3

テキストインデクスを定義した表に，adbunarchivechunk コマンドを実行する場合に加算します。

変数の説明

IMPORTBUF

「(a)　変数 IMPORTBUF の求め方」を参照して求めてください。

SORTIOBUF

「(b)　変数 SORTIOBUF の求め方」を参照して求めてください。

SORTBUF

「(c)　変数 SORTBUF の求め方」を参照して求めてください。

RTHD_DATALOAD

「(d)　変数 RTHD_DATALOAD の求め方」を参照して求めてください。

RTHD_IDXREC

「(e)　変数 RTHD_IDXREC の求め方」を参照して求めてください。

RTHD_IDXBUILD

「(f)　変数 RTHD_IDXBUILD の求め方」を参照して求めてください。

RTHD_RNGIDX

「(g)　変数 RTHD_RNGIDX の求め方」を参照して求めてください。

RTHD_TXTIDX

「(h)　変数 RTHD_TXTIDX の求め方」を参照して求めてください。

RTHD_ARCDIRPATH

「(i)　変数 RTHD_ARCDIRPATH の求め方」を参照して求めてください。

(a)　変数 IMPORTBUF の求め方

変数 *IMPORTBUF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IMPORTBUF＝
    ↑(IMP_SQBLK＋IMP_SQIO＋IMP_SQPGE＋IMP_SQHS)÷1,024↑
```

変数の説明

IMP_SQBLK

次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMP_SQBLK＝
   imp_blknum×(112＋↑IMP_BLKSZ÷imp_pagesize÷64↑×24)
```

imp_blknum

アンアーカイブチャンクオプションadb_unarcv_buff_blk_num の指定値

IMP_BLKSZ

次に示す計算式から求めてください。

計算式

```
IMP_BLKSZ＝ 4,096×1,024
```

imp_pagesize

次の DB エリアのページサイズのうち，最小の値（単位：バイト）

- 処理対象表が格納されている DB エリア
- 処理対象表に定義されている B-tree インデクスまたはテキストインデクスが格納されている DB エリア

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

IMP_SQIO

次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMP_SQIO＝ 568×imp_blknum
```

imp_blknum

アンアーカイブチャンクオプションadb_unarcv_buff_blk_num の指定値

IMP_SQPGE

次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMP_SQPGE＝
  (176＋imp_pagesize＋BUFLOG)×(IMP_BLKSZ÷imp_pagesize)×imp_blknum
```

imp_pagesize

次の DB エリアのページサイズのうち，最小の値（単位：バイト）

- 処理対象表が格納されている DB エリア
- 処理対象表に定義されている B-tree インデクスまたはテキストインデクスが格納されている DB エリア

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

BUFLOG

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFLOG* を参照してください。

IMP_BLKSZ

次に示す計算式から求めてください。

計算式

```
IMP_BLKSZ＝ 4,096×1,024
```

imp_blknum

アンアーカイブチャンクオプションadb_unarcv_buff_blk_num の指定値

IMP_SQHS

次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMP_SQHS＝ (8×imp_blknum)＋(40×imp_blknum)
```

imp_blknum

アンアーカイブチャンクオプションadb_unarcv_buff_blk_num の指定値

(b)　変数 SORTIOBUF の求め方

変数*SORTIOBUF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SORTIOBUF＝ sort_io_buff_size×sort_rthd
```

変数の説明

sort_io_buff_size

16 を代入してください。

sort_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

(c)　変数 SORTBUF の求め方

変数*SORTBUF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SORTBUF＝ （48＋sort_buff_size×1,024）×sort_rthd
```

変数の説明

sort_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

sort_buff_size

アンアーカイブチャンクオプションadb_unarcv_sort_buff_size の指定値※

注※

ソート処理では，作業用ワークファイルが作成されます。このファイルの容量を最小にする変数*sort_buff_size* の計算式を次に示します。変数*sort_buff_size* は，次に示す計算式を満たすようにしてください。ただし，メモリに余裕がない場合，またはほとんどのデータがインデクスキー順に並んでいる場合は，必要以上に大きな値を指定しないでください。

計算式

$$\text{sort_buff_size} \geqq \left\lceil \left(\frac{\text{REC_SIZE}+7}{2} + \sqrt{(\text{BLK}+8)\times\text{row_num}\times\text{KEY_INF} + \frac{\text{REC_SIZE}+7}{4}} + \text{COL_INFO} \right) \div 1{,}024 \right\rceil$$

REC_SIZE

次に示す計算式から求めてください。

計算式

```
REC_SIZE＝ （KEYSZ＋SYS1）＋SYS2
```

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

また，処理対象表に複数の B-tree インデクスが定義されている場合は，定義された B-tree インデクスの中で最大のキー長を使用してください。

SYS1

- 処理対象表に定義されたインデクスのインデクス構成列が固定長の場合

  10 を代入してください。
- 処理対象表に定義されたインデクスのインデクス構成列に可変長を含む場合

  12 を代入してください。

SYS2

- 処理対象表に定義されたインデクスが可変長の複数列インデクスの場合

  次に示す計算式から求めてください。

  計算式

  ```
  SYS2＝ （インデクス構成列数×4）
  ```
- そのほかのインデクスの場合

  0 を代入してください。

BLK

次に示す計算式から求めてください。

計算式

```
BLK＝ REC_SIZE＋（KEYSZ＋8）＋56
```

row_num

adbunarchivechunk コマンドで表に格納する行数

KEY_INF

次に示す計算式から求めてください。

計算式

```
KEY_INF＝ REC_SIZE＋（KEYSZ＋8）＋28
```

COL_INFO

次に示す計算式から求めてください。

計算式

```
COL_INFO＝ 2,112＋（MULTI_KEY_INFO×32）＋（KEYSZ＋8）
```

MULTI_KEY_INFO

- 処理対象表に定義された B-tree インデクスが可変長の複数列インデクスの場合
  次に示す計算式から求めてください。
  計算式

  ```
  MULTI_KEY_INFO＝ （インデクス構成列数×2）＋2
  ```

- そのほかの場合
  5 を代入してください。

(d) 変数 RTHD_DATALOAD の求め方

変数*RTHD_DATALOAD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_DATALOAD＝
   ↑↓（80＋192※1＋8×（col_num＋var_col_num）
   ＋32×var_col_num※2＋7）÷8↓×8÷1,024↑
```

注※1

処理対象表にレンジインデクスが定義されている場合に加算してください。

注※2

処理対象表が非 FIX 表の場合に加算してください。

変数の説明

col_num

処理対象表の列数

var_col_num

処理対象表にあるVARCHAR 型，およびVARBINARY 型の列数

(e) 変数 RTHD_IDXREC の求め方

変数*RTHD_IDXREC* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXREC＝
   ↑（386＋328×（idx_num－1）＋32×max_idx_col_num
   ＋（64＋buf_size×1,024）×2×idx_num）÷1,024↑
```

変数の説明

idx_num

処理対象表に定義された B-tree インデクスの数

max_idx_col_num

処理対象表に定義された B-tree インデクスのインデクス構成列数の最大値

buf_size

1,024 を代入してください。

(f)　変数 RTHD_IDXBUILD の求め方

変数 *RTHD_IDXBUILD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXBUILD = \max_{1 \leqq k \leqq idx_num} RTHD_IDXBUILD_MEM(k)$$

変数の説明

idx_num

処理対象表に定義された B-tree インデクスの数

RTHD_IDXBUILD_MEM(k)

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXBUILD_MEM(k) = \lceil (61,474 + \lceil MAX(255, KEYSZ) \div 2 \rceil \times 2 + 3,070 + \lceil KEYSZ + CTRL \div 8 \rceil \times 8 + MAX(255, KEYSZ) + 14 + \lfloor 8 + page_size \times 95 \div 100 \rfloor \times 2) \div 1,024 \rceil$$

page_size

B-tree インデクスが定義されているデータ用 DB エリアのページサイズ（単位：バイト）

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」にある「表 6-3　DB エリアのページサイズ」のデータ用 DB エリアのページサイズを参照して求めてください。

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL

- インデクス構成列のキー長がすべて固定長の場合
  10 バイトを代入してください。
- インデクス構成列のキー長に可変長を含む場合
  12 バイトを代入してください。

(g) 変数 RTHD_RNGIDX の求め方

変数 *RTHD_RNGIDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_RNGIDX＝ ↑（16＋128×rngidx_num）×load_rthd÷1,024↑
```

変数の説明

rngidx_num

処理対象表に定義されたレンジインデクスの数

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

(h) 変数 RTHD_TXTIDX の求め方

変数 *RTHD_TXTIDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_TXTIDX = \max_{1 \leqq k \leqq idx_num} RTHD_TXTIDXBUILD_MEM(k)$$

変数の説明

idx_num

処理対象表に定義されたテキストインデクスの数

RTHD_TXTIDXBUILD_MEM(k)

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_TXTIDXBUILD_MEM(k)＝
     buff_blk_size×buff_blk_num＋3,072＋↑(idx_div_num×24)÷1,024↑
     ＋txt_sort_buff_size×1,024
```

buff_blk_size

4,096 を代入してください。

buff_blk_num

アンアーカイブチャンクオプションadb_unarcv_buff_blk_num の指定値

idx_div_num

k 番目のテキストインデクスのインデクス構成列定義長

txt_sort_buff_size

アンアーカイブチャンクオプションadb_unarcv_txt_buff_size の指定値

(i)　変数 RTHD_ARCDIRPATH の求め方

変数*RTHD_ARCDIRPATH* には，次に示す値を代入してください。

値（単位：キロバイト）

```
RTHD_ARCDIRPATH= 1
```

(3)　ヒープメモリの所要量の求め方（adbunarchivechunk コマンド実行時）

adbunarchivechunk コマンド実行時のヒープメモリ（*HEAP_UNARCCKSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_UNARCCKSZ= ∑(HEAP_ZLIB+HEAP_SRTHMEM)※
```

注※

複数のadbunarchivechunk コマンドを同時に実行する場合は，各adbunarchivechunk コマンドが使用するヒープメモリの所要量を求めてください。そして，それぞれで求めたヒープメモリの所要量を合計してください。

変数の説明

HEAP_ZLIB

「(a)　変数 HEAP_ZLIB の求め方」を参照して求めてください。

HEAP_SRTHMEM

「(b)　変数 HEAP_SRTHMEM の求め方」を参照して求めてください

(a)　変数 HEAP_ZLIB の求め方

変数*HEAP_ZLIB* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_ZLIB＝ 42×load_rthd
```

変数の説明

load_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

(b)　変数 HEAP_SRTHMEM の求め方

変数 *HEAP_SRTHMEM* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_SRTHMEM＝ 505×sort_rthd
```

変数の説明

sort_rthd

次に示す計算式から求めてください。

アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値－1

6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方

adbreorgsystemdata コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

なお，ここでの「処理対象表」とは，「システム表（実表）」のことです。

■共有メモリ

- プロセス共通メモリ（*PROC_REORGSZ*）
- リアルスレッド固有メモリ（*RTHD_REORGSZ*）

■プロセスメモリ

- ヒープメモリ（*HEAP_REORGSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1) プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）

adbreorgsystemdata コマンド実行時のプロセス共通メモリ（*PROC_REORGSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_REORGSZ＝
    PROC_MNG＋PROC_IDX＋PROC_IDXBUILD
    ＋PROC_DBUPDINF＋PROC_RTHDUPDINF＋PROC_UNLOAD_BUFF
```

変数の説明

PROC_MNG

システム表の再編成の管理で使用するプロセス共通メモリ

「(a)　変数 PROC_MNG の求め方」を参照して求めてください。

PROC_IDX

B-tree インデクスを作成するときに使用するプロセス共通メモリ

「(b)　変数 PROC_IDX の求め方」を参照して求めてください。

PROC_IDXBUILD

B-tree インデクスを作成するときに使用するプロセス共通メモリ

「(c)　変数 PROC_IDXBUILD の求め方」を参照して求めてください。

PROC_DBUPDINF

DB エリア，表，インデクス，およびチャンク更新情報の格納で必要なプロセス共通メモリ

「(d)　変数 PROC_DBUPDINF の求め方」を参照して求めてください。

PROC_RTHDUPDINF

処理リアルスレッド更新情報

「(e)　変数 PROC_RTHDUPDINF の求め方」を参照して求めてください。

PROC_UNLOAD_BUFF

システム表の再編成時に使用するバッファ

「(f)　変数 PROC_UNLOAD_BUFF の求め方」を参照して求めてください。

(a) 変数 PROC_MNG の求め方

変数 *PROC_MNG* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_MNG＝
    3＋135＋DIC＋IOA＋LOD＋IMPORTBUF_CTL＋STS＋PAGEALLOC＋EXPF＋PROC_AUDINFSZ※
```

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

DIC

処理対象表の定義情報を取得するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
DIC＝ ↑（1,012＋512×col_num＋max_rowsz＋1,156×idx_num）÷1,024↑
```

col_num

処理対象表の列数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

max_rowsz

最大行長

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

idx_num

処理対象表に定義されたインデクス数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

表 6-15　システム表に対応する変数の値

項番	処理対象表名	変数 col_num の値	変数 var_col_num の値	変数 max_rowsz の値	変数 Σ(DATASIZE)の値	変数 idx_num の値	変数 max_idx_col_num の値	変数 max_keysz の値
1	STATUS_TABLES 表	12	2	318	1,556	1	2	203
2	STATUS_COLUMNS 表	11	6	32,588	257,088	1	3	304
3	STATUS_INDEXES 表	8	3	476	701	2	2	203
4	STATUS_CHUNKS 表	10	3	1,312	2,575	1	3	212
5	STATUS_SYNONYM_DICTIONARIES 表	10	6	2,108	4,152	1	1	121

IOA

データベースに格納するデータイメージを作成するためのメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IOA＝MAX（IOA_UNLOAD , IOA_RELOAD）
```

IOA_UNLOAD

アンロード処理で使用するメモリ

次に示す計算式から求めてください。

計算式

```
IOA_UNLOAD＝
    sql_size＋（scan_buff_size＋offset_area×2）×scan_rthd×2
```

sql_size

次に示す計算式から求めてください。

```
sql_size＝1＋1,024
```

scan_buff_size

次に示す計算式から求めてください。

```
scan_buff_size＝16×1,024
```

offset_area

次に示す計算式から求めてください。

```
offset_area＝
      （8×col_num×（scan_buff_size×1,024÷Σ(DATASIZE)））÷1,024
```

col_num

処理対象表の列数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

Σ（*DATASIZE*）

処理対象表にある各列のデータ型のデータを文字形式で表した場合の最大長の合計

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

scan_rthd

1 を代入してください。

IOA_RELOAD

リロード処理で使用するメモリ

次に示す計算式から求めてください。

計算式

```
IOA_RELOAD＝
 ↑（max_rowsz＋436＋448×col_num＋124×idx_num＋4×dba_num）
  ÷1,024↑×load_rthd
```

max_rowsz

最大行長

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

col_num

処理対象表の列数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

idx_num

処理対象表に定義されたインデクス数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

dba_num

1 を代入してください。

load_rthd

1 を代入してください。

LOD

リロード処理で必要となるメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
LOD＝
  ↑（（（read_size×read_buff_num）×1,024×load_rthd）
  ＋（（5,200＋input_edit_area）×load_rthd）
  ＋（511×input_file_num））÷1,024↑
```

read_size

1,024 を代入してください。

read_buff_num

3 を代入してください。

load_rthd

1 を代入してください。

input_edit_area

入力データの編集領域長

次に示す計算式から求めてください。

計算式

```
input_edit_area＝
MIN（MAX（↑input_recsize÷1,024↑×1,024 , 32,768） , 536,870,912）
```

input_recsize

入力レコード長

次に示す計算式から求めてください。

計算式

```
input_recsize＝Σ（DATASIZE） ＋ col_num×8×2
```

Σ（*DATASIZE*）

処理対象表にある各列のデータ型のデータを文字形式で表した場合の最大長の合計

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

col_num

処理対象表の列数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

input_file_num

1,024 を代入してください。

IMPORTBUF_CTL

システム表の再編成用バッファ制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
IMPORTBUF_CTL＝↑（248＋BUFBLK）÷1,024↑×load_rthd
```

BUFBLK

「6.3.4　通常運用時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（通常運用時）」の「(f)　変数 PROC_UPDSZ の求め方」の変数*BUFBLK* を参照してください。

load_rthd

1 を代入してください。

STS

ステータス情報を退避するためのメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
STS=
  ↑（256＋（1,185×idx_num）＋（2,067×idx_num×load_rthd）＋1,025）÷1,024↑
```

idx_num

処理対象表に定義されたインデクスの数

「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

load_rthd

1 を代入してください。

PAGEALLOC

リロード用ページ割り当て制御情報

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PAGEALLOC= MAX（dataload_rthd , dividx_rthd）×310
```

dataload_rthd

1 を代入してください。

dividx_rthd

1 を代入してください。

EXPF

ファイル処理で使用するメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
EXPF＝552＋512×exp_rthd
```

exp_rthd

1 を代入してください。

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数 *PROC_AUDINFSZ* を参照してください。

(b)　変数 PROC_IDX の求め方

変数 *PROC_IDX* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDX＝
  ↑（idxf_buff_size×1,024＋（13,151＋（33,928＋（13×max_keysz））
    ×idx_num×load_rthd）×cmd_d_opt
    ＋（152＋rd_buff_size×load_rthd×2×1,024＋2×ld_buff_size×1,024）
    ×dividx_rthd）÷1,024↑
```

変数の説明

idxf_buff_size

1,024 を代入してください。

max_keysz

処理対象表に定義されたインデクスの中で，最大のインデクスキー長（単位：バイト）

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

idx_num

処理対象表に定義されたインデクスの数

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

load_rthd

1 を代入してください。

cmd_d_opt

1 を代入してください。

rd_buff_size

1,024 を代入してください。

ld_buff_size

1,024 を代入してください。

dividx_rthd

1 を代入してください。

(c)　変数 PROC_IDXBUILD の求め方

変数 *PROC_IDXBUILD* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$PROC_IDXBUILD = \max_{k=1,\cdots idx_num} PROC_IDXCREATE_MEM(k)$$

変数の説明

idx_num

処理対象表に定義されたインデクスの数

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

PROC_IDXCREATE_MEM(k)

処理対象表に定義された *k* 番目のインデクスの一括作成に必要なメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_IDXCREATE_MEM(k)＝
  ↑（456＋256×idx_col_num＋（24＋144×↑idx_lv÷16↑）
   ×（dividx_rthd＋1）＋（KEYSZ＋（idx_col_num＋1）×2＋12）
   ×2＋page_size×2＋268）÷1,024↑
```

idx_col_num

インデクスのインデクス構成列の数

「表 6-16　システム表のインデクスに対応する変数 idx_col_num と KEYSZ の値」を参照して，インデクス名に対応する値を代入してください。

idx_lv

インデクスの段数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』の漸化式を計算して，*PIDX（n）* ＝ 1 となったときの *n* の値を代入してください。

ただし，*PIDX（1）* ＝ 1 の場合は，B-tree インデクスの段数は 2 としてください。

計算式を求める際，各変数には次の値を代入してください。

- *page_size*
  4,096 を代入してください。
- *pctfree*
  0 を代入してください。
- *dbarea_file_num*
  1 を代入してください。

dividx_rthd

1 を代入してください。

KEYSZ

インデクスのキー長（単位：バイト）

「表 6-16　システム表のインデクスに対応する変数 idx_col_num と KEYSZ の値」を参照して，インデクス名に対応する値を代入してください。

page_size

4,096 を代入してください。

表 6-16　システム表のインデクスに対応する変数 idx_col_num と KEYSZ の値

項番	処理対象表名	インデクス名	変数 idx_col_num の値	変数 KEYSZ の値
1	STATUS_TABLES 表	STATUSINDEXM01	2	203
2	STATUS_COLUMNS 表	STATUSINDEXM02	3	304
3	STATUS_INDEXES 表	STATUSINDEXM03	2	203
4		STATUSINDEXM04	2	203
5	STATUS_CHUNKS 表	STATUSINDEXM05	3	212
6	STATUS_SYNONYM_DICTIONARIES 表	STATUSINDEXS01	1	121

(d)　変数 PROC_DBUPDINF の求め方

変数 *PROC_DBUPDINF* は，次に示す計算式から求めてください。

マルチノード機能を使用しない場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
      ↑（（40＋↑（DBUPDINF_ENT_NUM÷1,024）↑×131,136）÷1,024）↑
```

マルチノード機能を使用する場合の計算式（単位：キロバイト）

```
PROC_DBUPDINF＝
      ↑（（80＋（↑（DBUPDINF_ENT_NUM÷1,024）↑＋1）×131,136）÷1,024）↑
```

変数の説明

DBUPDINF_ENT_NUM

DB エリア，表，インデクス，およびチャンク更新情報のエントリ数

マルチノード機能を使用するかどうかで，値が異なります。

マルチノード機能を使用しない場合

1 を代入してください。

マルチノード機能を使用する場合

次に示す計算式から求めてください。

計算式（単位：個数）

```
DBUPDINF_ENT_NUM＝5＋idx_num×10
```

idx_num

処理対象表に定義されたインデクスの数

(e)　変数 PROC_RTHDUPDINF の求め方

変数*PROC_RTHDUPDINF* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_RTHDUPDINF＝ ↑（664＋（192×rthd_num））÷1,024↑
```

変数の説明

rthd_num

1 を代入してください。

(f)　変数 PROC_UNLOAD_BUFF の求め方

変数*PROC_UNLOAD_BUFF* は，次に示す値を代入してください。

値（単位：キロバイト）

```
PROC_UNLOAD_BUFF＝ 8,192
```

(2)　リアルスレッド固有メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）

adbreorgsystemdata コマンド実行時に使用するリアルスレッド固有メモリ（*RTHD_REORGSZ*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_REORGSZ＝MAX（RTHD_UNLOADSZ , RTHD_RELOADSZ）
```

変数の説明

RTHD_UNLOADSZ

アンロード処理時にadbreorgsystemdata コマンドが使用するリアルスレッド固有メモリ

「(a)　変数 RTHD_UNLOADSZ の求め方」を参照して求めてください。

RTHD_RELOADSZ

リロード処理時にadbreorgsystemdata コマンドが使用するリアルスレッド固有メモリ

「(b)　変数 RTHD_RELOADSZ の求め方」を参照して求めてください。

(a)　変数 RTHD_UNLOADSZ の求め方

変数*RTHD_UNLOADSZ* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

処理対象表によって計算式が異なります。

・STATUS_TABLES表の場合
RTHD_UNLOADSZ＝ 809＋5×uthd_num＋6×MAX(128 ， uthd_num)＋SGTBL

・STATUS_COLUMNS表の場合
RTHD_UNLOADSZ＝ 33,452＋5×uthd_num＋511×MAX(128 ， uthd_num)＋33×SGTBL

・STATUS_INDEXES表の場合
RTHD_UNLOADSZ＝ 877＋5×uthd_num＋7×MAX(128 ， uthd_num)＋SGTBL

・STATUS_CHUNKS表の場合
RTHD_UNLOADSZ＝ 1,808＋5×uthd_num＋22×MAX(128 ， uthd_num)＋2×SGTBL

・STATUS_SYNONYM_DICTIONARIES表の場合
RTHD_UNLOADSZ＝ 2,612＋5×uthd_num＋35×MAX(128 ， uthd_num)＋3×SGTBL

変数の説明

uthd_num

サーバ定義adb_sys_uthd_num オペランドの指定値

SGTBL

処理対象表のセグメント数

次の表を参照して，処理対象表名に対応する値を求めてください。

表 6-17　システム表のセグメントサイズの見積もり

項番	処理対象表名	変数 SGTBL の求め方
1	STATUS_TABLES 表	↑(*STBLTABLESSIZE*÷64)↑
2	STATUS_COLUMNS 表	↑(*STBLCOLUMNSSIZE*÷64)↑
3	STATUS_INDEXES 表	↑(*STBLINDEXESSIZE*÷64)↑
4	STATUS_CHUNKS 表	↑(*STBLCHUNKSSIZE*÷64)↑
5	STATUS_SYNONYM_DICTIONARIES 表	↑(*STBLSYNONYMDICSIZE*÷64)↑

変数*STBLTABLESSIZE*，*STBLCOLUMNSSIZE*，*STBLINDEXESSIZE*，*STBLCHUNKSSIZE*，および*STBLSYNONYMDICSIZE* については，「5.12　システム表用 DB エリアの容量見積もり」を参照してください。

(b)　変数 RTHD_RELOADSZ の求め方

変数*RTHD_RELOADSZ* は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_RELOADSZ＝
        RELOADBUF＋SORTIOBUF＋SORTBUF
        ＋RTHD_DATALOAD＋RTHD_IDXREC＋RTHD_IDXBUILD
```

変数の説明

RELOADBUF

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RELOADBUF＝ ↑（*IMP_SQBLK*＋*IMP_SQIO*＋*IMP_SQPGE*＋*IMP_SQHS*）÷1,024↑

IMP_SQBLK

次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_SQBLK＝ *imp_blknum*×（112＋↑*IMP_BLKSZ*÷*imp_pagesize*÷64↑×24）

imp_blknum

64 を代入してください。

IMP_BLKSZ

次に示す計算式から求めてください。

IMP_BLKSZ＝ 4,096×1,024

imp_pagesize

4,096 を代入してください。

IMP_SQIO

次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_SQIO＝ 568×*imp_blknum*

IMP_SQPGE

次に示す計算式から求めてください。

計算式（単位：バイト）

IMP_SQPGE＝ （176＋*imp_pagesize*＋*BUFLOG*）×（*IMP_BLKSZ*÷*imp_pagesize*）×*imp_blknum*

BUFLOG

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFLOG* を参照してください。

IMP_SQHS

次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMP_SQHS＝ （8×imp_blknum）＋（40×imp_blknum）
```

SORTIOBUF

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SORTIOBUF＝ sort_io_buff_size×sort_rthd
```

sort_io_buff_size

16 を代入してください。

sort_rthd

1 を代入してください。

SORTBUF

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SORTBUF＝ （48＋ sort_buff_size×1,024）×sort_rthd
```

sort_buff_size

256 を代入してください。

RTHD_DATALOAD

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_DATALOAD＝
 ↑（↓（80＋8×（col_num＋var_col_num）＋32×var_col_num＋7）÷8↓×8）÷1,024↑
```

col_num

処理対象表の列数

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

var_col_num

処理対象表にあるVARCHAR 型，およびVARBINARY 型の列数

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

RTHD_IDXREC

次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_IDXREC＝
↑（386＋328×（*idx_num*−1）＋32×*max_idx_col_num*＋（64＋*buf_size*×1,024）×2×*idx_num*）÷1,024↑

idx_num

処理対象表に定義されたインデクスの数

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

max_idx_col_num

処理対象表に定義されたインデクスのインデクス構成列数の最大値

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

buf_size

1,024 を代入してください。

RTHD_IDXBUILD

次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$RTHD_IDXBUILD = \max_{1 \leqq k \leqq idx_num} RTHD_IDXBUILD_MEM(k)$$

idx_num

処理対象表に定義されたインデクスの数

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

RTHD_IDXBUILD_MEM（k）

処理対象表に定義された *k* 番目の B-tree インデクスの一括作成に必要なメモリ

次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_IDXBUILD_MEM（k）＝
    ↑（61,474＋↑MAX（255, KEYSZ）÷2↑×2＋3,070
    ＋↑KEYSZ＋CTRL÷8↑×8＋MAX（255, KEYSZ）
    ＋14＋↓8＋page_size×95÷100↓×2）÷1,024↑
```

KEYSZ

B-tree インデクスのキー長（単位：バイト）

「6.3.18　adbreorgsystemdata コマンド実行時のメモリ所要量の求め方」の「(1)　プロセス共通メモリの所要量の求め方（adbreorgsystemdata コマンド実行時）」の「(a)　変数 PROC_MNG の求め方」の「表 6-15　システム表に対応する変数の値」を参照して，処理対象表名に対応する値を代入してください。

CTRL

12 バイトを代入してください。

page_size

4,096 バイトを代入してください。

(3)　ヒープメモリの所要量の求め方（adbreorgsystemdata コマンド実行時）

adbreorgsystemdata コマンド実行時のヒープメモリ（*HEAP_REORGSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
HEAP_REORGSZ＝ 505×sort_rthd
```

変数の説明

sort_rthd

1 を代入してください。

6.3.19　adbclientdefmang コマンド実行時のメモリ所要量の求め方

adbclientdefmang コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_CLTDEFMNG*）
- リアルスレッド固有メモリ（*RTHD_CLTDEFMNG*）

■プロセスメモリ

- ヒープメモリ（*HEAP_CLTDEFMNG*）

それぞれのメモリ所要量の計算式を次に示します。

(1) プロセス共通メモリの所要量の求め方（adbclientdefmang コマンド実行時）

adbclientdefmang コマンド実行時のプロセス共通メモリ（*PROC_CLTDEFMNG*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_CLTDEFMNG ＝ ↑(*CMI* ＋ *CDM_WORK*)÷1,024↑ ＋ *PROC_AUDINFSZ*※

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

CMI

「6.3.3 HADB サーバ開始時のメモリ所要量の求め方」の「(3) プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(a) 変数 SCI の求め方」の変数 *CMI* を参照してください。

変数 *CMI* の計算式に出てくる各変数は，adbclientdefmang コマンド実行時のクライアント管理定義ファイルの内容によって変わります。

なお，adbclientdefmang コマンドの実行中は，次に示す変数 *CMI* のメモリ領域を同時に確保します。

- adbclientdefmang コマンド実行前に確保していた変数 *CMI* のメモリ領域

  変更前のクライアント管理定義ファイルの内容です。ただし，adbclientdefmang コマンド実行時に，クライアント定義の集中管理機能を使用していなかった場合は存在しません。

- adbclientdefmang コマンド実行時に確保した変数 *CMI* のメモリ領域

  変更後のクライアント管理定義ファイルの内容です。

また，adbclientdefmang コマンドの実行結果によって，解放されるメモリ領域が異なります。

- adbclientdefmang コマンドが正常終了した場合

  HADB サーバは，adbclientdefmang コマンド実行前に確保していた変数 *CMI* のメモリ領域を解放します。

- adbclientdefmang コマンドがエラーになった場合

  HADB サーバは，adbclientdefmang コマンド実行時に確保した変数 *CMI* のメモリ領域を解放します。

CDM_WORK

次に示す計算式から求めてください。

計算式（単位：バイト）

CDM_WORK ＝ ↑8×*authid_num*÷32↑×32 ＋ 64

authid_num

クライアント管理定義のadbclientmang オペランドに指定された認可識別子の総数

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照してください。

(2)　リアルスレッド固有メモリの所要量の求め方（adbclientdefmang コマンド実行時）

adbclientdefmang コマンド実行時のリアルスレッド固有メモリ（*RTHD_CLTDEFMNG*）の所要量は，次に示す計算式から求めてください。

変数*RTHD_CLTDEFMNG* は，マルチノード機能を使用する場合に求めてください。マルチノード機能を使用しない場合は求める必要はありません。

計算式（単位：キロバイト）

$$RTHD_CLTDEFMNG = \uparrow MLT_COMP \div 1{,}024 \uparrow$$

変数の説明

MLT_COMP

次に示す計算式から求めてください。

計算式（単位：バイト）

$$MLT_COMP = \uparrow (8 + 608 \times authid_num) \div 32 \uparrow \times 32 + 64$$

authid_num

クライアント管理定義のadbclientmang オペランドに指定された認可識別子の総数

(3)　ヒープメモリの所要量の求め方（adbclientdefmang コマンド実行時）

adbclientdefmang コマンド実行時のヒープメモリ（*HEAP_CLTDEFMNG*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$HEAP_CLTDEFMNG = \uparrow (MNGANA + CLTANA) \div 1{,}024 \uparrow$$

変数の説明

MNGANA

次に示す計算式から求めてください。

計算式（単位：バイト）

MNGANA ＝ 512×*cltmng_num*

cltmng_num

クライアント管理定義ファイルに指定されたクライアント管理定義のadbclientmang オペランドの総数

CLTANA

次に示す値を代入してください。

値（単位：バイト）

CLTANA ＝ 10,240

6.3.20　adbsyndict コマンド実行時のメモリ所要量の求め方

adbsyndict コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_SYNDICTSZ*）
- リアルスレッド固有メモリ（*RTHD_SYNDICTSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1)　プロセス共通メモリの所要量の求め方（adbsyndict コマンド実行時）

adbsyndict コマンド実行時のプロセス共通メモリ（*PROC_SYNDICTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

PROC_SYNDICTSZ ＝ SYNDICDEFSZ ＋ PROC_AUDINFSZ※

注※

監査証跡機能が有効な場合に加算してください。

変数の説明

SYNDICDEFSZ

同義語辞書の情報を管理するメモリ

次の値を代入してください。

値（単位：キロバイト）

```
SYNDICDEFSZ ＝ 3
```

PROC_AUDINFSZ

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」にある変数*PROC_AUDINFSZ* を参照して求めてください。

(2)　リアルスレッド固有メモリの所要量の求め方（adbsyndict コマンド実行時）

adbsyndict コマンド実行時のリアルスレッド固有メモリ（*RTHD_SYNDICTSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_SYNDICTSZ＝168,800＋↑（↑（syn_group_num÷100,000）↑×2,400÷1,024）↑＋SYNDICACCSZ
```

変数の説明

syn_group_num

同義語グループの数（単位：個）

SYNDICACCSZ

同義語辞書の情報を操作するメモリ

次の値を代入してください。

値（単位：キロバイト）

```
SYNDICACCSZ ＝ 6
```

6.3.21　adbaudittrail コマンド実行時のメモリ所要量の求め方

adbaudittrail コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。各メモリの所要量を求めてください。

■共有メモリ

- プロセス共通メモリ（*PROC_AUDTRAILSZ*）
- リアルスレッド固有メモリ（*RTHD_AUDTRAILSZ*）

それぞれのメモリ所要量の計算式を次に示します。

(1) プロセス共通メモリの所要量の求め方（adbaudittrail コマンド実行時）

adbaudittrail コマンド実行時のプロセス共通メモリ（*PROC_AUDTRAILSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
PROC_AUDTRAILSZ ＝ AUDINF ＋ AUDTBLDEFSZ
```

変数の説明

AUDINF

監査証跡の管理情報

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(r)　変数 AUDINF の求め方」を参照して求めてください。

監査証跡機能を有効にすると変数*AUDINF* で求めた値が確保され，無効にすると解放されます。

AUDTBLDEFSZ

監査対象定義の管理情報

--start オプションを指定してadbaudittrail コマンドを実行する場合だけ加算してください。--start オプション以外を指定する場合は，*AUDTBLDEFSZ* の加算は不要です。

次に示す値を代入してください。

値（単位：キロバイト）

```
AUDTBLDEFSZ＝ 4
```

(2) リアルスレッド固有メモリの所要量の求め方（adbaudittrail コマンド実行時）

adbaudittrail コマンド実行時のリアルスレッド固有メモリ（*RTHD_AUDTRAILSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
RTHD_AUDTRAILSZ ＝ AUDTHDINF ＋ AUDTBLACCSZ
```

変数の説明

AUDTHDINF

「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(4)　リアルスレッド固有メモリの所要量の求め方（HADB サーバ開始時）」の変数*AUDTHDINF* を参照して求めてください。

監査証跡機能を有効にすると変数*AUDTHDINF* で求めた値が確保され，無効にすると解放されます。

AUDTBLACCSZ

監査対象定義の操作情報

--start オプションを指定してadbaudittrail コマンドを実行する場合だけ加算してください。--start オプション以外を指定する場合は，*AUDTBLACCSZ* の加算は不要です。

次の値を代入してください。

値（単位：キロバイト）

AUDTBLACCSZ＝ 1

6.3.22 adbconvertaudittrailfile コマンド実行時のメモリ所要量の求め方

adbconvertaudittrailfile コマンドを実行する場合，HADB サーバは次に示すメモリを使用します。

■共有メモリ

- リアルスレッド固有メモリ（*RTHD_CONVERTAUDSZ*）

adbconvertaudittrailfile コマンド実行時に使用するリアルスレッド固有メモリ（*RTHD_CONVERTAUDSZ*）の所要量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

RTHD_CONVERTAUDSZ = *AUDREADSZ* + *AUDBLOCKSZ*

変数の説明

AUDREADSZ

監査証跡ファイルの管理領域

次に示す値を代入してください。

値（単位：キロバイト）

AUDREADSZ = 1

AUDBLOCKSZ

監査証跡ファイルの作業領域

次に示す値を代入してください。

値（単位：キロバイト）

AUDBLOCKSZ = 262,144

6.4 HADB サーバの使用時に単調増加するファイル

ここでは，HADB サーバの使用時に単調増加するファイルについて説明します。

HADB サーバの使用時に単調増加するファイルは，次に示すディレクトリに作成されます。

- サーバディレクトリ
- DB ディレクトリ
- 監査証跡の出力先ディレクトリ※

注※

監査証跡機能を使用する場合に必要なディレクトリです。

> **重要**
>
> 単調増加するファイルは，ファイルサイズおよびファイル数の上限がありません。そのため，単調増加するファイルによって，ディスクの空き容量が不足しないように注意する必要があります。

サーバディレクトリで単調増加するファイルを次の表に示します。

表 6-18 サーバディレクトリで単調増加するファイル

項番	ファイル名	ファイルの説明	ファイルの削除方法
1	$ADBDIR/spool/adbdump*YYYYMMDDhhmmss.サーバプロセスのプロセス ID*※1	HADB ダンプファイル	• HADB サーバの正常終了時に削除されます。 • adbinfosweep コマンドを実行すると削除されます
2	$ADBDIR/spool/adbdumperror*YYYYMMDDhhmmssSSSSSS_TTTTTTTTTTTTTTTTTTTT.サーバプロセスのプロセス ID*※1，※2		adbinfosweep コマンドを実行すると削除されます。

注※1

ファイル名の *YYYYMMDDhhmmss* は，ファイルを生成した時刻になります。

注※2

ファイル名の *SSSSSS* は，HADB ダンプファイルを生成した時刻のマイクロ秒になります。また，*TTTTTTTTTTTTTTTTTTTT* は，HADB サーバの内部矛盾エラーを検知したリアルスレッドのスレッド ID になります。

DB ディレクトリで単調増加するファイルを次の表に示します。

表 6-19　DB ディレクトリで単調増加するファイル

項番	ファイル名	ファイルの説明	ファイルの削除方法
1	$DBDIR/SPOOL/core.*サーバプロセスのプロセス ID*※1	障害情報（コアファイル）	adbinfosweep コマンドを実行すると削除されます。
2	$DBDIR/ADBWORK/*xxxxx*※2，※3	コマンド実行時に作成される作業用一時ファイル※4	実行したコマンドが完了すると削除されます。

注※1

サーバ定義adb_core_path オペランドを指定した場合は，$DBDIR/SPOOL ディレクトリ下ではなく，adb_core_path オペランドに指定したディレクトリ下に，障害情報（コアファイル）が出力されます。

注※2

ファイル名の *xxxxx* は，任意の値になります。

注※3

実行するコマンドの-w オプションを省略した場合に，このディレクトリに作業用一時ファイルが作成されます。実行するコマンドの-w オプションを指定した場合は，-w オプションに指定したディレクトリに，作業用一時ファイルが作成されます。

注※4

次のコマンドを実行するときに，作業用一時ファイルが作成されます。

- adbimport コマンド
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド

監査証跡の出力先ディレクトリで単調増加するファイルを次の表に示します。

表 6-20　監査証跡の出力先ディレクトリで単調増加するファイル

項番	ファイル名	ファイルの説明	ファイルの削除方法
1	$AUDITDIR/adbaud-*YYYYMMDD-hhmmss-nnn*.aud※1，※2	監査証跡ファイル※3	OS のcp コマンドとrm コマンドで，監査証跡の保存先ディレクトリに監査証跡ファイルを移動してください。 詳細については，「12.3.1　監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）」を参照してください。

注※1

- ファイル名の$AUDITDIR は，監査証跡の出力先ディレクトリを示します。サーバ定義の adb_audit_log_path オペランドに指定したパス名です。
- ファイル名の *YYYYMMDD* と *hhmmss* は，監査証跡ファイルをリネームした日時になります。
- ファイル名の *nnn* は，監査証跡ファイルをリネームした日時のミリ秒になります。

注※2

マルチノード機能を使用している場合，ファイル名は$AUDITDIR/adbaud-*YYYYMMDD-hhmmss-nnn-N*.aud になります。*N* は，監査証跡ファイルが出力されたノードのノード番号を示します。

注※3

サーバ定義のadb_audit_log_max_num オペランドに0 を指定した場合，またはadb_audit_log_max_num オペランドを省略した場合は，上限なく監査証跡ファイルが作成されます。

6.5　サーバメッセージログファイルの容量見積もり

ここでは，サーバメッセージログファイルの容量を見積もる方法について説明します。サーバメッセージログファイルについては，「10.4.1　メッセージの出力先」の「(3)　サーバメッセージログファイル」を参照してください。

サーバメッセージログファイルの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

サーバメッセージログファイルの容量＝　(*env_msglog_size*×1,024×4)

変数の説明

- *env_msglog_size*
  環境変数ADBMSGLOGSIZE の指定値
  指定していない場合は，16 を仮定してください。

6.6　障害情報（コアファイル）の容量見積もり

ここでは，障害情報（コアファイル）の容量を見積もる方法について説明します。

障害情報（コアファイル）の容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

障害情報（コアファイル）の容量＝ adb_core_size＋sysd_core_size

変数の説明

adb_core_size

次のどちらかのコアファイルのサイズです（単位：キロバイト）。この変数には，2,097,152 を代入してください。

- サーバ定義のadb_core_path オペランドに指定したディレクトリ下に格納されるコアファイル
- $DBDIR/SPOOL ディレクトリ下に格納されるコアファイル

sysd_core_size

コアファイルの出力が systemd-coredump によって制御されている場合に，HADB サーバが/var/lib/systemd/coredump ディレクトリ下に出力するコアファイルのサイズです（単位：キロバイト）。この変数に代入する値は，次の条件から選択してください。

- コアファイルの出力が systemd-coredump によって制御されていて，かつ LZ4 圧縮が有効の場合
  この変数には，20,972 を代入してください。
- コアファイルの出力が systemd-coredump によって制御されていて，かつ LZ4 圧縮が無効（無圧縮）の場合
  この変数には，2,097,152 を代入してください。
- コアファイルの出力が systemd-coredump によって制御されていない場合
  この変数には，0 を代入してください。

メモ

- コアファイルの出力が systemd-coredump によって制御されているかどうかを確認する方法は，「8.2.1　インストール前に実施する作業」の「(9)　設定ファイルの変更および作成（コアファイルの出力が systemd-coredump で制御されている場合）」を参照してください。
- systemd-coredump によるコアファイルの LZ4 圧縮が有効であるかどうかは，/etc/systemd/coredump.conf 内にあるCoredump セクション中のCompress オプションの指定値で確認できます。Compress オプションにyes を指定しているか，またはCompress オプションがコメントになっている場合は，LZ4 圧縮が有効になっています。Compress オプションにno を指定している場合は，LZ4 圧縮が無効（無圧縮）になっています。

6.7 HADB ダンプファイルの容量見積もり

HADB ダンプファイルは，HADB サーバのサーバプロセスが異常終了したときに出力されるトラブルシュート情報のことです。

HADB ダンプファイルの容量は，次に示す計算式から求めてください。

計算式（単位：メガバイト）

HADBダンプファイルの容量＝ 25,600＋*max_users*×106＋*rthd_num*×1,821

変数の説明

max_users

サーバ定義adb_sys_max_users オペランドの指定値

rthd_num

サーバ定義adb_sys_rthd_num オペランドの指定値

6.8　統計ログファイルの容量見積もり

ここでは，統計ログファイルの容量を見積もる方法について説明します。

統計ログファイルの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

統計ログファイルの容量 ＝ *sta_log_max_size* ×*sta_log_size_unit* ×1,024×4＋1,024

変数の説明

sta_log_max_size

サーバ定義adb_sta_log_max_size オペランドの指定値

「7.2.7　統計情報に関するオペランド（set 形式）」の adb_sta_log_max_size オペランドを参照してください。

sta_log_size_unit

- サーバ定義adb_sta_log_size_unit オペランドを省略した場合，またはサーバ定義adb_sta_log_size_unit オペランドにG を指定した場合

  1,024 としてください。

- サーバ定義adb_sta_log_size_unit オペランドにM を指定した場合

  1 としてください。

サーバ定義adb_sta_log_size_unit オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式）」の adb_sta_log_size_unit オペランドを参照してください。

なお，統計ログファイルの出力先は，サーバ定義adb_sta_log_path オペランドの指定有無によって異なります。

- サーバ定義adb_sta_log_path オペランドを指定していない場合

  統計ログファイルは，$ADBDIR/spool ディレクトリ下に出力されます。

- サーバ定義adb_sta_log_path オペランドを指定している場合

  統計ログファイルは，サーバ定義adb_sta_log_path オペランドに指定したディレクトリ下に出力されます。

サーバ定義adb_sta_log_path オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式）」の adb_sta_log_path オペランドを参照してください。

6.9　SQL トレースファイルの容量見積もり

ここでは，SQL トレースファイルの容量を見積もる方法について説明します。

SQL トレースファイルの容量は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
SQLトレースファイルの総容量＝ sqltxt_size×1,024×8
```

変数の説明

sqltxt_size

サーバ定義adb_sql_trc_txtfile_size オペランドの指定値

「7.2.5　SQL 文に関するオペランド（set 形式）」の adb_sql_trc_txtfile_size オペランドを参照してください。

6.10　更新行のカラム化機能に必要なファイルの容量見積もり

ここでは，更新行のカラム化機能に必要なファイル容量について説明します。

更新行のカラム化機能で必要なファイルの総容量（*COLUMNIZELOG*）は，次のとおりです。

ファイルの総容量（単位：キロバイト）

COLUMNIZELOG＝ 131,072

メモ

- 更新行のカラム化機能に必要なファイルは，更新行のカラム化機能が有効かどうかに関係なく，HADB サーバの開始時に作成されます。
- 更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

6.11　アクセスパス探索情報ログファイルの容量見積もり

ここでは，アクセスパス探索情報ログファイルの容量について説明します。

アクセスパス探索情報ログファイルの総容量（*OPTLOG*）は，次のとおりです。

ファイルの総容量（単位：キロバイト）

OPTLOG＝ 131,072 × 2

アクセスパス探索情報ログファイルは，システムが使用するファイルで，HADB サーバの開始時に作成されます。このファイルには，SQL 文の実行時にアクセスパスの構築に関する保守情報が出力されます。

6.12　システムログファイルの容量見積もり

ここでは，システムログファイルの容量の見積もり方法について説明します。

システムログファイルは，マスタログファイルおよびユーザログファイル群で構成されます。マスタログファイルは自動的に，ユーザログファイルはサーバ定義の指定値に従って，HADB サーバを開始したときにそれぞれ初期化容量分だけ，DB ディレクトリ下（$DBDIR/ADBSYS/ADBSLG）に作成されます。そのため，システムログファイルの容量を求める場合，マスタログファイルおよびユーザログファイルの容量をそれぞれ求める必要があります。

システムログの増加によって，システムログファイルの容量が初期化した容量を超えた場合，システムログファイルは自動的に拡張されます。システムログファイルが拡張すると更新性能への影響が大きいため，システムログファイルの容量見積もりを実施して，必要な初期化容量を指定してください。

システムログファイルの容量を求める計算式を次に示します。なお，システムログファイル中に余分な空き領域が発生することがあるため，ここで求めた容量の 1.2 倍以上の容量を準備してください。

計算式（単位：キロバイト）

システムログファイルの容量 ＝ *MSTLOG*＋*USRLOG*×*user_file_num*

変数の説明

- *MSTLOG*
  マスタログファイルの容量（単位：キロバイト）
  「6.12.1　マスタログファイルの容量の求め方」を参照して求めてください。
- *USRLOG*
  ユーザログファイルの容量（単位：キロバイト）
  「6.12.2　ユーザログファイルの容量の求め方」を参照して求めてください。
- *user_file_num*
  ユーザログファイル数（単位：個）
  「6.12.15　ユーザログファイル数の求め方」を参照して求めてください。

6.12.1　マスタログファイルの容量の求め方

マスタログファイルにはイベントログが出力されます。HADB サーバの稼働中はイベントログが出力され続けるため，マスタログファイルの容量は増加し続けます。ただし，HADB サーバの稼働中に，次に示す条件をすべて満たす契機がある場合，マスタログファイルの容量が自動的に縮小されます。

- マスタログファイルの容量が 2 メガバイトを超えている場合
- ほかの更新トランザクションが存在しないときに更新トランザクションが決着した場合

また，HADB サーバを再起動したときには，前回稼働時に出力されたイベントログはマスタログファイルから削除されます。

重要

マスタログファイルが作成されるディスクの空き容量が不足すると，トランザクションを決着または回復できなくなり，HADB サーバが異常終了することがあります。そのため，マスタログファイルの容量を正確に見積もって十分なディスク容量を確保するようにしてください。

マスタログファイルの容量は次に示す計算式から求めてください。

ヒント

計算式の変数の説明中に出てくる「見積もり期間」は，次のどちらかから求めてください。

- HADB サーバの稼働期間
- マスタログファイルの容量が自動的に縮小する契機が生じる期間

例えば，サーバのメンテナンスなどで HADB サーバを半年に一度停止することがある場合は，見積もり期間を半年としてください。マスタログファイルが自動的に縮小する契機が毎日ある場合は，見積もり期間を 1 日としてください。

計算式（単位：キロバイト）

マスタログファイルの容量＝

$$\mathrm{MAX}\left\{16,\left\lceil\frac{\left(\mathrm{tran_count}+\left(\sum_{i=1}^{\mathrm{utlexec_count}}\mathrm{UTi}\right)\right)\times 184+\mathrm{arcdir_remove_count}\times 712}{65,512}\right\rceil+1\right\}\times 64$$

変数の説明

tran_count

見積もり期間中の更新トランザクションの実行回数

arcdir_remove_count

見積もり期間中に，アーカイブディレクトリ下のファイルが削除される回数

アーカイブマルチチャンク表に対して，対象の SQL 文とコマンドを実行した場合に，アーカイブディレクトリ下が削除される回数を代入してください。アーカイブディレクトリ下が削除される回数を，次の表に示します。

表 6-21　アーカイブディレクトリ下が削除される回数

項番	実行する SQL 文とコマンド	アーカイブディレクトリ下が削除される回数
1	• DROP SCHEMA 文 • REVOKE 文のSCHEMA	削除対象のスキーマに存在するアーカイブマルチチャンク表数を代入してください。
2	• ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合） • DROP TABLE 文 • TRUNCATE TABLE 文	1 を代入してください。
3	PURGE CHUNK 文	削除対象のチャンク数を代入してください。
4	adbunarchivechunk コマンド	処理対象のチャンク数を代入してください。

utlexec_count

見積もり期間中のコマンドの実行回数

次に示すコマンドを実行した回数の合計になります。

- adbimport コマンド
- adbidxrebuild コマンド
- adbmodarea コマンド
- adbmergechunk コマンド
- adbgetcst コマンド
- adbchgchunkcomment コマンド
- adbchgchunkstatus コマンド
- adbarchivechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド
- adbsyndict コマンド

UTi

見積もり期間中に実行したコマンドごとのデータベース更新スレッド数

コマンドごとのデータベース更新スレッド数は，次に示す計算式から求めてください。

- 計算式（adbimport コマンドを実行したとき）

```
UTi（adbimportコマンド実行時）＝
  （インポートオプションadb_import_rthd_numの指定値＋1）×9
```

- 計算式（adbidxrebuild コマンドを実行したとき）

```
UTi（adbidxrebuildコマンド実行時）＝
  （インデクス再作成オプションadb_idxrebuild_rthd_num＋1）×6
```

- 計算式（adbmodarea コマンドを実行したとき）

```
UTi（adbmodareaコマンド実行時）＝ 2
```

- 計算式（adbmergechunk コマンドを実行したとき）

```
UTi（adbmergechunkコマンド実行時）＝
  （マージチャンクオプションadb_mergechunk_rthd_numの指定値＋1）×2
```

- 計算式（adbgetcst コマンドを実行したとき）

```
UTi（adbgetcstコマンド実行時）＝ 1
```

- 計算式（adbchgchunkcomment コマンドを実行したとき）

```
UTi（adbchgchunkcommentコマンド実行時）＝ 1
```

- 計算式（adbchgchunkstatus コマンドを実行したとき）

```
UTi（adbchgchunkstatusコマンド実行時）＝ 1
```

- 計算式（adbarchivechunk コマンドを実行したとき）

```
UTi（adbarchivechunkコマンド実行時）＝
  （アーカイブチャンクオプションadb_arcv_rthd_numの指定値＋1）×処理対象のチャンク数
```

- 計算式（adbunarchivechunk コマンドを実行したとき）

```
UTi（adbunarchivechunkコマンド実行時）＝
  （アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値＋1）×11
  ×処理対象のチャンク数
```

- 計算式（adbreorgsystemdata コマンドを実行したとき）

```
UTi（adbreorgsystemdataコマンド実行時）＝ 10
```

- 計算式（adbsyndict コマンドを実行したとき）

```
UTi（adbsyndictコマンド実行時）＝ 1
```

メモ

各コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

6.12.2　ユーザログファイルの容量の求め方

ユーザログファイルには，データベースの更新処理時にデータベースの更新履歴情報であるユーザログが格納されます。ロールバック処理が発生した場合，HADB サーバはユーザログを使用してトランザクションを回復します。

ユーザログファイルの容量は次に示す計算式から求めてください。

計算式（単位：キロバイト）

ユーザログファイルの容量＝

$$\mathrm{MAX}\left\{\mathrm{user_file_init_size}\times 1{,}024,\left(\left\uparrow\frac{\mathrm{max_user_log}}{65{,}512}\right\uparrow+1\right)\times 64\right\}$$

変数の説明

- *user_file_init_size*

  サーバ定義adb_log_usrfile_size オペランドの指定値

- *max_user_log*

  ユーザログの出力量が最も多いトランザクションのユーザログ容量（単位：バイト）

変数*max_user_log* を求める場合，まず，次に示す処理ごとにトランザクションのユーザログ容量を求めてください。その中で，最も大きいユーザログ容量の値を変数*max_user_log* に代入してください。

- 「6.12.3　定義系 SQL 実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.4　adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.5　adbidxrebuild コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.6　adbmergechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.8　TRUNCATE TABLE 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.9　PURGE CHUNK 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.10　adbchgchunkstatus コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.11　adbarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.12　adbunarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」
- 「6.12.13　adbreorgsystemdata コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」

- 「6.12.14　更新行のカラム化機能のメンテナンス処理に必要なユーザログ容量（変数 max_user_log）の求め方」

ヒント

ユーザログはトランザクション処理が終了したときに削除されます。そのため，ユーザログの出力量が最も多いトランザクションのユーザログ容量を基準にして計算してください。

メモ

上記以外の SQL 文およびコマンドについては，出力されるユーザログ容量がとても少ないため，見積もりは不要です。

6.12.3　定義系 SQL 実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

ここでは，次に示す定義系 SQL を実行したときに出力されるユーザログ容量の求め方について説明します。

- CREATE TABLE 文（実表の定義）
- CREATE INDEX 文（インデクスの定義）
- ALTER TABLE 文のCHANGE OPTION CHUNK（チャンク数の最大値の変更）
- ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）
- DROP USER 文（HADB ユーザの削除）
- DROP SCHEMA 文（スキーマの削除）
- DROP TABLE 文（実表の削除）
- DROP INDEX 文（インデクスの削除）
- REVOKE 文のSCHEMA（スキーマ操作権限の取り消し）

なお，上記以外の定義系 SQL については，ユーザログファイルの容量を超えるユーザログは出力されないため，見積もりは不要です。

(1)　CREATE TABLE 文実行時に出力されるユーザログ容量の求め方

CREATE TABLE 文実行時に出力されるユーザログ容量（変数 *CRTTBLLOG*）は，次に示す計算式から求めてください。

計算式（単位：バイト）

$$CRTTBLLOG = 1,296 + \left\{ dbarea_file_num \times 6 \right\} \times page_size$$

変数の説明

dbarea_file_num：定義対象表を格納するデータ用 DB エリアの DB エリアファイル数

page_size：定義対象表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

(2) CREATE INDEX 文実行時に出力されるユーザログ容量の求め方

CREATE INDEX 文実行時に出力されるユーザログ容量（変数 *CRTIDXLOG*）は，次に示す計算式から求めてください。

計算式（単位：バイト）

$$CRTIDXLOG = 812 + \left\{ dbarea_file_num \times 6 \right\} \times page_size$$

変数の説明

dbarea_file_num：定義対象インデクスを格納するデータ用 DB エリアの DB エリアファイル数

page_size：定義対象インデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

(3) ALTER TABLE 文の CHANGE OPTION CHUNK 実行時に出力されるユーザログ容量の求め方

ALTER TABLE 文のCHANGE OPTION CHUNK 実行時に出力されるユーザログ容量は，「(7) DROP TABLE 文実行時に出力されるユーザログ容量の求め方」を参照して求めてください。その際，変数 *SGDATA* に 0 を仮定してください。

また，処理対象表にインデクスが定義されている場合は，定義されているインデクスの数だけ，「(8) DROP INDEX 文実行時に出力されるユーザログ容量の求め方」を参照して求めてください。その際，変数 *SGIDX* に 0 を仮定してください。

それぞれで求めた値の合計値が，ALTER TABLE 文のCHANGE OPTION CHUNK 実行時に出力されるユーザログ容量になります。

(4) ALTER TABLE 文実行時に出力されるユーザログ容量の求め方（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）

ALTER TABLE 文でアーカイブマルチチャンク表をレギュラーマルチチャンク表に変更すると，ロケーション表とロケーション表に定義されたインデクスが削除されます。そのため，ロケーション表とロケーション表に定義されたインデクスが削除されるときに出力されるユーザログ容量を求める必要があります。

なお，ロケーション表とロケーション表に定義されたインデクスは，処理対象のアーカイブマルチチャンク表と同じデータ用 DB エリアに格納されています。

アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更するために，ALTER TABLE 文を実行する場合に出力されるユーザログ容量（変数*ALTTBLLOG*）は，次に示す計算式から求めてください。

計算式（単位：バイト）

$$ALTTBLLOG = DRPLTBLLOG + DRPLIDXLOG$$

変数の説明

DRPLTBLLOG

ロケーション表の削除ログ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$DRPLTBLLOG = 263,712 + \left\lceil \frac{\left(dbarea_file_num \times 6 + \left\lceil \frac{SGDATA}{SEGBF} \right\rceil + \left\lceil \frac{SGDATA}{\left\lceil \frac{page_size \times 125}{16} \right\rceil \times SEGBF} \right\rceil \times page_size \right)}{usrlog_file_num} \right\rceil$$

dbarea_file_num

削除対象のロケーション表を格納するデータ用 DB エリアのデータ用 DB エリアファイルの数

SGDATA

削除対象のロケーション表を格納するデータ用 DB エリアのセグメント数

SEGBF

削除対象のロケーション表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size

削除対象のロケーション表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

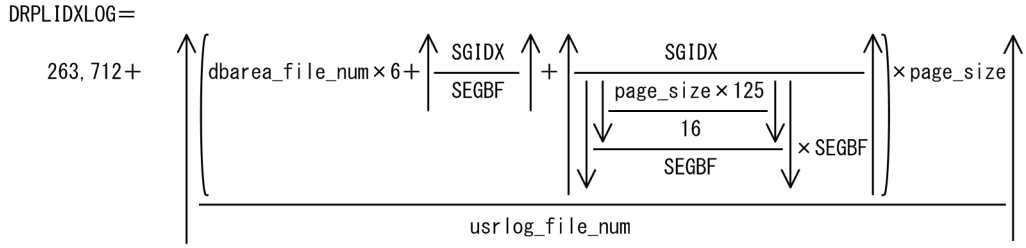
```
usrlog_file_num＝
    ALTER TABLE文※実行時に必要なユーザログファイル数－1

注※
  アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合に
  実行するALTER TABLE文が該当します。
```

DRPLIDXLOG

ロケーション表に定義されたインデクスの削除ログ

次に示す計算式から求めてください。

計算式（単位：バイト）



dbarea_file_num

削除対象のロケーション表に定義されたインデクスを格納するデータ用 DB エリアのデータ用 DB エリアファイルの数

SGIDX

削除対象のロケーション表に定義されたインデクスを格納するデータ用 DB エリアのセグメント数

SEGBF

削除対象のロケーション表に定義されたインデクスを格納するデータ用 DB エリアのセグメントブロック化係数

page_size

削除対象のロケーション表に定義されたインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    ALTER TABLE文※実行時に必要なユーザログファイル数－1

注※
  アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合に
  実行するALTER TABLE文が該当します。
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更するために，ALTER TABLE 文を実行する場合に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

(5)　DROP USER 文実行時に出力されるユーザログ容量の求め方

削除対象の HADB ユーザが所有するスキーマがある場合，「(6)　DROP SCHEMA 文実行時に出力されるユーザログ容量の求め方」を参照して，削除するスキーマについてユーザログ容量を求めてください。

(6)　DROP SCHEMA 文実行時に出力されるユーザログ容量の求め方

削除対象のスキーマに表を定義している場合は，「(7)　DROP TABLE 文実行時に出力されるユーザログ容量の求め方」を参照して，削除するすべての表について，それぞれユーザログ容量を求めてください。そのあとで，その合計値を求めてください。

削除対象のスキーマに表を定義していない場合は，ユーザログファイルの容量を超えるユーザログは出力されないため，見積もりは不要です。

(7)　DROP TABLE 文実行時に出力されるユーザログ容量の求め方

DROP TABLE 文実行時に出力されるユーザログ容量（変数*DRPTBLLOG*）は，次に示す計算式から求めてください。

計算式（単位：バイト）

$$DRPTBLLOG = 263,712 + \left\lceil \frac{\left(dbarea_file_num \times 6 + \left\lceil \frac{SGDATA}{SEGBF} \right\rceil + \left\lceil \frac{SGDATA}{\left\lceil \frac{page_size \times 125}{16} \right\rceil \times SEGBF} \right\rceil \times page_size \right)}{usrlog_file_num} \right\rceil$$

変数の説明

dbarea_file_num

削除対象表を格納するデータ用 DB エリアの DB エリアファイル数

SGDATA

削除対象表を格納するデータ用 DB エリアのセグメント数

SEGBF

削除対象表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size

削除対象表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    DROP TABLE文実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

DROP TABLE 文実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

重要

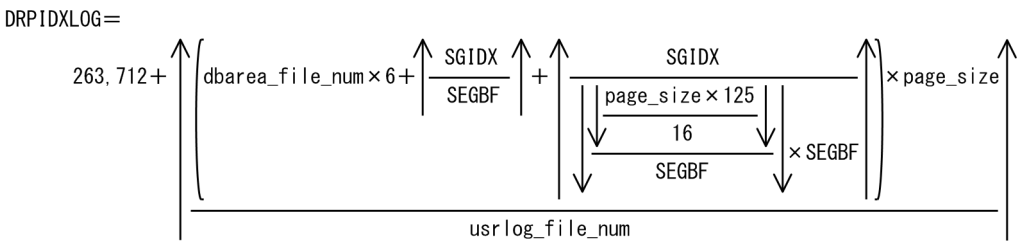
削除対象の表にインデクスが定義されている場合は，「(8)　DROP INDEX 文実行時に出力されるユーザログ容量の求め方」で求めた値を加算してください。

また，削除対象の表がアーカイブマルチチャンク表の場合は，「(4)　ALTER TABLE 文実行時に出力されるユーザログ容量の求め方（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）」で求めた値を加算してください。

(8)　DROP INDEX 文実行時に出力されるユーザログ容量の求め方

DROP INDEX 文実行時に出力されるユーザログ容量（変数 *DRPIDXLOG*）は，次に示す計算式から求めてください。

計算式（単位：バイト）



変数の説明

dbarea_file_num

削除対象インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX

削除対象インデクスを格納するデータ用 DB エリアのセグメント数

SEGBF

削除対象インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

page_size

削除対象インデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    DROP INDEX文実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

DROP INDEX 文実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

(9)　REVOKE 文の SCHEMA 実行時に出力されるユーザログ容量の求め方

REVOKE 文のSCHEMA 実行時に出力されるユーザログ容量は，「(6)　DROP SCHEMA 文実行時に出力されるユーザログ容量の求め方」を参照して求めてください。

6.12.4　adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbimport コマンド実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

```
adbimportコマンド実行時に出力されるユーザログ容量＝
        MAX{TBLPRGLOG＋IDXPRGLOG, IMPTBLLOG, IMPSTBLLOG＋IMPRNGIDXLOG, IMPIDXLOG＋IMPTIXLOG}
```

変数の説明

- *TBLPRGLOG*
  インポート対象表の行データ削除ログ
  「(1)　変数 TBLPRGLOG の求め方」を参照してください。
- *IDXPRGLOG*
  インポート対象表のインデクスデータ削除ログ
  「(2)　変数 IDXPRGLOG の求め方」を参照してください。
- *IMPTBLLOG*
  インポート対象表の行データ格納ログ
  「(3)　変数 IMPTBLLOG の求め方」を参照してください。
- *IMPSTBLLOG*
  STATUS_CHUNKS 表の更新ログ
  「(4)　変数 IMPSTBLLOG の求め方」を参照してください。
- *IMPRNGIDXLOG*
  インポート対象表のレンジインデクスデータ作成ログ
  「(5)　変数 IMPRNGIDXLOG の求め方」を参照してください。
- *IMPIDXLOG*

インポート対象表の B-tree インデクスデータ作成ログ
「(6)　変数 IMPIDXLOG の求め方」を参照してください。

- *IMPTIXLOG*
  インポート対象表のテキストインデクスデータ作成ログ
  「(7)　変数 IMPTIXLOG の求め方」を参照してください。

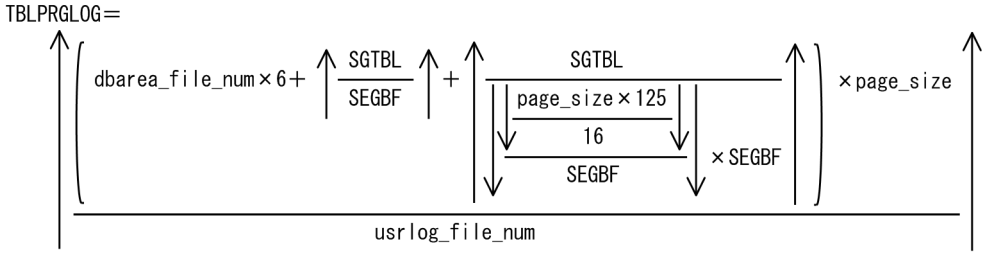
(1)　変数 TBLPRGLOG の求め方

変数 *TBLPRGLOG* は次に示す計算式から求めてください。

ただし，次に示す場合，見積もりは不要です。*TBLPRGLOG* に 0 を代入してください。

- シングルチャンク表に対して，-d オプションを省略してadbimport コマンドを実行する場合
- マルチチャンク表に対して，-d オプションおよび-b オプションを省略してadbimport コマンドを実行する場合
- マルチチャンク表に対して，-d オプションを省略して，かつ-b オプションを指定してadbimport コマンドを実行する場合

計算式（単位：バイト）



変数の説明

dbarea_file_num
　表を格納するデータ用 DB エリアの DB エリアファイル数

SGTBL
　表を格納するセグメント数

SEGBF
　表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size
　表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num
　次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbimportコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

adbimport コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(2)　変数 IDXPRGLOG の求め方

変数 *IDXPRGLOG* は次に示す計算式から求めてください。

ただし，次に示す場合，見積もりは不要です。*IDXPRGLOG* に 0 を代入してください。

- シングルチャンク表に対して，-d オプションを省略してadbimport コマンドを実行する場合
- マルチチャンク表に対して，-d オプションおよび-b オプションを省略してadbimport コマンドを実行する場合
- マルチチャンク表に対して，-d オプションを省略して，かつ-b オプションを指定してadbimport コマンドを実行する場合

計算式（単位：バイト）

$$IDXPRGLOG = \sum_{i=1}^{idx_num} \left\{ \left\lceil \frac{\left(dbarea_file_num(i) \times 6 + \left\lceil \frac{SGIDX(i)}{SEGBF(i)} \right\rceil + \left\lceil \frac{SGIDX(i)}{\left\lfloor \frac{\left\lfloor \frac{page_size(i) \times 125}{16} \right\rfloor}{SEGBF(i)} \right\rfloor \times SEGBF(i)} \right\rceil \times page_size(i) \right)}{usrlog_file_num} \right\rceil \right\}$$

変数の説明

idx_num

削除対象の表に定義されたインデクスの数

dbarea_file_num（*i*）

インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX（*i*）

インデクスごとの，インデクスを格納するセグメント数

SEGBF（*i*）

インデクスごとのデータ用 DB エリアのセグメントブロック化係数

page_size（*i*）

インデクスごとのデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbimportコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

adbimport コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(3)　変数 IMPTBLLOG の求め方

変数 *IMPTBLLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

IMPTBLLOG＝

$$\left(\text{dbarea_file_num} \times 6 + \left\uparrow \frac{\text{SGDATA}}{\text{SEGBF}} \right\uparrow + \left\uparrow \frac{\text{SGDATA}}{\left\downarrow \frac{\left\downarrow \frac{\text{page_size} \times 125}{16} \right\downarrow}{\text{SEGBF}} \right\downarrow \times \text{SEGBF}} \right\uparrow \right) \times \text{page_size}$$

$$\text{imp_load_rthd}$$

変数の説明

- *dbarea_file_num*

  表を格納するデータ用 DB エリアの DB エリアファイル数

- *SGDATA*

  インポート処理で確保する，表を格納するセグメント数

- *SEGBF*

  表を格納するデータ用 DB エリアのセグメントブロック化係数

- *page_size*

  表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

- *imp_load_rthd*

  次に示す計算式から求めてください。

  ```
  インポートオプションadb_import_rthd_numの指定値－1
  ```

  インポートオプションadb_import_rthd_num を省略した場合は，インポートオプション adb_import_rthd_num を省略したときに使用されるスレッド数の値を代入してください。詳細につ

いては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。また，確保するセグメント数については，インポートするデータサイズを基に見積もってください。

(4)　変数 IMPSTBLLOG の求め方

変数 *IMPSTBLLOG* は次に示す計算式から求めてください。

ただし，次に示す場合，見積もりは不要です。*IMPSTBLLOG* に 0 を代入してください。

- シングルチャンク表に対して，adbimport コマンドを実行する場合
- マルチチャンク表に対して，-d オプションおよび-b オプションを省略してadbimport コマンドを実行する場合

計算式（単位：バイト）

IMPSTBLLOG＝ *IMPSTBLPRGLOG*＋*IMPSTBLUPDLOG*

次に示す場合，*IMPSTBLPRGLOG* だけ見積もりが必要です。

- マルチチャンク表に対して，-d オプションを指定してadbimport コマンドを実行する場合

また，次に示す場合，*IMPSTBLUPDLOG* だけ見積もりが必要です。

- マルチチャンク表に対して，-d オプションを省略して，かつ-b オプションを指定してadbimport コマンドを実行する場合

変数の説明

IMPSTBLPRGLOG

STATUS_CHUNKS 表のデータを削除，追加するときの更新ログ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IMPSTBLPRGLOG = 488 \times \left\{ chunk_num + 1 \right\}$$

chunk_num

作成されたチャンク数

IMPSTBLUPDLOG

STATUS_CHUNKS 表のデータを更新，追加するときの更新ログ

次に示す値を代入してください。

値（単位：バイト）

IMPSTBLUPDLOG＝ 1,464

(5) 変数 IMPRNGIDXLOG の求め方

変数 *IMPRNGIDXLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

$$IMPRNGIDXLOG = MAX\left\{IMPRNGIDX(1)LOG,\ IMPRNGIDX(2)LOG,\ \cdots,\ IMPRNGIDX(n)LOG\right\}$$

変数の説明

- *n*

  インポート対象の表に定義されたレンジインデクスの数

- *IMPRNGIDX(i)LOG*

  インポート対象の表で i 番目に定義されたレンジインデクスのログサイズ

  次に示す計算式から求めてください。ただし，adbimport コマンド実行時の-d オプションの指定有無（作成モードまたは追加モード）によって計算式が異なります。

■計算式（-d オプションを指定する場合（作成モードの場合））（単位：バイト）

$$IMPRNGIDX(i)LOG = DIRRNGPGNO(i) \times page_size(i)$$

- *page_size(i)*

  インポート対象表で i 番目に定義されたレンジインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

- *DIRRNGPGNO(i)*

  インポート対象表で i 番目に定義されたレンジインデクスの管理部のページ数

  次に示す計算式から求めてください。

$$DIRRNGPGNO(i) = dbarea_file_num(i) \times 6 + \left\lceil \frac{SGRNGDATA(i)}{SEGBF(i)} \right\rceil + \left\lceil \frac{SGRNGDATA(i)}{\left\lfloor \frac{\left\lfloor \frac{page_size(i) \times 125}{16} \right\rfloor}{SEGBF(i)} \right\rfloor \times SEGBF(i)} \right\rceil$$

- *dbarea_file_num(i)*

  レンジインデクスを格納するデータ用 DB エリアの DB エリアファイル数（単位：個）

- *SGRNGDATA(i)*

  データインポート処理で確保するレンジインデクスのセグメント数

「5.8.6　レンジインデクスを格納するセグメント数の求め方」を参照してください。

- *SEGBF(i)*

  レンジインデクスを格納するデータ用 DB エリアのセグメントブロック化係数

  「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の変数「*SEGBF*」を参照してください。

- *page_size(i)*

  インポート対象表で i 番目に定義されたレンジインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

■計算式（-d オプションを省略する場合（追加モードの場合））（単位：バイト）

$$IMPRNGIDX(i)LOG = \left\{DIRRNGPGNO(i) + OLDSGRNGDATA \times SEGSIZE(i)\right\} \times page_size(i)$$

- *DIRRNGPGNO(i)*

  インポート対象表で i 番目に定義されたレンジインデクスの管理部のページ数

  『■計算式（-d オプションを指定する場合（作成モードの場合））』にある変数 *DIRRNGPGNO*（*i*）を参照してください。

- *OLDSGRNGDATA*

  データインポートする前に確保してあったレンジインデクスを格納するデータ用 DB エリアのセグメント数

  「5.8.6　レンジインデクスを格納するセグメント数の求め方」の変数 *RS*（*i*）を参照してください。

- *SEGSIZE(i)*

  インポート対象表で i 番目に定義されたレンジインデクスを格納するデータ用 DB エリアのセグメントサイズ

  次に示す計算式から求めてください。

  SEGSIZE(i)＝ *1セグメントの容量*（単位：バイト）÷*page_size(i)*

- *page_size(i)*

  インポート対象表で i 番目に定義されたレンジインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

(6)　変数 IMPIDXLOG の求め方

変数 *IMPIDXLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

$$IMPIDXLOG = IMPIDX(1)LOG + IMPIDX(2)LOG + \cdots + IMPIDX(n)LOG$$

変数の説明

- *n*

  インポート対象の表に定義された B-tree インデクスの数

- *IMPIDX*（*i*）*LOG*

  インポート対象の表で i 番目に定義された B-tree インデクスのログサイズ

  次に示す計算式から求めてください。

  計算式（単位：バイト）

$$IMPIDX(i)LOG = \left(\frac{(DIRPGNO(i) + unique_key_num) \times page_size(i)}{imp_dividx_rthd}\right)$$

- *unique_key_num*

  ユニークなキー値の数

  ・adbimport コマンドに-d オプションを指定した場合は 0 としてください。

  ・上記以外の場合は，インポート対象表に i 番目に定義された B-tree インデクスについて，インポートするデータで生成されるキー値から重複を排除した上で，インデクスにすでに格納されているキー値の最小値以上，かつ最大値以下である値の数としてください。

- *page_size*（*i*）

  B-tree インデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

- *imp_dividx_rthd*

  次に示す計算式から求めてください。

  インポートオプションadb_import_rthd_numの指定値 －1

  インポートオプションadb_import_rthd_num を省略した場合は，インポートオプション adb_import_rthd_num を省略したときに使用されるスレッド数の値を代入してください。詳細については，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

- *DIRPGNO*（*i*）

  インポート対象表で i 番目に定義された B-tree インデクスの管理部のページ数

  次に示す計算式から求めてください。

  計算式

$$DIRPGNO(i) = dbarea_file_num(i) \times 6 + \left\lceil \frac{SGIDX(i)}{SEGBF(i)} \right\rceil + \left\lceil \frac{SGIDX(i)}{\left\lfloor \frac{\left\lfloor \frac{page_size(i) \times 125}{16} \right\rfloor}{SEGBF(i)} \right\rfloor \times SEGBF(i)} \right\rceil$$

- *dbarea_file_num*（*i*）

  B-tree インデクスを格納するデータ用 DB エリアの DB エリアファイル数

- *SGIDX*（*i*）

  インポート処理で確保する，B-tree インデクスを格納するセグメント数

- *SEGBF*（*i*）

  B-tree インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。また，確保するセグメント数については，インポートするデータサイズを基に見積もってください。

(7)　変数 IMPTIXLOG の求め方

変数 *IMPTIXLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

```
IMPTIXLOG＝
     MAX｛IMPTIX(1)LOG，IMPTIX(2)LOG，…，IMPTIX(n)LOG｝
```

変数の説明

n

インポート対象の表に定義されたテキストインデクスの数

IMPTIX(i)LOG

インポート対象の表で i 番目に定義されたテキストインデクスのログサイズ

次に示す計算式から求めてください。ただし，adbimport コマンド実行時の-d オプションの指定有無（作成モードまたは追加モード）によって計算式が異なります。

■計算式（-d オプションを指定する場合（作成モードの場合））（単位：バイト）

```
IMPTIX(i)LOG＝
     DIRTIXPGNO(i)×page_size(i)
```

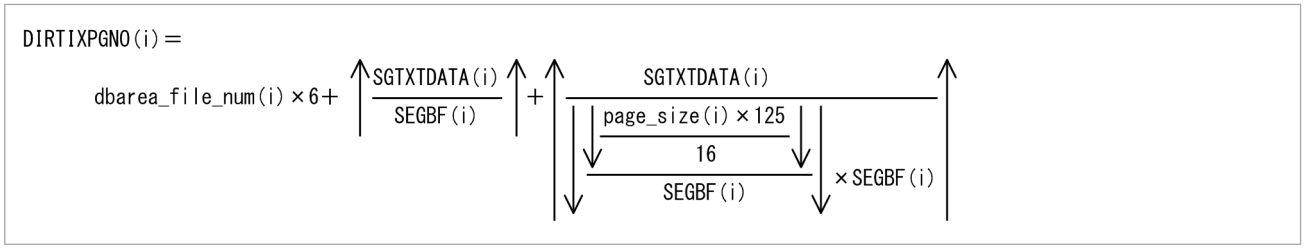
page_size(i)

インポート対象表で i 番目に定義されたテキストインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

DIRTIXPGNO(i)

インポート対象表で i 番目に定義されたテキストインデクスの管理部のページ数

次に示す計算式から求めてください。



dbarea_file_num(i)

テキストインデクスを格納するデータ用 DB エリアの DB エリアファイル数（単位：個）

SGTXTDATA(i)

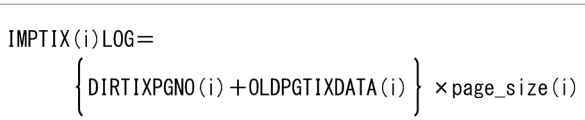
データインポート処理で確保するテキストインデクスのセグメント数

SEGBF(i)

テキストインデクスを格納するデータ用 DB エリアのセグメントブロック化係数

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の変数「*SEGBF*」を参照してください。

■計算式（-d オプションを省略する場合（追加モードの場合））（単位：バイト）



DIRTIXPGNO(i)

インポート対象表で i 番目に定義されたテキストインデクスの管理部のページ数

『■計算式（-d オプションを指定する場合（作成モードの場合））』にある変数*DIRTIXPGNO(i)*を参照してください。

OLDPGTIXDATA(i)

データインポートする前に確保してあったテキストインデクスの格納ページ数

次に示す計算式から求めてください。

$$OLDPGTIXDATA(i) = \uparrow TIP_STRSEG(i) \uparrow + \uparrow TIP_APPSEG(i) \uparrow$$

TIP_STRSEG(i)

「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」の「(1)　文字列管理部用のセグメントで使用する格納ページ数（変数 TIP_STRSEG（i））の求め方」を参照してください。

TIP_APPSEG(i)

「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」の「(2)　出現情報管理部用のセグメントで使用する格納ページ数（変数 TIP_APPSEG（i））の求め方」を参照してください。

6.12.5　adbidxrebuild コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbidxrebuild コマンド実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

```
adbidxrebuildコマンド実行時に出力されるユーザログ容量＝
                                MAX（IDXPRGLOG＋TBLPRGLOG※　，IDXMAKELOG）
```

変数の説明

IDXPRGLOG

インデクス再作成対象表のインデクスデータ削除ログ

「(1)　変数 IDXPRGLOG の求め方」を参照してください。

TBLPRGLOG

対象表でバックグラウンドインポートを実行した場合の行データ削除ログ

「(2)　変数 TBLPRGLOG の求め方」を参照してください。

IDXMAKELOG

インデクス再作成対象表のインデクスデータ作成ログ

「(3)　変数 IDXMAKELOG の求め方」を参照してください。

注※

変数 *TBLPRGLOG* は，-b オプションを指定したadbimport コマンドが中断したあとに，--force オプションを指定してadbidxrebuild コマンドを実行する場合に求めてください。

(1)　変数 IDXPRGLOG の求め方

変数 *IDXPRGLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

IDXPRGLOG＝

$$\sum_{i=1}^{idx_num} \left\{ \left\lceil \frac{\left(dbarea_file_num(i) \times 6 + \left\lceil \frac{SGIDX(i)}{SEGBF(i)} \right\rceil + \left\lceil \frac{SGIDX(i)}{\left\lfloor \frac{\left\lfloor \frac{page_size(i) \times 125}{16} \right\rfloor}{SEGBF(i)} \right\rfloor \times SEGBF(i)} \right\rceil \times page_size(i) \right)}{usrlog_file_num} \right\rceil \right\}$$

変数の説明

idx_num

インデクス再作成対象のインデクスの数

dbarea_file_num（*i*）

インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX（*i*）

インデクスを格納するセグメント数

SEGBF（*i*）

インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

page_size（*i*）

インデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbidxrebuildコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

adbidxrebuild コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(2)　変数 TBLPRGLOG の求め方

変数 *TBLPRGLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

TBLPRGLOG＝

$$\left\lceil \frac{\left(dbarea_file_num \times 6 + \left\lceil \frac{SGDATA}{SEGBF} \right\rceil + \left\lceil \frac{SGDATA}{\left\lceil \frac{page_size \times 125}{16} \right\rceil \times SEGBF} \right\rceil \right) \times page_size}{usrlog_file_num} \right\rceil$$

変数の説明

dbarea_file_num

表を格納するデータ用 DB エリアの DB エリアファイル数

SGDATA

バックグラウンドインポートで確保する表のセグメント数

バックグラウンドインポートで格納するデータ量を基に，バックグラウンドインポートで確保する表のセグメント数を見積もってください。

SEGBF

表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size

表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbidxrebuildコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

adbidxrebuild コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(3)　変数 IDXMAKELOG の求め方

変数 *IDXMAKELOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXMAKELOG = MAX\left\{IDXMAKE(1)LOG,\ IDXMAKE(2)LOG,\ \cdots,\ IDXMAKE(n)LOG\right\}$$

変数の説明

n

インデクス再作成対象のインデクスの数

IDXMAKE（*i*）*LOG*

インデクス再作成対象となる i 番目のインデクスのログサイズ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXMAKE(i)LOG = \left(\frac{DIRPGNO(i) \times page_size(i)}{rbld_dividx_rthd}\right)$$

page_size（*i*）

インデクス再作成対象となる i 番目のインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

rbld_dividx_rthd

次に示す計算式から求めてください。

↓（*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値* －1）÷2↓

インデクス再作成オプションadb_idxrebuild_rthd_num を省略した場合は，インデクス再作成オプションadb_idxrebuild_rthd_num を省略したときに使用されるスレッド数の値を代入してください。詳細については，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

DIRPGNO（*i*）

インデクス再作成対象となるｉ番目のインデクスの管理部のページ数

次に示す計算式から求めてください。

計算式

$$DIRPGNO(i) = dbarea_file_num(i) \times 6 + \left\uparrow \frac{SGIDX(i)}{SEGBF(i)} \right\uparrow + \left\uparrow \left\downarrow \frac{page_size(i) \times 125}{16} \right\downarrow \times SEGBF(i) \right\uparrow$$

dbarea_file_num（*i*）

インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX（*i*）

インデクス再作成処理でチャンクごとに確保する，インデクスを格納するセグメント数の最大値

次に示す計算式から求めてください。

計算式

$$SGIDX(i) = MAX\left\{CHUNKSGIDX(1)LOG,\ CHUNKSGIDX(2)LOG,\ \cdots,\ CHUNKSGIDX(m)LOG\right\}$$

- *m*

  インデクス再作成対象のインデクスの作成したチャンク数

- *CHUNKSGIDX*（*j*）*LOG*

  インデクス再作成対象となるインデクスのチャンクごとに確保するインデクスを格納するセグメント数

SEGBF（*i*）

インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

6.12.6　adbmergechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbmergechunk コマンド実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

```
adbmergechunkコマンド実行時に出力されるユーザログ容量＝
                                    MAX（MERPRGLOG ， MERMAKELOG＋MERLOCLOG）
```

変数の説明

MERPRGLOG

マージチャンク対象表に定義されているインデクスデータの削除ログ

「(1)　変数 MERPRGLOG の求め方」を参照してください。

MERMAKELOG

マージチャンク対象表に定義されているインデクスデータの作成ログ

「(2)　変数 MERMAKELOG の求め方」を参照してください。

MERLOCLOG

ロケーション表の更新ログ

「(3)　変数 MERLOCLOG の求め方」を参照してください。

(1)　変数 MERPRGLOG の求め方

変数 *MERPRGLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

MERPRGLOG＝

$$\sum_{i=1}^{idx_num} \left\{ \left(\left(dbarea_file_num(i) \times 6 + \left\uparrow \frac{SGIDX(i)}{SEGBF(i)} \right\uparrow \right) + \left\uparrow \frac{SGIDX(i)}{\left\downarrow \frac{\left\downarrow \frac{page_size(i) \times 125}{16} \right\downarrow}{SEGBF(i)} \right\downarrow \times SEGBF(i)} \right\uparrow \times page_size(i) \right) \Big/ usrlog_file_num \right\}$$

変数の説明

idx_num

マージチャンク対象表に定義されているインデクスの数

dbarea_file_num（*i*）

各インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX（*i*）

各インデクスを格納するセグメント数

SEGBF（*i*）

各インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

page_size（*i*）

各インデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbmergechunkコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

adbmergechunk コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(2)　変数 MERMAKELOG の求め方

変数 *MERMAKELOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

$$MERMAKELOG = MAX\left\{MERMAKE(1)LOG,\ MERMAKE(2)LOG,\ \cdots,\ MERMAKE(n)LOG\right\}$$

変数の説明

n

マージチャンク対象表に定義されているインデクスの数

MERMAKE（*i*）*LOG*

マージチャンク対象表に定義されている i 番目のインデクスのログサイズ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$MERMAKE(i)LOG = \left(\frac{DIRPGNO(i) \times page_size(i)}{merc_dividx_rthd}\right)$$

page_size（*i*）

マージチャンク対象表に定義されているｉ番目のインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

merc_dividx_rthd

次に示す計算式から求めてください。

↓（*マージチャンクオプションadb_mergechunk_rthd_numの指定値*－1）÷2↓

マージチャンクオプションadb_mergechunk_rthd_num を省略した場合は，マージチャンクオプションadb_mergechunk_rthd_num を省略したときに使用されるスレッド数の値を代入してください。詳細については，マニュアル『HADB　コマンドリファレンス』の『adbmergechunk（チャンクのマージ)』を参照してください。

DIRPGNO（*i*）

マージチャンク対象表に定義されているｉ番目のインデクスの管理部のページ数

次に示す計算式から求めてください。

計算式

$$DIRPGNO(i) = dbarea_file_num(i) \times 6 + \left\uparrow \frac{SGIDX(i)}{SEGBF(i)} \right\uparrow + \left\uparrow \left\downarrow SGIDX(i) - \left\downarrow \frac{page_size(i) \times 125}{16} \right\downarrow \times SEGBF(i) \right\downarrow \right\uparrow$$

dbarea_file_num（*i*）

インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX（*i*）

マージチャンク処理（インデクス再作成時）でチャンクごとに確保するインデクスのセグメント数の最大値

次に示す計算式から求めてください。

計算式

$$SGIDX(i) = MAX\left\{ CHUNKSGIDX(1),\ CHUNKSGIDX(2),\ \cdots,\ CHUNKSGIDX(m) \right\}$$

- *m*

  マージチャンク対象のチャンク数

- *CHUNKSGIDX*（*j*）

  再作成対象となるインデクスのチャンクごとに確保するインデクスのセグメント数

SEGBF（*i*）

インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

(3)　変数 MERLOCLOG の求め方

変数 *MERLOCLOG* は，次に示す計算式から求めてください。

なお，処理対象表がアーカイブマルチチャンク表以外の表である場合は，0 を代入してください。また，処理対象のチャンクがアーカイブ状態ではない場合も，0 を代入してください。

計算式（単位：バイト）

$$MERLOCLOG = UPDLOG + LBP \times \left(\uparrow \frac{84 + page_size}{8} \uparrow \times 8 + 4 \right)$$

変数の説明

UPDLOG

「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の『UPDATE 文（行の更新）を実行した場合に出力されるユーザログ容量』の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

VRWLOG

0 を代入してください。

TIXSLOG

0 を代入してください。

IDXSLOG

変数 *IDXSLOG* を求める場合，「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の「(3)　B-tree インデクスユーザログ容量の求め方」を参照してください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *idx_num*

  1 を代入してください。

- *IDXLOG(i)*

  次に示す計算式から求めてください。

  計算式

$$IDXLOG(i) = (296 + SPLITLOG(i)) \times LROWNUM + (\uparrow (84 + page_size) \div 8 \uparrow \times 8 + 4) \times \uparrow LROWNUM \div 256 \uparrow$$

- *SPLITLOG(i)*

「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の「(3)　B-tree インデクスユーザログ容量の求め方」の変数「*SPLITLOG (i)*」の『計算式（更新対象行のインデクスキーの重複数が 256 以上の場合)』を参照して求めてください。

LROWNUM

adbarchivechunk コマンドで作成されるアーカイブファイル数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合)」にある変数*LROWNUM* の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *archive_chunk_num*

  0 を代入してください。

- *merge_chunk_num*

  1 を代入してください。

upd_row_num

上記の変数*LROWNUM* で求めた値を代入してください。

SGMTLOG

次に示す計算式から求めてください。

計算式

```
SGMTLOG＝ 1,228×↑LBP÷SEGSIZE↑
```

LBP

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合)」にある変数*LBP(i)* の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *LROWNUM*

  上記の変数*LROWNUM* で求めた値を代入してください。

SEGSIZE

アーカイブマルチチャンク表を格納したデータ用 DB エリアのセグメントサイズ

次に示す計算式から求めてください。

計算式（単位：ページ数）

SEGSIZE＝ 1セグメントの容量（単位：バイト）÷*page_size*

page_size

アーカイブマルチチャンク表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方

INSERT 文（行の挿入），DELETE 文（行の削除）およびUPDATE 文（行の更新）の実行時に出力されるユーザログ容量の求め方について説明します。

まず，「(1)　基本行ユーザログ容量の求め方」から「(6)　セグメント割り当て時のユーザログ容量の求め方」までのユーザログ容量をそれぞれ求めてください。そのあとで，計算式 1～計算式 3 に次に示す変数を代入して計算してください。

計算式 1（INSERT 文を実行した場合に出力されるユーザログ容量）（単位：バイト）

```
計算式1（INSERT文を実行した場合に出力されるユーザログ容量）＝
　　（BRWLOG＋VRWLOG＋IDXSLOG＋TIXSLOG）×ins_row_num＋SGMTLOG
```

計算式 2（DELETE 文を実行した場合に出力されるユーザログ容量）（単位：バイト）

```
計算式2（DELETE文を実行した場合に出力されるユーザログ容量）＝
　　（BRWLOG＋VRWLOG＋COLUMN_DELLOG＋IDXSLOG）×del_row_num＋SGMTLOG
```

計算式 3（UPDATE 文を実行した場合に出力されるユーザログ容量）（単位：バイト）

```
計算式3（UPDATE文を実行した場合に出力されるユーザログ容量）＝
　　（BRWLOG＋VRWLOG＋COLUMN_DELLOG＋IDXSLOG＋TIXSLOG）×upd_row_num＋SGMTLOG
```

変数の説明

BRWLOG

基本行ユーザログ容量（単位：バイト）

BRWLOG については，「(1)　基本行ユーザログ容量の求め方」を参照して求めてください。

VRWLOG

分岐行ユーザログ容量（単位：バイト）

VRWLOG については，「(2)　分岐行ユーザログ容量の求め方」を参照して求めてください。

COLUMN_DELLOG

カラムストア表の無効情報管理用ページのログ容量（単位：バイト）

COLUMN_DELLOG については，「(5)　カラムストア表の無効情報管理用ページのログ容量の求め方」を参照して求めてください。

IDXSLOG

表に定義したすべての B-tree インデクスユーザログ容量の合計（単位：バイト）

IDXSLOG については，「(3)　B-tree インデクスユーザログ容量の求め方」を参照して求めてください。

TIXSLOG

表に定義したすべてのテキストインデクスユーザログ容量の合計（単位：バイト）

TIXSLOG については，「(4)　テキストインデクスユーザログ容量の求め方」を参照して求めてください。

SGMTLOG

セグメント割り当て時のユーザログ容量（単位：バイト）

SGMTLOG については，「(6)　セグメント割り当て時のユーザログ容量の求め方」を参照して求めてください。

ins_row_num

INSERT 文で挿入した行数

del_row_num

DELETE 文で削除した行数

upd_row_num

UPDATE 文で更新した行数

(1)　基本行ユーザログ容量の求め方

基本行ユーザログ容量（変数 *BRWLOG*）を次の表に示します。

表 6-22　基本行ユーザログ容量（変数 BRWLOG）

項番	操作内容	基本行ユーザログ容量（単位：バイト）
1	INSERT 文	92
2	DELETE 文	
3	UPDATE 文	184

(2)　分岐行ユーザログ容量の求め方

分岐行ユーザログ容量（変数 *VRWLOG*）の計算式を次の表に示します。

表 6-23　分岐行ユーザログ容量（変数 VRWLOG）

項番	操作内容	分岐行ユーザログ容量（単位：バイト）
1	INSERT 文	$92 \times \sum_{i=1}^{var_num} \left\uparrow \frac{var_size(i)}{page_size-70} \right\uparrow$
2	DELETE 文	項番 1 と同じです。
3	UPDATE 文	$184 \times \sum_{i=1}^{var_num_upd} \left\uparrow \frac{var_size(i)}{page_size-70} \right\uparrow$

変数の説明

var_num

分岐行として管理される列の列数（単位：個）

次に示す列の数を求めてください。

表オプションBRANCH ALL の指定ありの場合

- VARCHAR 型，およびVARBINARY 型の列の数

表オプションBRANCH ALL の指定なしの場合

次のどれかに該当するVARCHAR 型，およびVARBINARY 型の列の数の合計

- 列定義BRANCH にYES を指定した列
- 定義長 256 バイト以上で列定義BRANCH にAUTO を指定した列
- 定義長 256 バイト以上で列定義BRANCH を省略した列

var_num_upd

分岐行として管理される列のうち，更新する列数（単位：個）

var_size

分岐行として管理される列の平均データサイズ（単位：バイト）

「5.8.2　行の種別ごとの格納ページ数の求め方」の「(2)　分岐行用ページ数（変数 VP（i））の求め方」の「表 5-17　分岐行となる列のデータ長の一覧」を参照して求めてください。

page_size

データ用 DB エリアのページサイズ（単位：バイト）

(3)　B-tree インデクスユーザログ容量の求め方

B-tree インデクスユーザログ容量（変数 *IDXSLOG*）は，次に示す計算式から求めてください。

計算式

$$\text{インデクスユーザログ容量} = \sum_{i=1}^{idx_num} IDXLOG(i)$$

変数の説明

idx_num

更新対象列に定義された B-tree インデクスの数

IDXLOG（*i*）

更新対象表に定義された i 番目の B-tree インデクスに関するインデクスユーザログのサイズ（単位：バイト）

IDXLOG（*i*）は次に示す表に従って求めてください。

項番	更新種別	更新対象キーの更新前重複数	IDXLOG（i）の値
1	INSERT 文	0（新規キー）	AFTERLOG（i）＋ SPLITLOG（i）
2		1 以上 254 以下	項番 1 と同じです。
3		255	300＋AFTERLOG（i）
4		256 以上	項番 1 と同じです。
5	DELETE 文		BEFORELOG（i）
6	UPDATE 文		BEFORELOG（i）＋INSERT文実行時に出力されるﾕｰｻﾞﾛｸﾞ容量※

注※

更新対象行の更新後データを INSERT した場合と同じログサイズとなります。

AFTERLOG（*i*）

更新対象行の更新後データの更新論理ログサイズ（単位：バイト）

BEFORELOG（*i*）

更新対象行の更新前データの更新論理ログサイズ（単位：バイト）

AFTERLOG（*i*）および*BEFORELOG*（*i*）の長さは，次に示す計算式で求めてください。

計算式

$$\text{AFTERLOG(i)およびBEFORELOG(i)} = \left\lceil \frac{120 + (8 \times idx_col_num) + 4 + KEYSZDB}{8} \right\rceil \times 8 + 4$$

idx_col_num

対象となるインデクスのインデクス構成列数（単位：個）

KEYSZDB

対象となるインデクスキーのデータベース格納長（単位：バイト）

AFTERLOG（*i*）の場合は更新後のインデクスキー，*BEFORELOG*（*i*）の場合は更新前のインデクスキーになります。

SPLITLOG（*i*）

インデクスページスプリットが発生した場合のユーザログ容量（単位：バイト）

対象となるインデクスで，インデクスページスプリットが発生した場合のユーザログ容量は，次に示す計算式で求めてください。

計算式（更新対象行のインデクスキーの重複数が 254 以下の場合）

$$\text{SPLITLOG(i)} = 260 + \left(\left\lceil \frac{84 + page_size}{8} \right\rceil \times 8 + 4 \right) \times idx_level$$

計算式（更新対象行のインデクスキーの重複数が 256 以上の場合）

$$SPLITLOG(i)=260+\left(\left\uparrow\frac{84+page_size}{8}\right\uparrow\times 8+4\right)\times 2$$

page_size

対象となる B-tree インデクスが定義されたデータ用 DB エリアのページサイズ（単位：バイト）

idx_level

対象となる B-tree インデクスの段数（単位：個）

(4) テキストインデクスユーザログ容量の求め方

テキストインデクスユーザログ容量（変数 *TIXSLOG*）は，次に示す計算式から求めてください。

計算式

$$TIXSLOG=\sum_{i=1}^{tix_num}TIXLOG(i)$$

変数の説明

tix_num

更新対象列に定義されたテキストインデクスの数

TIXLOG(*i*)

更新対象表に定義された i 番目のテキストインデクスに関するテキストインデクスユーザログのサイズ（単位：バイト）

TIXLOG(*i*)は，次に示す表に従って求めてください。

表 6-24 TIXSLOG(i)の値

項番	更新種別	TIXLOG(i)の値
1	INSERT 文	*SPLITLOG*(*i*)
2	UPDATE 文	*SPLITLOG*(*i*)

SPLITLOG(*i*)

テキストインデクスのインデクスページスプリットのログサイズ

次に示す計算式から求めてください。

計算式（単位：バイト）

$$SPLITLOG(i)=260+\left(\left\uparrow\frac{84+page_size}{8}\right\uparrow\times 8+4\right)\times 2$$

page_size

対象となるテキストインデクスが定義されたデータ用 DB エリアのページサイズ（単位：バイト）

(5)　カラムストア表の無効情報管理用ページのログ容量の求め方

変数*COLUMN_DELLOG*は，次に示す計算式から求めてください。

なお，処理対象表がローストア表の場合は，0として見積もってください。

計算式（単位：バイト）

```
COLUMN_DELLOG ＝ 332＋2×page_size
```

変数の説明

page_size
　カラムストア表が定義されているDBエリアのページサイズ（単位：バイト）

(6)　セグメント割り当て時のユーザログ容量の求め方

INSERT文およびUPDATE文を実行して行やインデクスエントリが追加されたときに，行やインデクスエントリを使用中ページに格納できない場合，未使用ページに割り当てます。また，カラムストア形式で格納された行をUPDATE文またはDELETE文を実行して無効化した場合，無効情報管理用ページに対象のデータが無効となったことを示す情報を格納できないときも，未使用ページに割り当てます。

この際，セグメント内のすべてのページを使用している場合は，新規セグメントを割り当てます。

新規セグメント割り当て時のユーザログ容量（変数*SGMTLOG*）を次の表に示します。

割り当てるセグメント数については，INSERT文で追加する行データおよびUPDATE文で更新する行データを基に，「5.8.1　データ用DBエリアの総ページ数の求め方」，「5.8.2　行の種別ごとの格納ページ数の求め方」，「5.8.3　B-treeインデクスのセグメントごとの格納ページ数の求め方」および「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」を参照して求めてください。

処理対象表がカラムストア表の場合は，DELETE文で削除する行データ数も考慮して見積もってください。

表6-25　新規セグメント割り当て時のユーザログ容量

項番	操作内容	ユーザログ容量（単位：バイト）
1	INSERT文	1,228×セグメント割り当て数
2	UPDATE文	
3	DELETE文※	

注※
　削除対象表がローストア表の場合は，0として見積もってください。

6.12.8 TRUNCATE TABLE 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

TRUNCATE TABLE 文実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

TRUNCATE TABLE文実行時に出力されるﾕｰｻﾞﾛｸﾞ容量＝

$$263,712+\left\lceil\frac{\left(\text{dbarea_file_num}\times 5+\left\lceil\frac{\text{SGDATA}}{\text{SEGBF}}\right\rceil+\left\lceil\frac{\text{SGDATA}}{\left\lceil\frac{\text{page_size}\times 125}{16}\right\rceil\times\text{SEGBF}}\right\rceil\right)\times\text{page_size}}{\text{usrlog_file_num}}\right\rceil$$

＋TRUNCLOCLOG

変数の説明

dbarea_file_num

処理対象の実表を格納するデータ用 DB エリアの DB エリアファイル数

SGDATA

処理対象の実表を格納するデータ用 DB エリアのセグメント数

SEGBF

処理対象の実表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size

処理対象の実表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    TRUNCATE TABLE文実行時に必要なユーザログファイル数－1
```

TRUNCLOCLOG

ロケーション表の更新ログ

次に示す計算式から求めてください。

なお，処理対象表がアーカイブマルチチャンク表以外の表である場合は，0 を代入してください。また，処理対象表にアーカイブ状態のチャンクが存在しない場合も，0 を代入してください。

計算式（単位：バイト）

$$\text{TRUNCLOCLOG}=\text{DELLOG}+\text{LBP}\times\left(\left\lceil\frac{84+\text{page_size}}{8}\right\rceil\times 8+4\right)$$

DELLOG

「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の『DELETE 文（行の削除）を実行した場合に出力されるユーザログ容量』の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

VRWLOG

0 を代入してください。

IDXSLOG

変数*IDXSLOG* を求める場合，「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の「(3)　B-tree インデクスユーザログ容量の求め方」を参照してください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *idx_num*

  1 を代入してください。

- *IDXLOG(i)*

  次に示す計算式から求めてください。

  計算式

  ```
  IDXLOG(i)＝ 148×LROWNUM＋(↑(84＋page_size)÷8↑×8＋4)×↑LROWNUM÷256↑
  ```

LROWNUM

adbarchivechunk コマンドで作成されるアーカイブファイル数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」にある変数*LROWNUM* の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *archive_chunk_num*

  1 を代入してください。

- *merge_chunk_num*

  0 を代入してください。

LBP

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」にある変数*LBP(i)* の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *LROWNUM*

  上記の変数*LROWNUM* で求めた値を代入してください。

page_size

アーカイブマルチチャンク表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

del_row_num

上記の変数*LROWNUM* で求めた値を代入してください。

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

TRUNCATE TABLE 文実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

なお，TRUNCATE TABLE 文を実行する実表にインデクスが定義されている場合は，インデクスごとに次に示す計算式を求めてください。そして，求めた値を上記の計算式の結果に加算してください。

計算式（実表にインデクスが定義されている場合）（単位：バイト）

TRUNCATE TABLE文でｲﾝﾃﾞｸｽを削除するときのﾕｰｻﾞﾛｸﾞ容量＝

$$263,712 + \left\lceil \left(dbarea_file_num \times 5 + \frac{SGIDX}{SEGBF} \right) \uparrow + \left\lceil \frac{SGIDX}{\left\lceil \frac{page_size \times 125}{16} \right\rceil \downarrow \times SEGBF} \right\rceil \downarrow \times page_size \right\rceil \uparrow \Big/ usrlog_file_num$$

変数の説明

dbarea_file_num

処理対象のインデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGIDX

処理対象のインデクスを格納するデータ用 DB エリアのセグメント数

SEGBF

処理対象のインデクスを格納するデータ用 DB エリアのセグメントブロック化係数

page_size

処理対象のインデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    TRUNCATE TABLE文実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

TRUNCATE TABLE 文実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

6.12.9　PURGE CHUNK 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

PURGE CHUNK 文実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

PURGE CHUNK文実行時に出力されるﾕｰｻﾞﾛｸﾞ容量＝

$$\sum_{j=1}^{chunk_num} \left(TBLCHPRGLOG(j)+IDXCHPRGLOG(j)+STBLUPDLOG+PRGLOCLOG\right)$$

変数の説明

chunk_num

1 つのPURGE CHUNK 文で削除するチャンク数

TBLCHPRGLOG(j)

実表のチャンクが保持する行データの削除ログ

削除するチャンクごとに，次に示す計算式で求めてください。

計算式（単位：バイト）

TBLCHPRGLOG＝

$$\left\lceil \left\{ dbarea_file_num \times 6 + \left\lceil \frac{SGCHTBL}{SEGBF} \right\rceil + \left\lceil \frac{SGCHTBL}{\left\lceil \frac{page_size \times 125}{16} \right\rceil \times SEGBF} \right\rceil \times page_size \right\} \Big/ usrlog_file_num \right\rceil$$

dbarea_file_num

実表を格納するデータ用 DB エリアの DB エリアファイル数

SGCHTBL

実表のチャンクが確保するセグメント数

SEGBF

実表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size

実表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    PURGE CHUNK文実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

PURGE CHUNK 文実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

IDXCHPRGLOG(*i*)

処理対象表のチャンクごとのインデクスデータの削除ログ

削除するチャンクごとに，次に示す計算式で求めてください。

計算式（単位：バイト）

IDXCHPRGLOG＝

$$\sum_{i=1}^{idx_num} \left\{ \left\lceil \frac{\left(dbarea_file_num(i) \times 6 + \left\lceil \frac{SGCHIDX(i)}{SEGBF(i)} \right\rceil + \left\lceil \frac{SGCHIDX(i)}{\left\lfloor \frac{\left\lfloor \frac{page_size(i) \times 125}{16} \right\rfloor}{SEGBF(i)} \right\rfloor \times SEGBF(i)} \right\rceil \right) \times page_size(i)}{usrlog_file_num} \right\rceil \right\}$$

idx_num

削除対象の実表に定義されたインデクスの数

dbarea_file_num(*i*)

インデクスを格納するデータ用 DB エリアの DB エリアファイル数

SGCHIDX(*i*)

インデクスのチャンクが確保するセグメント数

SEGBF(*i*)

インデクスごとのデータ用 DB エリアのセグメントブロック化係数

page_size(*i*)

インデクスごとのデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    PURGE CHUNK文実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

PURGE CHUNK 文実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(1)　ユーザログファイル数を見積もる必要がある SQL 文」を参照してください。

STBLUPDLOG

システム表（実表）からデータを削除したときの更新ログ

変数*STBLUPDLOG*には，480 バイトを代入してください。

PRGLOCLOG

ロケーション表の更新ログ

次に示す計算式から求めてください。

なお，処理対象表がアーカイブマルチチャンク表以外の表である場合は，0 を代入してください。また，処理対象のチャンクがアーカイブ状態ではない場合も，0 を代入してください。

計算式（単位：バイト）

$$PRGLOCLOG = DELLOG + LBP \times \left(\left\uparrow \frac{84 + page_size}{8} \right\uparrow \times 8 + 4 \right)$$

DELLOG

「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の『DELETE 文（行の削除）を実行した場合に出力されるユーザログ容量』の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

VRWLOG

0 を代入してください。

IDXSLOG

変数*IDXSLOG*を求める場合，「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の「(3)　B-tree インデクスユーザログ容量の求め方」を参照してください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *idx_num*

  1 を代入してください。

- *IDXLOG(i)*

  次に示す計算式から求めてください。

  計算式

$$IDXLOG(i) = 148 \times LROWNUM + (\uparrow (84 + page_size) \div 8 \uparrow \times 8 + 4) \times \uparrow LROWNUM \div 256 \uparrow$$

LROWNUM

adbarchivechunk コマンドで作成されるアーカイブファイル数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」にある変数*LROWNUM*の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *archive_chunk_num*
  1 を代入してください。
- *merge_chunk_num*
  0 を代入してください。

LBP

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」にある変数*LBP(i)*の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *LROWNUM*
  上記の変数*LROWNUM* で求めた値を代入してください。

page_size

アーカイブマルチチャンク表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

del_row_num

上記の変数*LROWNUM* で求めた値を代入してください。

6.12.10　adbchgchunkstatus コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbchgchunkstatus コマンド実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

adbchgchunkstatusコマンド実行時に出力されるユーザログ容量＝
TBLCHKSTATLOG＋*IDXCHKSTATLOG*＋5,760,360×（1＋*idx_num*）

変数の説明

idx_num

状態変更を行うチャンク内の実表に定義されたインデクスの数

TBLCHKSTATLOG

状態変更を行うチャンク内の実表のログ容量

次に示す計算式から求めてください。

計算式（単位：バイト）

$$TBLCHKSTATLOG = dbarea_file_num \times \left\{ 2 + MIN\left(\left\lceil \frac{32,000}{\left\lfloor \frac{page_size}{160} \right\rfloor} \right\rceil , change_chunk_num \right) \right\} \times page_size$$

- *dbarea_file_num*
  実表を格納するデータ用 DB エリアの DB エリアファイル数
- *change_chunk_num*
  状態変更を行うチャンクの数
- *page_size*
  実表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

IDXCHKSTATLOG

状態変更を行うチャンク内の実表に定義されたインデクスのログ容量
次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXCHKSTATLOG = \sum_{i=1}^{idx_num} \left\{ dbarea_file_num(i) \times \left(2 + MIN \left(\left\lceil \frac{32,000}{\left\lfloor \frac{page_size(i)}{160} \right\rfloor} \right\rceil , change_chunk_num \right) \right) \times page_size(i) \right\}$$

- *idx_num*
  状態変更を行うチャンク内の実表に定義されたインデクスの数
- *dbarea_file_num*（*i*）
  各インデクスのデータ用 DB エリアの DB エリアファイル数
- *change_chunk_num*
  状態変更を行うチャンクの数
- *page_size*（*i*）
  各インデクスのデータ用 DB エリアのページサイズ（単位：バイト）

6.12.11 adbarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbarchivechunk コマンド実行時に出力されるユーザログ容量は，次に示す計算式から求めてください。

計算式（単位：バイト）

```
adbarchivechunkコマンド実行時に出力されるユーザログ容量＝
        TBLARCCHKLOG＋IDXARCCHKLOG＋STBLUPDLOG＋1,920,120×(1＋idx_num)＋ARCCHKLOCLOG
```

注

複数のチャンクに対してadbarchivechunk コマンドを実行する場合は，各チャンクの中で，最大のセグメント数を使用しているチャンクを基に求めてください。

変数の説明

TBLARCCHKLOG

adbarchivechunk コマンドの実行時に出力されるアーカイブマルチチャンク表に関するログ容量

「(1)　変数 TBLARCCHKLOG の求め方」を参照してください。

IDXARCCHKLOG

adbarchivechunk コマンドの実行時に出力されるアーカイブマルチチャンク表に定義されたインデクスに関するログ容量

「(2)　変数 IDXARCCHKLOG の求め方」を参照してください。

STBLUPDLOG

adbarchivechunk コマンドの実行時にシステム表（実表）のデータを更新した場合に出力されるログ容量

「(3)　変数 STBLUPDLOG の求め方」を参照してください。

idx_num

adbarchivechunk コマンドを実行するアーカイブマルチチャンク表に定義されたインデクス数

ARCCHKLOCLOG

ロケーション表の更新ログ

「(4)　変数 ARCCHKLOCLOG の求め方」を参照してください。

(1)　変数 TBLARCCHKLOG の求め方

変数 *TBLARCCHKLOG* は，次に示す計算式から求めてください。

計算式（単位：バイト）

$$TBLARCCHKLOG = \left\lceil \left(dbarea_file_num \times 6 + \left\lceil \frac{SGCHTBL}{SEGBF} \right\rceil + \left\lceil \frac{SGCHTBL}{\left\lceil \frac{page_size \times 125}{16} \right\rceil \times SEGBF} \right\rceil \right) \times page_size \right\rceil \Big/ usrlog_file_num$$

変数の説明

dbarea_file_num

アーカイブマルチチャンク表を格納するデータ用 DB エリアのデータ用 DB エリアファイル数

SGCHTBL

adbarchivechunk コマンドの処理対象チャンクのうち，アーカイブマルチチャンク表が使用するセグメント数

SEGBF

アーカイブマルチチャンク表を格納するデータ用 DB エリアのセグメントブロック化係数

page_size

アーカイブマルチチャンク表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbarchivechunkコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

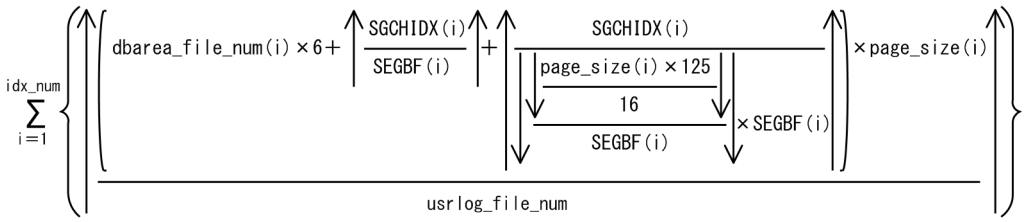
adbarchivechunk コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(2)　変数 IDXARCCHKLOG の求め方

変数 *IDXARCCHKLOG* は，次に示す計算式から求めてください。

計算式（単位：バイト）

IDXARCCHKLOG＝



変数の説明

idx_num

adbarchivechunk コマンドを実行するアーカイブマルチチャンク表に定義されたインデクス数

dbarea_file_num(i)

各インデクスが格納されているデータ用 DB エリアのデータ用 DB エリアファイル数

SGCHIDX(i)

adbarchivechunk コマンドの処理対象チャンクのうち，アーカイブマルチチャンク表に定義された各インデクスが使用するセグメント数

SEGBF(i)

アーカイブマルチチャンク表に定義された各インデクスを格納するデータ用 DB エリアのセグメントブロック化係数

page_size(i)

アーカイブマルチチャンク表に定義された各インデクスを格納するデータ用 DB エリアのページサイズ（単位：バイト）

usrlog_file_num

次に示す計算式から求めてください。

```
usrlog_file_num＝
    adbarchivechunkコマンド実行時に必要なユーザログファイル数－1
```

データ用 DB エリアのセグメント数，およびセグメントブロック化係数については，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」を参照してください。

adbarchivechunk コマンド実行時に必要なユーザログファイル数は，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

(3)　変数 STBLUPDLOG の求め方

変数 *STBLUPDLOG* は，次に示す値を代入してください。

値（単位：バイト）

```
STBLUPDLOG＝ 968
```

(4)　変数 ARCCHKLOCLOG の求め方

変数 *ARCCHKLOCLOG* は次に示す計算式から求めてください。

計算式（単位：バイト）

$$ARCCHKLOCLOG = INSLOG + LBP \times \left(\uparrow \frac{84 + page_size}{8} \uparrow \times 8 + 4 \right)$$

変数の説明

INSLOG

「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の『INSERT 文（行の挿入）を実行した場合に出力されるユーザログ容量』の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

VRWLOG

0 を代入してください。

TIXSLOG

0 を代入してください。

IDXSLOG

変数*IDXSLOG*を求める場合，「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の「(3)　B-tree インデクスユーザログ容量の求め方」を参照してください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *idx_num*

  1 を代入してください。

- *IDXLOG(i)*

  次に示す計算式から求めてください。

  計算式

```
IDXLOG(i)＝
        (148＋SPLITLOG(i))×LROWNUM＋(↑(84＋page_size)÷8↑×8＋4)×↑LROWNUM÷256↑
```

- *SPLITLOG(i)*

  「6.12.7　データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方」の「(3)　B-tree インデクスユーザログ容量の求め方」の変数「*SPLITLOG（i)*」の『計算式（更新対象行のインデクスキーの重複数が 256 以上の場合)』を参照して求めてください。

LROWNUM

adbarchivechunk コマンドで作成されるアーカイブファイル数（単位：個）

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合)」にある変数*LROWNUM* の計算式から求めてください。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *archive_chunk_num*

  1 を代入してください。

- *merge_chunk_num*

  0 を代入してください。

ins_row_num

上記の変数*LROWNUM* で求めた値を代入してください。

SGMTLOG

次に示す計算式から求めてください。

計算式

```
SGMTLOG＝ 1,228×↑LBP÷SEGSIZE↑
```

LBP

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合)」にある変数*LBP(i)*の計算式から求めてくださ

い。ただし，次に示す変数については，代入する値が異なります。対象の変数と代入する値を次に示します。

- *LROWNUM*

  上記の変数*LROWNUM*で求めた値を代入してください。

SEGSIZE

アーカイブマルチチャンク表を格納したデータ用 DB エリアのセグメントサイズ

次に示す計算式から求めてください。

計算式（単位：ページ数）

SEGSIZE＝ *1セグメントの容量*（単位：バイト）÷*page_size*

page_size

アーカイブマルチチャンク表を格納するデータ用 DB エリアのページサイズ（単位：バイト）

6.12.12 adbunarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbunarchivechunk コマンド実行時に出力されるユーザログ容量は，次に示す場合に出力されるユーザログ容量と同じです。

- アーカイブマルチチャンク表に対して，-d オプションを省略して，かつ-b オプションを指定して adbimport コマンドを実行する場合

  入力データは，アーカイブ状態を解除するチャンク内のデータです。

adbimport コマンド実行時に出力されるユーザログ容量の求め方については，「6.12.4 adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」を参照してください。

なお，複数のチャンクに対してadbunarchivechunk コマンドを実行する場合は，各チャンクの中で，最大のセグメント数を使用しているチャンクを基に求めてください。

また，使用するセグメント数を求める場合，adbdbstatus コマンドに-c archivechunk オプションを指定して「アーカイブ状態のチャンクのサマリ情報」を出力してください。次に示す出力項目から，セグメント数を確認できます。

- *Unarchive_table_size*

  アーカイブ状態になる前のチャンクで，表を格納するために使用していたセグメントの容量（単位：バイト）

- *Unarchive_index_size*

  アーカイブ状態になる前のチャンクで，インデクスを格納するために使用していたセグメントの容量（単位：バイト）

6.12.13　adbreorgsystemdata コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方

adbreorgsystemdata コマンド実行時に出力されるユーザログ容量は，次に示す場合に出力されるユーザログ容量と同じです。

- マルチチャンク表に対して，-d オプションを指定してadbimport コマンドを実行する場合

adbimport コマンド実行時に出力されるユーザログ容量の求め方については，「6.12.4　adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」を参照してください。ユーザログ容量（変数*max_user_log*）を求める際，各変数には次の値を代入してください。

IMPSTBLLOG

0 を代入してください。

IMPRNGIDXLOG

0 を代入してください。

IMPTIXLOG

0 を代入してください。

dbarea_file_num

1 を代入してください。

dbarea_file_num(i)

1 を代入してください。

SGTBL

再編成対象のシステム表（実表）に対して，adbdbstatus コマンドを実行し，「表のサマリ情報」を出力してください。そして，出力項目の「Used_segments」の値を代入してください。

STATUS_CHUNKS 表を再編成対象にした場合の，adbdbstatus コマンドの実行例を次に示します。

（例）

```
adbdbstatus -d summary -c table -n HADB.STATUS_CHUNKS
```

SEGBF

56 を代入してください。

SEGBF(i)

56 を代入してください。

page_size

4,096 を代入してください。

page_size(i)

4,096 を代入してください。

pctfree

0 を代入してください。

変数*pctfree* は，「6.12.4　adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方」の計算式を求める場合に，次の変数を求めるときに使用します。

- 「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」にある変数*IP_LOWER（i）*
- 「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」にある変数*IP_UPPER（i）*

usrlog_file_num

次の値を代入してください。

```
adbreorgsystemdataコマンド実行時のユーザログファイル数-1
```

adbreorgsystemdata コマンド実行時のユーザログファイル数については，「6.12.15　ユーザログファイル数の求め方」の「(2)　ユーザログファイル数を見積もる必要があるコマンド」を参照してください。

idx_num

再編成対象のシステム表（実表）に定義されているインデクス数を代入してください。

再編成対象のシステム表（実表）に定義されているインデクス数については，「付録 C.8　システム表（実表）の B-tree インデクス」を参照してください。

SGIDX(i)

再編成対象のシステム表（実表）に定義されているインデクスに対して，adbdbstatus コマンドを実行し，「インデクスのサマリ情報」を出力してください。そして，出力項目の「Used_segments」の値を代入してください。

adbdbstatus コマンドは定義されているインデクスごとに実行してください。

再編成対象のシステム表（実表）に定義されているインデクスについては，「付録 C.8　システム表（実表）の B-tree インデクス」を参照してください。

STATUS_CHUNKS 表のインデクス（STATUSINDEXM05）を再編成対象にした場合の，adbdbstatus コマンドの実行例を次に示します。

（例）

```
adbdbstatus -d summary -c index -n HADB.STATUSINDEXM05
```

SEGDATA

SGTBL の値を代入してください。

imp_load_rthd

1 を代入してください。

imp_dividx_rthd

1 を代入してください。

6.12.14　更新行のカラム化機能のメンテナンス処理に必要なユーザログ容量（変数 max_user_log）の求め方

更新行のカラム化機能を有効にする場合は，更新行のカラム化機能のメンテナンス処理に必要なユーザログ容量を求めてください。

更新行のカラム化機能のメンテナンス処理は，カラムストア表のチャンクごとに実行されます。そのため，更新行のカラム化機能の対象となるカラムストア表のチャンクごとにユーザログ容量（*COLUMNIZELOG*）を計算し，その中で最も大きい値を更新行のカラム化機能のメンテナンス処理に必要なユーザログ容量としてください。チャンクごとのユーザログ容量（*COLUMNIZELOG*）は，次に示す計算式から求めてください。

計算式（単位：バイト）

```
COLUMNIZELOG ＝ 1,428
              ＋（rowdata_sgmt_num×112＋var_sgmt_num×1,428
                  ＋dbarea_file_num×SEGSIZE×144）×COMPRESSION_RATE
              ＋dbarea_file_num×2,080
```

変数の説明

rowdata_sgmt_num

更新行のカラム化機能のメンテナンス処理の対象となるローデータ用セグメント数

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」にある変数 *UPDATESEGNUM（i）* を参照して，該当するチャンクのローデータ用セグメント数を求めてください。

var_sgmt_num

作成される分岐行を格納するセグメント数

該当するチャンクにINSERT 文またはUPDATE 文によって追加または更新されたデータを，adbimport コマンドでインポートすると仮定して，次に示す変数を参照して求めてください。

- 「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」にある変数 *ROWDATASEGNUM（i, k）*

dbarea_file_num

カラムストア表が格納される DB エリアの DB エリアファイル数

SEGSIZE

セグメントサイズ（単位：ページ数）

次に示す計算式から求めてください。

```
SEGSIZE＝ 1セグメントの容量（単位：バイト）÷page_size
```

page_size

データ用 DB エリアのページサイズ（単位：バイト）

COMPRESSION_RATE

元データの圧縮率

「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(g)　変数 SGCOLUMNTBL の求め方（マルチチャンク表の場合）」にある変数 *COMPRESSION_RATE* を参照して求めてください。

圧縮率が不明な場合は 0.5 を仮定してください。

6.12.15　ユーザログファイル数の求め方

ユーザログファイルは，特定の SQL 文やコマンドを実行した場合に，必要な数だけ割り当てられます。そして，割り当てられたユーザログファイルは，SQL 文とコマンドが終了したときに解放されます。そのため，ユーザログファイル数（「6.12　システムログファイルの容量見積もり」の変数 *user_file_num*）は，同時に実行する SQL 文とコマンドを考慮して見積もる必要があります。

複数の SQL 文とコマンドを同時に実行する場合は，次の手順で求めてください。

手順

1. 同時に実行するすべての SQL 文とコマンドのユーザログファイル数を求めてください。
2. 求めたすべてのユーザログファイル数を合計してください。
3. 合計したユーザログファイル数以上の値を，サーバ定義adb_log_usrfile_num オペランドに指定してください。

サーバ定義adb_log_usrfile_num オペランドについては，「7.2.3　システムログに関するオペランド（set 形式）」の adb_log_usrfile_num オペランドを参照してください。

(1)　ユーザログファイル数を見積もる必要がある SQL 文

ユーザログファイル数を見積もる必要がある SQL 文を次に示します。

■定義系 SQL

- ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）
- DROP INDEX 文
- DROP TABLE 文
- DROP SCHEMA 文
- DROP USER 文
- REVOKE 文のSCHEMA（スキーマ操作権限の取り消し）
- 上記以外の定義系 SQL

■操作系 SQL

- PURGE CHUNK 文
- TRUNCATE TABLE 文
- DELETE 文
- INSERT 文
- UPDATE 文

各 SQL 文のユーザログファイル数の求め方を次に示します。なお，求めた結果が 0 の場合は，1 を仮定してください。

■定義系 SQL の場合

- ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（ALTER TABLE文※実行時）＝ MIN（A , B）＋1

  A：サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値
  B：削除対象のロケーション表や，削除対象のロケーション表のインデクスを格納する
     データ用DBエリアのうち，データ用DBエリアファイル数の最大値

  注※：ALTER TABLE文でアーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合
```

- DROP INDEX 文実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（DROP INDEX文実行時）＝ MIN（A , B）＋1

  A：サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値
  B：処理対象のインデクスを格納するデータ用DBエリアのデータ用DBエリアファイル数
```

- DROP TABLE 文実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（DROP TABLE文実行時）＝  MIN（A , B）＋1

  A：サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値
  B：処理対象の表やインデクスを格納するデータ用DBエリアのうち，
     データ用DBエリアファイル数の最大値
```

- DROP SCHEMA 文実行時のユーザログファイル数の求め方（単位：個）

  処理対象のスキーマが包括するすべての表について，『DROP TABLE 文実行時のユーザログファイル数の求め方』を参照して求めてください。そして，求めた結果のうち，最も大きい値になります。

- DROP USER 文実行時のユーザログファイル数の求め方（単位：個）

  処理対象のユーザが所有するスキーマについて，『DROP SCHEMA 文実行時のユーザログファイル数の求め方』を参照して求めてください。

- REVOKE 文のSCHEMA 実行時のユーザログファイル数の求め方（単位：個）

  『DROP SCHEMA 文実行時のユーザログファイル数の求め方』を参照して求めてください。

- 上記以外の定義系 SQL 実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

■操作系 SQL の場合

- PURGE CHUNK 文実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（PURGE CHUNK文実行時）＝　MIN（A , B）＋1

  A：サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値
  B：処理対象の表やインデクスを格納するデータ用DBエリアのうち，
     データ用DBエリアファイル数の最大値
```

- TRUNCATE TABLE 文実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（TRUNCATE TABLE文実行時）＝　MIN（A , B）＋1

  A：サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値
  B：処理対象の表やインデクスを格納するデータ用DBエリアのうち，
     データ用DBエリアファイル数の最大値
```

- DELETE 文実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

- INSERT 文実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

- UPDATE 文実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

重要

クライアント定義adb_sql_exe_max_rthd_num オペランドを指定している場合，サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値ではなく，クライアント定義 adb_sql_exe_max_rthd_num オペランドの指定値を代入してください。

ただし，クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値が，サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値よりも大きい場合は，サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値を代入してください。

サーバ定義adb_sql_exe_max_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_max_rthd_num オペランドを参照してください。

(2)　ユーザログファイル数を見積もる必要があるコマンド

ユーザログファイル数を見積もる必要があるコマンドを次に示します。

- adbimport コマンド
- adbidxrebuild コマンド
- adbmodarea コマンド
- adbmergechunk コマンド

- adbgetcst コマンド
- adbchgchunkcomment コマンド
- adbchgchunkstatus コマンド
- adbarchivechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド
- adbsyndict コマンド

各コマンドのユーザログファイル数の求め方を次に示します。

- adbimport コマンド実行時のユーザログファイル数の求め方（単位：個）

  ```
  ユーザログファイル数（adbimportコマンド実行時）＝
      インポートオプションadb_import_rthd_numの指定値＋1
  ```

  インポートオプションadb_import_rthd_num については，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

- adbidxrebuild コマンド実行時のユーザログファイル数の求め方（単位：個）

  ```
  ユーザログファイル数（adbidxrebuildコマンド実行時）＝
      インデクス再作成オプションadb_idxrebuild_rthd_numの指定値＋1
  ```

  インデクス再作成オプションadb_idxrebuild_rthd_num については，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

- adbmodarea コマンド実行時のユーザログファイル数の求め方（単位：個）

  2 になります。

- adbmergechunk コマンド実行時のユーザログファイル数の求め方（単位：個）

  ```
  ユーザログファイル数（adbmergechunkコマンド実行時）＝
      マージチャンクオプションadb_mergechunk_rthd_numの指定値＋1
  ```

  マージチャンクオプションadb_mergechunk_rthd_num については，マニュアル『HADB　コマンドリファレンス』の『adbmergechunk（チャンクのマージ)』を参照してください。

- adbgetcst コマンド実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

- adbchgchunkcomment コマンド実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

- adbchgchunkstatus コマンド実行時のユーザログファイル数の求め方（単位：個）

  1 になります。

- adbarchivechunk コマンド実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（adbarchivechunkコマンド実行時）＝
    （アーカイブチャンクオプションadb_arcv_rthd_numの指定値＋1）×処理対象のチャンク数
```

アーカイブチャンクオプションadb_arcv_rthd_numについては，マニュアル『HADB　コマンドリファレンス』の『adbarchivechunk（チャンクのアーカイブ)』を参照してください。

- adbunarchivechunkコマンド実行時のユーザログファイル数の求め方（単位：個）

```
ユーザログファイル数（adbunarchivechunkコマンド実行時）＝
          （アンアーカイブチャンクオプションadb_unarcv_rthd_numの指定値＋1）
           ×11×処理対象のチャンク数
```

アンアーカイブチャンクオプションadb_unarcv_rthd_numについては，マニュアル『HADB　コマンドリファレンス』の『adbunarchivechunk（チャンクのアンアーカイブ)』を参照してください。

- adbreorgsystemdataコマンド実行時のユーザログファイル数の求め方（単位：個）

  5になります。

- adbsyndictコマンド実行時のユーザログファイル数の求め方（単位：個）

  1になります。

6.13 アーカイブディレクトリの容量見積もり

ここでは，アーカイブディレクトリの容量の求め方について説明します。アーカイブディレクトリの容量は，アーカイブマルチチャンク表を定義する場合に求める必要があります。

アーカイブディレクトリの容量（*ARCVDIRSIZE*）については，次に示す計算式から求めてください。

計算式（単位：ギガバイト）

$$ARCVDIRSIZE = \sum_{i=1}^{tbl_num_in_arcvdir} (ALLARCVCHKSIZE(i))$$

変数の説明

tbl_num_in_arcvdir

対象のアーカイブディレクトリを使用するアーカイブマルチチャンク表の数（単位：個）

対象のアーカイブディレクトリを，CREATE TABLE 文のチャンクアーカイブ指定のARCHIVEDIR に指定して，定義したアーカイブマルチチャンク表の数を代入してください。

ALLARCVCHKSIZE(i)

i 番目のアーカイブマルチチャンク表にある，アーカイブ状態のチャンクすべての容量を合計した値

変数*ALLARCVCHKSIZE*（アーカイブマルチチャンク表にある，アーカイブ状態のチャンクすべての容量）については，次に示す計算式から求めてください。

計算式（単位：ギガバイト）

$$ALLARCVCHKSIZE = \sum_{i=1}^{arcvchk_num_in_tbl} (ARCVCHKSIZE(i))$$

arcvchk_num_in_tbl

アーカイブマルチチャンク表にある，アーカイブ状態のチャンク数を合計した値（単位：個）

ARCVCHKSIZE(i)

アーカイブマルチチャンク表にある，i 番目のアーカイブ状態のチャンクの容量

変数*ARCVCHKSIZE*（アーカイブ状態のチャンクの容量）については，次に示す計算式から求めてください。

計算式（単位：ギガバイト）

$$ARCVCHKSIZE = chunk_data \times compression_rate$$

chunk_data

チャンクに格納される入力データの容量（単位：ギガバイト）

compression_rate

チャンクに格納される入力データの圧縮率

変数*compression_rate* については，入力データの一部を使用して，次に示す手順で求めてください。

手順

1. アーカイブマルチチャンク表を定義する

   アーカイブマルチチャンク表を定義する方法については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE TABLE（表の定義)』を参照してください。

2. adbimport コマンドに-b オプションを指定して，入力データの一部をアーカイブマルチチャンク表に格納する

   adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport (データのインポート)』を参照してください。

3. 手順 2.で作成したチャンクの容量をadbdbstatus コマンドで確認する

   adbdbstatus コマンドに-d used オプションと-c table オプションを指定して，「DB エリア，表とインデクスの使用量情報」を出力してください。出力項目の値を基に，「手順 2.で作成したチャンクの容量」を次に示す計算式から求めてください。

   計算式

   ```
   チャンクの容量＝ Used_segmentsの値×Segment_sizeの値×Page_sizeの値
   ```

   adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus (データベースの状態解析)』を参照してください。

4. 手順 2.で作成したチャンクをadbarchivechunk コマンドでアーカイブ状態にする

   adbarchivechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbarchivechunk（チャンクのアーカイブ)』を参照してください。

5. 手順 4.でアーカイブ状態にしたチャンクの容量をadbdbstatus コマンドで確認する

   adbdbstatus コマンドに-d used オプションと-c archivechunk オプションを指定して，「アーカイブ状態のチャンクの使用量情報」を出力してください。出力項目の「Archive_file_size」の値を確認して，アーカイブ状態にしたチャンクの容量を確認してください。

6. 手順 3.と手順 5.で確認したチャンクの容量を使用して圧縮率を求める

   次に示す計算式で求めてください。

   計算式

   ```
   compression_rate＝ after_comp_somedata÷before_comp_somedata
   ```

after_comp_somedata

手順 5.で確認したチャンクの容量（単位：バイト）

adbarchivechunk コマンドでアーカイブ状態にしたチャンクの容量です。アーカイブ状態のチャンクのため，入力データが圧縮されています。

before_comp_somedata

手順 3.で確認したチャンクの容量（単位：バイト）

adbarchivechunk コマンドでアーカイブ状態にする前のチャンクの容量です。アーカイブ状態ではないチャンクのため，入力データは圧縮されていません。

6.14　同義語辞書ファイルの格納ディレクトリの容量見積もり

ここでは，同義語辞書ファイルの格納ディレクトリの容量の求め方について説明します。同義語辞書ファイルの格納ディレクトリの容量は，同義語検索をする場合に求める必要があります。

> **メモ**
>
> 同義語辞書ファイルの格納ディレクトリとは，サーバ定義のadb_syndict_storage_path オペランドに指定するディレクトリのことです。

同義語辞書ファイルの格納ディレクトリの容量（*SYNDIRSIZE*）については，次に示す計算式から求めてください。

計算式（単位：ギガバイト）

$$SYNDIRSIZE = \sum_{i=1}^{syndict_num} (SYNDICTSIZE(i))$$

変数の説明

syndict_num

作成する同義語辞書の数（単位：個）

SYNDICTSIZE(i)

各同義語辞書の同義語辞書ファイルのサイズ

次に示す計算式から求めてください。

計算式（単位：ギガバイト）

$$SYNDICTSIZE = 513 + \frac{\frac{syn_num \times 32 + syn_list_file_size + 4 \times syn_group_num}{1,024}}{1,024 \times 1,024} + CORSYNSIZE$$

syn_num

同義語一覧定義ファイル中に指定した同義語の数を合計した値（単位：個）

syn_list_file_size

同義語一覧定義ファイルのサイズ（単位：バイト）

syn_group_num

同義語グループの数（単位：個）

CORSYNSIZE

表記ゆれ補正検索に対応した同義語辞書のサイズ

表記ゆれ補正オプションに「CORRECTIONRULE」を指定して，同義語辞書を作成する場合に加算してください。

次に示す計算式から求めてください。

計算式（単位：ギガバイト）

$$\mathrm{CORSYNSIZE} = \left(513 + \frac{\frac{\mathrm{syn_num} \times 32 + \mathrm{syn_list_file_size} \times 8 + 4 \times \mathrm{syn_group_num}}{1{,}024}}{1{,}024 \times 1{,}024}\right) \times 2$$

6.15　監査証跡の出力先ディレクトリの容量見積もり

ここでは，**監査証跡の出力先ディレクトリ**の容量の求め方について説明します。監査証跡の出力先ディレクトリの容量は，監査証跡機能を使用する場合に求める必要があります。

> **メモ**
>
> 監査証跡の出力先ディレクトリとは，サーバ定義のadb_audit_log_path オペランドに指定するディレクトリのことです。

監査証跡の出力先ディレクトリの容量（*AUDFILE*）については，次に示す計算式から求めてください。

計算式（単位：メガバイト）

AUDFILE ＝ *aud_log_max_size* × *aud_log_max_num*

変数の説明

aud_log_max_size

サーバ定義adb_audit_log_max_size オペランドの指定値

adb_audit_log_max_size オペランドについては，「7.2.9　監査証跡機能に関するオペランド（set 形式）」を参照してください。

aud_log_max_num

サーバ定義adb_audit_log_max_num オペランドの指定値

adb_audit_log_max_num オペランドについては，「7.2.9　監査証跡機能に関するオペランド（set 形式）」を参照してください。

なお，adb_audit_log_max_num オペランドに 0 を指定すると，上限なく監査証跡ファイルが作成されます。そのため，変数*AUDFILE* を求めることができません。adb_audit_log_max_num オペランドに 0 を指定する場合は，次に示す監査証跡機能の運用を考慮して，監査証跡の出力先ディレクトリの容量を求めてください。

- 出力される監査証跡のデータ容量と，作成される監査証跡ファイルの数
- 監査証跡ファイルを，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリに移動する頻度

監査証跡機能の運用については，「12.　監査証跡機能の運用」を参照してください。

6.16　統一フォーマット用監査証跡の出力先ディレクトリの容量見積もり

ここでは，**統一フォーマット用監査証跡の出力先ディレクトリ**の容量の求め方について説明します。統一フォーマット用監査証跡の出力先ディレクトリには，統一フォーマット用監査証跡ファイルが出力されます。

> **メモ**
>
> 統一フォーマット用監査証跡の出力先ディレクトリとは，adbconvertaudittrailfile コマンドの-d オプションに指定するディレクトリのことです。

統一フォーマット用監査証跡の出力先ディレクトリの総容量（*CMNAUDFILE*）については，次に示す計算式から求めてください。

計算式（単位：メガバイト）

CMNAUDFILE ＝ 512 × 4 × 2※

注※

HADB サーバが作成する統一フォーマット用監査証跡ファイルの容量は，1 ファイル当たり 512 メガバイトです。そして，最大 4 ファイルが作成されます。なお，古いファイルを削除して上書きする際に，一時的にディスク上にデータが残ることがあります。統一フォーマット用監査証跡の出力先ディレクトリの総容量を求める際には，一時的にデータが残ることを考慮するため，2 を乗算する必要があります。

6.17　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量見積もり

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量の求め方は，「6.14　同義語辞書ファイルの格納ディレクトリの容量見積もり」と同じです。マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量は，マルチノード機能の使用時に同義語検索をする場合に求める必要があります。

なお，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリは，全ノードで見積もりが必要になります。

> **メモ**
>
> マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリとは，サーバ定義の adb_syndict_node_storage_path オペランドに指定するディレクトリのことです。

6.18　adbexport コマンドが出力するファイルの容量見積もり

ここでは，adbexport コマンドが出力する出力データファイルの容量の求め方について説明します。

出力データファイルの総容量（*EXPORTFILESIZE*）については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

EXPORTFILESIZE ＝↑（*HEADERSIZE*×*file_num*＋*ROWSIZE*×*row_num*）÷1,024↑

変数の説明

HEADERSIZE

出力データファイルに出力する列名のサイズ

adbexport コマンド実行時に--with-column-name オプションを指定する場合に加算してください。--with-column-name オプションを指定しない場合は 0 となります。

次の計算式から求めてください。

計算式（単位：バイト）

HEADERSIZE ＝202×*column_num*

column_num

出力するデータの列数

file_num

出力データパスファイルに指定した出力データファイルの数

ROWSIZE

データ 1 行分のサイズ

次の計算式から求めてください。

計算式（単位：バイト）

$$ROWSIZE = column_num + \sum_{k=1}^{column_num} column_data_size(k)$$

column_data_size（k）

k 列目の列データ長

変数*column_data_size* に代入する値は，列データのデータ型によって異なります。次の表で該当する値を代入してください。

表 6-26　column_data_size に代入する値

項番	分類	データ型	データ長（単位：バイト）
1	数データ	INTEGER	20
2		SMALLINT	11

項番	分類	データ型	データ長（単位：バイト）
3		DECIMAL(*m*,*n*)※	*m* ＋ 2
4		NUMERIC(*m*,*n*)※	
5		DOUBLE PRECISION	24
6		FLOAT	
7	文字データ	CHARACTER(*n*)	2×*n* ＋ 2
8		VARCHAR(*n*)	2×*d* ＋ 2
9	日時データ	DATE	12
10		TIME(*p*)	11 ＋*p*
11		TIMESTAMP(*p*)	22 ＋*p*
12	バイナリデータ	BINARY(*n*)	2×*n* ＋ 2
13		VARBINARY(*n*)	2×*d* ＋ 2

（凡例）

m，*n*：正の整数

p：0，3，6，9または12

d：実際のデータ長

注※

全体の桁数が*m*桁で，小数点以下の桁数が*n*桁の固定小数点数を示します。*m*を省略した場合は38が仮定されます。

row_num

adbexportコマンドで出力する行数

6.19　実表の再編成に必要なファイルの容量見積もり

ここでは，実表の再編成に必要なファイルの容量の求め方について説明します。

実表の再編成を実施するには，adbexport コマンドで再編成対象の実表のデータを出力し，adbimport コマンドでデータを実表に再格納する必要があります。実表の再編成については，次の個所を参照してください。

- 「11.1.11　シングルチャンク表を再編成する方法」
- 「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）」
- 「11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合）」
- 「11.4.16　マルチチャンク表を再編成する方法（サンプルシェルを使用して再編成する場合）」

運用上，実表の再編成が必要になる場合には，再編成で使用するファイルの総容量（*REORGFILESIZE*）を見積もってください。そして，十分な空き領域を確保したディスクを準備しておいてください。

再編成に必要なファイルの総容量（*REORGFILESIZE*）については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

REORGFILESIZE ＝*EXPORTFILESIZE*＋*IMPORTWRKFILESIZE*

変数の説明

EXPORTFILESIZE

再編成対象の実表またはチャンクのデータを，adbexport コマンドで出力したときの出力データファイルの総容量（単位：キロバイト）

「6.18　adbexport コマンドが出力するファイルの容量見積もり」を参照して求めてください。

IMPORTWRKFILESIZE

再編成対象の実表またはチャンクのデータを，adbimport コマンドで格納したときの作業用一時ファイルの容量（単位：キロバイト）

「6.21.1　adbimport コマンド実行時の作業用一時ファイルの容量見積もり」の『■インデクス作成用の作業用一時ファイル』を参照して求めてください。

6.20　アンロードファイルの容量見積もり

ここでは，adbreorgsystemdata コマンド実行時に出力されるアンロードファイルの容量の求め方について説明します。アンロードファイルの容量は，再編成の対象とするシステム表（実表）の使用量で決まります。

アンロードファイルの容量（*UNLDFILESIZE*）については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

```
UNLDFILESIZE＝ ↑ROWSIZE×ROWNUM÷1,024↑＋1
```

変数の説明

ROWSIZE

アンロードファイル内の 1 行のデータ長（単位：バイト）

システム表（実表）ごとに異なります。次の表から求めてください。

表 6-27　システム表（実表）ごとのアンロードファイル内の 1 行のデータ長

項番	システム表（実表）	アンロードファイル内の 1 行のデータ長（単位：バイト）
1	STATUS_TABLES 表	417
2	STATUS_COLUMNS 表	32,721
3	STATUS_INDEXES 表	472
4	STATUS_CHUNKS 表	1,436
5	STATUS_SYNONYM_DICTIONARIES 表	2,237

ROWNUM

システム表（実表）の行数

SQL 文を実行して，再編成対象のシステム表（実表）の行数を取得してください。詳細については，「11.17.5　システム表の状態や使用量を確認する方法」の「(2)　システム表（実表）の格納効率を確認する」を参照してください。

6.21　コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，次に示すコマンドを実行したときに作成される作業用一時ファイルの容量の求め方について説明します。

- adbimport コマンド実行時
- adbidxrebuild コマンド実行時
- adbmergechunk コマンド実行時
- adbunarchivechunk コマンド実行時
- adbreorgsystemdata コマンド実行時
- adbsyndict コマンド実行時

上記に示すコマンドを同時に実行する場合，コマンドごとに作業用一時ファイルの容量を求めてください。そして，それぞれで求めた容量を合計してください。

なお，上記以外のコマンドを実行しても，作業用一時ファイルは作成されません。

6.21.1　adbimport コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，データをインポートするために，adbimport コマンド実行時に作成される作業用一時ファイルの容量の求め方について説明します。

作成される作業用一時ファイルは次のとおりです。

- インデクス作成用の作業用一時ファイル
- データ圧縮用の作業用一時ファイル

■インデクス作成用の作業用一時ファイル

インデクスを定義している表に対してadbimport コマンドを実行した場合，次に示すディレクトリ下にインデクス作成用の作業用一時ファイルが作成されます。

- -w オプションを指定した場合
  指定したディレクトリ下に作成されます。
- -w オプションを指定しない場合
  DB ディレクトリ下（$DBDIR/ADBWORK）に作成されます。

インデクス作成用の作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

インデクス作成用の作業用一時ファイルの容量＝

$$\left\lceil \sum_{i=1}^{idx_num} \left(RECFILE(i) + SORTFILE(i) + SORTWORK(i) + IDXWORK(i) \right) \div 1,024 \right\rceil + \left\lceil \sum_{i=1}^{tix_num} \left(TIXWORK(i) + TIXCORWORK(i) \right) \div 1,024 \right\rceil$$

変数の説明

idx_num

定義した B-tree インデクスの数

RECFILE(i)

インデクスレコードファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$RECFILE(i) = \left\{ KEYSZ(i) + CTRL(i) \right\} \times \left\{ row_num \right\} + 640$$

KEYSZ(i)

i 番目の B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL(i)

i 番目の B-tree インデクス制御情報

- インデクス構成列のキー長がすべて固定長の場合：10 バイト
- インデクス構成列のキー長に可変長を含む場合：12 バイト

row_num

`adbimport` コマンドで格納する行数

SORTFILE(i)

ソート結果用ファイル

次に示す計算式から求めてください。変数 *KEYSZ(i)*，変数 *CTRL(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTFILE(i) = \left\{KEYSZ(i) + CTRL(i)\right\} \times row_num + TCTRL(i)$$

TCTRL(i)

i 番目のテキストインデクス制御情報

- テキストインデクスの場合

  次に示す計算式から求めてください。

  計算式

  $$TCTRL(i) = (26 + 表が格納されるデータ用DBエリアのDBエリアファイル数 \times 72) \times 16 + 4$$

- テキストインデクスではない場合

  0 を代入してください。

SORTWORK(i)

ソート用ワークファイル

次に示す計算式から求めてください。変数 *RECFILE(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTWORK(i) = \left\{RECFILE(i) + 8 \times (row_num)\right\} \times 2$$

IDXWORK(i)

B-tree インデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXWORK(i) = \left\{12 + KEYSZDB(i)\right\} \times \left\{PIDX_LEAF + PIDX(2)\right\}$$

KEYSZDB(i)

i 番目のデータベース格納キー長（単位：バイト）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の変数 *KEYSZDB* を参照してください。ただし，可変長のデータは，常にキー長を定義長として求めてください。

PIDX_LEAF

B-tree インデクスの 1 段目のページ数（リーフページ）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i)）の求め方」の『計算式 1（PIDX_LEAF の求め方)』に従って求めてください。

PIDX(2)

B-tree インデクスの 2 段目のページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i)）の求め方」の『計算式 1（PIDX（k）の求め方)』に従って求めてください。

tix_num

テキストインデクスの数

TIXWORK(i)

テキストインデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXWORK(i)＝
    row_num×string_num(i)×172＋row_num×（define_num(i)＋12)
```

row_num

adbimport コマンドで格納する行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

define_num(i)

テキストインデクスのインデクス構成列の定義長

TIXCORWORK(i)

テキストインデクス作成用ワークファイル（表記ゆれ補正検索向け）

テキストインデクスの定義時に，「テキストインデクス表記ゆれ補正指定」を指定した場合に求める必要があります。

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXCORWORK(i)＝
    row_num×string_num(i)×202
```

row_num

adbimport コマンドで格納する行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

■データ圧縮用の作業用一時ファイル

カラムストア表へデータを格納する場合だけ，データ圧縮用の作業用一時ファイルが作成されます。

データ圧縮用の作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

データ圧縮用の作業用一時ファイルの容量＝

$$\left\lceil \left\{ \left(\sum_{i=1}^{col_num} \left(\left\downarrow \frac{col_size(i)+15}{16} \right\downarrow \times 16 \right) + \left\downarrow \frac{col_num+15}{16} \right\downarrow \times 16 \right) \times 262,144 \right\} \div 1,024 \right\rceil \times imp_load_rthd$$

変数の説明

col_num

処理対象表の列の総数（単位：個）

col_size(i)

処理対象表の各列のデータ長（単位：バイト）

各列のデータ長は，「表 6-9　各データ型のデータ長の一覧」を参照してください。

imp_load_rthd

次の計算式から求めてください。

インポートオプションadb_import_rthd_numの指定値 －1

6.21.2　adbidxrebuild コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，インデクスを再作成するために，adbidxrebuild コマンド実行時に作成される作業用一時ファイルの容量の求め方について説明します。

インデクスを定義している表に対してadbidxrebuild コマンドを実行した場合，次に示すディレクトリ下に作業用一時ファイルが作成されます。

- -w オプションを指定した場合
  指定したディレクトリ下に作成されます。
- -w オプションを指定しない場合
  DB ディレクトリ下（$DBDIR/ADBWORK）に作成されます。

作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

作業用一時ファイルの容量＝

$$\left\lceil \sum_{i=1}^{idx_num} \left(RECFILE(i) + SORTFILE(i) + SORTWORK(i) + IDXWORK(i) \right) \div 1,024 \right\rceil + \left\lceil \sum_{i=1}^{tix_num} \left(TIXWORK(i) + TIXCORWORK(i) \right) \div 1,024 \right\rceil$$

変数の説明

idx_num

定義した B-tree インデクスの数

RECFILE(i)

インデクスレコードファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$RECFILE(i) = \left\{ KEYSZ(i) + CTRL(i) \right\} \times \left\{ row_num \right\} + 640$$

KEYSZ(i)

i 番目の B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL(i)

i 番目の B-tree インデクス制御情報

- インデクス構成列のキー長がすべて固定長の場合：10 バイト
- インデクス構成列のキー長に可変長を含む場合：12 バイト

row_num

adbidxrebuild コマンドの対象表に格納されている行数

SORTFILE(i)

ソート結果用ファイル

次に示す計算式から求めてください。変数 *KEYSZ(i)*，変数 *CTRL(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTFILE(i) = \left\{KEYSZ(i) + CTRL(i)\right\} \times row_num + TCTRL(i)$$

TCTRL(i)

i 番目のテキストインデクス制御情報

- テキストインデクスの場合

  次に示す計算式から求めてください。

  計算式

  $$TCTRL(i) = (26 + \text{表が格納されるデータ用DBエリアのDBエリアファイル数} \times 72) \times 16 + 4$$

- テキストインデクスではない場合

  0 を代入してください。

SORTWORK(i)

ソート用ワークファイル

次に示す計算式から求めてください。変数 *RECFILE(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTWORK(i) = \left\{RECFILE(i) + 8 \times (row_num)\right\} \times 2$$

IDXWORK(i)

B-tree インデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXWORK(i) = \left\{12 + KEYSZDB(i)\right\} \times \left\{PIDX_LEAF + PIDX(2)\right\}$$

KEYSZDB(i)

i 番目のデータベース格納キー長（単位：バイト）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の変数 *KEYSZDB* を参照してください。ただし，可変長のデータは，常にキー長を定義長として求めてください。

PIDX_LEAF

B-tree インデクスの 1 段目のページ数（リーフページ）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i)）の求め方」の『計算式 1（PIDX_LEAF の求め方)』に従って求めてください。

PIDX(2)

B-tree インデクスの 2 段目のページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i)）の求め方」の『計算式 1（PIDX（k）の求め方)』に従って求めてください。

tix_num

テキストインデクスの数

TIXWORK(i)

テキストインデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXWORK(i)＝
    row_num×string_num(i)×172＋row_num×（define_num(i)＋12)
```

row_num

adbidxrebuild コマンドの対象表に格納されている行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

define_num(i)

テキストインデクスのインデクス構成列の定義長

TIXCORWORK(i)

テキストインデクス作成用ワークファイル（表記ゆれ補正検索向け）

テキストインデクスの定義時に，「テキストインデクス表記ゆれ補正指定」を指定した場合に求める必要があります。

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXCORWORK(i)＝
    row_num×string_num(i)×202
```

row_num

adbidxrebuild コマンドの対象表に格納されている行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

6.21.3 adbmergechunk コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，複数のチャンクをマージするために，adbmergechunk コマンド実行時に作成される作業用一時ファイルの容量の求め方について説明します。

インデクスを定義している表に対してadbmergechunk コマンドを実行した場合，次に示すディレクトリ下に作業用一時ファイルが作成されます。

- -w オプションを指定した場合
  指定したディレクトリ下に作成されます。
- -w オプションを指定しない場合
  DB ディレクトリ下（$DBDIR/ADBWORK）に作成されます。

作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

作業用一時ファイルの容量＝

$$\left\lceil \sum_{i=1}^{idx_num} \left(RECFILE(i) + SORTFILE(i) + SORTWORK(i) + IDXWORK(i) \right) \div 1,024 \right\rceil + \left\lceil \sum_{i=1}^{tix_num} \left(TIXWORK(i) + TIXCORWORK(i) \right) \div 1,024 \right\rceil$$

変数の説明

idx_num
定義した B-tree インデクスの数

RECFILE(i)
インデクスレコードファイル
次に示す計算式から求めてください。

計算式（単位：バイト）

$$RECFILE(i) = \left\{ KEYSZ(i) + CTRL(i) \right\} \times \left\{ row_num \right\} + 640$$

KEYSZ(i)
i 番目の B-tree インデクスのキー長（単位：バイト）
「5.8.4 B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL(i)

i 番目の B-tree インデクス制御情報

- インデクス構成列のキー長がすべて固定長の場合：10 バイト
- インデクス構成列のキー長に可変長を含む場合：12 バイト

row_num

adbmergechunk コマンドの対象表のマージ元チャンクに格納されている行数

SORTFILE(i)

ソート結果用ファイル

次に示す計算式から求めてください。変数 *KEYSZ(i)*，変数 *CTRL(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTFILE(i) = \left\{KEYSZ(i) + CTRL(i)\right\} \times row_num + TCTRL(i)$$

TCTRL(i)

i 番目のテキストインデクス制御情報

- テキストインデクスの場合

  次に示す計算式から求めてください。

  計算式

  $$TCTRL(i) = (26 + \text{表が格納されるデータ用DBエリアのDBエリアファイル数} \times 72) \times 16 + 4$$

- テキストインデクスではない場合

  0 を代入してください。

SORTWORK(i)

ソート用ワークファイル

次に示す計算式から求めてください。変数 *RECFILE(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTWORK(i) = \left\{RECFILE(i) + 8 \times (row_num)\right\} \times 2$$

IDXWORK(i)

B-tree インデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXWORK(i) = \left\{12 + KEYSZDB(i)\right\} \times \left\{PIDX_LEAF + PIDX(2)\right\}$$

KEYSZDB(i)

i 番目のデータベース格納キー長（単位：バイト）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の変数 *KEYSZDB* を参照してください。ただし，可変長のデータは，常にキー長を定義長として求めてください。

PIDX_LEAF

B-tree インデクスの 1 段目のページ数（リーフページ）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の『計算式 1（PIDX_LEAF の求め方)』に従って求めてください。

PIDX(2)

B-tree インデクスの 2 段目のページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』に従って求めてください。

tix_num

テキストインデクスの数

TIXWORK(i)

テキストインデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$TIXWORK(i) = row_num \times string_num(i) \times 172 + row_num \times (define_num(i) + 12)$$

row_num

adbmergechunk コマンドの対象表のマージ元チャンクに格納されている行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

define_num(i)

テキストインデクスのインデクス構成列の定義長

TIXCORWORK(i)

テキストインデクス作成用ワークファイル（表記ゆれ補正検索向け）

テキストインデクスの定義時に，「テキストインデクス表記ゆれ補正指定」を指定した場合に求める必要があります。

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXCORWORK(i)＝
    row_num×string_num(i)×202
```

row_num

adbmergechunk コマンドの対象表のマージ元チャンクに格納されている行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

6.21.4 adbunarchivechunk コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，チャンクのアーカイブ状態を解除するために，adbunarchivechunk コマンド実行時に作成される作業用一時ファイルの容量の求め方について説明します。

インデクスを定義している表に対してadbunarchivechunk コマンドを実行した場合，次に示すディレクトリ下に作業用一時ファイルが作成されます。

- -w オプションを指定した場合
  指定したディレクトリ下に作成されます。
- -w オプションを指定しない場合
  DB ディレクトリ下（$DBDIR/ADBWORK）に作成されます。

作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

作業用一時ファイルの容量＝

$$\left\uparrow \sum_{i=1}^{idx_num} \left(RECFILE(i)+SORTFILE(i)+SORTWORK(i)+IDXWORK(i)\right) \div 1,024 \right\uparrow$$

$$+ \left\uparrow \sum_{i=1}^{tix_num} \left(TIXWORK(i)+TIXCORWORK(i)\right) \div 1,024 \right\uparrow$$

変数の説明

idx_num

定義した B-tree インデクスの数

RECFILE(i)

インデクスレコードファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$RECFILE(i) = \left\{KEYSZ(i) + CTRL(i)\right\} \times \left\{row_num\right\} + 640$$

KEYSZ(i)

i 番目の B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL(i)

i 番目の B-tree インデクス制御情報

- インデクス構成列のキー長がすべて固定長の場合：10 バイト
- インデクス構成列のキー長に可変長を含む場合：12 バイト

row_num

adbunarchivechunk コマンドの対象表のチャンクに対応するアーカイブファイルに格納されている行数

SORTFILE(i)

ソート結果用ファイル

次に示す計算式から求めてください。変数 *KEYSZ(i)*，変数 *CTRL(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTFILE(i) = \left\{KEYSZ(i) + CTRL(i)\right\} \times row_num + TCTRL(i)$$

TCTRL(i)

i 番目のテキストインデクス制御情報

- テキストインデクスの場合

  次に示す計算式から求めてください。

  計算式

  $$TCTRL(i) = (26 + \text{表が格納されるデータ用DBエリアのDBエリアファイル数} \times 72) \times 16 + 4$$

- テキストインデクスではない場合

0 を代入してください。

SORTWORK(i)

ソート用ワークファイル

次に示す計算式から求めてください。変数*RECFILE(i)*および変数*row_num* については，上記のインデクスレコードファイル（変数*RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTWORK(i) = \left\{ RECFILE(i) + 8 \times (row_num) \right\} \times 2$$

IDXWORK(i)

B-tree インデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXWORK(i) = \left\{ 12 + KEYSZDB(i) \right\} \times \left\{ PIDX_LEAF + PIDX(2) \right\}$$

KEYSZDB(i)

i 番目のデータベース格納キー長（単位：バイト）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の変数*KEYSZDB* を参照してください。ただし，可変長のデータは，常にキー長を定義長として求めてください。

PIDX_LEAF

B-tree インデクスの 1 段目のページ数（リーフページ）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i））の求め方」の『計算式 1（PIDX_LEAF の求め方)』に従って求めてください。

PIDX(2)

B-tree インデクスの 2 段目のページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i））の求め方」の『計算式 1（PIDX（k）の求め方)』に従って求めてください。

tix_num

テキストインデクスの数

TIXWORK(i)

テキストインデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXWORK(i)＝
    row_num×string_num(i)×172＋row_num×（define_num(i)＋12）
```

row_num

adbunarchivechunk コマンドの対象表のチャンクに対応するアーカイブファイルに格納されている行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

define_num(i)

テキストインデクスのインデクス構成列の定義長

TIXCORWORK(i)

テキストインデクス作成用ワークファイル（表記ゆれ補正検索向け）

テキストインデクスの定義時に，「テキストインデクス表記ゆれ補正指定」を指定した場合に求める必要があります。

次に示す計算式から求めてください。

計算式（単位：バイト）

```
TIXCORWORK(i)＝
    row_num×string_num(i)×202
```

row_num

adbunarchivechunk コマンドの対象表のチャンクに対応するアーカイブファイルに格納されている行数

string_num(i)

テキストインデクスのインデクス構成列に格納されている平均文字数

6.21.5 adbreorgsystemdata コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，システム表（実表）を再編成するために，adbreorgsystemdata コマンド実行時に作成される作業用一時ファイルの容量の求め方について説明します。

インデクスを定義している表に対してadbreorgsystemdata コマンドを実行した場合，次に示すディレクトリ下に作業用一時ファイルが作成されます。

- -w オプションを指定した場合
  指定したディレクトリ下に作成されます。
- -w オプションを指定しない場合
  DB ディレクトリ下（$DBDIR/ADBWORK）に作成されます。

作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

作業用一時ファイルの容量＝

$$\left\lceil \sum_{i=1}^{idx_num} \left(RECFILE(i) + SORTFILE(i) + SORTWORK(i) + IDXWORK(i) \right) \div 1,024 \right\rceil$$

変数の説明

idx_num

再編成対象のシステム表（実表）に定義されている B-tree インデクスの数

再編成対象のシステム表（実表）に定義されている B-tree インデクスについては，「付録 C.8　システム表（実表）の B-tree インデクス」を参照してください。

RECFILE(i)

インデクスレコードファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$RECFILE(i) = \left\{ KEYSZ(i) + CTRL(i) \right\} \times \left\{ row_num \right\} + 640$$

KEYSZ(i)

i 番目の B-tree インデクスのキー長（単位：バイト）

「5.8.4　B-tree インデクスのキー長（KEYSZ）の求め方」を基に，B-tree インデクスのキー長を求めてください。

なお，可変長データの B-tree インデクスのキー長を求める際は，「表 5-19　B-tree インデクスのキー長一覧」中の「実際のデータ長」を「データの定義長」で計算してください。

CTRL(i)

i 番目の B-tree インデクス制御情報

- インデクス構成列のキー長がすべて固定長の場合：10 バイト
- インデクス構成列のキー長に可変長を含む場合：12 バイト

row_num

`adbreorgsystemdata` コマンドの対象となる表に格納されている行数

SORTFILE(i)

ソート結果用ファイル

次に示す計算式から求めてください。変数 *KEYSZ(i)*，変数 *CTRL(i)* および変数 *row_num* については，上記のインデクスレコードファイル（変数 *RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTFILE(i) = \left\{KEYSZ(i) + CTRL(i)\right\} \times row_num$$

SORTWORK(i)

ソート用ワークファイル

次に示す計算式から求めてください。変数*RECFILE(i)*および変数*row_num* については，上記のインデクスレコードファイル（変数*RECFILE(i)*）の計算式を参照してください。

計算式（単位：バイト）

$$SORTWORK(i) = \left\{RECFILE(i) + 8 \times (row_num)\right\} \times 2$$

IDXWORK(i)

B-tree インデクス作成用ワークファイル

次に示す計算式から求めてください。

計算式（単位：バイト）

$$IDXWORK(i) = \left\{12 + KEYSZDB(i)\right\} \times \left\{PIDX_LEAF + PIDX(2)\right\}$$

KEYSZDB(i)

i 番目のデータベース格納キー長（単位：バイト）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i)）の求め方」の変数*KEYSZDB* を参照してください。ただし，可変長のデータは，常にキー長を定義長として求めてください。

PIDX_LEAF

B-tree インデクスの 1 段目のページ数（リーフページ）

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(1)　下位ページ用のセグメントで使用する格納ページ数（変数 IP_LOWER（i)）の求め方」の『計算式 1（PIDX_LEAF の求め方)』に従って求めてください。

PIDX(2)

B-tree インデクスの 2 段目のページ数

「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」の「(2)　上位ページ用のセグメントで使用する格納ページ数（変数 IP_UPPER（i)）の求め方」の『計算式 1（PIDX（k）の求め方)』に従って求めてください。

6.21.6　adbsyndict コマンド実行時の作業用一時ファイルの容量見積もり

ここでは，adbsyndict コマンドの実行時に作成される作業用一時ファイルの容量の求め方について説明します。

作業用一時ファイルの容量については，次に示す計算式から求めてください。

計算式（単位：キロバイト）

$$\text{作業用一時ファイルの容量} = \frac{\text{syn_list_file_size} \times 8}{1,024}$$

変数の説明

syn_list_file_size

同義語一覧定義ファイルのサイズ（単位：バイト）

6.22　コマンド実行時に増加するデータ量の見積もり

ここでは，次に示すコマンドを実行したときに増加するデータ量の求め方について説明します。

- adbmergechunk コマンド

6.22.1　adbmergechunk コマンド実行時に増加するデータ量の見積もり

マージチャンク処理の対象表にインデクスを定義している場合，adbmergechunk コマンドの実行中，およびadbmergechunk コマンドの中断時に，インデクスのデータ量が増加します。

そのため，adbmergechunk コマンドを実行する場合は，「5.8.3　B-tree インデクスのセグメントごとの格納ページ数の求め方」，「5.8.5　テキストインデクスのセグメントごとの格納ページ数の求め方」および「5.8.6　レンジインデクスを格納するセグメント数の求め方」で見積もった値に，マージ先チャンクのインデクス再作成に必要なデータ量を加算してください。インデクス再作成時に必要なデータ量については，マージ元チャンクのインデクスのデータ量の合計値と同等と仮定して計算してください。

adbmergechunk コマンド実行時にインデクスのデータ量が増加する理由については，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」の「(3)　adbmergechunk コマンドによるチャンクのマージ」のメモを参照してください。

6.23　コマンドを同時実行する際の考慮点

ここでは，次に示すコマンドを同時実行する際の考慮点について説明します。

- adbarchivechunk コマンド
- adbchgchunkcomment コマンド
- adbchgchunkstatus コマンド
- adbdbstatus コマンド
- adbexport コマンド
- adbgetcst コマンド
- adbidxrebuild コマンド
- adbimport コマンド
- adbmergechunk コマンド
- adbmodbuff コマンド
- adbreorgsystemdata コマンド
- adbsql コマンド
- adbstat コマンド
- adbsyndict コマンド
- adbunarchivechunk コマンド

なお，これらのコマンドを同時実行する場合，同時に実行する AP もあわせて考慮する必要があります。

6.23.1　コマンドの最大同時実行数について

「6.23　コマンドを同時実行する際の考慮点」で説明したコマンドの最大同時実行数は，サーバ定義 adb_sys_max_users オペランドの値と，HADB サーバに接続中の AP の数によって決まります。コマンドの最大同時実行数を求める計算式を次に示します。

計算式

コマンドの最大同時実行数 ＝ MAX_USERS － AP_CNCT_NUM

変数の説明

MAX_USERS

サーバ定義adb_sys_max_users オペランドに指定した最大同時接続数

「7.2.1　システム構成に関するオペランド（set 形式）」の adb_sys_max_users オペランドを参照してください。

AP_CNCT_NUM

HADB サーバに接続中の AP の数

HADB サーバに接続中の AP は，adbls -d cnct コマンドで確認できます。adbls -d cnct コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示)』を参照してください。

上記の計算式で求めた値を超えた数のコマンドを同時に実行できません。最大同時実行数の上限を超えた場合，エラーとなります（KFAA30932-E メッセージが出力されます)。

なお，adbstat コマンドについては，最大同時実行数の上限はありません。

6.23.2　コマンドの実行時に使用する処理リアルスレッド数に関する考慮点

(1)　コマンドの実行時に使用する処理リアルスレッド数の最大値

「表 6-28　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧」に記載しているコマンドを実行する場合，次の計算式で求める「コマンドの実行時に使用する処理リアルスレッド数」が確保できるようにしてください。

計算式

コマンドの実行時に使用する処理リアルスレッド数の最大値＝
SYS_RTHD_NUM - AP_USE_RTHD_NUM

複数のコマンドを同時実行する場合，同時実行するすべてのコマンドで必要な処理リアルスレッド数の合計値となります。

変数の説明

SYS_RTHD_NUM

サーバ定義adb_sys_rthd_num オペランドの指定値

adb_sys_rthd_num オペランドには，SQL 文およびコマンドの処理を行う際に使用する処理リアルスレッド数の最大値を指定します。adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式)」の adb_sys_rthd_num オペランドを参照してください。

AP_USE_RTHD_NUM

AP の実行時に使用する処理リアルスレッド数の合計

AP の実行時に使用する処理リアルスレッド数については，次に示すオペランドを参照してください。

- サーバ定義adb_sql_exe_max_rthd_num オペランド

  「7.2.2　性能に関するオペランド（set 形式)」の adb_sql_exe_max_rthd_num オペランドを参照してください。

- クライアント定義adb_sql_exe_max_rthd_num オペランド

  マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

(2) コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプション

コマンドの実行時に使用する処理リアルスレッド数は，次の表に示すオペランドおよびコマンドオプションで指定できます。

表 6-28 コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧

項番	コマンド名	オペランド名	コマンドオプション名	最低限必要な処理リアルスレッド数
1	adbarchivechunk コマンド	サーバ定義 adb_sql_exe_max_rthd_num オペランド	アーカイブチャンクオプション adb_arcv_rthd_num	3
2	adbexport コマンド		エクスポートオプション adb_export_rthd_num	3
3	adbgetcst コマンド		コスト情報収集オプション adb_getcst_rthd_num	2
4	adbidxrebuild コマンド		インデクス再作成オプション adb_idxrebuild_rthd_num	3
5	adbimport コマンド		インポートオプション adb_import_rthd_num	2
6	adbmergechunk コマンド		マージチャンクオプション adb_mergechunk_rthd_num	3
7	adbunarchivechunk コマンド		アンアーカイブチャンクオプション adb_unarcv_rthd_num	2
8	adbsql コマンド	• サーバ定義 adb_sql_exe_max_rthd_num オペランド • クライアント定義 adb_sql_exe_max_rthd_num オペランド	なし	0
9	adbreorgsystemdata コマンド※	なし	なし	3

注※

adbreorgsystemdata コマンドが使用する処理リアルスレッド数は，3 です。コマンドオプションによる指定はできません。

adb_sql_exe_max_rthd_num オペランドには，最大 SQL 処理リアルスレッド数を指定します。また，コマンドオプションを指定している場合は，adb_sql_exe_max_rthd_num オペランドではなく，コマンドオプションの指定が優先されます。

コマンドを同時実行する場合，「(1)　コマンドの実行時に使用する処理リアルスレッド数の最大値」で求めたコマンドの実行時に使用する処理リアルスレッド数の最大値を超えると，処理リアルスレッドの確保待ちが発生するおそれがあります。発生した場合，ほかのコマンドや AP が終了して必要な処理リアルスレッド数を確保したあとに，コマンドの処理を続行します。

そのため，「表 6-28　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧」で説明したコマンドを同時実行する場合，「表 6-28　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧」で説明したオペランドおよびコマンドオプションで，適切な処理リアルスレッド数を指定するようにしてください。

メモ

処理リアルスレッドの確保待ち状態になったコマンドについては，adbls -d cnct コマンドで確認できます。adbls -d cnct コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示)』を参照してください。

(3)　コマンドを同時実行するときの処理リアルスレッド数の目安

コマンドの実行時に使用する処理リアルスレッド数を適切に見積もらないと，コマンドの実行性能に影響を及ぼします。処理リアルスレッド数の目安を次に示します。

- **コマンドの同時実行数が少ない期間**

  「(1)　コマンドの実行時に使用する処理リアルスレッド数の最大値」で求めたコマンドの実行時に使用する処理リアルスレッド数の最大値の 8 割程度を使用するようにしてください。

- **コマンドの同時実行数が多い期間**

  コマンドの同時実行時，処理リアルスレッドの確保待ちが発生しても，運用上問題とならないように，処理リアルスレッド数を使用するようにしてください。

6.23.3　コマンド同時実行時の排他制御に関する考慮点

HADB サーバは，「6.23　コマンドを同時実行する際の考慮点」で示すコマンドを実行すると，「2.10.4 確保される排他資源と排他制御モード」に従って，排他資源を確保します。

複数のコマンドを同時に実行する場合，排他資源が確保される際に競合が発生しないよう注意してください。排他資源が確保される際に競合が発生した場合，通常は，エラーとなります（KFAA50290-E メッセージが出力されます)。

ただし，一部例外があります（エラーとならないで，排他資源を確保できるまで待つことがあります)。詳細については，「2.10.4　確保される排他資源と排他制御モード」を参照してください。

6.23.4　コマンド同時実行時のメモリ所要量に関する考慮点

コマンドを同時実行する場合に使用するメモリ所要量は，同時実行するコマンドが使用するメモリ所要量の合計になります。

対象のコマンドが使用するメモリ所要量については，「6.3　HADB サーバのメモリ所要量の見積もり」を参照してください。

6.23.5　コマンド同時実行時のシステムログファイルの容量に関する考慮点

コマンドを同時実行する場合に使用するシステムログファイルの容量は，同時実行するコマンドが使用するシステムログファイルの容量の合計になります。

対象のコマンドが使用するシステムログファイルの容量については，「6.12　システムログファイルの容量見積もり」を参照してください。

6.24　クライアントグループ機能を適用する際の考慮点

ここでは，クライアントグループ機能を適用する際の考慮点について説明します。

6.24.1　グループの接続数を設定する際の考慮点

クライアントグループ機能を適用すると，HADB クライアントおよびコマンドが使用できる接続数を，グループごとに設定できます。クライアントグループ機能で設定できる接続数の種別を，次に示します。

- 最大同時接続数
- 接続保証数

クライアントグループ機能の最大同時接続数および接続保証数については，「2.12.3　グループごとの接続数・処理リアルスレッド数の設定」を参照してください。

(1)　グループの最大同時接続数および接続保証数を設定する際の考慮点

クライアントグループ機能を適用して，グループの最大同時接続数および接続保証数を設定する際には，次に示す点を考慮する必要があります。

- HADB サーバの最大同時接続数をどうするか

  サーバ定義adb_sys_max_users オペランドの指定値を検討します。サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set 形式）」の adb_sys_max_users オペランドを参照してください。
- グループが使用する最大同時接続数および接続保証数を，どれだけ確保するか

  グループの最大同時接続数および接続保証数は，サーバ定義adbcltgrp オペランドで指定できます。サーバ定義adbcltgrp オペランドについては，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」を参照してください。
- 無所属の（グループに所属していない）HADB クライアントやコマンドが使用する接続数は，どの程度必要になるか

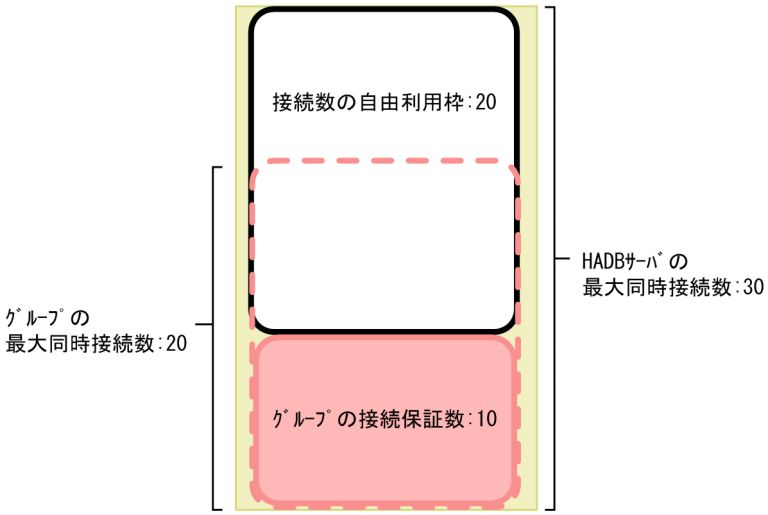
  HADB サーバの最大同時接続数と，各グループの最大同時接続数および接続保証数に従って，無所属の HADB クライアントやコマンドが使用できる接続数が決まります。

  無所属の HADB クライアントやコマンドが使用できる接続数は，「**接続数の自由利用枠**」の範囲となります。接続数の自由利用枠は，HADB サーバの最大同時接続数から各グループの接続保証数を除いた残りの接続数になります。

  この関係性を考慮した上で，各グループの最大同時接続数および接続保証数に適切な値を設定してください。

クライアントグループ機能を適用した場合の，接続数の使用範囲の概要について，次の図に示します。

図 6-7　クライアントグループ機能を適用した場合の，接続数の使用範囲の概要



(2)　クライアントグループ機能を適用した場合の接続数の求め方

クライアントグループ機能を適用した場合の接続数の求め方について，説明します。

■グループの最大同時接続数と接続保証数の求め方

グループの最大同時接続数および接続保証数は，サーバ定義adbcltgrp オペランドの指定値およびサーバ定義adb_sys_max_users オペランドの指定値に従って決まります。グループの最大同時接続数および接続保証数に設定した範囲で，グループに所属している HADB クライアントとコマンドは接続数を使用できます。グループの最大同時接続数および接続保証数について，次の表に示します。

表 6-29　グループの最大同時接続数と接続保証数

項番	接続数の種別	各種別の値
1	最大同時接続数	次の計算式で求めた値になります。 MIN（A ，（B－C）） 変数の説明 A:所属しているグループを設定した サーバ定義adbcltgrpオペランドの-mオプションの指定値 B：サーバ定義adb_sys_max_usersオペランドの指定値 C：ほかのサーバ定義adbcltgrpオペランドの -uオプションの指定値の合計
2	接続保証数	次の値になります。 所属しているグループを設定したサーバ定義adbcltgrp オペランドの-u オプションの指定値

■接続数の自由利用枠の求め方

接続数の自由利用枠は，グループに所属しているかどうかに関係なく，すべての HADB クライアントやコマンドが使用できる接続数です。接続数の自由利用枠は，次の計算式で求められます。

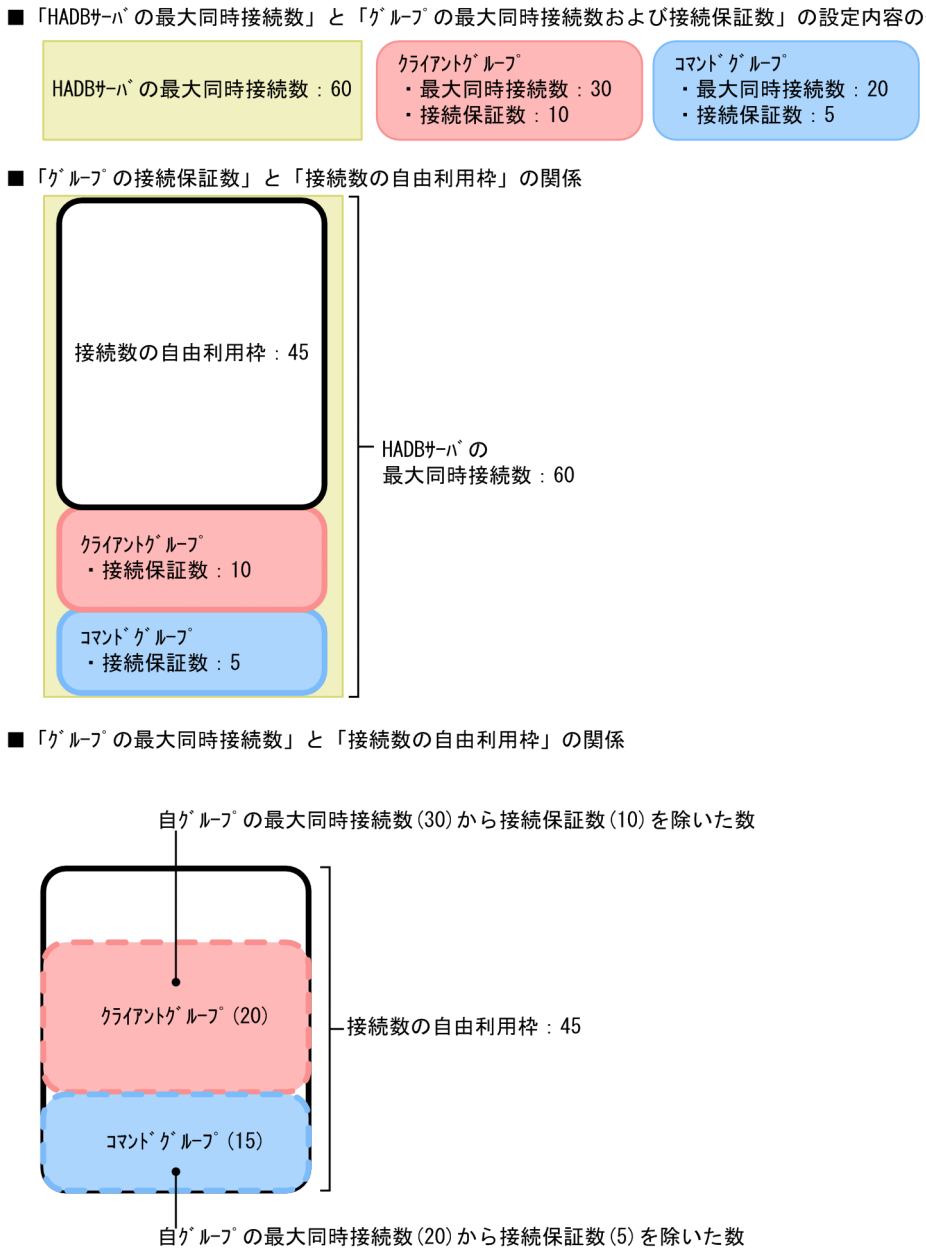
計算式

```
A－B

  変数の説明
    A：サーバ定義adb_sys_max_usersオペランドの指定値
    B：サーバ定義adbcltgrpオペランドの
       -uオプションの指定値の合計
```

(3)　クライアントグループ機能を適用した場合の接続数の関係

クライアントグループ機能を適用した場合に，「(2)　クライアントグループ機能を適用した場合の接続数の求め方」で示した接続数の種別の値が，それぞれどのように関係しているかについて，次の図に示します。

図 6-8　クライアントグループ機能を適用した場合の接続数の関係



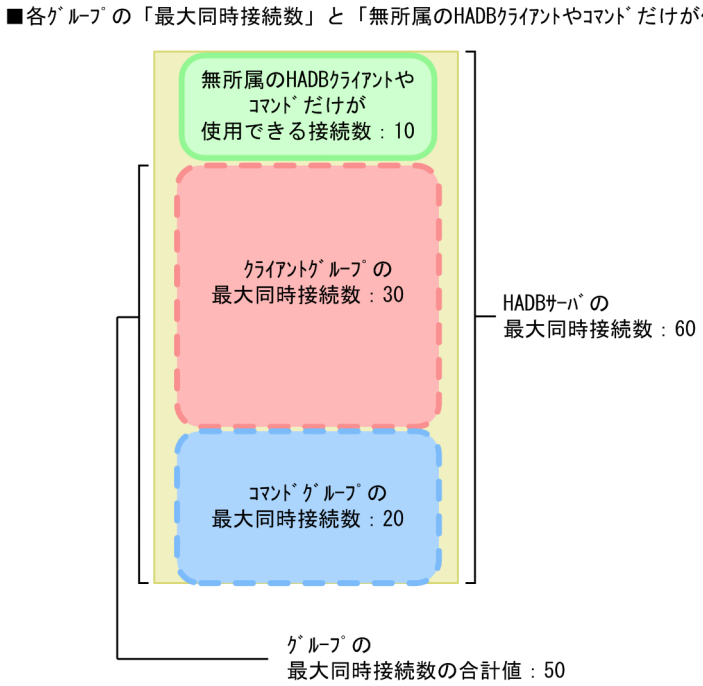
［説明］

- HADB サーバの最大同時接続数（60）から，クライアントグループの接続保証数（10）とコマンドグループの接続保証数（5）が確保されます。そのため，接続保証数で確保されていない残りの接続数が，接続数の自由利用枠（45）となります。接続数の自由利用枠は，グループに所属しているかどうかに関係なく，すべての HADB クライアントとコマンドが使用できる接続数になります。
- 接続数の自由利用枠（45）の中で，各グループは，自グループの最大同時接続数から接続保証数を除いた数を上限として，接続数を使用できます。この例の場合，クライアントグループは，自グループの最大同時接続数（30）から接続保証数（10）を除いた数（20）を上限として，接続数の自由利用枠を使用できます。また，コマンドグループは，自グループの最大同時接続数（20）から接続保証数（5）を除いた数（15）を上限として，接続数の自由利用枠を使用できます。

メモ

グループの最大同時接続数の合計値によっては，接続数の自由利用枠の一部が「無所属の HADB クライアントやコマンドだけが使用できる接続数」になることがあります。「図 6-8　クライアントグループ機能を適用した場合の接続数の関係」の設定内容を基に，無所属の HADB クライアントやコマンドだけが使用できる接続数と，グループの最大同時接続数の合計値との関係性について，次の図に示します。

図 6-9　無所属の HADB クライアントやコマンドだけが使用できる接続数と，グループの最大同時接続数の合計値との関係性



［説明］

グループに設定した最大同時接続数の合計値が，HADB サーバの最大同時接続数より少ない場合，グループが常に使用しない接続数が発生します。この接続数が，「無所属の HADB クライアントやコマンドだけが使用できる接続数」になります。

この例では，各グループの最大同時接続数の合計値（50）が，HADB サーバの最大同時接続数（60）より少ないため，グループが常に使用しない接続数（10）が発生します。この接続数（10）は，無所属の HADB クライアントやコマンドだけが使用できます。

(4)　最大同時接続数に関する警告メッセージの出力契機の設定

サーバ定義のadbcltgrp オペランドで-w オプションを指定すると，グループが使用できる最大同時接続数の残り枠が少なくなったときに，警告メッセージKFAA40020-W を出力できます。adbcltgrp オペランドについては，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」のadbcltgrp オペランドを参照してください。

6.24.2　グループの処理リアルスレッド数を設定する際の考慮点

クライアントグループ機能を適用すると，HADB クライアントおよびコマンドが使用できる処理リアルスレッド数を，グループごとに設定できます。クライアントグループ機能で設定できる処理リアルスレッド数の種別を，次に示します。

- グループで使用できる処理リアルスレッドの最大数
- グループで使用できる処理リアルスレッドの保証数

グループで使用できる処理リアルスレッドの最大数および保証数については，「2.12.3　グループごとの接続数・処理リアルスレッド数の設定」を参照してください。

(1)　グループで使用できる処理リアルスレッドの最大数および保証数を設定する際の考慮点

クライアントグループ機能を適用して，グループで使用できる処理リアルスレッドの最大数および保証数を設定する際には，次に示す点を考慮する必要があります。

- HADB サーバの処理リアルスレッド数をどうするか

  サーバ定義adb_sys_rthd_num オペランドの指定値を検討します。サーバ定義adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」のadb_sys_rthd_num オペランドを参照してください。
- グループが使用する処理リアルスレッドの最大数および保証数を，どれだけ確保するか

  グループで使用できる処理リアルスレッドの最大数および保証数は，サーバ定義adbcltgrp オペランドで指定できます。サーバ定義adbcltgrp オペランドについては，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」を参照してください。
- 無所属の（グループに所属していない）HADB クライアントやコマンドが使用する処理リアルスレッド数は，どの程度必要になるか

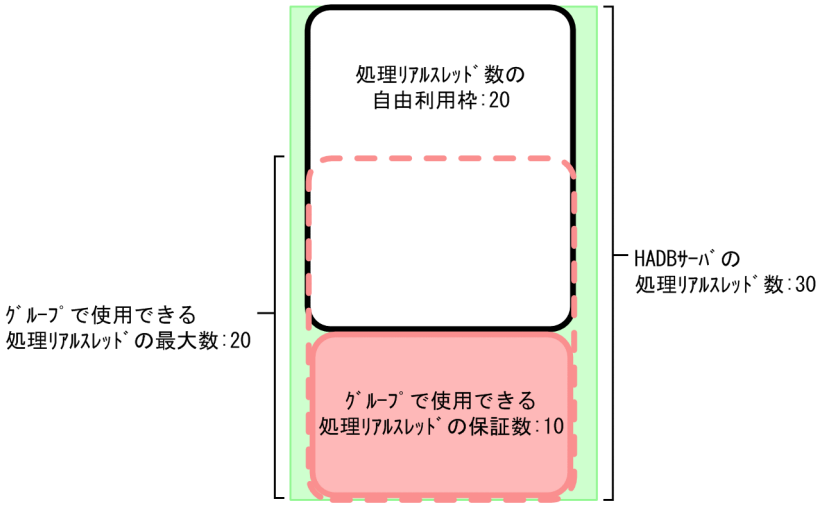
HADB サーバの処理リアルスレッド数と，各グループで使用できる処理リアルスレッドの最大数および保証数に従って，無所属の HADB クライアントやコマンドが使用できる処理リアルスレッド数が決まります。

無所属の HADB クライアントやコマンドが使用できる処理リアルスレッド数は，「**処理リアルスレッド数の自由利用枠**」の範囲となります。処理リアルスレッド数の自由利用枠は，HADB サーバの処理リアルスレッド数から，各グループで使用できる処理リアルスレッドの保証数を除いた残りの処理リアルスレッド数になります。

この関係性を考慮した上で，各グループで使用できる処理リアルスレッドの最大数および保証数に適切な値を設定してください。

クライアントグループ機能を適用した場合の，処理リアルスレッド数の使用範囲の概要について，次の図に示します。

図 6-10　クライアントグループ機能を適用した場合の，処理リアルスレッド数の使用範囲の概要



(2)　クライアントグループ機能を適用した場合の処理リアルスレッド数の求め方

クライアントグループ機能を適用した場合の処理リアルスレッド数の求め方について，説明します。

■グループで使用できる処理リアルスレッドの最大数および保証数の求め方

グループで使用できる処理リアルスレッドの最大数および保証数は，サーバ定義adbcltgrp オペランドの指定値およびサーバ定義adb_sys_rthd_num オペランドの指定値に従って決まります。グループで使用できる処理リアルスレッドの最大数および保証数に設定した範囲で，グループに所属している HADB クライアントとコマンドは処理リアルスレッド数を使用できます。グループで使用できる処理リアルスレッドの最大数および保証数について，次の表に示します。

表 6-30　グループで使用できる処理リアルスレッドの最大数および保証数

項番	処理リアルスレッド数の種別	各種別の値
1	処理リアルスレッドの最大数※	次の計算式で求めた値になります。 MIN（A，（B−C）） 変数の説明 A：所属しているグループを設定した サーバ定義adbcltgrpオペランドの-rオプションの指定値 B：サーバ定義adb_sys_rthd_numオペランドの指定値 C：ほかのサーバ定義adbcltgrpオペランドの-eオプションの指定値の合計
2	処理リアルスレッドの保証数	次の値になります。 所属しているグループを設定したサーバ定義adbcltgrp オペランドの-e オプションの指定値

注※

次のコマンドは，*最低限必要な処理リアルスレッド数*が確保されないと実行できません。そのため，処理リアルスレッド数の最大数は，*最低限必要な処理リアルスレッド数*以上の値が確保されるように調整する必要があります。もし，*最低限必要な処理リアルスレッド数*以上の値が確保されない場合は，次のコマンドを実行してもエラーとなるおそれがあります。

- adbimport コマンド
- adbidxrebuild コマンド
- adbgetcst コマンド
- adbexport コマンド
- adbmergechunk コマンド
- adbarchivechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド

各コマンドの*最低限必要な処理リアルスレッド数*は，「6.23.2　コマンドの実行時に使用する処理リアルスレッド数に関する考慮点」の「(2)　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプション」の「表 6-28　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧」を参照してください。

■処理リアルスレッド数の自由利用枠の求め方

処理リアルスレッド数の自由利用枠は，グループに所属しているかどうかに関係なく，すべての HADB クライアントやコマンドが使用できる処理リアルスレッド数です。処理リアルスレッド数の自由利用枠は，次の計算式で求められます。

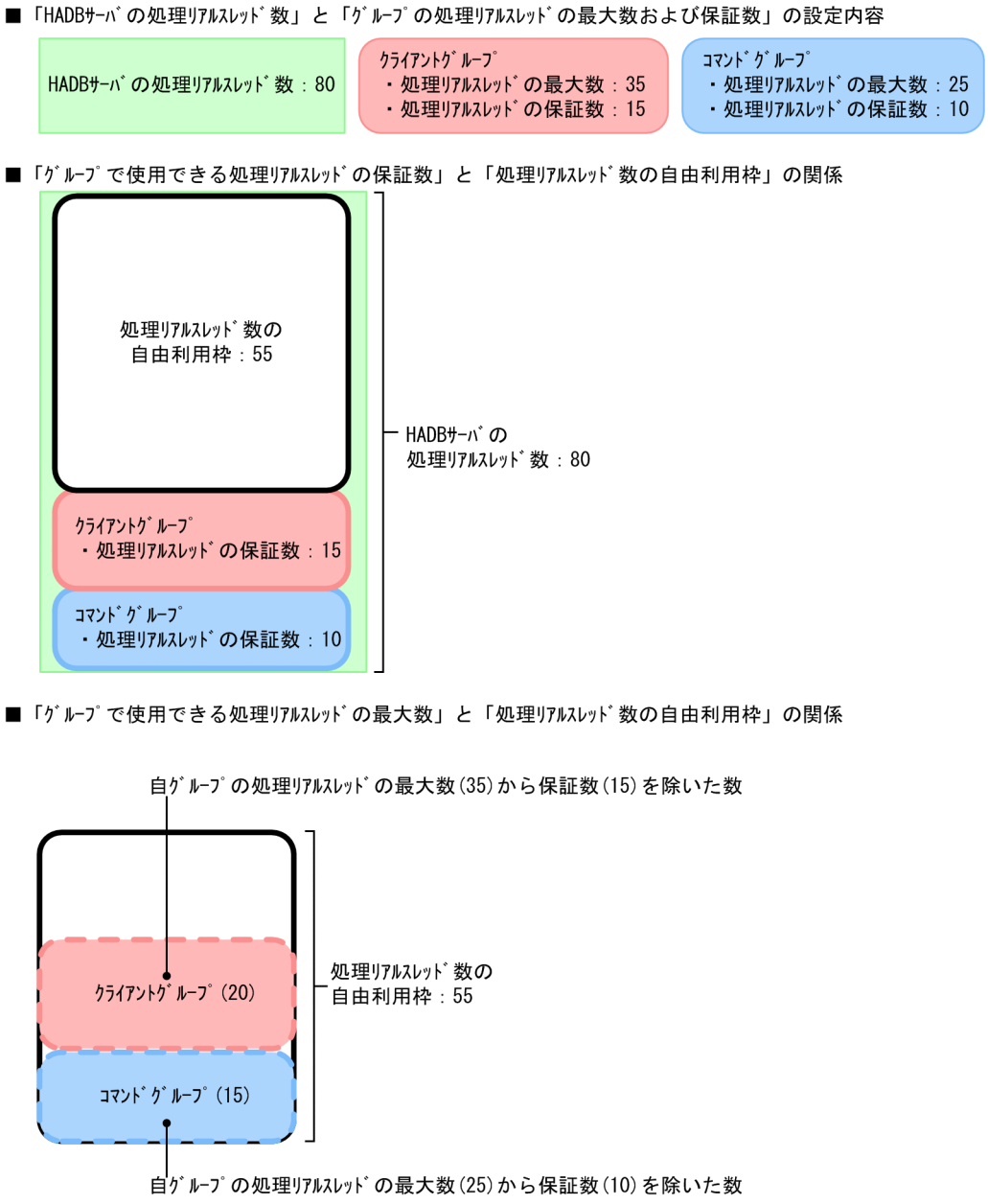
計算式

```
A－B

  変数の説明
    A：サーバ定義adb_sys_rthd_numオペランドの指定値
    B：サーバ定義adbcltgrpオペランドの
       -eオプションの指定値の合計
```

(3)　クライアントグループ機能を適用した場合の処理リアルスレッド数の関係

クライアントグループ機能を適用した場合に，「(2)　クライアントグループ機能を適用した場合の処理リアルスレッド数の求め方」で示した処理リアルスレッド数の種別の値が，それぞれどのように関係しているかについて，次の図に示します。

図 6-11　クライアントグループ機能を適用した場合の処理リアルスレッド数の関係



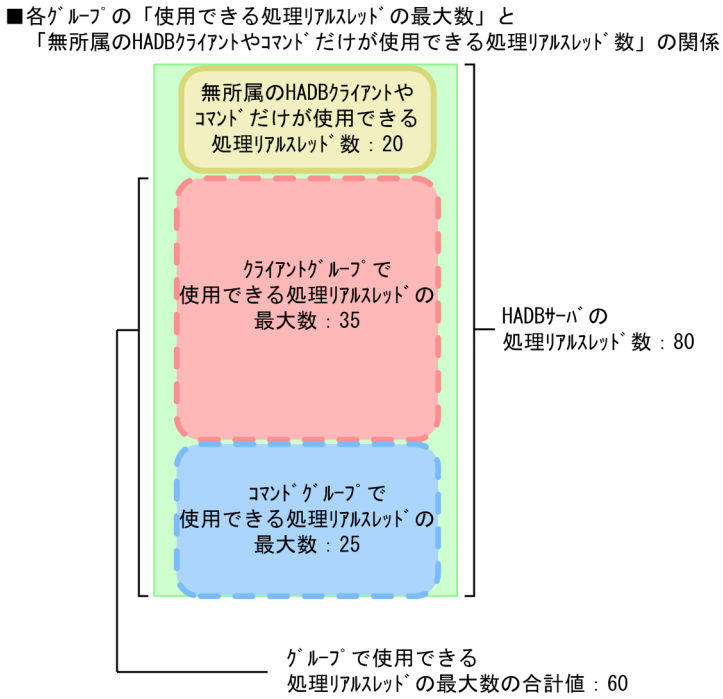
［説明］

- HADB サーバの処理リアルスレッド数（80）から，クライアントグループの処理リアルスレッドの保証数（15）とコマンドグループの処理リアルスレッドの保証数（10）が確保されます。そのため，処理リアルスレッドの保証数で確保されていない残りの処理リアルスレッド数が，処理リアルスレッド数の自由利用枠（55）となります。処理リアルスレッド数の自由利用枠は，グループに所属しているかどうかに関係なく，すべての HADB クライアントとコマンドが使用できる処理リアルスレッド数になります。
- 処理リアルスレッド数の自由利用枠（55）の中で，各グループは，自グループで使用できる処理リアルスレッドの最大数から保証数を除いた数を上限として，処理リアルスレッド数を使用できます。この例の場合，クライアントグループは，自グループで使用できる処理リアルスレッドの最大数（35）から保証数（15）を除いた数（20）を上限として，処理リアルスレッド数の自由利用枠を使用できます。また，コマンドグループは，自グループで使用できる処理リアルスレッドの最大数（25）から保証数（10）を除いた数（15）を上限として，処理リアルスレッド数の自由利用枠を使用できます。

メモ

グループで使用できる処理リアルスレッドの最大数の合計値によっては，処理リアルスレッド数の自由利用枠の一部が「無所属の HADB クライアントやコマンドだけが使用できる処理リアルスレッド数」になることがあります。「図 6-11　クライアントグループ機能を適用した場合の処理リアルスレッド数の関係」の設定内容を基に，無所属の HADB クライアントやコマンドだけが使用できる処理リアルスレッド数と，グループで使用できる処理リアルスレッドの最大数の合計値との関係性について，次の図に示します。

図 6-12　無所属の HADB クライアントやコマンドだけが使用できる処理リアルスレッド数と，グループで使用できる処理リアルスレッドの最大数の合計値との関係性



［説明］

グループで使用できる処理リアルスレッドの最大数の合計値が，HADB サーバの処理リアルスレッド数より少ない場合，グループが常に使用しない処理リアルスレッド数が発生します。この処理リアルスレッド数が，「無所属の HADB クライアントやコマンドだけが使用できる処理リアルスレッド数」になります。

この例では，各グループで使用できる処理リアルスレッドの最大数の合計値（60）が，HADB サーバの処理リアルスレッド数（80）より少ないため，グループが常に使用しない処理リアルスレッド数（20）が発生します。この処理リアルスレッド数（20）は，無所属の HADB クライアントやコマンドだけが使用できます。

6.24.3　クライアントグループ機能で使用される接続数と処理リアルスレッド数の関係

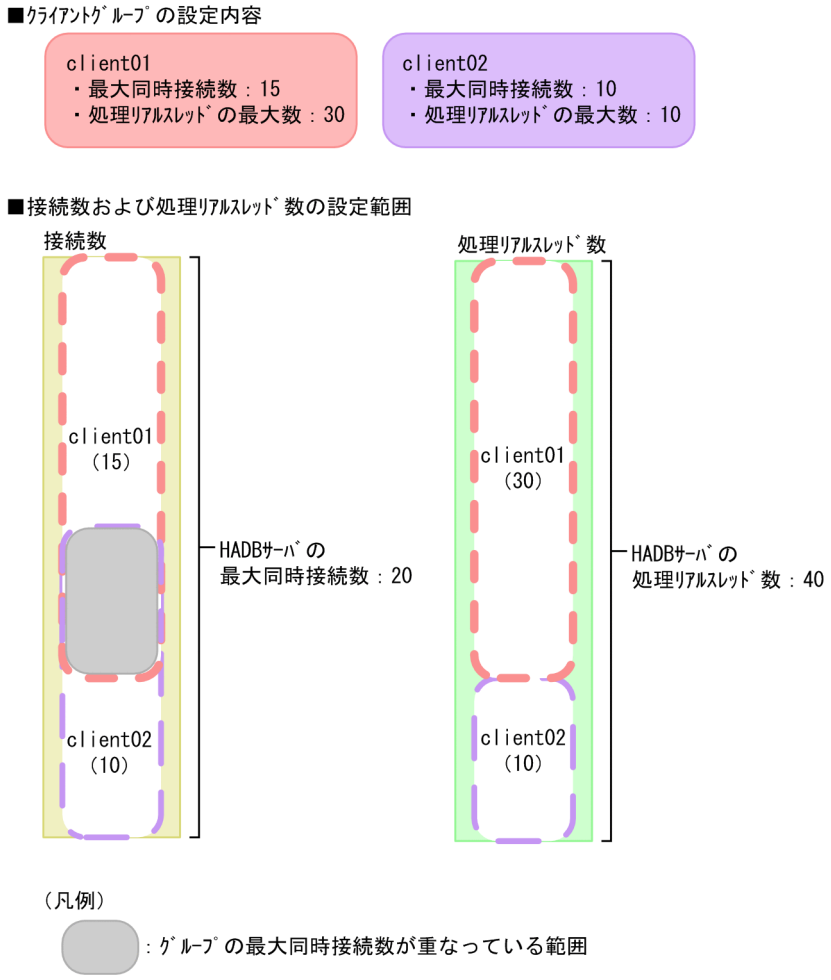
ここでは，クライアントグループ機能を適用した場合に使用される接続数と処理リアルスレッド数の関係について，設定例を基に説明します。

(1)　設定例 1（使用できる接続数と処理リアルスレッド数の上限を設定する）

ここでは，クライアントグループ機能を適用して，グループの最大同時接続数と，グループで使用できる処理リアルスレッドの最大数を設定する例について，説明します。

グループの最大同時接続数と，グループで使用できる処理リアルスレッドの最大数を設定することで，特定のグループだけが，接続数や処理リアルスレッド数を占有することを防げます。

図 6-13　使用できる接続数と処理リアルスレッド数の上限を設定した例



[説明]

- 各グループに設定した最大同時接続数に従って，対象のグループは接続数を使用します。最大同時接続数を超えて，接続数を使用できません。そのため，1 つのグループが，接続数すべてを占有することを防げます。処理リアルスレッド数についても同様です。

  この例の場合，クライアントグループ「client01」は，グループの最大同時接続数（15），およびグループで使用できる処理リアルスレッドの最大数（30）が使用できる上限となります（「client01」が，すべての接続数と処理リアルスレッド数を占有しません）。

- この例のように，各グループに設定した最大同時接続数の範囲が重なっている場合，重なっている範囲の接続数については，ほかのグループが接続数を使用していない分だけ使用できます。仮に，クライアントグループ「client01」が接続数（13）を使用している場合，クライアントグループ「client02」は接続数（7）を使用できます。

  各グループに設定した，使用できる処理リアルスレッドの最大数の範囲が重なっている場合についても，同様です。

- この例では，各クライアントグループに接続保証数を設定していません（接続保証数（0）を指定しています）。そのため，クライアントグループ「client01」および「client02」が接続数を使用していない場合は，無所属の HADB クライアントやコマンドが，接続数を使用できます。無所属の HADB クライアントやコマンドが接続数を使用していると，各グループは，それぞれの最大同時接続数の上限値まで使用できないことがあります。処理リアルスレッド数についても同様です。

この例で設定したクライアントグループの指定例を，次に示します。

■サーバ定義adbcltgrp オペランドの指定例

```
adbcltgrp -g client01 -m 15 -u 0 -r 30 -e 0
adbcltgrp -g client02 -m 10 -u 0 -r 10 -e 0
```

■クライアント定義adb_clt_group_name オペランドの指定例（クライアントグループ「client01」に所属させる場合）

```
adb_clt_group_name = client01
```

■クライアント定義adb_clt_group_name オペランドの指定例（クライアントグループ「client02」に所属させる場合）

```
adb_clt_group_name = client02
```

メモ

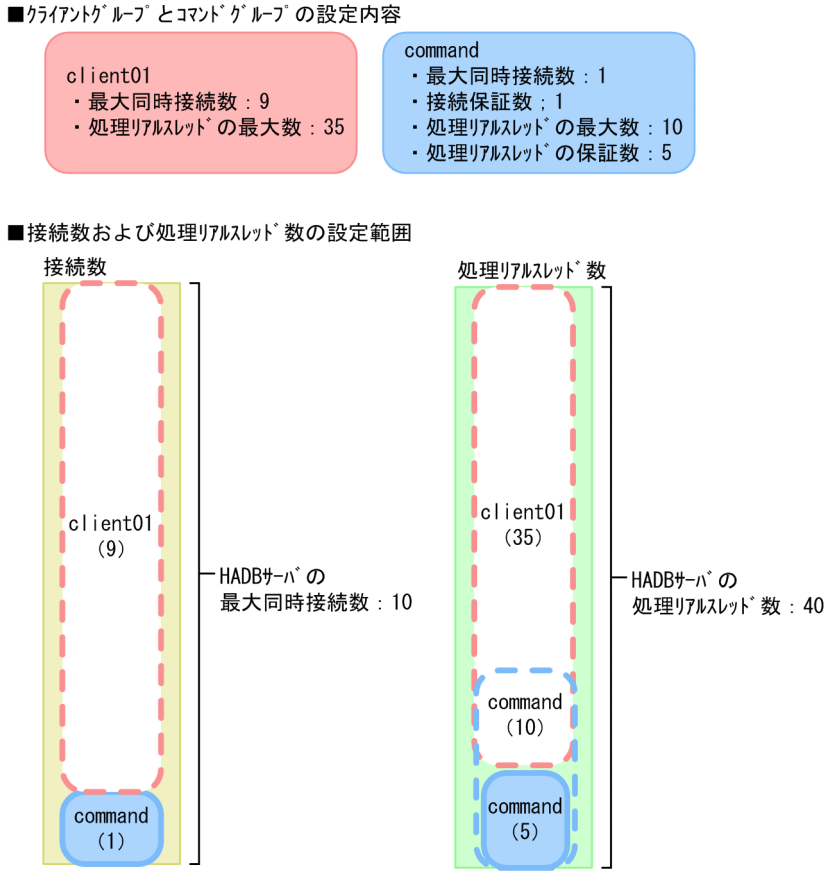
- サーバ定義adbcltgrp オペランドについては，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」の adbcltgrp（クライアントグループを設定する場合），または adbcltgrp（コマンドグループを設定する場合）を参照してください。
- クライアント定義adb_clt_group_name オペランドについては，マニュアル『HADB　AP 開発ガイド』の『システム構成に関するオペランド』を参照してください。

(2)　設定例 2（最低限使用できる接続数と処理リアルスレッド数を設定する）

ここでは，クライアントグループ機能を適用して，グループの最大同時接続数，接続保証数，グループで使用できる処理リアルスレッドの最大数，および使用できる処理リアルスレッドの保証数を設定する例について，説明します。

グループの最大同時接続数と，グループで使用できる処理リアルスレッドの最大数を設定することで，特定のグループだけが，接続数や処理リアルスレッド数を占有することを防げます。また，グループの接続保証数と，グループで使用できる処理リアルスレッドの保証数を設定することで，自グループ以外の HADB クライアントの影響を受けなくなります。

図 6-14　最低限使用できる接続数と処理リアルスレッド数を設定した例



[説明]

コマンドグループ「command」は，設定した最大同時接続数と接続保証数に従って，常に 1 つの接続数を確保します。そのため，「command」以外の HADB クライアントの影響を受けません。クライアントグループ「client01」は，設定した最大同時接続数に従って，接続数を使用します。

また，「command」は，設定した処理リアルスレッドの保証数に従って，常に 5 個の処理リアルスレッド数を確保します。さらに，「command」には，処理リアルスレッドの最大数に 10 を指定しています。そのため，「client01」が使用していない処理リアルスレッドがあれば，追加で 5 個使用できます。

なお，「command」および「client01」が接続数や処理リアルスレッド数を使用していない場合は，無所属の HADB クライアントは，接続数を最大 9 個，処理リアルスレッド数を最大 35 個まで使用できます。

この例で設定したクライアントグループとコマンドグループの指定例を，次に示します。

■サーバ定義adbcltgrp オペランドの指定例

```
adbcltgrp -g client01 -m 9 -u 0 -r 35 -e 0
adbcltgrp -g command -m 1 -u 1 -r 10 -e 5
```

■クライアント定義adb_clt_group_name オペランドの指定例（クライアントグループ「client01」に所属させる場合）

```
adb_clt_group_name = client01
```

■クライアント定義adb_clt_group_name オペランドの指定例（コマンドグループ「command」に所属させる場合）

```
adb_clt_group_name = command
```

メモ

- サーバ定義adbcltgrp オペランドについては，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」の adbcltgrp（クライアントグループを設定する場合），または adbcltgrp（コマンドグループを設定する場合）を参照してください。
- クライアント定義adb_clt_group_name オペランドについては，マニュアル『HADB　AP 開発ガイド』の『システム構成に関するオペランド』を参照してください。

6.25　作業表に関する見積もり

ここでは，次の項目の見積もり方について説明します。

- グローバル作業表用のグローバルバッファのページ数
- ローカル作業表用バッファのページ数
- ハッシュテーブルを使用した検索時に作成される作業表の数

6.25.1　グローバル作業表用のグローバルバッファのページ数の見積もり

ここでは，グローバル作業表用のグローバルバッファのページ数の見積もり方について説明します。

ここで求めた値を，サーバ定義adb_dbbuff_wrktbl_glb_blk_num オペランドに指定してください。
adb_dbbuff_wrktbl_glb_blk_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_dbbuff_wrktbl_glb_blk_num オペランドを参照してください。

グローバル作業表用のグローバルバッファのページ数の見積もり方を次に示します。

見積もり方

1. 作業表の構成列に関する情報を求める

   マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『SQL を実行した場合に作成される作業表について』を参照してください。表中の『作業表が作成される SQL』のうち，次に示す項目について，作業表の構成列のデータ型，データ長，および作業表への格納行数を求めてください。

   - FROM 句に複数の表参照を指定した場合
   - 導出表を指定した場合
   - ビュー表を指定した場合
   - WITH 句を指定した場合
   - 表関数導出表を指定した場合
   - 結合表を指定した場合
   - 副問合せを指定した場合
   - 再帰的問合せの指定がある場合

2. グローバル作業表用のグローバルバッファのページ数を求める

   「5.9.1　作業表用 DB エリアの総ページ数の求め方」の計算式を使用して，グローバル作業表用のグローバルバッファのページ数を求めてください。

   手順 1.で求めた値を，計算式に代入する変数としてください。

作業表のページ数が作業表用のバッファサイズ内で収まらない場合，収まらない分の作業表が作業表用 DB エリアファイルに出力されるため，処理性能低下の原因となります。そのため，作業表用のバッファサイズを適切に見積もる必要があります。

6.25.2　ローカル作業表用バッファのページ数の見積もり

ここでは，ローカル作業表が使用するローカル作業表用バッファのページ数の見積もり方について説明します。ローカル作業表とは，リアルスレッドごとに作成される，リアルスレッド固有の作業表のことです。

ローカル作業表用バッファのページ数を求めたら，求めた値以上の値をサーバ定義の adb_dbbuff_wrktbl_clt_blk_num オペランドに指定してください。詳細については，「7.2.2　性能に関するオペランド（set 形式）」の adb_dbbuff_wrktbl_clt_blk_num オペランドを参照してください。

また，必要に応じて次に示すオペランドおよびオプションにも指定してください。

- クライアント定義の adb_dbbuff_wrktbl_clt_blk_num オペランド
  マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。
- エクスポートオプション adb_export_wrktbl_blk_num
  マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート）』の『adbexport コマンドの指定形式』を参照してください。

ローカル作業表用バッファのページ数を求める計算式を次に示します。

計算式（単位：ページ数）

$$\text{ローカル作業表用バッファのページ数} = \mathrm{MAX}\left\{A\ ,\ B\right\}$$

変数の説明

A：SQL 文の実行時に作成される作業表の数×2

B：サーバ定義の adb_sys_uthd_num オペランドの値

SQL 文の実行時に作成される作業表の数については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』を参照して求めてください。作業表の数を求める際の注意点を次に示します。

- 作成される作業表の数は，マニュアル『HADB　AP 開発ガイド』の『作成される作業表の個数』で説明している例を参照して，SQL 文ごとに求めてください。求めた数のうち，最大の数を代入して計算してください。
- 外への参照列を含む副問合せがある場合，マニュアル『HADB　AP 開発ガイド』の『作成される作業表の個数』で説明している例を参照して，作成される作業表の数を求めてください。求めた値を S とした場合，次に示す計算式で最終的に作成される作業表の数を計算します。

計算式（単位：個数）

$$作成される作業表の数 = S \times MIN\left\{T, U\right\}$$

変数の説明

T：サーバ定義のadb_sys_uthd_num オペランドの値

U：外への参照列として参照されている列を検索する問合せ指定の副問合せ評価前の結果の行数

- ハッシュテーブルを使用した検索を実行する場合，次の計算式で求めた値を，SQL 文の実行時に作成される作業表の数に加算してください。

```
2×ハッシュテーブルを使用した検索時に作成される作業表の数＋1
```

ハッシュテーブルを使用した検索時に作成される作業表の数の求め方については，「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

SQL 文の実行時に作成される作業表の数の計算例を次に示します。

(例)

次に示す SQL 文の実行時に作成される作業表の数を求めます。

```
SELECT "C1","C2","C3" FROM "T1"
  WHERE "C1"=ANY(                                    ...1
                SELECT COUNT(DISTINCT "C3") FROM "T2"
                  WHERE "T1"."C1"="C2" GROUP BY "C1")    ...2
                  ORDER BY "C1","C2","C3"                ...3
```

[説明]

1. 限定述語のための作業表が作成されます。ただし，外への参照列（"T1"."C1"）を含む副問合せのため，作成される作業表の数は次の計算式で求めます。

$$1 \times MIN\left\{Y, Z\right\}$$

変数の説明
Y：サーバ定義のadb_sys_uthd_numオペランドの値
Z：表T1の行数
この例の場合，「外への参照列として参照されている列を
検索する問合せ指定の副問合せ評価前の結果の行数」が，
「SELECT "C1","C2","C3" FROM "T1"」の結果の行数となるため，
表T1の行数となります。

2. GROUP BY 句のソート処理で使うための作業表が作成されます。ただし，外への参照列（"T1"."C1"）を含む副問合せのため，作成される作業表の数は次の計算式で求めます。

```
1×MIN{ Y , Z }

 変数の説明
     Y：サーバ定義のadb_sys_uthd_numオペランドの値
     Z：表T1の行数
          この例の場合，「外への参照列として参照されている列を
          検索する問合せ指定の副問合せ評価前の結果の行数」が，
          「SELECT "C1","C2","C3" FROM "T1"」の結果の行数となるため，
          表T1の行数となります。
```

3. ORDER BY 句のソート処理で使う作業表が作成されます。この場合，作成される作業表は 1 つです。

上記の SQL 文の実行時に作成される作業表の数は，1～3 の合計となります。

6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数

ここでは，ハッシュテーブルを使用した検索時に作成される作業表の数の見積もり方について説明します。

次に示すどれかに該当する場合，ハッシュテーブルを使用した検索が行われます。

- 表の結合方式でハッシュジョインが適用された場合
- 副問合せの処理方式でハッシュ実行が適用された場合
- グループ化の処理方式，DISTINCT 集合関数の処理方式でグローバルハッシュグループ化が適用された場合
- SELECT DISTINCT の処理方式でハッシュ実行が適用された場合
- 集合演算の処理方式でハッシュ実行が適用された場合

サーバ定義およびクライアント定義のadb_sql_exe_hashtbl_area_size オペランドの値に従って，ハッシュテーブル領域が確保されて，ハッシュジョイン処理が行われます。このとき，確保されたハッシュテーブル領域が不足した場合，作業表が作成されます。

ハッシュテーブルを使用した検索時に作成される作業表の数は，実行する SQL 文ごとに次の計算式で求めてください。

計算式

```
ﾊｯｼｭﾃｰﾌﾞﾙを使用した検索時に作成される作業表の数＝
                    MAX( ↑((A＋32)×B×2)÷D↑ , 121 )×C
```

変数の説明

変数の説明中に出てくる列長およびデータ長については，「表 6-9　各データ型のデータ長の一覧」を参照して求めてください。

A：ハッシュテーブル最大行長（単位：バイト）

各問合せ式本体で，次の単位ごとにハッシュテーブル行長を求めてください。その中で最大の値が，ハッシュテーブル最大行長になります。

- 複数の表を＝で結合している問合せ指定

  ＝で結合している各表に対して，選択式および探索条件中に指定したすべての列について，列長の合計を求めてください。

  なお，＝の結合条件の左右に列指定を含む値式を指定している場合は，その値式の結果のデータ長も合計に加えてください。

  表ごとに求めた値のうち，最小値を除いたすべての値の合計がハッシュテーブル行長になります。

  ただし，2 表だけを＝で結合していて，その結合の左右に指定した値式の結果のデータ型またはデータ長が異なる場合は，すべての値の合計を求めてください。その際，最小値は除かないでください。

  また，アクセスパスの表示結果から，ハッシュテーブル行長を求めることができます。表の結合方式がHASH JOIN となっているツリーの問合せに対して，上記のことを実施し，ハッシュテーブル行長を求めてください。

- 限定述語の表副問合せ

  次に示す 4 つの値を合計したものが，ハッシュテーブル行長になります。

  - 副問合せの選択式の結果のデータ長の合計値
  - 副問合せに指定した集合関数の結果のデータ長の合計値
  - 探索条件中の外への参照列を含む述語に指定したすべての列の列長の合計値
  - 限定述語またはIN 述語の右側の表副問合せの選択式の結果と，その述語の左側に指定している列のデータ型またはデータ長が異なる場合に，その述語の左側に指定している列の列長

  なお，アクセスパスの表示結果から，ハッシュテーブル行長を求めることができます。副問合せの処理方式がSUBQUERY HASH となっているツリーの問合せに対して，上記の 4 つの値を合計したものが，ハッシュテーブル行長になります。

- IN 述語中の表副問合せ

  ハッシュテーブル行長の求め方は，『限定述語の表副問合せ』を参照してください。

- 限定述語の表副問合せおよびIN 述語中の表副問合せ以外で，探索条件中に＝で外への参照列を指定している副問合せ

  ハッシュテーブル行長の求め方は，『限定述語の表副問合せ』を参照してください。

- GROUP BY 句を含む問合せ指定

  グループ化列の列長，および集合関数の結果のデータ長をそれぞれ求めてください。求めた値を合計したものが，ハッシュテーブル行長になります。

  なお，アクセスパスの表示結果から，ハッシュテーブル行長を求めることができます。グループ化の処理方式がGLOBAL HASH GROUPING となっているツリーの問合せに対して，グループ化列の列長，および集合関数の結果のデータ長をそれぞれ求めてください。求めた値を合計したものが，ハッシュテーブル行長になります。

- DISTINCT 集合関数を含む問合せ指定

  グループ化列の列長，DISTINCT 集合関数の引数に指定した値式の結果のデータ長，および集合関数（DISTINCT 集合関数を除く）の結果のデータ長をそれぞれ求めてください。求めた値を合計したものが，ハッシュテーブル行長になります。

- SELECT DISTINCT が指定された問合せ指定

  SELECT DISTINCT の選択式のデータ長の合計が，ハッシュテーブル行長になります。

  なお，アクセスパスの表示結果から，ハッシュテーブル行長を求めることができます。SELECT DISTINCT の処理方式がハッシュ実行となっているツリーの問合せに対して，SELECT DISTINCT の選択式のデータ長をそれぞれ求めてください。求めた値を合計したものが，ハッシュテーブル行長になります。

- UNION，またはUNION DISTINCT が指定された問合せ式本体

  問合せ式本体の結果導出される表の列の列長を求めてください。求めた値を合計したものが，ハッシュテーブル行長になります。

  なお，アクセスパスの表示結果から，ハッシュテーブル行長を求めることができます。集合演算の処理方式がハッシュ実行となっているツリーの問合せに対して，問合せ式本体の結果導出される表の列の列長をそれぞれ求めてください。求めた値を合計したものが，ハッシュテーブル行長になります。

B：中間検索結果最大件数

次の計算式から求めてください。

```
MAX（A , B）

    A：一時中間検索結果の最大件数
    B：2,048×（ｸﾗｲｱﾝﾄ定義adb_sql_exe_max_rthd_numｵﾍﾟﾗﾝﾄﾞの値）
       ×（ｸﾗｲｱﾝﾄ定義adb_sql_exe_max_rthd_numｵﾍﾟﾗﾝﾄﾞの値）
```

各問合せ式本体で，次の単位ごとに一時中間検索結果の最大件数を求めてください。その中での最大件数が，一時中間検索結果の最大件数になります。

- 複数の表を＝で結合している問合せ指定

  ＝の結合それぞれに対して，結合の左右それぞれの検索結果件数のうち，小さい方の値を求めます。求めた結合ごとの値のうち，最大の値が中間検索結果件数になります。

  なお，アクセスパスの表示結果から，中間検索結果件数を求めることができます。表の結合方式がHASH JOIN となっているツリーの問合せに対して，外表の結果件数を求めてください。求めた件数のうち，最大の件数が中間検索結果件数になります。

- 限定述語の表副問合せ

  副問合せの結果の行数が中間検索結果件数になります。

  なお，アクセスパスの表示結果から，中間検索結果件数を求めることができます。副問合せの処理方式がSUBQUERY HASH となっているツリーの問合せに対して，副問合せの結果件数を求めてください。求めた件数のうち，最大の件数が中間検索結果件数になります。

- IN 述語中の表副問合せ

  中間検索結果件数の求め方は，『限定述語の表副問合せ』を参照してください。

- 限定述語の表副問合せおよびIN 述語中の表副問合せ以外で，探索条件中に＝で外への参照列を指定している副問合せ

  中間検索結果件数の求め方は，『限定述語の表副問合せ』を参照してください。

- GROUP BY 句を含む問合せ指定

  グループ化を適用したあとの結果件数が，中間検索結果件数になります。

  なお，アクセスパスの表示結果から，中間検索結果件数を求めることができます。グループ化の処理方式がGLOBAL HASH GROUPING となっているツリーの問合せに対して，グループ化を適用したあとの結果件数を求めてください。求めた件数のうち，最大の件数が中間検索結果件数になります。

- DISTINCT 集合関数を含む問合せ指定

  重複排除後の結果件数が中間検索結果件数になります。問合せ指定にGROUP BY 句を指定する場合は，DISTINCT 集合関数の引数もグループ化列と見なし，そのグループ化を適用したあとの結果件数を目安にしてください。

- SELECT DISTINCT が指定された問合せ指定

  重複排除後の結果件数が中間検索結果件数になります。

- UNION，またはUNION DISTINCT が指定された問合せ式本体

  集合演算後の結果件数が中間検索結果件数になります。

C：ハッシュテーブル数

次の計算式から求めてください。

```
ﾊｯｼｭﾃｰﾌﾞﾙ数＝
    {(＝で結合している表数)－(＝で結合している問合せ指定数)}
    ＋(限定述語の数)
    ＋(右側に副問合せを指定したIN述語の数)
    ＋(限定述語および右側に副問合せを指定したIN述語の副問合せ以外で，
       探索条件中に＝で外への参照列を指定している副問合せ数)
    ＋(GROUP BY句の指定数※)
    ＋(DISTINCT集合関数の指定数)
    ＋(FULL OUTER JOINの指定数)
    ＋(SELECT DISTINCTの指定数)
    ＋(UNION，またはUNION DISTINCTの指定数)
```

注※

SQL パラレル実行機能が適用される検索系 SQL の場合は，*GROUP BY 句の指定数*×2 としてください。

ハッシュテーブル数は，アクセスパスの表示結果からも求めることができます。アクセスパスを表示できる場合は，アクセスパスの表示結果と次の計算式からハッシュテーブル数を求めてください。

```
ﾊｯｼｭﾃｰﾌﾞﾙ数＝
    HASH JOIN（表の結合方式）の表示数※
    ＋SUBQUERY HASH（副問合せの処理方式）の表示数※
    ＋GLOBAL HASH GROUPING（グループ化の処理方式）の表示数※
    ＋GLOBAL HASH UNIQUE（重複排除の処理方式）の表示数※
```

注※

アクセスパスの表示結果の<<Tree View>>に表示される情報から求めてください。例えば，<<Tree View>>中にHASH JOIN が 2 回表示されている場合，「HASH JOIN（表の結合方式）の表示数」は 2 と計算してください。

D：ハッシュテーブル領域

次の計算式から求めてください。

```
ﾊｯｼｭﾃｰﾌﾞﾙ領域＝
    adb_sql_exe_hashtbl_area_sizeオペランドの指定値×1,024×1,024
```

クライアント定義のadb_sql_exe_hashtbl_area_size オペランドの指定値が，サーバ定義の adb_sql_exe_hashtbl_area_size オペランドの指定値よりも小さい場合は，クライアント定義の指定値を代入してください。それ以外は，サーバ定義の指定値を代入してください。

なお，adb_sql_exe_hashtbl_area_size オペランドの指定値が0 の場合，ハッシュテーブルを使用した検索は適用されません。

7 サーバ定義の設計

この章では，サーバ定義のオペランドの指定形式，内容，および文法規則について説明します。

7.1　サーバ定義のオペランドの指定形式

この節では，サーバ定義の各オペランドの指定形式について説明します。

サーバ定義の各オペランドの内容については，「7.2　サーバ定義のオペランドの内容」を参照してください。

なお，サーバ定義の作成方法および変更方法については，「8.5　サーバ定義の作成および変更方法」を参照してください。

また，サーバ定義の記述形式には，set 形式とコマンド形式があります。サーバ定義の各オペランドを指定する際，サーバ定義の記述形式についても確認してください。set 形式およびコマンド形式については，「7.3.1　サーバ定義の文法規則の詳細」の「(2)　記述形式」を参照してください。

> **メモ**
>
> クライアント定義のオペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』を参照してください。

サーバ定義の各オペランドの指定形式を次の表に示します。

なお，網掛けが付いているオペランド（色が付いているオペランド）は，指定必須のオペランドです。また，表中の「番号」は，各オペランドに付けた番号と対応しています。

番号	オペランドおよびオプションの指定形式	オペランドの分類
1	**set adb_db_path** = ***DBディレクトリ名***	システム構成
2	〔set adb_rpc_port = *HADB サーバのポート番号*〕	
3	〔set adb_sys_max_users = *最大同時接続数*〕	
4	〔set adb_core_path = *障害情報（コアファイル）の出力先ディレクトリ名*〕	
5	〔set adb_blk_path_wrk = *作業表用 DB エリアファイルのブロックスペシャルファイルのパス名*〕	
6	〔set adb_dbarea_wrk_page_size = *作業表用 DB エリアのページサイズ*〕	
7	〔set adb_sys_rthd_num = *処理リアルスレッド数*〕	性能
8	〔set adb_sys_uthd_num = *1 リアルスレッドで生成できる擬似スレッド数*〕	
9	〔set adb_sys_memory_limit = *HADB サーバが使用するメモリの最大サイズ*〕	
10	〔set adb_sql_exe_max_rthd_num = *最大 SQL 処理リアルスレッド数*〕	
11	〔set adb_sql_exe_hashgrp_area_size = *ハッシュグループ化領域サイズ*〕	
12	〔set adb_sql_exe_hashtbl_area_size = *ハッシュテーブル領域サイズ*〕	
13	〔set adb_sql_exe_hashflt_area_size = *ハッシュフィルタ領域サイズ*〕	
14	〔set adb_sql_opt_hashjoin_div = {Y｜N}〕	

番号	オペランドおよびオプションの指定形式	オペランドの分類
15	〔set adb_sql_opt_drvtbl_grping_prior = {LOCAL｜GLOBAL}〕	
16	〔set adb_sys_rthd_area_max = *確保できるリアルスレッド固有メモリの最大サイズ*〕	
17	〔set adb_sys_proc_area_max = *確保できるプロセス共通メモリの最大サイズ*〕	
18	〔set adb_sys_shm_huge_page_size = *HugePages の 1 ページ当たりのページサイズの倍数*〕	
19	〔set adb_dbbuff_wrktbl_glb_blk_num = *グローバル作業表用のグローバルバッファのページ数*〕	
20	〔set adb_dbbuff_wrktbl_clt_blk_num = *ローカル作業表用バッファのページ数*〕	
21	〔set adb_sql_tbldef_cache_size = *表定義プールサイズ*〕	
22	〔set adb_log_usrfile_num = *ユーザログファイル数*〕	システムログ
23	〔set adb_log_usrfile_size = *ユーザログファイルの初期化容量*〔,*ユーザログファイルの容量の縮小契機*〕〕	
24	〔set adb_log_usrbuf_num = *ユーザログバッファ面数*〕	
25	〔set adb_log_rec_msg_interval =*データベースの回復処理の経過時間を示すメッセージの出力間隔*〕	
26	〔set adb_sys_max_users_wrn_pnt = *最大同時接続数に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕〕	状態監視
27	〔set adb_sys_memory_limit_wrn_pnt = *HADB サーバのメモリ使用量に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕〕	
28	〔set adb_dbarea_wrn_pnt_use_rate = *データ用 DB エリアの使用率*〔,*データ用 DB エリアの使用率*〔,*データ用 DB エリアの使用率*〕〕〕	
29	〔set adb_dbarea_wrn_pnt_except_name = *データ用 DB エリア名*〔,*データ用 DB エリア名*〕…〕	
30	〔set adb_rpc_wait_time = *応答待ち時間の上限*〕	
31	〔set adb_rpc_tcp_keepalive_time = *KeepAlive プローブ開始までの無通信時間*〕	
32	〔set adb_sql_prep_delrsvd_words = *削除する予約語*〔, *削除する予約語*〕…〕	SQL 文
33	〔set adb_sys_trn_iso_lv = {READ_COMMITTED｜REPEATABLE_READ}〕	
34	〔set adb_sql_text_out = {Y｜N}〕	
35	〔set adb_sql_trc_out = {Y｜N}〕	
36	〔set adb_sql_trc_param = {Y｜N}〕	
37	〔set adb_sql_trc_accesspath = {Y｜N}〕	
38	〔set adb_sql_trc_level = {SQL｜CALL}〕	
39	〔set adb_sql_trc_txtfile_size = *SQL トレースファイルの最大容量*〕	
40	〔set adb_sql_order_mode = {BYTE｜ISO}〕	
41	〔set adb_sql_prep_dec_div_rs_prior = {INTEGRAL PART｜FRACTIONAL_PART}〕	

番号	オペランドおよびオプションの指定形式	オペランドの分類
42	〔set adb_sql_default_dbarea_shared = *表またはインデクスを格納するデフォルトのデータ用 DB エリア名*〕	
43	〔set adb_sql_rngidx_preread = {ALL | NO | *レンジインデクス名* 〔, *レンジインデクス名*〕…} 〕	レンジインデクス
44	〔set adb_sta_log_max_size = *統計ログファイルの最大容量*〕	統計情報
45	〔set adb_sta_log_path = *統計ログファイルの出力先ディレクトリ名*〕	
46	〔set adb_sta_log_size_unit = {G | M} 〕	
47	〔set adb_syndict_storage_path = *同義語辞書ファイルの格納ディレクトリ*〕	同義語検索
48	〔set adb_syndict_node_storage_path = *マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ*〕	
49	〔set adb_audit_log_path = *監査証跡の出力先ディレクトリ*〕	監査証跡機能
50	〔set adb_audit_log_max_size = *監査証跡ファイルの最大容量*〕	
51	〔set adb_audit_log_max_num = *監査証跡ファイルの最大世代数*〕	
52	〔set adb_sys_multi_node_info = *ノードのホスト名* 〔:*ノードのポート番号*〕, *ノードのホスト名* 〔:*ノードのポート番号*〕 〔,*ノードのホスト名* 〔:*ノードのポート番号*〕 〕…〕	マルチノード機能
53	〔set adb_sys_multi_worker_node_info = *ワーカーノードのホスト名* 〔:*ワーカーノードのポート番号*〕 〕	
54	〔set adb_sys_multi_max_worker_nodes = *ワーカーノードの最大数*〕	
55	〔set adb_sys_node_start_wait_time = *他ノードの起動完了待ち時間の上限*〕	
56	〔set adb_sys_max_parallel_exec_num = *SQL パラレル実行機能を適用する検索系 SQL の最大同時実行数*〕	
57	{ { 〔adbbuff -g *グローバルバッファ名* {-n *DB エリア名* 〔, *DB エリア名*〕… | -o } 〔-p *グローバルバッファのページ数*〕 〔-a *レンジインデクス専用のグローバルバッファのページ数*〕 〔-v *テーブルスキャンバッファに使用するメモリサイズ* 〔, *リアルスレッドごとに使用するメモリサイズの上限値*〕 〕 〔-k *セグメントを一括読み込みするためのテーブルスキャンバッファのバッファ面数*〕 〕 } }	グローバルバッファ
58	{ { 〔adbcltgrp -g *クライアントグループ名* 〔-m *設定したクライアントグループの最大同時接続数*〕 〔-u *設定したクライアントグループの接続保証数*〕 〔-r *設定したクライアントグループで使用できる処理リアルスレッドの最大数*〕 〔-e *設定したクライアントグループで使用できる処理リアルスレッドの保証数*〕	クライアントグループ機能

番号	オペランドおよびオプションの指定形式	オペランドの分類
	〔-w *設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕〕 }}	
59	〔adbcltgrp -g command 〔-m *設定したコマンドグループの最大同時接続数*〕 〔-u *設定したコマンドグループの接続保証数*〕 〔-r *設定したコマンドグループで使用できる処理リアルスレッドの最大数*〕 〔-e *設定したコマンドグループで使用できる処理リアルスレッドの保証数*〕 〔-w *設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕〕	

7.2　サーバ定義のオペランドの内容

この節では，サーバ定義の各オペランドの内容について説明します。

サーバ定義の各オペランドの指定形式については，「7.1　サーバ定義のオペランドの指定形式」を参照してください。

なお，サーバ定義の作成方法および変更方法については，「8.5　サーバ定義の作成および変更方法」を参照してください。

また，サーバ定義の記述形式には，set 形式とコマンド形式があります。サーバ定義の各オペランドを指定する際，サーバ定義の記述形式についても確認してください。set 形式およびコマンド形式については，「7.3.1　サーバ定義の文法規則の詳細」の「(2)　記述形式」を参照してください。

> **メモ**
>
> クライアント定義のオペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』を参照してください。

サーバ定義の各オペランドの内容については，次項以降を参照してください。

7.2.1　システム構成に関するオペランド（set 形式）

[1]　adb_db_path = *DB ディレクトリ名*

～〈パス名〉((2～510 バイト))

DB ディレクトリ名を絶対パス名で指定します。このオペランドは必ず指定してください。ここで指定した DB ディレクトリ名と，adbinit コマンドで指定する DB ディレクトリ名は必ず一致させてください。

このオペランドに指定するパス名の規則を次に示します。

- ルートディレクトリは指定できません。
- サーバディレクトリ（サブディレクトリも含む）とは異なるディレクトリを指定してください。
- 半角の英数字，アンダスコア（_），ハイフン（-），シャープ（#），アットマーク（@），スラッシュ（/）だけで構成されるパス名を指定してください。

[2]　adb_rpc_port = *HADB サーバのポート番号*

～〈整数〉((5,001～65,535))《23,650》

HADB サーバと HADB クライアント間の通信で使用する，HADB サーバのポート番号を指定します。

［マルチノード機能］

マルチノード機能を使用する場合は，プライマリノードおよびセカンダリノードの HADB サーバでこのオペランドの指定値を同じにしてください。ワーカーノードの HADB サーバでは，このオペランドの指定値を同じにする必要はありません。

[3] adb_sys_max_users = *最大同時接続数*

～〈整数〉((1～1,024))《10》

HADB サーバに接続する HADB クライアントの最大同時接続数を指定します。

クライアントグループ機能を適用する場合は，各グループの接続保証数の合計値よりも，このオペランドの指定値を大きくしてください。各グループの接続保証数の合計値の方が大きい場合は，HADB サーバの開始時にエラーとなります。

各グループの接続保証数については，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」のadbcltgrp オペランドの-u オプションの次の説明を参照してください。

- 「-u *設定したクライアントグループの接続保証数*」
- 「-u *設定したコマンドグループの接続保証数*」

［マルチノード機能］

- マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。
- このオペランドに指定された値が，マルチノード構成の HADB サーバに対する最大同時接続数になります。

[4] adb_core_path = *障害情報（コアファイル）の出力先ディレクトリ名*

～〈パス名〉((2～510 バイト))

障害情報（コアファイル）の出力先ディレクトリ名を絶対パスで指定します。このオペランドで指定したディレクトリ下に，障害情報（コアファイル）が出力されます。

通常，このオペランドは指定不要です。このオペランドを省略した場合，障害情報（コアファイル）の出力先ディレクトリは$DBDIR/SPOOL になります。

$DBDIR/SPOOL ディレクトリ下以外に，障害情報（コアファイル）を出力したい場合，このオペランドを指定してください。ただし，サーバディレクトリ下のconf ディレクトリ（$ADBDIR/conf），およびそのサブディレクトリは指定しないでください。

重要

- このオペランドを指定する場合，HADB 管理者にアクセス権限（読み取り権限，書き込み権限，および実行権限）が付与されたディレクトリを指定するようにしてください。
- コアファイルの出力が systemd-coredump によって制御されている場合，コアファイルは/var/lib/systemd/coredump ディレクトリ下に出力されます（デフォルトの設定ではLZ4 形式で圧縮されて出力されます）。この場合，adb_core_path オペランドに指定したディレクトリは，トラブルシュート情報の収集時にコアファイル展開用の一時ディレクトリとして使用されます。adb_core_path オペランドを省略したときは，$DBDIR/SPOOL ディレクトリがコアファイル展開用の一時ディレクトリになります。

コアファイルの出力が systemd-coredump によって制御されているかどうかを確認する方法については，「8.2.1　インストール前に実施する作業」の「(9)　設定ファイルの変更および作成（コアファイルの出力が systemd-coredump で制御されている場合）」を参照してください。

なお，コアファイルの出力が systemd-coredump によって制御されている場合であっても，adbinfoget -g コマンドの実行によって出力されたコアファイルは，このオペランドに指定したディレクトリ下または$DBDIR/SPOOL ディレクトリ下に出力されます。このオペランドの指定とコアファイルの出力先の関係を次の表に示します。

表 7-1　adb_core_path オペランドの指定とコアファイルの出力先の関係

コアファイルの出力制御	コアファイルの出力契機	adb_core_path オペランドの指定	コアファイルの出力先
カーネル単体	プロセスダウンなど	指定あり	adb_core_path オペランドに指定したディレクトリ下
		指定なし	$DBDIR/SPOOL ディレクトリ下
	adbinfoget -g コマンドの実行	指定あり	adb_core_path オペランドに指定したディレクトリ下
		指定なし	$DBDIR/SPOOL ディレクトリ下
systemd-coredump	プロセスダウンなど	指定あり	/var/lib/systemd/coredump ディレクトリ下
		指定なし	
	adbinfoget -g コマンドの実行	指定あり	adb_core_path オペランドに指定したディレクトリ下
		指定なし	$DBDIR/SPOOL ディレクトリ下

[5]　adb_blk_path_wrk = *作業表用 DB エリアファイルのブロックスペシャルファイルのパス名*

～〈パス名〉((2～255 バイト))

HADB サーバの開始時に，作業表用 DB エリアファイルに割り当てるブロックスペシャルファイルのパス名を指定します。

このオペランドの指定値，および既存の作業表用 DB エリアファイル（ADBWRK）の状態に従って，作業表用 DB エリアファイルを変更します。

既存の作業表用 DB エリアファイルを変更する場合は，変更後にファイルを初期化します。既存の作業表用 DB エリアファイルを変更しない場合は，ファイルの内容を参照して初期化していなければ初期化します。詳細を次の表に示します。

表 7-2　adb_blk_path_wrk オペランドと作業表用 DB エリアファイルの関係

サーバ定義 adb_blk_path_wrk オペランドの指定	既存の作業表用 DB エリアファイル		
	なし	レギュラーファイル	ブロックスペシャルファイル
指定を省略	レギュラーファイルを作成します。	変更しません（既存のレギュラーファイルのままとなります）。	変更しません（既存のブロックスペシャルファイルのままとなります）。
パス名を指定	adb_blk_path_wrk オペランドに指定したブロックスペシャルファイルを使用します。		adb_blk_path_wrk オペランドに指定したブロックスペシャルファイルを使用します。ただし，既存と同じブロックスペシャルファイル名を指定した場合は変更しません。

［マルチノード機能］

マルチノード機能を使用する場合で，作業表用 DB エリアファイルがノード固有となっていないときは，次のどちらかを実施してください。

- このオペランドに，ほかのノードと共有しないブロックスペシャルファイルを指定する
- 既存の作業表用 DB エリアファイルを削除して，このオペランドを省略する

[6]　adb_dbarea_wrk_page_size = *作業表用 DB エリアのページサイズ*

～〈整数〉((32～32,768))（単位：キロバイト）

作業表用 DB エリアのページサイズを，adbinit コマンド実行時に指定したページサイズから，任意のページサイズに切り替えたい場合に，切り替え後のページサイズを指定します。このオペランドに 32 で割り切れない値を指定した場合，自動的に 32 キロバイト単位に切り上げられます。

通常，このオペランドは指定不要です。このオペランドを省略した場合は，adbinit コマンド実行時に指定したページサイズが適用されます。

ページサイズを大きくすると，メモリ不足の原因となる可能性があります。そのため，このオペランドでページサイズを大きくする場合は，メモリ所要量を見直してください。メモリ所要量の見積もりについては，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」および「6.3.4　通常運用時のメモリ所要量の求め方」を参照してください。

稼働中の HADB サーバで適用されているページサイズは，次のコマンドで調べることができます。

- adbls -d gbuf コマンド

  コマンドの詳細については，マニュアル『HADB　コマンドリファレンス』の『adbls -d gbuf（グローバルバッファの情報表示)』を参照してください。

- adbls -d lbuf コマンド

  コマンドの詳細については，マニュアル『HADB　コマンドリファレンス』の『adbls -d lbuf（ローカル作業表用バッファの情報表示)』を参照してください。

また，adbinit コマンド実行時に指定した作業表用 DB エリアのページサイズは，ディクショナリ表の検索で調べることができます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の

検索」の「(37)　adbinit コマンド実行時に指定した作業表用 DB エリアのページサイズを調べる方法」を参照してください。

［マルチノード機能］

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。

7.2.2　性能に関するオペランド（set 形式）

［7］ adb_sys_rthd_num = *処理リアルスレッド数*

～〈整数〉((1～4,096))《40》

処理リアルスレッド数を指定します。

処理リアルスレッドは，SQL 文，および次に示す対象のコマンドを実行したときに，HADB サーバが使用するリアルスレッドのことです。

■対象のコマンド

- adbimport コマンド
- adbgetcst コマンド
- adbidxrebuild コマンド
- adbexport コマンド
- adbmergechunk コマンド
- adbarchivechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド

このオペランドには次に示す計算式から求められる値を指定してください。クライアントグループ機能を適用する場合も同様です。

計算式

$A \times B \times C$

変数の説明

A：同時に SQL 文を実行するコネクション数

B：adb_sql_exe_max_rthd_num オペランドの値

C：1 コネクション内で同時に実行するSELECT 文の最大数

なお，対象のコマンドを実行する場合で，かつクライアントグループ機能を適用しない場合は，コマンドを実行するために*最低限必要な処理リアルスレッド数*以上の値を，このオペランドに指定してください。*最低限必要な処理リアルスレッド数*以上の値を指定しないと，対象のコマンドを実行してもエラーとなるおそれがあります。各コマンドを実行するために*最低限必要な処理リアルスレッド数*は，「6.23.2 コマンドの実行時に使用する処理リアルスレッド数に関する考慮点」の「(2)　コマンドの実行時に使

用する処理リアルスレッド数を指定するオペランドとコマンドオプション」の「表 6-28　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧」を参照してください。

また，クライアントグループ機能を適用する場合は，各グループで使用できる処理リアルスレッドの保証数の合計値よりも，このオペランドの指定値を大きくしてください。各グループで使用できる処理リアルスレッドの保証数の合計値の方が大きい場合は，HADB サーバの開始時にエラーとなります。

各グループで使用できる処理リアルスレッドの保証数については，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」のadbcltgrp オペランドの-e オプションの次の説明を参照してください。

- 「-e *設定したクライアントグループで使用できる処理リアルスレッドの保証数*」
- 「-e *設定したコマンドグループで使用できる処理リアルスレッドの保証数*」

> **重要**
>
> このオペランドは，メモリ所要量への影響が大きいオペランドです。このオペランドの指定値を変更した場合は，必ずメモリ所要量の見積もりを行ってください。メモリ所要量の見積もりについては，「6.3　HADB サーバのメモリ所要量の見積もり」を参照してください。

[8]　adb_sys_uthd_num = *1 リアルスレッドで生成できる擬似スレッド数*

～〈整数〉((0～4,096))《128》

1 リアルスレッドで生成できる擬似スレッド数を指定します。

このオペランドには，基本的にデフォルト値を指定してください。

> **重要**
>
> このオペランドに0 を指定すると，非順序実行方式を適用しないで SQL 文を実行します。また，コネクションスレッドだけで SQL 文を実行します。

[9]　adb_sys_memory_limit = *HADB サーバが使用するメモリの最大サイズ*

～〈単位付き整数〉((5,120MB～1,099,511,627,776MB))

HADB サーバが使用するメモリ（共有メモリおよびプロセスメモリ）の最大サイズを単位付き整数で指定します。

> **重要**
>
> バージョン 03-01 以降の HADB サーバでは，このオペランドを指定してください。このオペランドと，サーバ定義adb_sys_proc_area_max オペランドおよびadb_sys_rthd_area_max オペランドは，対の関係になります。どちらか片方しか指定できません。そのため，このオペランドを指定する場合は，サーバ定義adb_sys_proc_area_max オペランドおよびadb_sys_rthd_area_max オペランドは指定しないでください。

■adb_sys_memory_limit オペランドに指定する値の目安

このオペランドには，「6.3　HADB サーバのメモリ所要量の見積もり」の『HADB サーバが使用するメモリの最大サイズ』で求めた値以上の値を指定してください。または，HADB サーバで使用できるメモリの最大サイズを指定してください。

ただし，サーバ定義adb_sys_shm_huge_page_size オペランドに0 以外を指定した場合（HADB サーバが使用する共有メモリにHugePages を適用する場合）は，サーバ定義adb_sys_memory_limit オペランドには次の計算式を満たす値を指定してください。次の計算式を満たさない値を指定すると，HADB サーバの開始時にエラーとなります。

サーバ定義adb_sys_memory_limitオペランドの指定値
　≦ *カーネルパラメタvm.nr_hugepages* × *HugePagesの1ページ当たりのページサイズ*

*HugePages の 1 ページ当たりのページサイズ*の確認方法については，サーバ定義adb_sys_shm_huge_page_size オペランドを参照してください。

■adb_sys_memory_limit オペランドに指定する単位の規則

- このオペランドに指定できる単位は，MB（メガバイト），GB（ギガバイト）およびTB（テラバイト）です。

  単位を指定した場合の指定例

```
set adb_sys_memory_limit = 196608MB
```

```
set adb_sys_memory_limit = 128GB
```

```
set adb_sys_memory_limit = 64TB
```

- 単位を省略した場合（整数だけを指定した場合）は，MB（メガバイト）が仮定されます。

  単位を省略した場合の指定例

  この場合は，196608MB が指定されたと仮定されます。

```
set adb_sys_memory_limit = 196608
```

- 各単位の指定範囲は次のとおりです。

  MB（メガバイト）の場合

  5,120MB～1,099,511,627,776MB

  GB（ギガバイト）の場合

  5GB～1,073,741,824GB

  TB（テラバイト）の場合

  1TB～1,048,576TB

■adb_sys_memory_limit オペランドを指定しない場合に仮定される値

このオペランドを指定しない場合，サーバ定義adb_sys_proc_area_max オペランドの指定値に従って，プロセス共通メモリの最大サイズが決まります。また，サーバ定義adb_sys_rthd_area_max オペランドの指定値に従って，1 リアルスレッド当たりのリアルスレッド固有メモリの最大サイズが決まります。

このオペランドを指定すると，次の表に従って，HADB サーバが使用するメモリ（共有メモリおよびプロセスメモリ）の最大使用量を制御します。

表 7-3　adb_sys_memory_limit オペランドが制御するメモリ領域の一覧

項番	HADB サーバが使用するメモリ領域			関連するサーバ定義のオペランド
1	共有メモリ	共有メモリ管理		－
2		プロセス共通メモリ※1	HADB サーバ制御用メモリ	－
3			その他（SQL 文実行用メモリ，コマンド実行用メモリなど）	－
4		リアルスレッド固有メモリ※2	ローカル作業表用バッファ	adb_dbbuff_wrktbl_clt_blk_num
5			ハッシュグループ化領域	adb_sql_exe_hashgrp_area_size
6			ハッシュテーブル領域	adb_sql_exe_hashtbl_area_size
7			ハッシュフィルタ領域	adb_sql_exe_hashflt_area_size
8			その他（SQL 文実行用ワークメモリ，コマンド実行用メモリなど）	－
9		グローバルバッファページ	グローバルバッファ	adbbuff -p
10			レンジインデクス用バッファ	adbbuff -a
11			テーブルスキャン用バッファ	• adbbuff -v • adbbuff -k
12			グローバル作業表用のグローバルバッファ	adb_dbbuff_wrktbl_glb_blk_num
13	プロセスメモリ	ヒープメモリ		－

（凡例）

－：関連するサーバ定義のオペランドはありません。

注※1

サーバ定義adb_sys_proc_area_max オペランドが関連します。

注※2

サーバ定義adb_sys_rthd_area_max オペランドが関連します。

[10] adb_sql_exe_max_rthd_num = *最大 SQL 処理リアルスレッド数*

～〈整数〉((0～4,096))《4》

SQL 処理リアルスレッド数の最大値を指定します。

SQL 処理リアルスレッドは，SQL 文実行時に使用する処理リアルスレッドです。

このオペランドには，HADB サーバをインストールしているマシンの CPU のコア数以下の値を指定してください。指定値を決める際には，次に示す『■adb_sql_exe_max_rthd_num オペランドを指定する際の考慮点』を確認してください。また，同時に実行する SQL 文の数を考慮してください。

> **重要**
>
> このオペランドに指定した値を変更した場合，adb_sys_rthd_num オペランドに指定した値も見直してください。このオペランドにadb_sys_rthd_num オペランドの値より大きな値を指定した場合，adb_sys_rthd_num オペランドの値が仮定されます。

■adb_sql_exe_max_rthd_num オペランドを指定する際の考慮点

次に示す計算式を満たす場合，このオペランドで指定した処理リアルスレッド数を SQL 文実行時に使用できないことがあります。その場合，SQL 文は，このオペランドで指定した処理リアルスレッド数が確保できるようになるまで待ち状態となります。クライアントグループ機能を適用しているかどうかによって，計算式が異なります。

計算式（クライアントグループ機能を適用していない場合）

```
A ＜ B×C×D
```

計算式（クライアントグループ機能を適用している場合）

```
E ＜ F×C×D

  注
    設定したクライアントグループおよびコマンドグループの中で，
    計算式を満たすグループが対象となります。
```

変数の説明

A：adb_sys_rthd_num オペランドの値

B：adb_sys_max_users オペランドの値

C：adb_sql_exe_max_rthd_num オペランドの値

D：1 コネクション内で同時に実行するSELECT 文の最大数（0 の場合は 1 を代入する）

E：設定したクライアントグループまたはコマンドグループで使用できる処理リアルスレッドの保証数

F：設定したクライアントグループまたはコマンドグループの最大同時接続数

このオペランドと，クライアント定義adb_sql_exe_max_rthd_num オペランドを両方指定した場合は，クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値が優先されます。ただし，クライアント定義adb_sql_exe_max_rthd_num オペランドの指定値が，このオペランドの指定値よりも大きい場合は優先されません（このオペランドの指定値が仮定されます）。

クライアント定義adb_sql_exe_max_rthd_num オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

また，設定したクライアントグループおよびコマンドグループの中で，使用できる処理リアルスレッドの最大数よりも，adb_sql_exe_max_rthd_num オペランドの値が大きいグループがある場合，そのグループのadb_sql_exe_max_rthd_num オペランドの値には，使用できる処理リアルスレッドの最大数が仮定されます。

使用できる処理リアルスレッドの最大数については，「7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）」のadbcltgrp オペランドの-r オプションを参照してください。

このオペランドの指定値と，SQL 文の実行時に使用する SQL 処理リアルスレッド数の関係を次の表に示します。

表 7-4　SQL 文の実行時に使用する SQL 処理リアルスレッド数

項番	実行する SQL 文		adb_sql_exe_max_rthd_num オペランドの指定値		
			0 を指定	1 を指定	2 以上を指定
1	定義系 SQL	ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）※1	使用する SQL 処理リアルスレッド数は 0 になります。	使用する SQL 処理リアルスレッド数は 0 になります。	使用する SQL 処理リアルスレッド数は，adb_sql_exe_max_rthd_num オペランドの指定値となります。
2		DROP INDEX 文※1			
3		DROP SCHEMA 文※1			
4		DROP TABLE 文※1			
5		DROP USER 文※1			
6		REVOKE 文のSCHEMA（スキーマ操作権限の取り消し）※1			
7	操作系 SQL	PURGE CHUNK 文※1			使用する SQL 処理リアルスレッド数は，計算式※2 で求めた値となります。
8		TRUNCATE TABLE 文※1			
9		SELECT 文※3		使用する SQL 処理リアルスレッド数は 1 になります。	使用する SQL 処理リアルスレッド数は，adb_sql_exe_max_rthd_num オペランドの指定値となります。
10	上記以外の SQL 文※4			使用する SQL 処理リアルスレッド数は 0 になります。	使用する SQL 処理リアルスレッド数は 0 になります。

注※1

このオペランドに1 以下を指定した場合は，コネクションスレッドだけで SQL 文を実行します。このオペランドに2 以上を指定した場合は，セグメントの解放処理を行うときに，表で示す SQL 処理リアルスレッド数を使用して SQL 文を実行します。

注※2

使用する SQL 処理リアルスレッド数は，次の計算式で求めた値になります。

```
MIN（A , B）

  A：adb_sql_exe_max_rthd_numオペランドの指定値
  B：MAX（C , D）

  C：処理対象表を格納するデータ用DBエリアの
     データ用DBエリアファイル数
  D：処理対象表に定義したインデクスを格納する
     データ用DBエリアのデータ用DBエリアファイル数※

  注※
    インデクスが複数定義されている場合は，
    各インデクスを格納するデータ用DBエリアのうち，
    データ用DBエリアファイル数が最も大きい値を代入する
```

注※3

このオペランドに0 を指定した場合は，非順序実行方式を適用しないで SQL 文を実行します。このオペランドに1 以上を指定した場合は，表で示す SQL 処理リアルスレッド数を使用して，非順序実行方式を適用して SQL 文を実行します。

注※4

このオペランドの指定値に関係なく，コネクションスレッドだけで SQL 文を実行します。

重要

サーバ定義adb_sys_uthd_num オペランドに0 を指定した場合は，このオペランドに0 を指定した場合と同じように SQL 文が実行されます。

[11] adb_sql_exe_hashgrp_area_size = *ハッシュグループ化領域サイズ*

～〈整数〉((0，4～1,000,000))《4,800》（単位：キロバイト）

ハッシュグループ化領域をキロバイト単位で指定します。

ローカルハッシュグループ化処理（GROUP BY 句によるグループ分けにハッシングが使用される）が適用される SQL 文を実行すると，SQL 処理リアルスレッドごとに，リアルスレッド固有メモリに対して，指定した値でハッシュグループ化領域が確保されます。

0 を指定すると，ローカルハッシュグループ化処理は実行されません。また，4 より小さい値を指定した場合には，4 が指定されたと仮定されます。

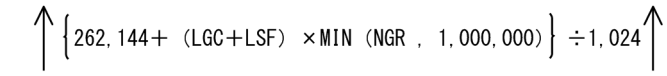
ハッシュグループ化領域は，ローカルハッシュグループ化処理が適用される SQL 文ごと，その SQL 文で指定されたGROUP BY 句ごとに指定の値で確保されます。

指定した値がローカルハッシュグループ化処理に十分な領域サイズでなかった場合，指定されたサイズで処理できる範囲でハッシュグループ化されます。処理できなかった行は，作業表用 DB エリアに作業表として格納されます。

ハッシュグループ化領域サイズの値は，次に示す計算式で求めた値を目安としてください。なお，GROUP BY 句が指定された問合せ指定が複数ある SQL 文の場合は，各問合せについて求めた値の最大値を目安としてください。

計算式

adb_sql_exe_hashgrp_area_sizeオペランド＝



変数の説明

- *LGC*

  グループ化列のデータ長の合計値

  「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」の「表 6-9　各データ型のデータ長の一覧」を参照してください。

- *LSF*

  集合関数の結果のデータ長

- *NGR*

  グループ分け結果の行数

このオペランドと，クライアント定義adb_sql_exe_hashgrp_area_size オペランドを両方指定した場合は，クライアント定義adb_sql_exe_hashgrp_area_size オペランドの指定値が優先されます。クライアント定義adb_sql_exe_hashgrp_area_size オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

[12]　adb_sql_exe_hashtbl_area_size = *ハッシュテーブル領域サイズ*

～〈整数〉((0～4,194,304))《2,000》(単位：メガバイト)

ハッシュテーブル領域のサイズをメガバイト単位で指定します。

ハッシュ検索が適用される SQL 文が実行されると，SQL 処理リアルスレッドごとに，リアルスレッド固有メモリに対して，次に示す計算式に従ってハッシュテーブル領域が確保されます。

計算式

↑このオペランドの指定値÷adb_sql_exe_max_rthd_numオペランドの指定値↑

なお，adb_sql_exe_max_rthd_num オペランドの指定値が0 の場合，ハッシュ検索は適用されません。

メモ

ハッシュ検索とはハッシュテーブル領域を使用する処理のことです。次に示す処理が該当します。

- 表の結合方式のハッシュジョイン
- グループ化の処理方式のグローバルハッシュグループ化
- 副問合せの処理方式のハッシュ実行
- SELECT DISTINCT の処理方式のハッシュ実行

- 集合演算の処理方式のハッシュ実行

上記のハッシュ検索の各処理については，マニュアル『HADB　AP 開発ガイド』の次の個所を参照してください。

- 『表の結合方式』
- 『副問合せの処理方式』
- 『グループ化の処理方式』
- 『SELECT DISTINCT の処理方式』
- 『集合演算の処理方式』

ハッシュテーブル領域は，ハッシュ検索が適用される SQL 文ごとに確保されます。

このオペランドに0 を指定した場合，ハッシュ検索は適用されません。なお，次に示す場合に限り，このオペランドに0 を指定してください。

このオペランドに 0 を指定してもよいケース

次に示すオペランドの指定値を大きくできないため，メモリ不足によるエラーを回避できない場合（KFAA30919-E，またはKFAA30930-E メッセージが出力されるとき）

- adb_sys_rthd_area_max オペランド
- adb_sys_memory_limit オペランド
- adb_dbbuff_wrktbl_clt_blk_num オペランド

指定した値がハッシュ検索に十分な領域サイズでなかった場合，指定されたサイズで処理できる範囲でハッシュテーブルが作成されます。ハッシュテーブルで処理できなかった行は，作業表用 DB エリアに作業表として格納されます。

ハッシュテーブル領域サイズの値は，次の計算式で求めた値を目安としてください。

計算式

$$\left\lceil \frac{\text{最大SQL文単位ハッシュテーブル領域サイズ}}{\text{最大SQL処理リアルスレッド数}} \right\rceil \times \text{最大SQL処理リアルスレッド数}$$

変数の説明

- *最大 SQL 文単位ハッシュテーブル領域サイズ*

  実行する各 SQL 文について求めた SQL 文単位ハッシュテーブル領域サイズの最大値です。SQL 文単位ハッシュテーブル領域サイズは，次の計算式で求めてください。

  ```
  ハッシュテーブル数×ハッシュテーブル最大行長×中間検索結果最大件数
  ```

  ハッシュテーブル数，ハッシュテーブル最大行長，および中間検索結果最大件数については，「6.25.3　ハッシュテーブルを使用した検索時に作成される作業表の数」を参照してください。

- *最大 SQL 処理リアルスレッド数*

サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値です。ただし，
adb_sql_exe_max_rthd_num オペランドの指定値が0 の場合は，1 と仮定して計算してください。

このオペランドと，クライアント定義adb_sql_exe_hashtbl_area_size オペランドを両方指定した場合は，クライアント定義adb_sql_exe_hashtbl_area_size オペランドの指定値が優先されます。ただし，クライアント定義adb_sql_exe_hashtbl_area_size オペランドの指定値が，このオペランドの指定値よりも大きい場合は優先されません（このオペランドの指定値が仮定されます）。クライアント定義adb_sql_exe_hashtbl_area_size オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

[13] adb_sql_exe_hashflt_area_size = *ハッシュフィルタ領域サイズ*

～〈整数〉((0～419,430))《サーバ定義の adb_sql_exe_hashtbl_area_size オペランドの値÷10》(単位：メガバイト)

ハッシュフィルタを作成するハッシュフィルタ領域のサイズをメガバイト単位で指定します。次の処理を実行する際に，ハッシュフィルタが適用されます。

- 表の結合方式のハッシュジョイン
- 副問合せの処理方式のハッシュ実行

ハッシュジョイン，ハッシュ実行，ハッシュフィルタについては，マニュアル『HADB　AP 開発ガイド』の『ハッシュジョインとは』および『副問合せの処理方式』を参照してください。

ハッシュフィルタ領域は，ハッシュジョインまたはハッシュ実行の際，ハッシュフィルタが適用されるSQL 文ごとに確保されます。ハッシュフィルタが適用される SQL 文が実行された場合，SQL 処理リアルスレッドごとに，リアルスレッド固有メモリに対して，次に示す計算式に従ってハッシュフィルタ領域が確保されます。

↑*このオペランドの指定値*÷*adb_sql_exe_max_rthd_numオペランドの指定値*↑

指定値の目安と調整

- 次の計算式で求めた値を，このオペランドの指定値の目安としてください。

  ハッシュフィルタ領域サイズ＝
  ↑*adb_sql_exe_hashtbl_area_sizeオペランドの指定値（ハッシュテーブル領域サイズ）*÷10↑

- ハッシュジョインおよびハッシュ実行の際にハッシュフィルタを適用した結果，処理性能が低下した場合に限り，このオペランドに0 を指定してください。
- このオペランドの指定値が小さく，ハッシュフィルタ領域サイズが十分に確保できない場合，ハッシュジョインおよびハッシュ実行の際にハッシュフィルタは適用されません。ハッシュフィルタが適用されたかどうかは，アクセスパスの統計情報のHashtbl_filter_num，およびHashtbl_sum_filter_check_cnt の値で確認できます。

  このオペランドに0 以外を指定した状態で，Hashtbl_filter_num の値に0 以外が出力され，かつ，Hashtbl_sum_filter_check_cnt に0 が出力された場合，ハッシュフィルタ領域サイズが不足しています。ハッシュフィルタが適用されていない状態のため，このオペランドの指定値を大きくしてください。アクセスパスの統計情報については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」の「(2)　アクセスパスの統計情報に出力される項目」の「(e)　ハッシュテーブル領域に関する情報」を参照してください。

留意事項

- このオペランドに0 を指定した場合，ハッシュジョインおよびハッシュ実行の際，ハッシュフィルタは適用されません。
- adb_sql_exe_max_rthd_num オペランドに0 を指定した場合，ハッシュジョインおよびハッシュ実行は適用されません。
- このオペランドと，クライアント定義のadb_sql_exe_hashflt_area_size オペランドの両方を指定した場合は，クライアント定義のadb_sql_exe_hashflt_area_size オペランドの指定値が優先されます。ただし，クライアント定義のadb_sql_exe_hashflt_area_size オペランドの指定値が，このオペランドの指定値よりも大きい場合は優先されません（このオペランドの指定値が仮定されます)。クライアント定義のadb_sql_exe_hashflt_area_size オペランドについては，マニュアル『HADB AP 開発ガイド』の『性能に関するオペランド』を参照してください。

[14] adb_sql_opt_hashjoin_div = {Y | N}

ハッシュジョインの分割実行を適用するかどうかを指定します。ハッシュジョインについては，マニュアル『HADB　AP 開発ガイド』の『ハッシュジョインとは』を参照してください。

- Y

  ハッシュジョインの分割実行を HADB が適用可能と判断した場合は，ハッシュジョインの分割実行を適用することがあります。

- N

  ハッシュジョインの分割実行を適用しません。

通常，このオペランドは指定不要です。

ハッシュジョインの分割実行を適用すると問題がある場合に，このオペランドにN を指定してください。

すべての SQL 文に対してハッシュ検索を実行するのに十分なハッシュテーブル領域サイズがadb_sql_exe_hashtbl_area_size オペランドに指定されている場合は，ハッシュジョインの分割実行を適用する必要はありません。このような場合にハッシュジョインの分割実行が適用されると，処理時間が長くなります。そのときは，このオペランドにN を指定してください。

なお，SQL 文の実行時にハッシュジョインの分割実行が適用されるかどうかは，アクセスパス情報で確認できます。詳細については，マニュアル『HADB　AP 開発ガイド』の『ツリー表示に出力される情報』の『表の結合方式』を参照してください。

[マルチノード機能]

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにすることを推奨します。各ノードでこのオペランドの指定値が異なる場合，SQL 文を実行するノードごとに異なるアクセスパスになるおそれがあります。

[15] adb_sql_opt_drvtbl_grping_prior = {LOCAL | GLOBAL}

次の条件をすべて満たす SQL 文を実行する場合に，優先するグループ化の処理方式を指定します。

- FROM 句に表参照を 1 つだけ指定している
- 上記の表参照が，展開されない内部導出表である
- 最も外側の問合せ指定にGROUP BY 句を指定している

通常，このオペランドは指定不要です。

SQL 文の処理に使用できる SQL 処理リアルスレッド数が多い場合（目安は 30 以上）にGLOBAL を指定すると，SQL 文の実行時間が短くなることがあります。

- LOCAL

  グループ化の処理方式にローカルハッシュグループ化を優先します。

- GLOBAL

  グループ化の処理方式にグローバルハッシュグループ化を優先します。

このオペランドの指定値とグループ化の処理方式の関係を次の例で説明します。

(例)

```
SELECT "C2"
    FROM (SELECT "C1",COUNT(*) FROM "T1" GROUP BY "C1") AS "DT1" ("C1","C2")
    GROUP BY "C2"
```

このオペランドにLOCAL を指定して上記の SQL 文を実行した場合は，外側のグループ化（下線部分のグループ化）の処理方式にローカルハッシュグループ化が適用されます。

このオペランドにGLOBAL を指定して上記の SQL 文を実行した場合は，外側のグループ化（下線部分のグループ化）の処理方式にグローバルハッシュグループ化が適用されます。

[マルチノード機能]

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにすることを推奨します。各ノードでこのオペランドの指定値が異なる場合，SQL 文を実行するノードごとに異なるアクセスパスになるおそれがあります。

[16] adb_sys_rthd_area_max = *確保できるリアルスレッド固有メモリの最大サイズ*

～〈整数〉((1,024～2,000,000,000))《1,024》(単位：メガバイト)

確保できる 1 リアルスレッド当たりのリアルスレッド固有メモリの最大サイズをメガバイト単位で指定します。

バージョン 03-01 以降の HADB サーバでは，このオペランドは指定不要です。その代わり，サーバ定義adb_sys_memory_limit オペランドを指定してください。このオペランドを指定した場合，サーバ定義adb_sys_memory_limit オペランドは指定できません。

もし，このオペランドを指定する場合は，次に示す計算式で求めた値を指定してください。

計算式

adb_sys_rthd_area_maxオペランド＝

$$\mathrm{MAX}\left\{1{,}024\ ,\ \left(\mathrm{MACHINE_MEMORY_SIZE}-(\mathrm{SHMMAN}+\mathrm{SHM_BUFGLOBAL})\div 1{,}024\right)\div(\mathrm{RTHD_NUM}+1)\right\}$$

変数の説明

- *MACHINE_MEMORY_SIZE*

  HADB サーバをインストールしたサーバマシンに搭載された物理メモリサイズ（単位：メガバイト）

- *SHMMAN*

HADB サーバ開始時に取得する HADB で使用する共有メモリの情報を格納する領域の領域サイズ（単位：キロバイト）

変数*SHMMAN* については，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(1) 共有メモリ管理の所要量の求め方（HADB サーバ開始時）」を参照してください。

- *SHM_BUFGLOBAL*

  HADB サーバ開始時に取得するグローバルバッファ領域の領域サイズ（単位：キロバイト）

  変数*SHM_BUFGLOBAL* については，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」を参照してください。

- *RTHD_NUM*

  HADB サーバで使用するリアルスレッド数（単位：個）

  次に示す計算式で求めてください。マルチノード機能を使用するかどうかで，計算式が異なります。

  - マルチノード機能を使用しない場合

    RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10

  - マルチノード機能を使用する場合（ワーカーノードを使用しない場合）

    RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10
    ＋（*サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）
    ×（*サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）＋5

  - マルチノード機能を使用する場合（ワーカーノードを使用する場合）

    ・プライマリノードまたはセカンダリノードの場合

    RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10
    ＋（*サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）
    ×（*サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）＋5
    ＋*サーバ定義のadb_sys_multi_max_worker_nodesオペランドの指定値*
    ×（*サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）

    ・ワーカーノードの場合

    RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10
    ＋（*サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）
    ×（*サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）＋5
    ＋*サーバ定義のadb_sys_multi_max_worker_nodesオペランドの指定値*
    ×*サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値*×2＋2

[17]　adb_sys_proc_area_max = *確保できるプロセス共通メモリの最大サイズ*

～〈整数〉((5,120～2,000,000,000))《8,192》（単位：メガバイト）

確保できるプロセス共通メモリの最大サイズをメガバイト単位で指定します。

バージョン 03-01 以降の HADB サーバでは，このオペランドは指定不要です。その代わり，サーバ定義adb_sys_memory_limit オペランドを指定してください。このオペランドを指定した場合，サーバ定義adb_sys_memory_limit オペランドは指定できません。

もし，このオペランドを指定する場合は，次に示す計算式で求めた値を指定してください。

計算式

adb_sys_proc_area_maxオペランド＝

$$\mathrm{MAX}\left\{5,120\ ,\ \left(\mathrm{MACHINE_MEMORY_SIZE}-(\mathrm{SHMMAN}+\mathrm{SHM_BUFGLOBAL})\div 1,024\right)\div(\mathrm{RTHD_NUM}+1)\right\}$$

変数の説明

- *MACHINE_MEMORY_SIZE*

  HADB サーバをインストールしたサーバマシンに搭載された物理メモリサイズ（単位：メガバイト）

- *SHMMAN*

  HADB サーバ開始時に取得する HADB で使用する共有メモリの情報を格納する領域の領域サイズ（単位：キロバイト）

  変数*SHMMAN* については，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(1) 共有メモリ管理の所要量の求め方（HADB サーバ開始時)」を参照してください。

- *SHM_BUFGLOBAL*

  HADB サーバ開始時に取得するグローバルバッファ領域の領域サイズ（単位：キロバイト）

  変数*SHM_BUFGLOBAL* については，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時)」を参照してください。

- *RTHD_NUM*

  HADB サーバで使用するリアルスレッド数（単位：個）

  次に示す計算式で求めてください。マルチノード機能を使用するかどうかで，計算式が異なります。

  - **マルチノード機能を使用しない場合**

    RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10

  - **マルチノード機能を使用する場合（ワーカーノードを使用しない場合）**

    RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10
    ＋（*サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）
    ×（*サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）＋5

  - **マルチノード機能を使用する場合（ワーカーノードを使用する場合）**

    ・プライマリノードまたはセカンダリノードの場合

```
RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10
    ＋（サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数－1）
    ×（サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値×2＋2）＋5
    ＋サーバ定義のadb_sys_multi_max_worker_nodesオペランドの指定値
    ×（サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値×2＋2）
```

・ワーカーノードの場合

```
RTHD_NUM＝サーバ定義のadb_sys_max_usersオペランドの指定値
    ＋サーバ定義のadb_sys_rthd_numオペランドの指定値＋10
    ＋（サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数－1）
    ×（サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値×2＋2）＋5
    ＋サーバ定義のadb_sys_multi_max_worker_nodesオペランドの指定値
    ×サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値×2＋2
```

[18] adb_sys_shm_huge_page_size = *HugePages の 1 ページ当たりのページサイズの倍数*

～〈整数〉((0～2,147,483,647))《0》(単位：キロバイト)

HADB サーバが使用する共有メモリにHugePages を適用する場合，HugePages の 1 ページ当たりのページサイズの倍数をキロバイト単位で指定します。

このオペランドには，次のコマンドで確認した値（HugePages の 1 ページ当たりのページサイズ）を指定することを推奨します。コマンドで値を確認するときは，値の単位に注意してください。このオペランドにHugePages の 1 ページ当たりのページサイズの倍数以外の値を指定すると，共有メモリの確保に失敗し，HADB サーバが開始できなくなります。

実行するコマンド

```
grep Hugepagesize /proc/meminfo
```

実行結果の例

```
Hugepagesize:       2048 kB
```

この例では，2048 を指定することを推奨します。

このオペランドに0 を指定すると，HADB サーバが使用する共有メモリにHugePages は適用されません。

なお，HADB サーバが使用する共有メモリにHugePages を適用するためには，カーネルパラメタを設定する必要があります。カーネルパラメタの設定については，「6.2　カーネルパラメタの見積もり」を参照してください。

[19] adb_dbbuff_wrktbl_glb_blk_num = *グローバル作業表用のグローバルバッファのページ数*

～〈整数〉((5～100,000,000))《128》

グローバル作業表用のグローバルバッファのページ数を指定します。

このオペランドを指定して確保されたグローバルバッファは，グローバル作業表を作成する SQL 文を実行したときだけ使用されます。

このオペランドの指定値の見積もり方法については，「6.25.1　グローバル作業表用のグローバルバッファのページ数の見積もり」を参照してください。

また，グローバル作業表については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『作業表の種類』を参照してください。

[20] adb_dbbuff_wrktbl_clt_blk_num = *ローカル作業表用バッファのページ数*

～〈整数〉((5～100,000,000))《256》

ローカル作業表用バッファのページ数を指定します。

このオペランドを指定して確保されたローカル作業表用バッファは，ローカル作業表を作成する SQL 文を実行したときだけ使用されます。また，このオペランドで指定したページ数分のローカル作業表用バッファは，リアルスレッドごとに確保されます。

このオペランドの指定値の見積もり方法については，「6.25.2　ローカル作業表用バッファのページ数の見積もり」を参照してください。

ローカル作業表については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『作業表の種類』を参照してください。

メモ

- このオペランドと，クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドを両方指定した場合は，クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値が優先されます。クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。
- このオペランドと，エクスポートオプションadb_export_wrktbl_blk_num を両方指定した場合は，エクスポートオプションadb_export_wrktbl_blk_num の指定値が優先されます。エクスポートオプションadb_export_wrktbl_blk_num については，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』の『adbexport コマンドの指定形式』の『エクスポートオプションの形式』を参照してください。
- 実際に確保されたローカル作業表用バッファのページ数は，adbls -d lbuf コマンドで確認できます。adbls -d lbuf コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d lbuf（ローカル作業表用バッファの情報表示)』を参照してください。
- このオペランドで指定したローカル作業表用バッファのページ数は，HADB サーバの稼働中に，adbmodbuff コマンドで変更できます。adbmodbuff コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodbuff（バッファの変更)』を参照してください。

[21] adb_sql_tbldef_cache_size = *表定義プールサイズ*

～〈整数〉((500～2,048,000))《5,120》(単位：キロバイト)

表定義情報を格納する表定義プールサイズをキロバイト単位で指定します。

通常，このオペランドは指定不要です。コスト情報を使用して運用する場合は，以降の説明を読み，適切な値を指定してください。

表定義情報とは，表にアクセスするために必要な情報で，表の情報だけではなく，列やインデクスの情報も含みます。表定義情報の作成時にはディクショナリ表へのアクセスが発生します。一度使用された表定義情報は，表定義プール（キャッシュ）に登録され，LRU 方式で管理されます。

表定義プールに表定義情報が登録されていると，次にその表定義情報が必要になったとき，表定義プールから取り出して利用できます（ディクショナリ表にアクセスして表定義情報を作成する必要がなくなります）。したがって，表定義プールを使用することで，I/O 回数の削減や，CPU 使用時間の短縮が期待できます。

表へのアクセス時に必要な表定義情報が表定義プールに存在しない場合は，表定義情報が作成され，表定義プールに登録されます。表定義プール上に空き領域がない場合は，使用されていないほかの表定義情報が表定義プールから掃き出されたあとで，表定義情報が登録されます。なお，表定義プール上に空き領域が確保できなかったときは，表定義情報が登録されません。

このため，HADB で実行されるトランザクション中で使用する表の表定義情報をすべて格納できるように，次の計算式で求めた表定義プールサイズを指定してください。表定義プールの空き領域が不足して表定義情報の掃き出しが発生した場合，SQL 文の性能に影響するおそれがあります。

計算式

$$\text{adb_sql_tbldef_cache_size オペランド} = \left\lceil \frac{\sum_{i=1}^{m} Wi}{1,024} \right\rceil \times 1.2$$

変数の説明

- *m*

  頻繁にアクセスする表の数

- *Wi*

  表定義情報長（単位：バイト）

  次に示す計算式で求めてください。

  計算式

$$Wi = 1,012 + (512 + 32,008^{※1}) \times col_num + max_rowsz + 1,156 \times idx_num + 15,864^{※2}$$

  注※1

  コスト情報を収集したことがある表の場合，加算してください。

  コスト情報を収集したことがある表かどうかを確認する方法については，「付録 C.9　システム表の検索」の「(1)　コスト情報を収集した全実表の表名と収集日時を調べる場合」を参照してください。

  注※2

  処理対象表がアーカイブマルチチャンク表の場合，加算してください。

  col_num

  表の列数

max_rowsz

最大行長（単位：バイト）

最大行長については，「5.8.2　行の種別ごとの格納ページ数の求め方」の「(1)　基本行用ページ数（変数 BP（i））の求め方」の行長*ROWSZ* の計算式を基に求めてください。

idx_num

表に定義されているインデクス数

7.2.3　システムログに関するオペランド（set 形式）

[22]　adb_log_usrfile_num = *ユーザログファイル数*

～〈整数〉((0～10,240))《同時に表を更新できるリアルスレッド数》

特定の SQL 文およびコマンドの実行時に必要なユーザログファイルの数を指定します。

指定値の分だけ，HADB サーバ開始時にユーザログファイルが作成されます。このオペランドの指定値については，「6.12.15　ユーザログファイル数の求め方」を参照してください。

留意事項を次に示します。

- この指定を省略した場合，または0 を指定した場合は，同時に表を更新できるリアルスレッド数が仮定されます。その結果，adb_log_usrfile_size オペランドに指定した容量のユーザログファイルが必要な分だけ作成されるようになります。そのため，ユーザログファイル数の不足によるエラーは発生しません。
- このオペランドに1 を指定した場合は，自動的に2 が仮定されます。そのため，ユーザログファイルが 2 つ作成されます。
- 同時に表を更新できるリアルスレッド数より大きな値をこのオペランドに指定しても，同時に表を更新できるリアルスレッド数がこのオペランドの指定値として仮定されます。
- 作成されたユーザログファイルは，データベースの初期設定を再度実行するまで削除されません。そのため，次に示す計算式で求めた値以上の空き領域が，DB ディレクトリ下のシステムディレクトリに存在することを確認してください。

  計算式

```
A×B

変数の説明
  A：ユーザログファイル数（adb_log_usrfile_numオペランドの値）
  B：ユーザログファイル初期化容量（adb_log_usrfile_sizeオペランドの値）
```

なお，ここでの「データベースの初期設定」とは，マルチノード機能を使用していない場合は，adbinit コマンドの実行を意味しています。マルチノード機能を使用している場合は，データベース作成時の「16.3.8　データベースの作成」の「(2)　ノード切り替え対象のファイルシステムの初期化」で説明している作業の実行を意味しています。

［マルチノード機能］

マルチノード機能を使用する場合は，このオペランドは指定不要です。仮に指定値として有効な値を指定しても，プライマリノードおよびセカンダリノードでは，このオペランドに0 が指定されたと仮定されます。そして，プライマリノードおよび各セカンダリノードで必要なユーザログファイル数が自動計算されて，最も大きい値が有効となります。

ワーカーノードでは，このオペランドの指定値に関係なく，ユーザログファイルは 2 個生成されます。

[23] adb_log_usrfile_size =*ユーザログファイルの初期化容量*〔,*ユーザログファイルの容量の縮小契機*〕

ユーザログファイルの初期化容量とユーザログファイルの容量の縮小契機を指定します。

ユーザログファイルの初期化容量：

～〈整数〉((16～1,048,576))《16》（単位：メガバイト）

ユーザログファイルの初期化容量をメガバイト単位で指定します。

このオペランドの指定値については，「6.12.2　ユーザログファイルの容量の求め方」を参照してください。

［マルチノード機能］

- プライマリノードおよびセカンダリノードでは，プライマリノードおよび各セカンダリノードに指定した*ユーザログファイルの初期化容量*のうち，最も大きい値が有効となります。
- ワーカーノードでは，このオペランドの指定値に関係なく，*ユーザログファイルの初期化容量*は 16 メガバイトになります。

ユーザログファイルの容量の縮小契機：

～〈整数〉((0，32～2,097,152))《ユーザログファイルの初期化容量×2》（単位：メガバイト）

ユーザログファイルの容量を縮小する契機をメガバイト単位で指定します。更新トランザクションの決着時，更新トランザクションが使用していたユーザログファイルの容量が，ここで指定した容量を超えている場合，ユーザログファイルを初期化容量に縮小します。通常は，このオペランドの指定は不要です。

規則を次に示します。

- *ユーザログファイルの容量の縮小契機*に0 を指定した場合，ユーザログファイルの容量の縮小は行われません。
- 「*ユーザログファイルの容量の縮小契機*＜*ユーザログファイルの初期化容量*」の場合，デフォルト値（*ユーザログファイルの初期化容量*×2）が仮定されます。

［マルチノード機能］

- プライマリノードおよびセカンダリノードでは，プライマリノードおよび各セカンダリノードに指定した*ユーザログファイルの容量の縮小契機*のうち，最も大きい値が有効となります。
- 「有効となった*ユーザログファイルの容量の縮小契機*＜有効となった*ユーザログファイルの初期化容量*」の場合，有効となったユーザログファイルの初期化容量の 2 倍の容量が，*ユーザログファイルの容量の縮小契機*として仮定されます。
- プライマリノードおよび全セカンダリノードで*ユーザログファイルの容量の縮小契機*に0 を指定した場合，ユーザログファイルの容量の縮小は行われません。

- ワーカーノードでは，このオペランドの指定値に関係なく，*ユーザログファイルの容量の縮小契機*は 32 メガバイトになります。

[24] adb_log_usrbuf_num = *ユーザログバッファ面数*

～〈整数〉((64～12,582,912))《64》

ユーザログファイルにアクセスするために使用するバッファの面数を指定します。

通常は，このオペランドの指定は不要です。指定値を大きくすると，ユーザログファイルへのアクセス速度が向上することがあります。

なお，HADB サーバは，ユーザログファイルごとにユーザログバッファを割り当てます。

[マルチノード機能]

- プライマリノードおよびセカンダリノードでは，プライマリノードおよび各セカンダリノードに指定した*ユーザログバッファ面数*のうち，最も大きい値が有効となります。
- ワーカーノードでは，このオペランドの指定値に関係なく，*ユーザログバッファ面数*は 64 になります。

[25] adb_log_rec_msg_interval = *データベースの回復処理の経過時間を示すメッセージの出力間隔*

～〈整数〉((0～60))《5》(単位：分)

データベースの回復処理の経過時間を示すメッセージの出力間隔を，分単位で指定します。

通常は，このオペランドの指定は不要です。

データベースの回復処理時に，このオペランドの指定値に従って，次に示す 2 つのメッセージが出力されます。

■KFAA81215-I メッセージが出力されるケース

次のような処理が発生してデータベースの回復処理が開始された場合，このオペランドの指定値に従って，データベースの回復処理の経過時間を示すKFAA81215-I メッセージが，一定の間隔で出力されます。

- HADB サーバの再開始処理
- トランザクションのロールバック処理

例えば，このオペランドに3 を指定した場合，データベースの回復処理が開始されてから 3 分が経過するたびにKFAA81215-I メッセージが出力されます。KFAA81215-I メッセージの内容を確認することで，データベースの回復処理に関する進捗状況を把握できます。

■KFAA81211-I メッセージが出力されるケース

データベースの回復処理中に，回復処理に時間が掛かったトランザクションが存在する場合は，KFAA81211-I メッセージも出力されます。

例えば，このオペランドに3 を指定した場合，1 つのトランザクションの回復処理が 3 分以上掛かったときに，そのトランザクションの回復処理が終了するときにKFAA81211-I メッセージが出力されます。KFAA81211-I メッセージには，そのトランザクションの回復処理に関する情報が出力されます。この出力情報を，トランザクションの回復処理に時間が掛かった原因の判断材料の 1 つにしてください。

なお，このオペランドに0 を指定した場合は，KFAA81211-I メッセージ，およびKFAA81215-I メッセージは出力されません。

7.2.4 状態監視に関するオペランド（set 形式）

[26] adb_sys_max_users_wrn_pnt = *最大同時接続数に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕

このオペランドを指定すると，adb_sys_max_users オペランドで指定した最大同時接続数の残り枠が少なくなったときに，警告メッセージKFAA40020-W が出力されます。

最大同時接続数に関する警告メッセージの出力契機：

～〈整数〉((0～100))《0》(単位：%)

警告メッセージKFAA40020-W の出力契機を，adb_sys_max_users オペランドの指定値（最大同時接続数）に対する比率で指定します。例えば，adb_sys_max_users オペランドに200 を指定している場合に，*最大同時接続数に関する警告メッセージの出力契機*に90 を指定すると，HADB サーバへの接続数が 180（200 の 90%の値）以上になったときに警告メッセージが出力されます。

なお，*最大同時接続数に関する警告メッセージの出力契機*に0 を指定した場合，警告メッセージは出力されません。

警告メッセージの出力済み状態のリセット契機：

～〈整数〉((0～99))(単位：%)

警告メッセージKFAA40020-W の出力済み状態のリセット契機を，adb_sys_max_users オペランドの指定値（最大同時接続数）に対する比率で指定します。警告メッセージは一度出力されると，警告メッセージ出力済み状態となり，HADB サーバへの接続数がある一定の値を下回らないかぎり警告メッセージは再出力されません。

メモ

adb_sys_max_users_wrn_pnt オペランドの指定値と警告メッセージの出力の関係を次に示します。

```
adb_sys_max_users = 200
adb_sys_max_users_wrn_pnt = 90,70
```

上記のようにオペランドを指定した場合，HADB サーバへの接続数が 180（200 の 90%の値）以上になったときに警告メッセージが出力されます。その後，HADB サーバへの接続数がいったん 170 に下がったあとに再度 180 以上になっても，警告メッセージは出力されません。

HADB サーバへの接続数が，140（200 の 70%の値）以下に下がった場合は，警告メッセージの出力済み状態がリセットされるため，HADB サーバへの接続数が再度 180 以上になったときに警告メッセージが出力されます。

警告メッセージの出力済み状態のリセット契機に関する留意事項を次に示します。

- リセット契機の指定を省略した場合，「*最大同時接続数に関する警告メッセージの出力契機*－30」の値が仮定されます。その値がマイナスになる場合は，0 が仮定されます。
- 「*最大同時接続数に関する警告メッセージの出力契機* ＞ *警告メッセージの出力済み状態のリセット契機*」となるように指定してください。この条件を満たさない場合，リセット契機の指定は無効になります（警告メッセージKFAA40022-W が出力されます）。このとき，リセット契機には，「*最大同時接続数に関する警告メッセージの出力契機*－30」が仮定されます。
- *最大同時接続数に関する警告メッセージの出力契機*に0 を指定した場合，リセット契機の指定は無効になります。

［マルチノード機能］

- マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。
- 警告メッセージKFAA40020-W は，プライマリノードに出力されます。

[27] adb_sys_memory_limit_wrn_pnt = *HADB サーバのメモリ使用量に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕

このオペランドを指定すると，HADB サーバが使用できるメモリの残容量が少なくなったときに，警告メッセージKFAA40021-W が出力されます。

なお，adb_sys_memory_limit オペランドが指定されている場合に限り，adb_sys_memory_limit_wrn_pnt オペランドの指定が有効になります。

HADB サーバのメモリ使用量に関する警告メッセージの出力契機：

～〈整数〉((0～100))《0》(単位：%)

警告メッセージKFAA40021-W の出力契機を，adb_sys_memory_limit オペランドの指定値（HADB サーバが使用するメモリの最大サイズ）に対する比率で指定します。例えば，adb_sys_memory_limit オペランドに64,000 メガバイトを指定している場合に，*HADB サーバのメモリ使用量に関する警告メッセージの出力契機*に90 を指定すると，HADB サーバのメモリ使用量が 57,600 メガバイト（64,000 の 90%の値）以上になったときに警告メッセージが出力されます。

重要

HADB サーバの開始直後にadbls -d mem -a コマンドを実行し，その実行結果に出力されるTOTAL_USE_PER の値を確認してください。このオペランドには，TOTAL_USE_PER の値を超える値を指定してください。TOTAL_USE_PER には，adb_sys_memory_limit オペランドの指定値に対する現在の HADB サーバのメモリ使用量の割合（単位：%）が表示されます。

TOTAL_USE_PER の値以下の値を指定すると，HADB サーバの開始時点で警告メッセージの出力条件を満たしてしまうことがあります。この場合，HADB サーバの開始時に警告メッセージが出力され，HADB の稼働中には警告メッセージが出力されません。

なお，*HADB サーバのメモリ使用量に関する警告メッセージの出力契機*に0 を指定した場合，警告メッセージは出力されません。

警告メッセージの出力済み状態のリセット契機：

～〈整数〉((0～99))（単位：%）

警告メッセージKFAA40021-W の出力済み状態のリセット契機を，adb_sys_memory_limit オペランドの指定値（HADB サーバが使用するメモリの最大サイズ）に対する比率で指定します。警告メッセージは一度出力されると，警告メッセージ出力済み状態となり，HADB サーバのメモリ使用量がある一定の値を下回らないかぎり警告メッセージは再出力されません。

メモ

adb_sys_memory_limit_wrn_pnt オペランドの指定値と警告メッセージの出力の関係を次に示します。

```
adb_sys_memory_limit = 64,000
adb_sys_memory_limit_wrn_pnt = 90,70
```

上記のようにオペランドを指定した場合，HADB サーバのメモリ使用量が 57,600 メガバイト（64,000 の 90%の値）以上になったときに警告メッセージが出力されます。その後，HADB サーバのメモリ使用量がいったん 51,200 メガバイト（64,000 の 80%の値）に下がったあとに再度 57,600 メガバイト以上になっても，警告メッセージは出力されません。

HADB サーバのメモリ使用量が，44,800 メガバイト（64,000 の 70%の値）以下に下がった場合は，警告メッセージの出力済み状態がリセットされるため，HADB サーバのメモリ使用量が再度 57,600 メガバイト以上になったときに警告メッセージが出力されます。

警告メッセージの出力済み状態のリセット契機に関する留意事項を次に示します。

- リセット契機の指定を省略した場合，「*HADB サーバのメモリ使用量に関する警告メッセージの出力契機* －30」の値が仮定されます。その値がマイナスになる場合は，0 が仮定されます。
- 「*HADB サーバのメモリ使用量に関する警告メッセージの出力契機* ＞ *警告メッセージの出力済み状態のリセット契機*」となるように指定してください。この条件を満たさない場合，リセット契機の指定は無効になります（警告メッセージKFAA40022-W が出力されます）。このとき，リセット契機には，「*HADB サーバのメモリ使用量に関する警告メッセージの出力契機* －30」が仮定されます。
- *HADB サーバのメモリ使用量に関する警告メッセージの出力契機*に0 を指定した場合，リセット契機の指定は無効になります。

[マルチノード機能]

警告メッセージKFAA40021-W は，全ノードに出力されることがあります。

[28] adb_dbarea_wrn_pnt_use_rate = *データ用 DB エリアの使用率*〔,*データ用 DB エリアの使用率*〔,*データ用 DB エリアの使用率*〕〕

～〈整数〉((0～99))《0》(単位：%)

データ用 DB エリアの使用率を監視する場合にこのオペランドを指定します。データ用 DB エリアの使用率がこのオペランドの指定値に達したときに，警告メッセージKFAA41230-W がメッセージログファイルに出力されます。このオペランドの指定例を次に示します。

(例)

```
set adb_dbarea_wrn_pnt_use_rate = 70,80,90
```

上記のように指定した場合，データ用 DB エリアの使用率が 70%，80%，90%に達したときに警告メッセージが出力されます。

なお，このオペランドに「90,70,80」と指定しても，「70%，80%，90%」に達したときに警告メッセージが出力されます。

重要

- レギュラーファイルで構成されているデータ用 DB エリアは，使用率の監視対象にはなりません。
- データ用 DB エリア以外の DB エリア（ディクショナリ用 DB エリアなど）は，使用率の監視対象にはなりません。

なお，このオペランドを省略した場合，またはこのオペランドに0 を指定した場合は，データ用 DB エリアの使用率は監視されません。

データ用 DB エリアの使用率は，次に示す計算式で算出されます。

データ用DBエリアの使用率＝↓割り当て済みのセグメント数÷割り当て可能なセグメント数↓

割り当て可能なセグメント数は，HADB サーバの開始時にデータ用 DB エリアを構成するデータ用 DB エリアファイルの容量を基に算出されます。

[マルチノード機能]

- マルチノード機能を使用する場合，プライマリノードのこのオペランドの指定値が適用されます。そのため，プライマリノードの切り替えを想定して，セカンダリノードにもこのオペランドを指定し，プライマリノードと同じ値を指定してください。プライマリノードとセカンダリノードでオペランドの指定値が異なると，プライマリノードの切り替えが発生したときに，監視対象とする使用率が変わります。

  (例)

  - ノード A の指定値：80,90
  - ノード B の指定値：85,95

  上記の場合，ノード A がプライマリノードのときは，使用率が 80%，90%に達したときに警告メッセージが出力されます。プライマリノードの切り替えが発生してノード B がプライマリノードになったときは，使用率が 85%，95%に達したときに警告メッセージが出力されます。

なお，ワーカーノードはプライマリノードに切り替わることがないため，このオペランドを指定する必要はありません。

- 警告メッセージKFAA41230-Wは，プライマリノードだけに出力されます。

[29] adb_dbarea_wrn_pnt_except_name = *データ用DBエリア名*〔,*データ用DBエリア名*〕…

～〈文字列〉((1～30バイト))

データ用DBエリアの使用率の監視対象から外したいデータ用DBエリアがある場合にこのオペランドを指定します。

adb_dbarea_wrn_pnt_use_rateオペランドを指定すると，ブロックスペシャルファイルで構成されているすべてのデータ用DBエリアが使用率の監視対象になります。この場合，警告メッセージKFAA41230-Wが頻繁に出力されることがあります。出力される警告メッセージの量を減らしたい場合にこのオペランドを指定してください。

例えば，次の条件をすべて満たす場合，警告メッセージが頻繁に出力されることがあります。このようなときにこのオペランドを指定して，警告メッセージの出力を抑止してください。

- データ用DBエリアの使用率が常に安定している（例えば，80%くらいで安定しているとする）
- データの削除と追加を頻繁に繰り返す
- 「adb_dbarea_wrn_pnt_use_rate = 70,80,90」と指定している

上記の場合，データを削除して追加するたびに，データ用DBエリアの使用率が80%を超えて警告メッセージが出力されます。

[マルチノード機能]

マルチノード機能を使用する場合，プライマリノードのこのオペランドの指定値が適用されます。そのため，プライマリノードの切り替えを想定して，セカンダリノードにもこのオペランドを指定し，プライマリノードと同じデータ用DBエリアを指定してください。プライマリノードとセカンダリノードで指定したデータ用DBエリアが異なると，プライマリノードの切り替えが発生したときに，監視対象外とするデータ用DBエリアが変わります。

なお，ワーカーノードはプライマリノードに切り替わることがないため，このオペランドを指定する必要はありません。

[30] adb_rpc_wait_time = *応答待ち時間の上限*

～〈整数〉((0～86,400))《1,800》(単位：秒)

HADBクライアントからの処理要求に応答してから，次の処理要求を受けるまでの待ち時間の上限を秒単位で指定します。

ここで指定した待ち時間を超えても，HADBクライアントから処理要求がなかった場合，HADBクライアントとの接続を切り離します。トランザクション処理中の場合は，そのトランザクションをロールバックしてからHADBクライアントとの接続を切り離します。

なお，このオペランドに0を指定した場合，待ち時間の上限は設定されません。

[マルチノード機能]

プライマリノードの切り替えが発生した時点で，HADBサーバに接続だけしていてトランザクションを実行していないAPは，HADBサーバから切り離されます。

[31] adb_rpc_tcp_keepalive_time = *KeepAlive プローブ開始までの無通信時間*

～〈整数〉((0～120))《10》(単位：分)

HADB クライアントと確立したコネクションについて，KeepAlive プローブによる生存確認を開始するまでの無通信時間を，分単位で指定します。

通常は，このオペランドの指定は不要です。このオペランドを指定すると，OS の KeepAlive 制御（KeepAlive プローブを開始するまでの無通信時間）を変更できます。

KeepAlive プローブに対する応答がない場合は，生存確認を 30 秒ごとに 4 回繰り返します。それでも応答がない場合は，HADB クライアントとのコネクションを切断します。

なお，このオペランドに0 を指定した場合は，KeepAlive プローブによる生存確認を行いません。

メモ

KeepAlive は，一定時間ごとに送信するプローブパケットに対する応答有無によって，コネクションの生存を確認する機能です。また，**プローブパケット**は，通信回線の生存を確認するために送信するパケットのことです。

7.2.5 SQL 文に関するオペランド（set 形式）

[32] adb_sql_prep_delrsvd_words = *削除する予約語*〔,*削除する予約語*〕…

～〈文字列〉((1～100 バイト))

HADB の予約語から削除する語を指定します。削除する予約語は 999 個まで指定できます。

削除する予約語は，¥”（バックスラッシュと二重引用符）で囲んだ英大文字で指定してください。または，¥”（バックスラッシュと二重引用符）で囲まない英大文字，英小文字で指定してください。指定した予約語が正しく削除される指定例を次に示します。

（例）予約語「ABS」を削除する場合の指定例

- adb_sql_prep_delrsvd_words = ¥”ABS¥”
- adb_sql_prep_delrsvd_words = ABS
- adb_sql_prep_delrsvd_words = abs

メモ

文字列を¥"（バックスラッシュと二重引用符）で囲むと，英大文字と英小文字が区別されます。したがって，「¥”abs¥”」のように英小文字を¥”（バックスラッシュと二重引用符）で囲んで指定すると，文字列が英小文字として扱われます。しかし，HADB には英小文字の予約語はないため，英小文字を¥”（バックスラッシュと二重引用符）で囲んで指定しても予約語は削除されません。

なお，adb_sql_prep_delrsvd_words オペランドで削除できる予約語はあらかじめ決まっています。指定した単語が予約語ではない，または削除できない予約語の場合，警告メッセージ（KFAA51111-W メッセージ）が出力されます。

削除できない予約語については，マニュアル『HADB　SQL リファレンス』の『予約語の一覧』を参照してください。

［マルチノード機能］

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。その際，予約語の指定順序も同じにしてください。

[33] adb_sys_trn_iso_lv = {READ_COMMITTED | REPEATABLE_READ}

適用するトランザクション隔離性水準を指定します。トランザクション隔離性水準については，「2.9.2 HADB がサポートしているトランザクション隔離性水準」を参照してください。

- READ_COMMITTED

  トランザクション隔離性水準にREAD COMMITTED を適用します。

- REPEATABLE_READ

  トランザクション隔離性水準にREPEATABLE READ を適用します。

［マルチノード機能］

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。

[34] adb_sql_text_out = {Y | N}

HADB サーバが受け付けたすべての SQL 文をサーバメッセージログファイルに出力するかどうかを指定します。

Y を指定した場合，各 SQL 文の先頭から 2,048 バイト分まで出力されます。

- Y

  SQL 文をサーバメッセージログファイルに出力します。

- N

  SQL 文をサーバメッセージログファイルに出力しません。

このオペランドにN を指定した場合，またはこのオペランドを省略した場合でも，クライアント定義の adb_clt_sql_text_out オペランドにY を指定したときは，SQL 文をサーバメッセージログファイルに出力します。

adb_clt_sql_text_out オペランドについては，マニュアル『HADB　AP 開発ガイド』の『SQL に関するオペランド』を参照してください。

[35] adb_sql_trc_out = {Y | N}

SQL 文を実行したときに，SQL トレース情報を出力するかどうかを指定します。

- Y

  SQL トレース情報を SQL トレースファイルに出力します。

- N

  SQL トレース情報を SQL トレースファイルに出力しません。

なお，このオペランドにN を指定した場合，次に示すオペランドの指定値に関係なく，SQL トレース情報は出力されません。

- サーバ定義adb_sql_trc_param オペランド
- サーバ定義adb_sql_trc_accesspath オペランド
- サーバ定義adb_sql_trc_level オペランド
- サーバ定義adb_sql_trc_txtfile_size オペランド

[36] adb_sql_trc_param = {Y | N}

SQL トレース情報に？パラメタの情報を出力するかどうかを指定します。

- Y

  SQL トレース情報に？パラメタの情報を出力します。

- N

  SQL トレース情報に？パラメタの情報を出力しません。

なお，このオペランドにY を指定しても，サーバ定義adb_sql_trc_out オペランドにN を指定した場合，？パラメタの情報は出力されません。

[37] adb_sql_trc_accesspath = {Y | N}

SQL トレース情報にアクセスパス情報およびアクセスパスの統計情報を出力するかどうかを指定します。

- Y

  SQL トレース情報にアクセスパス情報およびアクセスパスの統計情報を出力します。

- N

  SQL トレース情報にアクセスパス情報およびアクセスパスの統計情報を出力しません。

なお，このオペランドにY を指定しても，サーバ定義adb_sql_trc_out オペランドにN を指定した場合，アクセスパス情報およびアクセスパスの統計情報は出力されません。

[38] adb_sql_trc_level = {SQL | CALL}

SQL トレース情報を出力する単位を指定します。

- SQL

  SQL トレース情報を SQL 文単位で出力します。

- CALL

  SQL トレース情報をコール単位で出力します。

[39] adb_sql_trc_txtfile_size = *SQL トレースファイルの最大容量*

～〈整数〉((32～1,024))《256》(単位：メガバイト)

SQL トレースファイル 1 つ当たりの最大容量をメガバイト単位で指定します。SQL トレースファイルは 8 つ作成されます。

通常は，このオペランドの指定は不要です。SQL トレースファイルの最大容量を変更したい場合に，このオペランドを指定してください。

なお，このオペランドで指定した SQL トレースファイルの最大容量は，HADB サーバを起動したときに有効となります。ただし，このオペランドの指定値を小さくした場合，HADB サーバを起動した直後には，このオペランドの指定値よりも容量が大きい SQL トレースファイルが存在することがありま

す。この場合は，SQL トレースファイルの切り替えが発生して，対象の SQL トレースファイルに SQL トレース情報が出力されるときに，このオペランドで指定した SQL トレースファイルの最大容量が有効となります。

[40] adb_sql_order_mode = {BYTE | ISO}

ORDER BY 句を指定したSELECT 文の検索結果を並び替える際の，文字データの並び替え順序を指定します。このオペランドを省略した場合は，BYTE が仮定されます。

- BYTE

  文字データをバイトコード順に並び替えます。

- ISO

  文字データをソートコード順（ISO/IEC14651:2011 準拠）に並び替えます。

  ただし，環境変数ADBLANG にSJIS を指定している場合は，このオペランドにISO を指定できません。

[マルチノード機能]

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。

なお，文字データの並び替え順序は，クライアント定義adb_clt_sql_order_mode オペランド，およびコネクション属性でも指定できます。それぞれ指定した場合は，次の表に示す優先順位に従って指定が決まります（優先順位の番号が小さいほど，優先順位が高くなります）。

表 7-5　文字データの並び替え順序の優先順位

優先順位	文字データの並び替え順序の指定個所
1	コネクション属性
2	クライアント定義adb_clt_sql_order_mode オペランド
3	サーバ定義adb_sql_order_mode オペランド

[説明]

例えば，サーバ定義adb_sql_order_mode オペランドにISO を指定し，クライアント定義 adb_clt_sql_order_mode オペランドにBYTE を指定した場合，AP から実行するSELECT 文（ORDER BY 句を指定）には，BYTE が適用されます。

なお，上記の表中のすべての個所で指定を省略した場合は，BYTE が仮定されます。

[41] adb_sql_prep_dec_div_rs_prior = {INTEGRAL_PART | FRACTIONAL_PART}

SQL 文中に指定した除算（四則演算）の結果のデータ型がDECIMAL 型の場合，除算結果の位取りの最小値を指定します。

INTEGRAL_PART：

除算結果の位取りの最小値を 0 とします。INTEGRAL_PART を指定した場合，整数部の桁数が優先されます。そのため，除算結果が大きな数値となる可能性がある SQL 文を実行する際に，オーバフローエラーの発生を極力回避したい場合は，INTEGRAL_PART を指定してください。

FRACTIONAL_PART：

除算の第 1 演算項（被除数）の位取りを，除算結果の位取りの最小値とします。FRACTIONAL_PART を指定した場合，第 1 演算項の小数部の桁数が，除算結果の小数部の桁数の最小値になります。

第 1 演算項をDECIMAL(*p1*,*s1*)，第 2 演算項をDECIMAL(*p2*,*s2*)とした場合，除算結果はDECIMAL(38,*s*)となります。

- INTEGRAL_PART を指定した場合

  $s = \mathrm{MAX}\ \{0,\ 38 - (p1 - s1 + s2)\}$

- FRACTIONAL_PART を指定した場合

  $s = \mathrm{MAX}\ \{s1,\ 38 - (p1 - s1 + s2)\}$

このオペランドの指定値による除算結果の違いの例を次に示します。

(例)

表T1

C1列 DECIMAL(38,4)	C2列 DECIMAL(10,5)
30.5256	0.05223

次のSELECT 文を実行したとします。

```
SELECT "C1"/"C2" AS "除算結果" FROM "T1"
```

- INTEGRAL_PART を指定した場合の除算結果

  584.

- FRACTIONAL_PART を指定した場合の除算結果

  584.4457

ビュー表を検索したときの除算結果の位取りの例については，マニュアル『HADB　SQL リファレンス』の『除算結果のデータ型が DECIMAL 型の場合の留意事項』を参照してください。

[マルチノード機能]

マルチノード機能を使用する場合は，全ノードの HADB サーバでこのオペランドの指定値を同じにしてください。

[42] adb_sql_default_dbarea_shared = *表またはインデクスを格納するデフォルトのデータ用 DB エリア名*

～〈文字列〉((1～30 バイト))

データ用 DB エリア名を指定します。ここで指定したデータ用 DB エリアは，次の SQL 文で，表またはインデクスを格納する DB エリアの指定を省略したときに，表またはインデクスを格納するデフォルトの DB エリアになります。

- CREATE TABLE 文
- CREATE INDEX 文
- ALTER TABLE 文

DB エリア名を指定する際の名称規則については，マニュアル『HADB　SQL リファレンス』の『名前の指定』を参照してください。

なお，次のどちらかに該当する場合，HADB サーバの開始時に警告メッセージが出力されます。警告の内容を対処すると，このオペランドの指定値が有効になります。

- このオペランドにデータ用 DB エリアではない DB エリアを指定した場合
- このオペランドに存在しない DB エリアを指定した場合

[マルチノード機能]

マルチノード機能を使用する場合は，プライマリノードおよびセカンダリノードの HADB サーバでこのオペランドの指定値を同じにしてください。

ワーカーノードの HADB サーバには，このオペランドを指定する必要はありません。

7.2.6　レンジインデクスに関するオペランド（set 形式）

[43]　adb_sql_rngidx_preread = {ALL | NO | *レンジインデクス名*〔, *レンジインデクス名*〕…}

HADB サーバの開始時に一括先読みするレンジインデクスを指定します。レンジインデクスの一括先読みについては，「5.5.3　レンジインデクスの一括先読み」を参照してください。

ALL：

全レンジインデクスを一括先読みします。

NO：

レンジインデクスの一括先読みを行いません。

レンジインデクス名：

～〈文字列〉((1～205 バイト))

一括先読みするレンジインデクスを指定します。レンジインデクス名は 8,192 個まで指定できます。レンジインデクスの指定順に一括先読みされます。

> **重要**
>
> レンジインデクス名は，「*スキーマ名.インデクス識別子*」の形式で指定してください。スキーマ名を省略することはできません。
>
> （例）¥"ADBUSER01¥".¥"RINDX01¥"

留意事項を次に示します。

- レンジインデクスを一括先読みするには，adbbuff オペランドの-a オプション（レンジインデクス専用のグローバルバッファのページ数）の指定が必要です。そのため，一括先読み対象となるレンジインデクスを格納している DB エリアにグローバルバッファを割り当てる際は，必ず-a オプションを指定してください。

- レンジインデクスを一括先読みする際，1 回のページ要求で，複数のページを読み込みます。そのため，adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す 2 つの情報が一致しないことがあります。

  ・DBbuff_page_rng_request_cnt（レンジインデクスバッファのページ要求回数）

  ・DBbuff_page_rng_pagein_cnt（レンジインデクスバッファのページイン回数）

  adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』を参照してください。
- レンジインデクスの一括先読み中にadbbuff オペランドの-a オプションで指定したページが満杯になった場合，一部のレンジインデクスだけが読み込まれた状態になります。
- 未完状態のレンジインデクスは一括先読みされません。
- 指定したレンジインデクス名に構文的な誤りなどがある場合，そのレンジインデクスは一括先読みされません。そのほかのレンジインデクスについては，一括先読みされます。

7.2.7　統計情報に関するオペランド（set 形式）

[44]　adb_sta_log_max_size = *統計ログファイルの最大容量*

〜〈整数〉((1〜16))《4》(単位：ギガバイト※)

統計ログファイル 1 つ当たりの最大容量をギガバイト単位で指定します。統計ログファイルは 4 つ作成されます。

通常は，このオペランドの指定は不要です。統計ログファイルの最大容量を小さくしたい場合に，このオペランドを指定してください。

なお，このオペランドで指定した統計ログファイルの最大容量は，統計ログファイルが切り替わったときに有効となります。そのため，このオペランドを指定しても，統計ログファイルの切り替えが発生するまでは，統計ログファイルの最大容量がこのオペランドの指定値と異なります。

注※

小容量での統計ログファイルの切り替えテストを実施する場合，adb_sta_log_size_unit オペランドで統計ログファイル最大容量の単位を小さくする（メガバイトに変更する）ことができます。

[45]　adb_sta_log_path = *統計ログファイルの出力先ディレクトリ名*

〜〈パス名〉((2〜510 バイト))

統計ログファイルの出力先ディレクトリ名を絶対パスで指定します。

このオペランドで指定したディレクトリ下に，統計ログファイルが出力されます。

通常，このオペランドは指定不要です。このオペランドを省略した場合，統計ログファイルの出力先ディレクトリは，$ADBDIR/spool になります。

$ADBDIR/spool ディレクトリ下以外に，統計ログファイルを出力したい場合に，このオペランドを指定してください。

このオペランドを指定する場合の注意事項を次に示します。

- 統計ログファイルだけを格納する専用のディレクトリを指定してください。

- HADB 管理者に，アクセス権限（読み取り権限，書き込み権限，および実行権限）が付与されたディレクトリを指定してください。
- 十分な空き容量が確保されているディスク上のディレクトリを指定してください。

  adb_sta_log_max_size オペランド，およびadb_sta_log_size_unit オペランドを省略した場合，統計ログファイル 1 つ当たりの最大容量は 4 ギガバイトです。また，統計ログファイルは 4 つ作成されます。そのため，16 ギガバイト以上の容量が必要です。

  > **メモ**
  >
  > 統計ログファイル 1 つ当たりの最大容量は，サーバ定義 adb_sta_log_max_size オペランド，および adb_sta_log_size_unit オペランドで変更できます。

- 統計ログファイルの出力先ディレクトリを変更した場合，変更前の統計ログファイルに格納されている統計情報を参照できなくなります。そのため，統計ログファイルの出力先ディレクトリを変更する前に，adbstat コマンドを実行して統計情報を出力しておいてください。

[46] adb_sta_log_size_unit = {G | M}

adb_sta_log_max_size オペランドで指定した統計ログファイルの最大容量の，単位を切り替える場合に設定します。

通常の運用時には，このオペランドを省略してください。このオペランドは統計ログファイルの切り替えテストなどで，最大容量の単位をメガバイト単位に変更したい場合に使用します。このオペランドを指定していると，警告メッセージKFAA52201-W が出力されます。

- G

  統計ログファイルの最大容量の単位をギガバイトとします。

- M

  統計ログファイルの最大容量の単位をメガバイトとします。

7.2.8 同義語検索に関するオペランド（set 形式）

[47] adb_syndict_storage_path = *同義語辞書ファイルの格納ディレクトリ*

～〈パス名〉((2～510 バイト))

同義語辞書ファイルの格納ディレクトリを絶対パス名で指定します。同義語検索を行う場合は，このオペランドを指定してください。

このオペランドを指定する場合は，「11.16.1 同義語検索を行うための準備をする」の「(3) 同義語辞書ファイルの格納ディレクトリに必要な容量を見積もる」～「(6) サーバ定義を修正する」を参照してください。

[マルチノード機能]

マルチノード機能を使用する場合は，プライマリノードおよびセカンダリノードの HADB サーバでこのオペランドの指定値を同じにしてください。

ワーカーノードの HADB サーバには，このオペランドを指定する必要はありません。

メモ

- このオペランドを省略した状態でadbsyndict コマンドを実行した場合，adbsyndict コマンドがエラーになります。
- このオペランドの指定内容に誤りがあった場合，HADB サーバの開始時にエラーになります。

[48] adb_syndict_node_storage_path = *マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ*

～〈パス名〉((2～510 バイト))

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを絶対パス名で指定します。マルチノード機能の使用時に同義語検索を行う場合に，このオペランドを指定してください。このオペランドを指定する場合は，「16.26.1　同義語検索を行うための準備をする」の「(6)　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する」を参照してください。

重要

このオペランドの指定を変更した場合，adbsyndict コマンドを実行して同義語辞書ファイルの同期を取る必要があります。詳細については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

メモ

- このオペランドを省略した状態でadbsyndict コマンド（同義語辞書の登録または更新，同義語辞書ファイルの同期）を実行した場合，adbsyndict コマンドがエラーになります。
- このオペランドの指定内容に誤りがあった場合，HADB サーバの開始時にエラーになります。
- 次の条件をすべて満たす場合，警告メッセージKFAA51525-W が出力されます。
  - ・マルチノード機能を使用している
  - ・adb_syndict_storage_path オペランドを指定している
  - ・adb_syndict_node_storage_path オペランドを指定していない

7.2.9　監査証跡機能に関するオペランド（set 形式）

[49] adb_audit_log_path = *監査証跡の出力先ディレクトリ*

～〈パス名〉((2～510 バイト))

監査証跡の出力先ディレクトリを絶対パス名で指定します。監査証跡機能を使用する場合，このオペランドは必ず指定してください。

[50] adb_audit_log_max_size = *監査証跡ファイルの最大容量*

～〈整数〉((256～1,024))《256》(単位：メガバイト)

監査証跡ファイル 1 つ当たりの最大容量をメガバイト単位で指定します。

通常，このオペランドの指定は不要です。監査証跡ファイルの最大容量を大きくしたい場合に，このオペランドを指定してください。

なお，JP1/Audit と連携して監査証跡を管理する場合は，このオペランドにはデフォルト値を指定してください。監査証跡ファイルの最大容量が大きいと，JP1/Audit による監査証跡の収集漏れが発生するおそれがあります。

[51] adb_audit_log_max_num = *監査証跡ファイルの最大世代数*

～〈整数〉((0，4～10,000))《0》

監査証跡の出力先ディレクトリ下に作成される監査証跡ファイルの最大世代数を指定します。

このオペランドの指定値が，監査証跡の出力先ディレクトリ下に作成される監査証跡ファイル数の上限値になります。

通常，このオペランドの指定は不要です。作成される監査証跡ファイル数の上限を設定したい場合に，このオペランドを指定してください。監査証跡ファイル数が上限を超えた場合，いちばん古い監査証跡ファイルが削除されます。

このオペランドを省略した場合，またはこのオペランドに0 を指定した場合は，上限なく監査証跡ファイルが作成されます。

また，このオペランドに1～3 を指定した場合は，4 を指定したと仮定されます。その際，KFAA51413-W メッセージが出力されます。

このオペランドの指定値は，HADB サーバを開始したときに有効となります。

重要

このオペランドの指定値を変更する場合は，監査証跡の出力先ディレクトリに格納されている監査証跡ファイル数を確認してください。そして，確認した監査証跡ファイル数よりも大きい値を，このオペランドに指定してください。

監査証跡の出力先ディレクトリに格納されている監査証跡ファイル数よりも小さい値をこのオペランドに指定した場合は，HADB サーバは開始できません（エラーとなります）。

7.2.10 マルチノード機能に関するオペランド（set 形式）

マルチノード機能に関するオペランドは，ノード種別によって指定要否が次のように異なります。

オペランド名	ノード種別		
	プライマリノード	セカンダリノード	ワーカーノード
adb_sys_multi_node_info	○	○	○
adb_sys_multi_worker_node_info※1	×	×	○

オペランド名	ノード種別		
	プライマリノード	セカンダリノード	ワーカーノード
adb_sys_multi_max_worker_nodes※1	○	○	○
adb_sys_node_start_wait_time	△	△	△
adb_sys_max_parallel_exec_num※2	○	○	○

（凡例）

○：指定が必要です。

△：基本的には指定不要です。デフォルト値から指定を変更する場合に指定します。

×：指定不要です。

注※1

ワーカーノードを使用する場合に指定するオペランドです。

注※2

SQL パラレル実行機能を使用する場合に指定するオペランドです。

[52] adb_sys_multi_node_info = *ノードのホスト名*〔:*ノードのポート番号*〕,*ノードのホスト名*〔:*ノードのポート番号*〕〔,*ノードのホスト名*〔:*ノードのポート番号*〕〕…

～〈文字列〉((1～261 バイト))

プライマリノードおよび各セカンダリノードの HADB サーバ間通信で使用する HADB サーバのホスト名，およびポート番号を「*ホスト名*:*ポート番号*」の形式で指定します。このオペランドを指定することで，マルチノード機能が使用できます。マルチノード機能については，「16.　マルチノード機能の運用」を参照してください。

重要

- このオペランドには，プライマリノードと全セカンダリノードのホスト名とポート番号を指定します。ワーカーノードのホスト名とポート番号は指定しません。ワーカーノードのホスト名とポート番号は，adb_sys_multi_worker_node_info オペランドで指定します。
- このオペランドは，全ノードのサーバ定義に指定してください。また，指定内容は同じにしてください。
- HADB サーバ間で使用するネットワークと，HA モニタの監視パスのネットワークは，必ず共通のネットワークを使用してください。詳細については，「16.2.3　ネットワーク構成」の「(2)　ノード間ネットワーク」を参照してください。

指定規則を次に示します。

- *ノードのホスト名*〔:*ノードのポート番号*〕は，4 個まで指定できます。
- このオペランドには，必ず自ノード（このサーバ定義を指定して開始する HADB サーバ）のホスト名，およびポート番号も指定してください。

- マルチノード機能で使用する全ノードに対して，このオペランドには同じ値を指定してください。その際，指定順も同じにしてください。
- HADB サーバ間通信で使用するポート番号は省略できます。指定を省略した場合は，23651 が仮定されます。

このオペランドの指定例を次に示します。この例では，セカンダリノードを 2 つ設定するものとします。

```
set adb_sys_multi_node_info = 172.16.0.11:23651, ¥  ...1
                              172.16.0.12:23651, ¥  ...2
                              172.16.0.13:23651     ...3
```

[説明]

1. プライマリノードのホスト名とポート番号を指定します。
2. 1 つ目のセカンダリノードのホスト名とポート番号を指定します。
3. 2 つ目のセカンダリノードのホスト名とポート番号を指定します。

[53] adb_sys_multi_worker_node_info = *ワーカーノードのホスト名*〔:*ワーカーノードのポート番号*〕

～〈文字列〉((1～261 バイト))

ワーカーノードが HADB サーバ間通信で使用する HADB サーバのホスト名，およびポート番号を「*ホスト名*:*ポート番号*」の形式で指定します。このオペランドを指定したノードが，ワーカーノードになります。

重要

- このオペランドは，ワーカーノードのサーバ定義に指定してください。
- このオペランドには，自ワーカーノード（このサーバ定義を指定するワーカーノード）のホスト名とポート番号だけを指定します。

留意事項を次に示します。

- ワーカーノードのポート番号の指定は省略できます。指定を省略した場合は，23651 が仮定されます。
- マルチノード機能を使用しない場合，このオペランドの指定値は無効になります。

[54] adb_sys_multi_max_worker_nodes = *ワーカーノードの最大数*

～〈符号なし整数〉((1～8))

マルチノード構成に追加できるワーカーノードの最大数を指定します。このオペランドには，次の計算式で求めた値を指定してください。

adb_sys_multi_max_worker_nodesオペランドの指定値
＝ *A* －（*adb_sys_multi_node_infoオペランドに指定したノード数*－1）

A：システムの負荷がピークのときに必要となるノード数

重要

- このオペランドは，全ノードのサーバ定義に指定してください。また，指定値は同じにしてください。

- このオペランドを省略した場合，マルチノード構成にワーカーノードを追加することはできません。

なお，マルチノード機能を使用しない場合，このオペランドの指定値は無効になります。

[55] adb_sys_node_start_wait_time = *他ノードの起動完了待ち時間の上限*

～〈整数〉((1～7,200))《300》(単位：秒)

マルチノード機能を使用する場合で，各ノードの起動が完了するまでの待ち時間の上限を，秒単位で指定します。

このオペランドには，基本的にデフォルト値を指定してください。

[56] adb_sys_max_parallel_exec_num = *SQL パラレル実行機能を適用する検索系 SQL の最大同時実行数*

～〈符号なし整数〉((0～サーバ定義 adb_sys_max_users オペランドの値))《サーバ定義 adb_sys_max_users オペランドの値》

SQL パラレル実行機能を適用する検索系 SQL の最大同時実行数を指定します。SQL パラレル実行機能を適用した検索系 SQL の同時実行数がこのオペランドの指定値を超えると，新たに実行しようとする SQL パラレル実行機能を適用した検索系 SQL はエラーになります。

重要

- このオペランドは，全ノードのサーバ定義に指定してください。また，指定値は同じにしてください。
- このオペランドに0 を指定した場合，次の指定をしても検索系 SQL の実行時に SQL パラレル実行機能は適用されません。
  - クライアント定義のadb_clt_sql_parallel_exec オペランドにY を指定する
  - SQL 文に SQL パラレル実行指定を指定する

次に示すどちらかの条件を満たす場合は，このオペランドを省略してデフォルト値を適用することを推奨します。

- SQL パラレル実行機能を使用しない場合
- SQL パラレル実行機能を適用した検索系 SQL の最大同時実行数が明確にわからない場合

なお，マルチノード機能を使用しない場合，このオペランドの指定値は無効になります。

7.2.11 グローバルバッファに関するオペランドおよびオプション（コマンド形式）

[57] adbbuff

データ用 DB エリアに割り当てるグローバルバッファの内容を定義します。このオペランドは，複数指定できます。

このオペランドを指定する場合，次に示す 2 つの変数の合計値が，「6.3　HADB サーバのメモリ所要量の見積もり」で求めた HADB サーバが使用するメモリの最大サイズの 3 割～4 割程度の値となるよう，見積もってください。

- 「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」の変数*SHM_BUFGLOBAL*
- 「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時）」の「(d)　変数 BUFGLOBAL の求め方」の変数*BUFGLOBAL*

なお，マスタディレクトリ用 DB エリア，ディクショナリ用 DB エリア，システム表用 DB エリアおよび作業表用 DB エリアには，自動的にグローバルバッファが割り当てられるため，このオペランドで定義する必要はありません。

また，このオペランドで割り当てたグローバルバッファの内容については，`adbls -d gbuf` コマンドで確認できます。`adbls -d gbuf` コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d gbuf（グローバルバッファの情報表示）』を参照してください。

メモ

作業表用 DB エリアに割り当てられるバッファには，グローバル作業表用のグローバルバッファ，およびローカル作業表用バッファがあります。指定方法を次に示します。

- **グローバル作業表用のグローバルバッファの指定方法**

  「7.2.2　性能に関するオペランド（set 形式）」のサーバ定義 adb_dbbuff_wrktbl_glb_blk_num オペランドで，グローバル作業表用のグローバルバッファのページ数を指定できます。グローバル作業表用のグローバルバッファ名は，`ADBWRK` になります。

- **ローカル作業表用バッファの指定方法**

  「7.2.2　性能に関するオペランド（set 形式）」のサーバ定義 adb_dbbuff_wrktbl_clt_blk_num オペランドで，ローカル作業表用バッファのページ数を指定できます。

  また，クライアント定義`adb_dbbuff_wrktbl_clt_blk_num` オペランド，およびエクスポートオプション`adb_export_wrktbl_blk_num` でも，ローカル作業表用バッファのページ数を指定できます。

  クライアント定義`adb_dbbuff_wrktbl_clt_blk_num` オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。エクスポートオプション`adb_export_wrktbl_blk_num` については，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート）』の『adbexport コマンドの指定形式』の『エクスポートオプションの形式』を参照してください。

`-g` *グローバルバッファ名*

～〈グローバルバッファ名〉((1～30 バイト))

HADB サーバ内で一意なグローバルバッファ名を指定します。

グローバルバッファ名の名称規則を次に示します。

- 指定できる文字は，半角数字，半角英大文字，半角英小文字，半角カタカナ文字，半角空白，半角下線（_），および半角ハイフン（-）になります。また，全角文字も指定できます。ただし，全角空白は指定できません。
- 半角文字と全角文字を混在させて指定できます。
- 半角英大文字および半角英小文字は，常に区別して扱われます。
- 先頭の文字には，半角英大文字，半角英小文字，半角カタカナ大文字（ア～ン，ヲ）を指定してください。または，全角文字を指定してください。
- 最後の文字に，半角空白は指定できません。
- 半角空白と半角ハイフン（-）を含む場合は，まず，¥”（バックスラッシュと二重引用符）でグローバルバッファ名を囲んでください。次に，それらを”（二重引用符）で囲んでください。

  指定例

```
adbbuff -g ”¥”Sample Buffer01¥”” -n AREA01 -p 1024
adbbuff -g ”¥”Sample-Buffer02¥”” -n AREA02 -p 1024
```

なお，このオプションに指定するグローバルバッファ名には，「##ADBOTHER」を含む名称を指定しないでください。「##ADBOTHER」を含む名称は，自動的にグローバルバッファを割り当てるときに HADB サーバが使用することがあります。そのため，「##ADBOTHER」を含む名称を指定した場合，グローバルバッファ名で区別できなくなるおそれがあります。

ヒント

■グローバルバッファの割り当て方法（-n オプション，-o オプション）の使い分け

グローバルバッファを割り当てる DB エリアを指定する場合，通常は-n オプションで指定してください。-n オプションで指定していないすべての DB エリアに一括でグローバルバッファを割り当てたい場合，-o オプションを指定してください。

-n *DB エリア名*〔，*DB エリア名*〕…

～〈文字列〉((1～30 バイト))

グローバルバッファを割り当てる DB エリア名を指定します。

DB エリア名の名称規則については，マニュアル『HADB　SQL リファレンス』の『名前の指定』を参照してください。

このオプションを指定するときの注意事項を次に示します。

- **DB エリア名を複数指定する場合**

  このオプションに DB エリア名を複数指定した場合，1 つのグローバルバッファに複数の DB エリアを割り当てられます。DB エリア名を複数指定する場合，ページサイズが同じ DB エリアだけを指定してください。ページサイズが異なる DB エリアを指定した場合，エラーとなります。

- **カラムストア表が格納されているデータ用 DB エリアを指定する場合**

  カラムストア表が格納されているデータ用 DB エリアを指定する場合，このオプションにそのデータ用 DB エリア名だけを指定したadbbuff オペランドを定義することを推奨します。つまり，カラ

ムストア表が格納されているデータ用 DB エリアに対して，専用のグローバルバッファを割り当てるようにしてください。

1 つのグローバルバッファに対して，カラムストア表が格納されているデータ用 DB エリア以外に，ローストア表やインデクスが格納されているデータ用 DB エリアなども一緒に割り当てた場合，ローストア表の検索性能やインデクスを使用した検索性能が低下するおそれがあります。

- **ディクショナリ用 DB エリアを指定する場合**

  ディクショナリ用 DB エリアにグローバルバッファを割り当てるようメッセージが出力された場合には，このオプションにADBDIC だけを指定したadbbuff オペランドを新たに定義してください。通常は，このオプションにディクショナリ用 DB エリアを指定する必要はありません。

- **システム表用 DB エリアを指定する場合**

  システム表用 DB エリアにグローバルバッファを割り当てるようメッセージが出力された場合には，このオプションにADBSTBL だけを指定したadbbuff オペランドを新たに定義してください。通常は，このオプションにシステム表用 DB エリアを指定する必要はありません。

なお，-n オプションでグローバルバッファを割り当てなかった DB エリアについては，自動的に-o オプションを指定したときと，同様のグローバルバッファが割り当てられます。このとき-g オプションには，「##ADBOTHER」が指定されます。

-o

-n オプションで指定していないすべての DB エリアにグローバルバッファを割り当てます。

このオプションを指定したadbbuff オペランドは 1 つだけ定義できます。

このオプションを指定した場合，同じページサイズの DB エリアごとにまとめられて，それぞれに 1 つのグローバルバッファが割り当てられます。各グローバルバッファ名は，「-g *グローバルバッファ名*」＋「#*nnnnnnnnnn*」（*n* はページサイズを 10 桁の 10 進数で表現したもの）になります。例を次に示します。

(例)

- adbbuff オペランドの指定

```
adbbuff -g gbuf01 -o
```

- -n オプションでグローバルバッファを割り当てていない DB エリア

  ・ページサイズが 4 キロバイトの DB エリア：ADBDIC，ADBSTBL，ADBU00011，ADBU00012

  ・ページサイズが 32 キロバイトの DB エリア：ADBU00013

[説明]

この場合，HADB サーバによって，次のように 2 つのグローバルバッファが作成されます。

- gbuf01#0000004096

  上記のグローバルバッファには，ADBDIC，ADBSTBL，ADBU00011，およびADBU00012 が割り当てられます。

- gbuf01#0000032768

  上記のグローバルバッファには，ADBU00013 が割り当てられます。

なお，-n オプションですべての DB エリアにグローバルバッファを割り当てた場合，このオプションを指定しても無視されます。

-p *グローバルバッファのページ数*

～〈整数〉((0～2,147,483,647))《0》

DB エリアに割り当てるグローバルバッファのページ数を指定します。

このオプションを省略した場合，またはこのオプションに0 を指定した場合は，次に示す計算式で求めた値が仮定されます。

計算式

グローバルバッファのページ数＝
(*adb_sys_max_usersオペランドの指定値*＋*adb_sys_rthd_numオペランドの指定値*
＋10＋*multi_num*＋*worker_num*)
×（*adb_sys_uthd_numオペランドの指定値*＋2）

変数の説明

- *multi_num*

  マルチノード機能を使用する場合に以下の値を加算します。

  （*adb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）
  ×（*adb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）＋5

- *worker_num*

  マルチノード機能のワーカーノードを使用する場合に以下の値を加算します。

  - adbbuff オペランドを指定するノードがプライマリノードまたはセカンダリノードの場合

    adb_sys_multi_max_worker_nodesオペランドの指定値
    ×（*adb_sys_max_parallel_exec_numオペランドの指定値*×2＋2）

  - adbbuff オペランドを指定するノードがワーカーノードの場合

    adb_sys_multi_max_worker_nodesオペランドの指定値
    ×*adb_sys_max_parallel_exec_num オペランドの指定値*×2＋2

-a *レンジインデクス専用のグローバルバッファのページ数*

～〈整数〉((0～2,147,483,647))

このオプションは，レンジインデクスに関するオプションです。

レンジインデクス専用のグローバルバッファのページ数を指定します。

次に示す計算式から求めた値を指定することを推奨します。なお，マルチチャンク表の場合は，チャンクごとに次に示す計算式で値を求めてください。そして，求めた値を合計した値をこのオプションに指定してください。

計算式

$$\sum_{i=1}^{rngidx_num}\left(SGRI(i)\times SEGSIZE(i)+MIN\left(SAUSEDPAGENUM(i)\ ,\ SEGSIZE(i)\right)+\left\downarrow\frac{SAUSEDPAGENUM(i)}{SEGSIZE(i)}\right\downarrow\times SEGSIZE(i)\right)$$

rngidx_num

このadbbuff オペランドに指定した DB エリアに格納されるレンジインデクス数

SGRI(i)

「5.8.6　レンジインデクスを格納するセグメント数の求め方」の「(1)　変数 SGRI の求め方」を参照してください。

SEGSIZE(i)

各レンジインデクスが格納される DB エリアのセグメントサイズ（単位：ページ数）

次に示す計算式で求めてください。

SEGSIZE(i)＝ 1セグメントの容量（単位：バイト）÷*page_size(i)*

page_size(i)

各レンジインデクスが格納される DB エリアのページサイズ（単位：バイト）

SAUSEDPAGENUM(i)

各レンジインデクスが使用しているページ数（単位：ページ）

次に示す計算式で求めてください。

$$SAUSEDPAGENUM(i)=\left\uparrow\frac{PTNUM(i)\times 65}{(page_size(i)-64)\times 8}\right\uparrow+1$$

PTNUM(i)

各レンジインデクスが管理しているポインタの数（単位：個）

次に示す計算式で求めてください。

$$PTNUM(i)=\ tbl_dbareafile_num(i)+\left\downarrow\frac{tbl_dbarea_initsize(i)}{16,384}\right\downarrow\times 8,192$$

tbl_dbareafile_num(i)

各レンジインデクスを定義した表が格納された DB エリアファイルの数（単位：個）

tbl_dbarea_initsize(i)

各レンジインデクスを定義した表が格納された DB エリアの初期確保サイズ（単位：ギガバイト）

対象の DB エリアに対して，次に示すコマンドを実行したときに，初期確保サイズのオプションに指定した値を代入してください。

- adbinit コマンドの初期設定オプションadbinitdbarea の-i オプションに指定した値
- adbmodarea コマンドの DB エリア追加・変更オプションadbaddarea の-i オプションに指定した値

- adbmodarea コマンドの DB エリア追加・変更オプションadbexpandarea の-i オプションに指定した値

> **メモ**
>
> - adbinit コマンドの初期設定オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』の『adbinit コマンドの指定形式』の『初期設定オプションの形式』を参照してください。
> - adbmodarea コマンドの DB エリア追加・変更オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』の『adbmodarea コマンドの指定形式』の『DB エリア追加・変更オプションの形式』を参照してください。

留意事項を次に示します。

- -a オプションのグローバルバッファのページは，-p オプションのグローバルバッファのページとは別に確保されます。
- -a オプションを指定した場合，レンジインデクスは-p オプションで指定したグローバルバッファのページを使用しません。
- -a オプションを省略した場合，レンジインデクスは-p オプションで指定したグローバルバッファのページを使用します。

-v *テーブルスキャンバッファに使用するメモリサイズ*〔,*リアルスレッドごとに使用するメモリサイズの上限値*〕

このオプションを指定すると，テーブルスキャンの処理速度を向上させるためのテーブルスキャンバッファが確保されます。

確保されたテーブルスキャンバッファは，次のセグメントに対して適用されます。

- 基本行用セグメント（ローストア形式の表の場合）
- カラムデータ用セグメント（カラムストア形式の表の場合）

上記以外のセグメントに対しては適用されません。

> **重要**
>
> バージョン 04-01 以降の HADB サーバで，テーブルスキャンバッファを確保したい場合は，このオプションを指定してください。-k オプションは指定しないでください。
>
> また，このオプションと-k オプションの指定をどちらも省略した場合，テーブルスキャンバッファは確保されません。

テーブルスキャンバッファに使用するメモリサイズ：

～〈整数〉((50～100,000,000))（単位：メガバイト）

テーブルスキャン実行時に使用されるテーブルスキャンバッファに割り当てるメモリサイズを，メガバイト単位で指定してください。

HADB サーバの起動時に，このオプションの指定値に従って，テーブルスキャンバッファに割り当てるメモリサイズが確保されます。

テーブルスキャンバッファは，テーブルスキャンの処理速度を向上させるためのバッファです。テーブルスキャンバッファに，データベース上の複数ページを一括読み込みすると，I/O 回数を削減できます。HADB サーバは，読み込んだテーブルスキャンバッファ上のページに対して，グローバルバッファと同様にページ単位でアクセスします。

このオプションを指定すると，テーブルスキャンバッファが確保されます。そのため，テーブルスキャン実行時にグローバルバッファではなく，テーブルスキャンバッファが使用されます。テーブルスキャンバッファが使用されると，次の SQL 文の実行速度が向上する可能性があります。

- 単純なテーブルスキャンとなる SQL 文
- 最も外側の問合せ指定がテーブルスキャンとなるような SQL 文

なお，テーブルスキャン実行時に確保したテーブルスキャンバッファが不足した場合は，グローバルバッファが使用されます。

また，SQL 文の実行中に，テーブルスキャンが複数回実行される場合は，その SQL 文を実行するリアルスレッドが切り出したメモリが再利用されます。例えば，SQL 文の実行時に，複数の表に対してテーブルスキャンが実行される場合などが該当します。各 SQL 文を実行するリアルスレッドが切り出したメモリは，カーソルのクローズ時に返却されます。

このオプションを指定する場合，次に示す 2 つの計算式から求めた「最小メモリサイズ」以上，「最大メモリサイズ」以内の値を指定してください。

計算式（-v オプションのテーブルスキャンバッファに使用するメモリサイズの指定値）

```
最小メモリサイズ ≦ A ≦ 最大メモリサイズ
```

A＝ -vオプションの*テーブルスキャンバッファに使用するメモリサイズ*の指定値
最小メモリサイズ＝ MAX（↑sql_rthd_num×uthd_num÷4↑ , 128）×MAX_SEGSIZE_MB
最大メモリサイズ＝ MAX（rthd_num×uthd_num×table_num×3 , 128）×MAX_SEGSIZE_MB

変数の説明

sql_rthd_num
　サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値

uthd_num
　サーバ定義adb_sys_uthd_numオペランドの指定値

rthd_num
　サーバ定義adb_sys_rthd_numオペランドの指定値

table_num
　adbbuffオペランドの-nオプションに指定したDBエリア名に定義されている表の総数

MAX_SEGSIZE_MB
　adbbuffオペランドの-nオプションに指定したDBエリアの1セグメントの容量（単位：メガバイト）
　adbbuffオペランドの-nオプションに複数のDBエリアを指定する場合は，
　1セグメントの容量の最大値を代入してください。

メモ

このオプションに指定したメモリサイズには，次の 2 つが含まれます。

- データベースからの読み込みに使用するグローバルバッファページ
- テーブルスキャンバッファの管理に使用するプロセス共通メモリ

データベースからの読み込みに使用するグローバルバッファページのページサイズについては，「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(2)　グローバルバッファページの所要量の求め方（HADB サーバ開始時）」の変数 *SCANPAGES* を参照してください。

リアルスレッドごとに使用するメモリサイズの上限値：

～〈整数〉((50～100,000,000))（単位：メガバイト）

各リアルスレッドが使用できるテーブルスキャンバッファのメモリサイズの上限を，メガバイト単位で指定してください。

指定を省略した場合，メモリサイズの上限は設定されません。

このオプションを指定すると，各 SQL 文を実行するリアルスレッドが切り出したテーブルスキャンバッファのメモリサイズが上限に達した場合，上限以降はテーブルスキャンバッファではなくグローバルバッファが使われます。

同時に実行するテーブルスキャンを含む SQL 文の数がわからない場合には，より効率的にテーブルスキャンバッファを使えるようにするために，指定を省略することを推奨します。

同時に実行するテーブルスキャンを含む SQL 文の数がわかっており，それらの SQL 文の間で均等にテーブルスキャンバッファが使われるようにしたい場合は，次の計算式で求めた値を指定してください。

計算式（-v オプションの*リアルスレッドごとに使用するメモリサイズの上限値*に指定する値）

```
-vオプションのリアルスレッドごとに使用するメモリサイズの上限値に指定する値＝
    -vオプションのテーブルスキャンバッファに使用するメモリサイズの指定値÷
     （同時に実行するテーブルスキャンを含むSQL文の数
      ×サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値）
```

-k *セグメントを一括読み込みするためのテーブルスキャンバッファのバッファ面数*

～〈整数〉((1～20,000,000))

重要

バージョン 04-01 以降の HADB サーバでは，このオプションは指定不要です。テーブルスキャンバッファを確保したい場合は，-v オプションを指定してください。

このオプションを指定した場合，-v オプションは指定できません。どちらも指定した場合，HADB サーバを開始できません（KFAA51223-E メッセージが出力されます）。

このオプションと-v オプションの指定をどちらも省略した場合，テーブルスキャンバッファは確保されません。

また，このオプションの指定値が 12 以下の場合，テーブルスキャンバッファは確保されません（KFAA51204-W メッセージが出力されます）。

テーブルスキャン実行時に，セグメントを一括読み込みするためのテーブルスキャンバッファのバッファ面数を指定します。テーブルスキャンバッファの 1 面のサイズは 4 メガバイトです。

テーブルスキャンバッファは，テーブルスキャンの処理速度を向上させるためのバッファです。テーブルスキャンバッファに，データベース上の複数のページをセグメント単位に一括読み込みすると，I/O回数を削減できます。これによって，処理速度が向上する可能性があります。

HADB サーバは，セグメント単位で読み込んだテーブルスキャンバッファ上のページに対して，グローバルバッファと同様にページ単位でアクセスします。

このオプションを指定すると，テーブルスキャンバッファを確保します。そのため，テーブルスキャン実行時にグローバルバッファではなく，テーブルスキャンバッファを使用します。テーブルスキャンバッファを使用すると，次の SQL 文の実行速度が向上する可能性があります。

- 単純なテーブルスキャンとなる SQL 文
- 最も外側の問合せ指定がテーブルスキャンとなるような SQL 文

このオプションを指定する場合，次に示す 2 つの計算式から求めた「最小バッファ面数」以上，「最大バッファ面数」以内の値を指定してください。

計算式（-k オプションの指定値）

```
最小バッファ面数 ≦ -kオプションの指定値 ≦ 最大バッファ面数

最小バッファ面数＝ MAX（↑sql_rthd_num×uthd_num÷16↑ , 32）×MAX_SEGSIZE_MB
最大バッファ面数＝ MAX（↑rthd_num×uthd_num×table_num×3÷4↑ , 32）×MAX_SEGSIZE_MB

変数の説明
   sql_rthd_num
       サーバ定義adb_sql_exe_max_rthd_numオペランドの指定値
   uthd_num
       サーバ定義adb_sys_uthd_numオペランドの指定値
   rthd_num
       サーバ定義adb_sys_rthd_numオペランドの指定値
   table_num
       adbbuffオペランドの-nオプションに指定したDBエリア名に定義されている表の中で，
       テーブルスキャンの対象となっている表の総数
   MAX_SEGSIZE_MB
       adbbuffオペランドの-nオプションに指定したDBエリアの1セグメントの容量（単位：メガバイト）
       adbbuffオペランドの-nオプションに複数のDBエリアを指定する場合は，
       1セグメントの容量の最大値を代入してください。
```

また，このオプションを指定する場合，次に示す計算式を満たすようにサーバ定義を指定することを推奨します。左辺と右辺の値が，できるだけ同じ値に近づくよう調整してください。

計算式

```
RTHD ≦ MAXUSER×SQLTHD

変数の説明
   RTHD：     adb_sys_rthd_numオペランドの指定値
   MAXUSER：  adb_sys_max_usersオペランドの指定値
   SQLTHD：   adb_sql_exe_max_rthd_numオペランドの指定値
```

なお，テーブルスキャン実行時に確保したテーブルスキャンバッファが不足した場合は，グローバルバッファが使用されます。テーブルスキャンバッファが十分に確保されているかどうかについては，adbstat コマンドで確認できます。詳細については，「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」を参照してください。

また，テーブルスキャンについては，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『表の検索方式』を参照してください。

7.2.12　クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）

クライアントグループ機能に関するオペランドおよびオプションの内容について説明します。

クライアントグループ機能では，次に示すグループを設定できます。

- クライアントグループ
- コマンドグループ

各グループを設定する際には，サーバ定義adbcltgrp オペランドを指定します。クライアントグループを設定したadbcltgrp オペランドは複数指定できます。ただし，コマンドグループを設定したadbcltgrp オペランドについては，1 つだけ指定できます。

メモ

adbcltgrp オペランドで設定したグループについては，adbls -d cltgrp コマンドで確認できます。adbls -d cltgrp コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cltgrp（クライアントグループ・コマンドグループの情報表示)』を参照してください。

adbcltgrp オペランドの指定数の上限値は，次に示す計算式で求めた値になります。

計算式（単位：個）

```
サーバ定義adbcltgrpオペランドの指定数※の上限値＝
            サーバ定義adb_sys_max_usersオペランドの指定値＋1

注※
  クライアントグループを設定したadbcltgrpオペランド，および
  コマンドグループを設定したadbcltgrpオペランドの数の合計値です。
```

[マルチノード機能]

マルチノード機能を使用する場合，全ノードに対して，このオペランドには同じ値を指定してください。ただし，-r オプションおよび-e オプションは，ノードごとに異なる値を指定できます。ノードごとに異なる値を指定する場合は，サーバ定義adb_sys_rthd_num オペランドの指定値に従って，-r オプションおよび-e オプションの指定値を決定してください。

メモ

- クライアントグループおよびコマンドグループについては，「2.12.1　クライアントグループ機能の概要」を参照してください。

- サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set 形式）」の adb_sys_max_users オペランドを参照してください。
- サーバ定義adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドを参照してください。

ここでは，設定するグループごとに，サーバ定義adbcltgrp オペランドおよびオプションの内容を説明します。

[58] adbcltgrp（クライアントグループを設定する場合）

クライアントグループを設定します。クライアントグループを設定することで，グループに所属するHADB クライアントが使用する接続数や処理リアルスレッド数を指定できます。

-g *クライアントグループ名*

～〈文字列〉((1～30 バイト))

HADB サーバ内で一意なクライアントグループ名を指定します。

指定できる文字は，半角数字，半角英大文字，半角英小文字，または半角下線（_）になります。なお，先頭の文字には，半角英大文字，半角英小文字，または半角下線（_）のどれかを指定してください。

このオプションで指定したクライアントグループ名は，クライアント定義adb_clt_group_name オペランドに指定する必要があります。

> **メモ**
>
> クライアント定義adb_clt_group_name オペランドについては，マニュアル『HADB　AP 開発ガイド』の『システム構成に関するオペランド』を参照してください。

-m *設定したクライアントグループの最大同時接続数*

～〈整数〉((0～1,024))《adb_sys_max_users の指定値－ほかの adbcltgrp の-u オプションの指定値の合計》

設定したクライアントグループの最大同時接続数を指定します。

このオプションに指定した値が，設定したグループで使用できる接続数の上限になります。指定値を超えて，設定したグループが接続することはありません。

このオプションを省略した場合，次に示す計算式で求めた値が仮定されます。

計算式

```
A－B

  変数の説明
    A：サーバ定義adb_sys_max_usersオペランドの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-uオプションの指定値の合計※

  注※
    自グループ以外でサーバ定義adbcltgrpオペランドを指定していない場合，0になります。
```

また，このオプションの指定値が，上記の計算式で求めた値より大きい場合も，上記の計算式で求めた値が仮定されます。

サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set形式）」の adb_sys_max_users オペランドを参照してください。

> **重要**
>
> このオプションに0 を指定した場合，対象のグループに所属する HADB クライアントは，HADB サーバに接続できません。

-u *設定したクライアントグループの接続保証数*

～〈整数〉((0～1,024))《0》

設定したクライアントグループの接続保証数を指定します。

このオプションの指定値に従って，対象のグループが最低限使用する接続数を常に確保します。そのため，自グループ以外の HADB クライアントやコマンドの接続状況に影響されることなく，指定した分だけ常に接続できます。

このオプションには，-m オプションの指定値以下の値を指定してください。また，次の計算式を満たす値を指定してください（これらの条件を満たさない値を指定すると，HADB サーバの開始時にエラーとなります）。

計算式

```
A+B ≦ C

  変数の説明
    A：このオプションの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-uオプションの指定値の合計※
    C：サーバ定義adb_sys_max_usersオペランドの指定値

  注※
    自グループ以外で，サーバ定義adbcltgrpオペランドを指定していない場合，0になります。
```

サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set形式）」の adb_sys_max_users オペランドを参照してください。

> **重要**
>
> 「*すべてのサーバ定義 adbcltgrp オペランドの-u オプションの指定値を合計した値*」が，「*サーバ定義 adb_sys_max_users オペランドの指定値*」と同じである場合，次のどちらかに該当する HADB クライアントは，HADB サーバに接続できません。
>
> - 対象の HADB クライアントが，サーバ定義adbcltgrp オペランドの-u オプションに0 を指定しているグループに所属しているとき
> - 対象の HADB クライアントが，グループに所属していないとき

-r *設定したクライアントグループで使用できる処理リアルスレッドの最大数*

～〈整数〉((0～4,096))《adb_sys_rthd_num の指定値－ほかの adbcltgrp の-e オプションの指定値の合計》

設定したクライアントグループで使用できる処理リアルスレッドの最大数を指定します。

このオプションに指定した値が，設定したグループで使用できる処理リアルスレッド数の上限になります。指定値を超えて，設定したグループが処理リアルスレッドを使用することはありません。

このオプションを省略した場合，次に示す計算式で求めた値が仮定されます。

計算式

```
A－B

  変数の説明
    A：サーバ定義adb_sys_rthd_numオペランドの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-eオプションの指定値の合計※

  注※
    自グループ以外で，サーバ定義adbcltgrpオペランドを指定していない場合，0になります。
```

また，このオプションの指定値が，上記の計算式で求めた値より大きい場合も，上記の計算式で求めた値が仮定されます。

サーバ定義adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドを参照してください。

重要

このオプションに0 を指定した場合，対象のグループに所属する HADB クライアントに適用されるadb_sql_exe_max_rthd_num オペランドには，0 が仮定されます。そのため，対象の HADB クライアントは，常にコネクションスレッドだけで SQL 文を実行します。

サーバ定義adb_sql_exe_max_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_max_rthd_num オペランドを参照してください。

クライアント定義adb_sql_exe_max_rthd_num オペランドについては，マニュアル『HADB AP 開発ガイド』の『性能に関するオペランド』を参照してください。

-e *設定したクライアントグループで使用できる処理リアルスレッドの保証数*

～〈整数〉((0～4,096))《0》

設定したクライアントグループで使用できる処理リアルスレッドの保証数を指定します。

このオプションの指定値に従って，対象のグループが最低限使用する処理リアルスレッド数を常に確保します。そのため，自グループ以外の HADB クライアントやコマンドの使用状況に影響されることなく，指定した分だけ常に処理リアルスレッドを使用できます。

このオプションには，-r オプションの指定値以下の値を指定してください。また，次の計算式を満たす値を指定してください（これらの条件を満たさない値を指定すると，HADB サーバの開始時にエラーとなります）。

計算式

```
A+B ≦ C

  変数の説明
    A：このオプションの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-eオプションの指定値の合計※
    C：サーバ定義adb_sys_rthd_numオペランドの指定値

  注※
    自グループ以外で，サーバ定義adbcltgrpオペランドを指定していない場合，0になります。
```

サーバ定義adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドを参照してください。

重要

「*すべてのサーバ定義 adbcltgrp オペランドの-e オプションの指定値を合計した値*」が，「*サーバ定義 adb_sys_rthd_num オペランドの指定値*」と同じである場合，次のどちらかに該当する HADB クライアントは，adb_sql_exe_max_rthd_num オペランドに0 が仮定されます。

- 対象の HADB クライアントが，サーバ定義adbcltgrp オペランドの-e オプションに0 を指定しているグループに所属しているとき
- 対象の HADB クライアントが，グループに所属していないとき

この場合，対象の HADB クライアントは，常にコネクションスレッドだけで SQL 文を実行します。

サーバ定義adb_sql_exe_max_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_max_rthd_num オペランドを参照してください。

クライアント定義adb_sql_exe_max_rthd_num オペランドについては，マニュアル『HADB AP 開発ガイド』の『性能に関するオペランド』を参照してください。

-w *設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕

このオプションを指定すると，-m オプションで指定した最大同時接続数の残り枠が少なくなったときに，警告メッセージKFAA40020-W を出力できます。

なお，-m オプションに0 を指定した場合（0 が仮定された場合も含む），-w オプションの指定は無効になります。

設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機：

～〈整数〉((0～100))《0》(単位：%)

警告メッセージKFAA40020-W の出力契機を，-m オプションの指定値（クライアントグループに適用される最大同時接続数）に対する比率で指定します。例えば，-m オプションに20 を指定している場合に，警告メッセージの出力契機に90 を指定すると，HADB サーバへの接続数が 18（20 の 90%の値）以上になったときに警告メッセージが出力されます。

なお，警告メッセージの出力契機に0 を指定した場合，警告メッセージは出力されません。

重要

ほかのグループと自由利用枠を共有している場合（-m オプションと-u オプションの指定値が異なる場合），ほかのグループによって自由利用枠が使用されていると，警告メッセージが出力される前に，最大同時接続数超過のエラーが発生する可能性があります。

警告メッセージの出力済み状態のリセット契機：

～〈整数〉((0～99))（単位：%）

警告メッセージKFAA40020-W の出力済み状態のリセット契機を，-m オプションの指定値（クライアントグループに適用される最大同時接続数）に対する比率で指定します。警告メッセージは一度出力されると，警告メッセージ出力済み状態となり，HADB サーバへの接続数がある一定の値を下回らないかぎり警告メッセージは再出力されません。

メモ

-w オプションの指定値と警告メッセージの出力の関係を次に示します。

```
adbcltgrp -g ...
          -m 20
          -w 90,70
```

上記のように-m オプションと-w オプションを指定した場合，HADB サーバへの接続数が 18（20 の 90%の値）以上になったときに警告メッセージが出力されます。その後，HADB サーバへの接続数がいったん 17 に下がったあとに再度 18 以上になっても，警告メッセージは出力されません。

HADB サーバへの接続数が，14（20 の 70%の値）以下に下がった場合は，警告メッセージの出力済み状態がリセットされるため，HADB サーバへの接続数が再度 18 以上になったときに警告メッセージが出力されます。

警告メッセージの出力済み状態のリセット契機に関する留意事項を次に示します。

- リセット契機の指定を省略した場合，「*設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機*－30」の値が仮定されます。その値がマイナスになる場合は，0 が仮定されます。
- 「*設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機* ＞ *警告メッセージの出力済み状態のリセット契機*」となるように指定してください。この条件を満たさない場合，リセット契機の指定は無効になります（警告メッセージKFAA41000-W が出力されます）。このとき，リセット契機には，「*設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機*－30」の値が仮定されます。
- 「*設定したクライアントグループの最大同時接続数に関する警告メッセージの出力契機*」に0 を指定した場合，リセット契機の指定は無効になります。
- クライアントグループに属していない HADB クライアントについては，adb_sys_max_users_wrn_pnt オペランドの指定値が適用されます。

- クライアントグループを定義した場合，システム全体でadb_sys_max_users_wrn_pnt オペランドに指定した比率以上になっても警告メッセージは出力されません。各クライアントグループに指定した比率以上になったときに警告メッセージが出力されます。

［マルチノード機能］

警告メッセージKFAA40020-W は，プライマリノードに出力されます。

[59] adbcltgrp（コマンドグループを設定する場合）

コマンドグループを設定します。コマンドグループを設定することで，グループに所属するコマンドが使用する接続数や処理リアルスレッド数を指定できます。コマンドグループを設定する場合は，このオペランドは 1 つだけ指定できます。

-g command

コマンドグループを設定します。

このオプションを指定したadbcltgrp オペランドを指定すると，「HADB サーバに接続するコマンド」は，コマンドグループに所属することになります。HADB サーバに接続するコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧』を参照してください。

> **メモ**
>
> コマンドグループには，「HADB サーバに接続するコマンド」だけでなく，HADB クライアントも所属できます。コマンドグループに HADB クライアントを所属させたい場合は，クライアント定義adb_clt_group_name オペランドに「command」を指定してください。
>
> クライアント定義adb_clt_group_name オペランドについては，マニュアル『HADB　AP 開発ガイド』の『システム構成に関するオペランド』を参照してください。

-m *設定したコマンドグループの最大同時接続数*

～〈整数〉((0～1,024))《adb_sys_max_users の指定値－ほかの adbcltgrp の-u オプションの指定値の合計》

設定したコマンドグループの最大同時接続数を指定します。

このオプションに指定した値が，設定したグループで使用できる接続数の上限になります。指定値を超えて，設定したグループが接続することはありません。

このオプションを省略した場合，次に示す計算式で求めた値が仮定されます。

計算式

```
A－B

  変数の説明
    A：サーバ定義adb_sys_max_usersオペランドの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-uオプションの指定値の合計※

  注※
    自グループ以外のサーバ定義adbcltgrpオペランドを1つも設定していない場合，0になります。
```

また，このオプションの指定値が，上記の計算式で求めた値より大きい場合も，上記の計算式で求めた値が仮定されます。

サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set形式）」の adb_sys_max_users オペランドを参照してください。

> **重要**
>
> このオプションに0 を指定した場合，グループに所属するコマンドは，HADB サーバに接続できません。

-u *設定したコマンドグループの接続保証数*

～〈整数〉((0～1,024))《0》

設定したコマンドグループの接続保証数を指定します。

このオプションの指定値に従って，コマンドグループが最低限使用する接続数を常に確保します。そのため，自グループ以外の HADB クライアントやコマンドの接続状況に影響されることなく，指定した分だけ常に接続できます。

このオプションには，-m オプションの指定値以下の値を指定してください。また，次の計算式を満たす値を指定してください（これらの条件を満たさない値を指定すると，HADB サーバの開始時にエラーとなります）。

```
A+B ≦ C

  変数の説明
    A：このオプションの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-uオプションの指定値の合計※
    C：サーバ定義adb_sys_max_usersオペランドの指定値

  注※
    自グループ以外で，サーバ定義adbcltgrpオペランドを1つも設定していない場合，0になります。
```

サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set形式）」の adb_sys_max_users オペランドを参照してください。

> **重要**
>
> 「*すべてのサーバ定義 adbcltgrp オペランドの-u オプションの指定値を合計した値*」が，「*サーバ定義 adb_sys_max_users オペランドの指定値*」と同じである場合，次のどちらかに該当するコマンドは，HADB サーバに接続できません。
>
> - 対象のコマンドが，サーバ定義adbcltgrp オペランドの-u オプションに0 を指定しているグループに所属しているとき
> - 対象のコマンドが，グループに所属していないとき

-r *設定したコマンドグループで使用できる処理リアルスレッドの最大数*

～〈整数〉((0～4,096))《adb_sys_rthd_num の指定値－ほかの adbcltgrp の-e オプションの指定値の合計》

設定したコマンドグループで使用できる処理リアルスレッドの最大数を指定します。

このオプションに指定した値が，設定したグループで使用できる処理リアルスレッド数の上限になります。指定値を超えて，設定したグループが処理リアルスレッドを使用することはありません。

このオプションを省略した場合，次に示す計算式で求めた値が仮定されます。

計算式

```
A－B

  変数の説明
    A：サーバ定義adb_sys_rthd_numオペランドの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-eオプションの指定値の合計※

  注※
    自グループ以外で，サーバ定義adbcltgrpオペランドを1つも設定していない場合，0になります。
```

また，このオプションの指定値が，上記の計算式で求めた値より大きい場合も，上記の計算式で求めた値が仮定されます。

サーバ定義adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドを参照してください。

なお，コマンドグループを設定する場合で，かつ「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドにある『対象のコマンド』を実行する場合は，コマンドを実行するために*最低限必要な処理リアルスレッド数*以上の値を，このオプションに指定してください。*最低限必要な処理リアルスレッド数*以上の値を指定しないと，対象のコマンドを実行してもエラーとなるおそれがあります。各コマンドを実行するために*最低限必要な処理リアルスレッド数*は，「6.23.2　コマンドの実行時に使用する処理リアルスレッド数に関する考慮点」の「(2)　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプション」の「表 6-28　コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプションの一覧」を参照してください。

重要

このオプションに0 を指定した場合，コマンドグループに所属するコマンドに適用される adb_sys_rthd_num オペランドには，0 が仮定されます。そのため，対象のコマンドを実行すると，処理リアルスレッド数が不足して，エラーとなります。

-e *設定したコマンドグループで使用できる処理リアルスレッドの保証数*

～〈整数〉((0～4,096))《0》

設定したコマンドグループで使用できる処理リアルスレッドの保証数を指定します。

このオプションの指定値に従って，対象のグループが最低限使用する処理リアルスレッド数を常に確保します。そのため，自グループ以外の HADB クライアントやコマンドの使用状況に影響されることなく，指定した分だけ常に処理リアルスレッドを使用できます。

このオプションには，-r オプションの指定値以下の値を指定してください。また，次の計算式を満たす値を指定してください（これらの条件を満たさない値を指定すると，HADB サーバの開始時にエラーとなります）。

計算式

```
A+B ≦ C

  変数の説明
    A：このオプションの指定値
    B：自グループ以外のサーバ定義adbcltgrpオペランドの-eオプションの指定値の合計※
    C：サーバ定義adb_sys_rthd_numオペランドの指定値

  注※
    自グループ以外で，サーバ定義adbcltgrpオペランドを1つも設定していない場合，0になります。
```

サーバ定義adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドを参照してください。

重要

「*すべてのサーバ定義 adbcltgrp オペランドの-e オプションの指定値を合計した値*」が，「*サーバ定義 adb_sys_rthd_num オペランドの指定値*」と同じである場合，次のどちらかに該当するコマンドは，adb_sys_rthd_num オペランドに0 が仮定されます。

- 対象のコマンドが，サーバ定義adbcltgrp オペランドの-e オプションに0 を指定しているグループに所属しているとき
- 対象のコマンドが，グループに所属していないとき

そのため，対象のコマンドを実行すると，処理リアルスレッド数が不足して，エラーとなります。

-w *設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機*〔,*警告メッセージの出力済み状態のリセット契機*〕

このオプションを指定すると，-m オプションで指定した最大同時接続数の残り枠が少なくなったときに，警告メッセージKFAA40020-W を出力できます。

なお，-m オプションに0 を指定した場合（0 が仮定された場合も含む），-w オプションの指定は無効になります。

設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機：

～〈整数〉((0～100))《0》(単位：%)

警告メッセージKFAA40020-W の出力契機を，-m オプションの指定値（コマンドグループに適用される最大同時接続数）に対する比率で指定します。例えば，-m オプションに20 を指定している場合に，警告メッセージの出力契機に90 を指定すると，HADB サーバへの接続数が 18（20 の 90%の値）以上になったときに警告メッセージが出力されます。

なお，警告メッセージの出力契機に0 を指定した場合，警告メッセージは出力されません。

重要

ほかのグループと自由利用枠を共有している場合（-m オプションと-u オプションの指定値が異なる場合），ほかのグループによって自由利用枠が使用されていると，警告メッセージが出力される前に，最大同時接続数超過のエラーが発生する可能性があります。

警告メッセージの出力済み状態のリセット契機：

～〈整数〉((0～99))（単位：%）

警告メッセージKFAA40020-Wの出力済み状態のリセット契機を，-mオプションの指定値（コマンドグループに適用される最大同時接続数）に対する比率で指定します。警告メッセージは一度出力されると，警告メッセージ出力済み状態となり，HADBサーバへの接続数がある一定の値を下回らないかぎり警告メッセージは再出力されません。

メモ

-wオプションの指定値と警告メッセージの出力の関係を次に示します。

```
adbcltgrp -g ...
          -m 20
          -w 90,70
```

上記のように-mオプションと-wオプションを指定した場合，HADBサーバへの接続数が18（20の90%の値）以上になったときに警告メッセージが出力されます。その後，HADBサーバへの接続数がいったん17に下がったあとに再度18以上になっても，警告メッセージは出力されません。

HADBサーバへの接続数が，14（20の70%の値）以下に下がった場合は，警告メッセージの出力済み状態がリセットされるため，HADBサーバへの接続数が再度18以上になったときに警告メッセージが出力されます。

警告メッセージの出力済み状態のリセット契機に関する留意事項を次に示します。

- リセット契機の指定を省略した場合，「*設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機*－30」の値が仮定されます。その値がマイナスになる場合は，0が仮定されます。
- 「*設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機* ＞ *警告メッセージの出力済み状態のリセット契機*」となるように指定してください。この条件を満たさない場合，リセット契機の指定は無効になります（警告メッセージKFAA41000-Wが出力されます）。このとき，リセット契機には，「*設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機*－30」の値が仮定されます。
- 「*設定したコマンドグループの最大同時接続数に関する警告メッセージの出力契機*」に0を指定した場合，リセット契機の指定は無効になります。
- コマンドグループに属していないコマンドについては，adb_sys_max_users_wrn_pntオペランドの指定値が適用されます。
- コマンドグループを定義した場合，システム全体でadb_sys_max_users_wrn_pntオペランドに指定した比率以上になっても警告メッセージは出力されません。コマンドグループまたは各クライアントグループに指定した比率以上になったときに警告メッセージが出力されます。

7.3　サーバ定義の文法規則

この節では，サーバ定義を作成する場合，および変更する場合の文法規則について説明します。

7.3.1　サーバ定義の文法規則の詳細

サーバ定義を作成および変更する場合の文法規則の詳細を次に示します。

サーバ定義の作成方法および変更方法については，「8.5　サーバ定義の作成および変更方法」を参照してください。

また，サーバ定義のオペランドの指定形式と内容については，次の項目を参照してください。

- 「7.1　サーバ定義のオペランドの指定形式」
- 「7.2　サーバ定義のオペランドの内容」

(1)　記述順序

サーバ定義の，定義間の記述順序は任意です。

(2)　記述形式

サーバ定義の記述形式には次に示す 2 つの形式があります。

- set 形式
- コマンド形式

それぞれの形式について説明します。

(a)　set 形式

set 形式では，*オペランド*に*値*を設定します。

形式

```
set オペランド = 値
```

オペランド

set 形式のオペランド名です。

値

オペランド名に対する指定値です。英数字の文字列で構成されます。

指定例を次に示します。

指定例

```
set adb_sys_max_users = 1024
```

同じオペランドを複数指定した場合，1 つ目に指定した値が有効となり，2 つ目以降の値は無視されます。

(b) コマンド形式

コマンド形式のオペランドの指定形式を次に示します。

形式

```
オペランド名 オプション
```

オペランド名

コマンド形式のオペランド名です。

オプション

ハイフン（-）で始まる文字列です。形式 1 と形式 2 の 2 つがあります。

形式 1

```
オプション
```

形式 2

```
オプション オプション引数
```

オプション

ハイフン（-）と 1 文字の英字で構成されます。英大文字と英小文字は区別されます。

オプション引数

オプションに対する引数です。英数字の文字列で構成されます。

指定例を次に示します。

指定例

```
adbbuff -g BUFFER -n ADBUTBL01 -p 1024
```

(3) 改行コード

サーバ定義で使用できる改行コードを次に示します。

- LF（Line Feed）
- NL（New Line）
- CR +LF（Carriage Return + Line Feed）

なお，CR（Carriage Return）だけを使用することはできません。CR だけが使用されていた場合，HADB サーバを開始するときにエラーとなります。

(4) 行の継続

定義の 1 行の長さは最大 4,000 バイトです。4,000 バイトを超える指定の場合は，複数行に分けて記述してください。

複数に分けた行を継続行として認識させる場合，行の末尾に継続文字として¥を入力して，その直後で改行してください。継続行の先頭に半角スペースがある場合は，半角スペースも含めて継続されます。

ただし，¥と改行が連続していない場合は継続されません。指定例を次に示します。

改行した行が継続行として認識される例

```
adbbuff -g BUFFER ¥↓
        -n ADBU01
```

改行した行が継続行として認識されない例

```
adbbuff -g BUFFER ¥△↓
        -n ADBU01
```

（凡例）

↓：改行

△：半角スペース

(5) 文字列の識別規則

文字列中に二重引用符（ ” ）を指定したい場合，「¥”」としてください。「”AREA01”」として認識される指定例を次に示します。

指定例

```
adbbuff -g BUFFER -n ¥”AREA01¥” -p 1024
```

なお，二重引用符で囲まないと，英大文字と英小文字が区別されないオペランドがあります（adbbuff オペランドなど)。詳細については，各オペランドを参照願います。

(6) コメント

サーバ定義にコメントを記入する場合は，コメントの先頭にシャープ（#）を記述してください。

形式 1 のように，行の先頭にシャープ（#）を入力すると，行全体がコメントとして認識されます。

形式 1

```
#コメント…
```

形式 2 のように，記述した定義の後ろにコメントを入力したい場合は，シャープ（#）の前に半角スペースを入力してください。半角スペースがないと，コメントとして認識されません。

形式 2

```
定義の記述△#コメント…
```

（凡例）

△：半角スペース

なお，二重引用符（ ”）で囲んだ文字列の中にシャープ（#）がある場合，コメントとして認識されません。シャープ（#）およびシャープ（#）以降の文字列は，文字として認識されます。

「#」および「#」以降も文字として認識される例

```
set adb_db_path = ”TESTDB△#abc”
```

（凡例）

△：半角スペース

また，シャープ（#）がある行は，シャープ（#）以降の文字列がすべてコメントとして認識されます。そのため，複数に分けた行を継続行として認識させる際，コメントの記載位置に注意してください。継続行がコメントとして認識されてしまう例を次に示します。

継続行がコメントとして認識される例

```
adbbuff -g BUFFER△#コメント ¥↓
        -n DBAREA01
```

（凡例）

↓：改行

△：半角スペース

上記の場合，「△#コメント」は-n DBAREA01 の後ろに記載するようにしてください。

(7) BOM

BOM（Byte Order Mark）は，サーバ定義では使用できません。

(8) コンマ

set 形式の値やコマンド形式のオプション引数で，複数の文字列を指定する場合は，文字列と文字列の間をコンマで区切ってください。ただし，コンマで区切る場合は，文字列とコンマの間に半角スペースおよび改行を入力しないでください。HADB サーバを開始するときにエラーとなります。

コンマを使った指定例を次に示します。

正しい指定例

```
set adb_sql_prep_delrsvd_words = ABS,BEFORE,CALL,DATA

set adb_sql_prep_delrsvd_words = ABS,△BEFORE,△△CALL,△△△DATA
```

複数の文字列を指定する場合は，文字列と文字列の間をコンマで区切ってください。また，コンマと文字列の間には，半角スペースを入力できます。

エラーとなる指定例 1

```
set adb_sql_prep_delrsvd_words = ABS△,BEFORE△,CALL△,DATA
```

上記のように，先行する文字列とコンマの間には，半角スペースは入力できません。文字列中に半角スペースが含まれているため，指定値が不正と認識され，エラーとなります。

エラーとなる指定例 2

```
set adb_sql_prep_delrsvd_words = ABS,BEFORE↓
                                ,CALL,DATA
```

上記のように，文字列とコンマの間では，改行できません。エラーとなります。

改行する場合，複数に分けた行を継続行として認識させる必要があります。継続文字として¥を入力して，その直後で改行してください。指定例を次に示します。

指定例（継続行として認識させる場合）

```
set adb_sql_prep_delrsvd_words = ABS,BEFORE¥↓
                                ,CALL,DATA

set adb_sql_prep_delrsvd_words = ABS,BEFORE,¥↓
                                CALL,DATA
```

コンマと文字列の間に継続文字として¥を入力して，その直後で改行してください。どちらの場合でも，継続行として認識されます。

（凡例）

△：半角スペース

↓：改行

8 システム構築

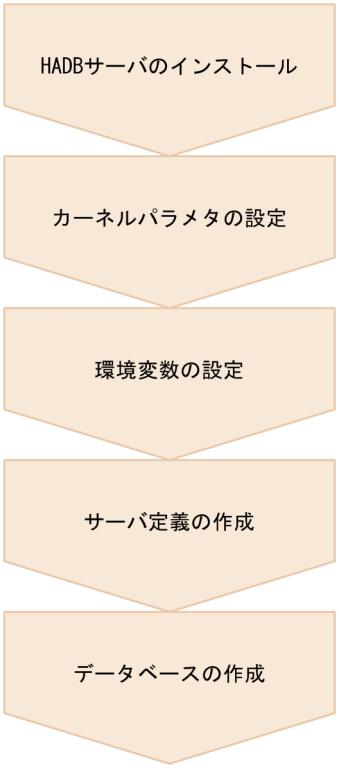
この章では，HADB のシステム構築について説明します。

体験環境を構築する手順については，「3.　体験環境構築ナビ」で説明しています。こちらもあわせて読むことを推奨します。

8.1　システム構築の手順

ここでは，HADB サーバのシステム構築について説明します。HADB クライアントのシステム構築については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。HADB サーバのシステム構築の手順を次の図に示します。

図 8-1　システム構築の手順



[説明]

1. HADB サーバのインストール

   HADB サーバをインストールします。インストール方法については，「8.2　インストール」を参照してください。

2. カーネルパラメタの設定

   カーネルパラメタを設定します。カーネルパラメタの設定については，「8.3　カーネルパラメタの設定」を参照してください。

3. 環境変数の設定

   環境変数を設定します。環境変数の設定については，「8.4　環境変数の設定」を参照してください。

4. サーバ定義の作成

   サーバ定義を作成します。サーバ定義の作成については，「7.　サーバ定義の設計」，および「8.5　サーバ定義の作成および変更方法」を参照してください。

5. データベースの作成

データベースを作成します。データベースの作成手順については，「9.1　データベースの作成手順」を参照してください。

8.2　インストール

ここでは，HADB サーバのインストール方法について説明します。

HADB クライアントのインストール方法については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントのインストールおよびアンインストール』を参照してください。

8.2.1　インストール前に実施する作業

HADB サーバをインストールする前に，**スーパユーザ**で次の作業を実施してください。

- 前提パッケージの確認※
- HADB 管理グループの設定
- HADB 管理者の設定
- HADB 管理グループに所属する OS ユーザの設定
- サーバディレクトリを格納するディレクトリの作成
- サーバディレクトリを格納するディレクトリの設定の変更
- 通信情報ファイルを格納するディレクトリの作成
- 設定ファイルの変更（RHEL 7 以降の場合）
- 設定ファイルの変更および作成（コアファイルの出力が systemd-coredump で制御されている場合）
- syslog のアクセス権限の変更

注※

スーパユーザではなく，一般ユーザ（管理者権限を持っていない OS ユーザ）でも実施できます。

上記の各作業の詳細を以降で説明します。

(1)　前提パッケージの確認

HADB サーバを動作させるには，次の表に示すパッケージ（x86_64 版）が OS にインストールされている必要があります。

表 8-1　前提パッケージ

項番	パッケージ名	Linux のバージョン		
		RHEL 6	RHEL 7	RHEL 8
1	bash	○	○	○
2	binutils	○	○	○
3	compat-openssl10	−	−	○

項番	パッケージ名	Linux のバージョン		
		RHEL 6	RHEL 7	RHEL 8
4	coreutils	○	○	○
5	file	○	○	○
6	findutils	○	○	○
7	gawk	○	○	○
8	gdb	○	○	○
9	glibc	○	○	○
10	glibc-common	○	○	○
11	grep	○	○	○
12	hostname	−	○	○
13	iproute	○	○	○
14	libaio	○	○	○
15	libuuid	○	○	○
16	lvm2	○	○	○
17	net-tools	○	−	−
18	openssl	○	−	−
19	openssl-libs	−	○	−
20	procps	○	−	−
21	procps-ng	−	○	○
22	sed	○	○	○
23	sysstat	○	○	○
24	tar	○	○	○
25	util-linux	−	○	○
26	util-linux-ng	○	−	−
27	zlib	○	○	○

（凡例）

○：この Linux のバージョンでは必要なパッケージです。

－：この Linux のバージョンでは不要なパッケージです。

■インストール済みのパッケージを確認する方法

次のコマンドを実行すると，OS にインストール済みのパッケージの一覧が表示されます。

```
yum list installed
```

インストールされていないパッケージがある場合は，そのパッケージを OS にインストールしてください。パッケージのインストール方法については，OS のマニュアルを参照してください。

■特定のパッケージがインストール済みかどうかを確認する方法

（例）

パッケージlibaio がインストール済みかどうかを確認します。

```
yum list installed | grep libaio
```

実行結果にパッケージlibaio が表示された場合は，パッケージlibaio はインストール済みです。実行結果にパッケージlibaio が表示されなかった場合は，パッケージlibaio はインストールされていません。

(2) HADB 管理グループの設定

HADB 専用のグループ（**HADB 管理グループ**）を，サーバマシンの OS に設定してください。HADB 管理グループは，OS のgroupadd コマンドで設定します。

HADB 管理グループを設定することで，HADB サーバを運用するためのディレクトリ下（サーバディレクトリ下および DB ディレクトリ下）に作成されるファイルに対して，HADB 管理グループ以外の OS ユーザからのアクセスを制限できます。そのため，セキュリティを強化できます。

■HADB 管理グループの設定例

サーバマシンの OS に HADB 管理グループとして，adbgroup を設定します。

```
groupadd adbgroup
```

なお，OS に設定されている既存のグループを，HADB 管理グループとして使用することもできます。

(3) HADB 管理者の設定

HADB 管理グループを設定したら，HADB サーバを管理する OS ユーザ（**HADB 管理者**）を，サーバマシンの OS に設定してください。HADB 管理者は，OS のuseradd コマンドで設定します。HADB 管理者の名称は，32 バイト以内としてください。

HADB 管理者とは，OS に設定するユーザであり，また HADB を管理するシステム管理者を指します。HADB 管理者には，HADB のすべてのコマンドを実行する権限があります。また，サーバディレクトリ下および DB ディレクトリ下のファイルの所有者になります。なお，HADB 管理者として設定した OS ユーザが，HADB 管理者として HADB サーバに認識されるタイミングは，HADB サーバをインストールしたときです。

HADB 管理者のユーザ名を OS に設定する際，HADB 管理者が所属するプライマリグループに HADB 管理グループを設定することで，HADB 管理者には HADB サーバの各種ファイルおよびディレクトリの所有者としてのアクセス権が与えられます。これによって，ほかの OS ユーザからの書き込みを禁止できます。

HADB 管理者のユーザ名を OS に設定したら，必ずパスワードを設定してください。パスワードは，OS のpasswd コマンドで設定します。

■HADB 管理者の設定例

サーバマシンの OS に HADB 管理者として，adbmanager を設定します。その際，プライマリグループとして HADB 管理グループ（adbgroup）を設定します。

```
useradd -g adbgroup adbmanager
```

(4) HADB 管理グループに所属する OS ユーザの設定

HADB 管理者を設定したら，HADB 管理者とは別に，HADB サーバを管理する OS ユーザ（**HADB 管理グループに所属する OS ユーザ**）を，サーバマシンの OS に設定することを推奨します。

HADB 管理グループに所属する OS ユーザには，HADB のコマンドを実行する権限が与えられます（一部のコマンドは除きます）。また，HADB 管理者が所有するサーバディレクトリ下および DB ディレクトリ下のファイルにも，アクセスできるようになります。HADB 管理グループに所属する OS ユーザを設定することで，HADB 管理者以外の OS ユーザでも HADB のコマンドを実行できるようになり，HADB サーバを運用する際の利便性が向上します。

HADB 管理グループに所属する OS ユーザは，OS のuseradd コマンドで設定します。HADB 管理グループに所属する OS ユーザの名称は，32 バイト以内としてください。また，HADB 管理グループに所属する OS ユーザのユーザ名を OS に設定する場合，HADB 管理グループをプライマリグループとして設定してください。

HADB 管理グループに所属する OS ユーザのユーザ名を OS に設定したら，必ずパスワードを設定してください。パスワードは，OS のpasswd コマンドで設定します。

■HADB 管理グループに所属する OS ユーザの設定例

サーバマシンの OS に HADB 管理グループに所属する OS ユーザとして，adbgroupuser01 を設定します。その際，プライマリグループとして HADB 管理グループ（adbgroup）を設定します。

```
useradd -g adbgroup adbgroupuser01
```

メモ

HADB 管理グループに所属する OS ユーザが実行できる HADB のコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧と共通規則』の『コマンドの一覧』を参照してください。

(5) サーバディレクトリを格納するディレクトリの作成

スーパユーザで，HADB サーバをインストールする際に必要な**サーバディレクトリを格納するディレクトリ**を作成してください。サーバディレクトリを格納するディレクトリは，OS のmkdir コマンドで作成します。

サーバディレクトリを格納するディレクトリのパス名は，「/HADB」としてください。

■サーバディレクトリを格納するディレクトリの作成例

サーバディレクトリを格納するディレクトリ（/HADB）を作成します。

```
mkdir /HADB
```

(6) サーバディレクトリを格納するディレクトリの設定の変更

サーバディレクトリを格納するディレクトリ（/HADB）を作成したら，スーパユーザで，次の設定を変更してください。

- サーバディレクトリを格納するディレクトリ（/HADB）の所有者とグループを変更する

  ディレクトリの所有者を，HADB 管理者（adbmanager）に変更してください。また，ディレクトリのグループを，HADB 管理グループ（adbgroup）に変更してください。ディレクトリの所有者とグループは，OS のchown コマンドで変更します。

- サーバディレクトリを格納するディレクトリ（/HADB）に書き込み権限を付与する

  HADB 管理者（adbmanager）が書き込みできるように，ディレクトリに書き込み権限を付与してください。ディレクトリの書き込み権限は，OS のchmod コマンドで付与します。

■サーバディレクトリを格納するディレクトリの設定の変更例

サーバディレクトリを格納するディレクトリ（/HADB）の所有者，およびグループを変更し，ディレクトリに書き込み権限を付与します。

```
chown adbmanager:adbgroup /HADB
chmod 755 /HADB
```

(7) 通信情報ファイルを格納するディレクトリの作成

HADB サーバを動作させるために必要な通信情報ファイルを格納するディレクトリを作成してください。通信情報ファイルを格納するディレクトリは，OS のmkdir コマンドで作成します。

通信情報ファイルを格納するディレクトリは，HADB サーバをインストールするサーバマシンの OS のバージョンによって異なります。

(a) RHEL 6 の場合

通信情報ファイルを格納するディレクトリのパス名，および付与するアクセス権限を次の表に示します。

表 8-2 通信情報ファイルを格納するディレクトリの情報（RHEL 6 の場合）

項番	作成するディレクトリのパス名	所有者	アクセス権限
1	/dev/HADB/pth	スーパユーザ	777
2	/lib/udev/devices/HADB/pth		

■通信情報ファイルを格納するディレクトリの作成例

通信情報ファイルを格納するディレクトリ（/dev/HADB/pth および/lib/udev/devices/HADB/pth）を作成します。その際，アクセス権限として777 を付与します。

```
mkdir -p -v -m 777 /dev/HADB/pth

mkdir -p -v -m 777 /lib/udev/devices/HADB/pth
```

(b)　RHEL 7 以降の場合

通信情報ファイルを格納するディレクトリのパス名，および付与するアクセス権限を次の表に示します。

表 8-3　通信情報ファイルを格納するディレクトリの情報（RHEL 7 以降の場合）

項番	作成するディレクトリのパス名	所有者	アクセス権限
1	/dev/HADB/pth	スーパユーザ	777

また，次に示す設定ファイルを作成する必要があります。

表 8-4　設定ファイルの情報（RHEL 7 以降の場合）

項番	作成する設定ファイル	所有者	アクセス権限
1	/etc/tmpfiles.d/dev-HADB-pth.conf	スーパユーザ	644（デフォルト）

設定ファイル（dev-HADB-pth.conf）に入力する内容を次に示します。

```
# Type Path Mode UID  GID  Age Argument
d /dev/HADB/pth 0777 root root - -
```

■通信情報ファイルを格納するディレクトリの作成例

通信情報ファイルを格納するディレクトリ（/dev/HADB/pth）を作成します。その際，アクセス権限として777 を付与します。

```
mkdir -p -v -m 777 /dev/HADB/pth
```

■設定ファイルの作成例

設定ファイル（dev-HADB-pth.conf）を作成します。

次に示す OS のコマンドを実行してください。

```
vi /etc/tmpfiles.d/dev-HADB-pth.conf
```

vi コマンドを実行したら，[I] キーを押してください。そのあとで，次に示す内容を入力してください。

```
# Type Path Mode UID  GID  Age Argument
d /dev/HADB/pth 0777 root root - -
```

入力が完了したら，[Esc] キーを押してください。そのあとで，次に示すコマンドを入力して，[Enter] キーを押してください。

```
:wq
```

(8) 設定ファイルの変更（RHEL 7 以降の場合）

RHEL 7 以降に HADB サーバをインストールする場合は，HADB 管理者が自身に関するログにアクセスできるように，次の表に示す設定ファイルの内容を変更します。

表 8-5 変更が必要な設定ファイル（RHEL 7 以降の場合）

項番	変更が必要な設定ファイル	所有者	アクセス権限
1	/etc/systemd/journald.conf	スーパユーザ	644（デフォルト）

上記の設定ファイル中の[Journal]セクションのStorage オプションを次のように変更してください。

＜変更前＞

```
[Journal]
#Storage=auto
```

＜変更後＞

```
[Journal]
Storage=persistent
```

変更後の設定を有効にするために，次のコマンドを実行してください。

```
systemctl restart systemd-journald
```

(9) 設定ファイルの変更および作成（コアファイルの出力が systemd-coredump で制御されている場合）

コアファイルの出力が systemd-coredump で制御されている場合は，ここで説明する作業を実施してください。次のコマンドを実行すると，systemd-coredump が有効かどうかを確認できます。

```
cat /proc/sys/kernel/core_pattern
```

上記のコマンドの実行結果に，次のように systemd-coredump へのパスが含まれている場合は，systemd-coredump が有効な状態です。

```
|/usr/lib/systemd/systemd-coredump %P %u %g %s %t %c %h %e
```

メモ

RHEL 8 では systemd-coredump がデフォルトで有効になっています。

■/etc/systemd/coredump.conf の変更

コアファイルの出力容量を増やすために，次の表に示す設定ファイルの内容を変更します。

表 8-6　変更が必要な設定ファイル

項番	変更が必要な設定ファイル	所有者	アクセス権限
1	/etc/systemd/coredump.conf	スーパユーザ	644（デフォルト）

上記の設定ファイル中の[Coredump]セクションのProcessSizeMax オプション，およびExternalSizeMax オプションを次のように変更します。

＜変更前＞

```
[Coredump]
#ProcessSizeMax=2G
#ExternalSizeMax=2G
```

＜変更後＞

```
[Coredump]
ProcessSizeMax=15E
ExternalSizeMax=15E
```

■/etc/systemd/system/systemd-coredump@.service の作成

systemd-coredump の実行時間を十分に確保するために，次の表に示す設定ファイルを作成してください。

表 8-7　作成が必要な設定ファイル

項番	作成が必要な設定ファイル	所有者	アクセス権限
1	/etc/systemd/system/systemd-coredump@.service	スーパユーザ	644（デフォルト）

上記の設定ファイルは，次の場所からコピーしてください。

```
cp /usr/lib/systemd/system/systemd-coredump@.service /etc/systemd/system/systemd-coredump
@.service
```

上記の設定ファイル中の[Service]セクションのRuntimeMaxSec オプションを，次のどちらかのように変更してください。

＜変更前＞

```
[Service]
RuntimeMaxSec=5min
```

＜変更後 1 ＞

```
[Service]
#RuntimeMaxSec=5min
```

＜変更後 2 ＞

```
[Service]
RuntimeMaxSec=infinity
```

(10) syslog のアクセス権限の変更

HADB サーバのメッセージが出力される syslog のアクセス権限を変更してください。syslog のアクセス権限は，OS のchmod コマンドで変更できます。

syslog のアクセス権限を変更することで，adbinfoget コマンドでトラブルシュート情報を取得するときに，syslog を取得できるようになります。

syslog のデフォルトのパス名，および付与するアクセス権限を次の表に示します。

表 8-8 アクセス権限を変更する syslog の情報

項番	syslog のデフォルトのパス名	所有者	アクセス権限
1	/var/log/messages※	スーパユーザ	604

注※

HADB サーバのメッセージが出力される syslog を，/var/log/messages から変更している場合は，変更先の syslog に対してアクセス権限を付与してください。

■syslog のアクセス権限の変更例

syslog（/var/log/messages）のアクセス権限を604 に変更します。

```
chmod 604 /var/log/messages
```

8.2.2 インストール手順

ここでは，HADB サーバをインストールする手順を説明します。HADB サーバをインストールする場合，**HADB 管理者として設定した OS ユーザ**で実行してください。

(1) インストールデータを格納するディレクトリの作成

HADB サーバのインストールデータを格納するディレクトリを作成してください。

```
mkdir /home/adbmanager/install
```

上記の例では，HADB サーバのインストールデータを格納するディレクトリとして/home/adbmanager/install を作成しています。

(2)　インストールデータを格納するディレクトリへの書き込み権限の付与

インストールデータを格納するディレクトリ（この例では/home/adbmanager/install）に対して，HADB管理者が書き込みできるように書き込み権限を付与してください。

```
chmod 755 /home/adbmanager/install
```

(3)　CD-ROM ファイルシステムのマウント

HADB サーバのインストールデータが格納されている CD-ROM ファイルシステムを自動マウントしてください。

CD-ROM ファイルシステムが自動マウントされない場合，次のコマンドを実行して CD-ROM ファイルシステムをマウントしてください。

```
mount /dev/cdrom /media
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

> **重要**
>
> CD-ROM のディレクトリ名やファイル名は，ご使用のサーバマシンによっては，ここの説明内容と表示が異なることがあります。OS の ls コマンドを実行して，表示されたディレクトリ名をそのまま入力してください。

(4)　インストールデータのコピー

マウントした CD-ROM ファイルシステムに格納されている次のファイルを，「(1)　インストールデータを格納するディレクトリの作成」で作成したディレクトリにコピーしてください。

- tar.gz 形式のファイル
- adbinstall コマンドの実行形式ファイル
- !8A9_HADB ファイル

```
cp /media/hitachi_advanced_data_binder_server-$VER.tar.gz /home/adbmanager/install   ...1
cp /media/adbinstall /home/adbmanager/install                                        ...2
cp '/media/!8A9_HADB' /home/adbmanager/install                                       ...3
```

[説明]

1. tar.gz 形式のファイル（HADB サーバのプログラムの圧縮ファイル）をコピーします。
2. adbinstall コマンドの実行形式ファイル（HADB サーバのインストールコマンド）をコピーします。
3. !8A9_HADB ファイルをコピーします。

上記の例では，すべてのファイルを/home/adbmanager/installディレクトリ下にコピーしています。

下線部分は，CD-ROMファイルシステムのマウントディレクトリ名です。環境によって異なります。

$VER は，HADBサーバのバージョンとリリース番号です。

> **重要**
>
> 上記の3つのファイルは，必ず同じディレクトリにコピーしてください。異なるディレクトリにコピーすると，HADBサーバのインストールが実行できません。

(5) コピーしたファイルの確認

「(4) インストールデータのコピー」でコピーしたファイルが破損していないことを確認してください。

```
cd /home/adbmanager/install
sha256sum -c /media/adbhashfile_server_sha256.txt
```

下線部分は，CD-ROMファイルシステムのマウントディレクトリ名です。環境によって異なります。

ファイルが破損しているおそれがある（ファイルのハッシュ値が異なる）場合は，エラーを示すメッセージが出力されます。この場合，「(4) インストールデータのコピー」から作業をし直してください。

出力されるメッセージの詳細については，OSのマニュアルを参照してください。

(6) 権限の付与

HADB管理者がadbinstallコマンド（HADBサーバのインストールコマンド）を実行できるように，adbinstallコマンドに対する実行権限を付与してください。

また，tar.gz形式のファイルに対する読み取り権限も付与してください。

```
chmod 500 /home/adbmanager/install/adbinstall
chmod 400 /home/adbmanager/install/hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，HADBサーバのバージョンとリリース番号です。

(7) HADBサーバのインストール

adbinstallコマンドを実行して，HADBサーバをインストールしてください。

```
/home/adbmanager/install/adbinstall -s /HADB/server
```

下線部分には，「(1) インストールデータを格納するディレクトリの作成」で作成したディレクトリを指定します。

-s オプションに指定したディレクトリ下（上記の場合，/HADB/server ディレクトリ下）に，HADB サーバがインストールされます。

メモ

- HADB サーバをインストールしたディレクトリが，**サーバディレクトリ**となります。サーバディレクトリとは，HADB サーバがインストールされるディレクトリのことであり，1 つのサーバプロセスに関するファイル群が格納されます。

  上記の例の場合，/HADB/server ディレクトリがサーバディレクトリになります。
- サーバディレクトリのパス名は，スラッシュ（/）で始まる 118 バイト以内の文字列としてください。
- サーバディレクトリのパス名に使用できる文字については，『はじめに』の「■このマニュアルで使用する構文要素記号」の『〈パス名〉』を参照してください。
- -s オプションに指定したディレクトリが存在しないときは，adbinstall コマンドを実行すると，自動的にディレクトリが作成されます。

重要

HADB サーバをインストールするためにadbinstall コマンドを実行した OS ユーザが，HADB 管理者として HADB サーバに認識されます。

■KFAA91553-E メッセージが出力された場合の対処方法

- サーバディレクトリには，必ず HADB 管理者が書き込みできるディレクトリを指定してください。HADB 管理者が書き込みできないディレクトリを-s オプションに指定した場合，KFAA91553-E メッセージが出力されてエラーになります。KFAA91553-E メッセージが出力された場合は，サーバディレクトリに対する書き込み権限を付与してください。
- インストールデータを格納するディレクトリに，HADB 管理者が書き込みできない場合，KFAA91553-E メッセージが出力されてエラーになります。KFAA91553-E メッセージが出力された場合は，インストールデータを格納するディレクトリに対する書き込み権限を付与してください。

■KFAA91558-W メッセージが出力された場合の対処方法

HADB 管理者として設定した OS ユーザではなく，root でadbinstall コマンドを実行した場合は，警告メッセージ（KFAA91558-W メッセージ）が出力されます。通常，HADB 管理者として設定した OS ユーザで，adbinstall コマンドを実行します。そのため，KFAA91558-W メッセージが出力された場合は，root でadbinstall コマンドを実行して問題がないかどうかを確認してください。

問題がある場合は，KFAA91558-W メッセージが出力されたあとに出力されるKFAA91559-Q メッセージの入力要求に対して，n またはN を入力してください。そのあとで，HADB 管理者として設定した OS ユーザでadbinstall コマンドを実行してください。

なお，root 以外のスーパユーザでadbinstall コマンドを実行した場合は，KFAA91558-W メッセージは出力されません。

メモ

root は，OS の id -u コマンドを実行して表示される値が0 のユーザを指します。また，OS のsu コマンドを使用して，ほかの OS ユーザからroot に切り替えたあとで，OS の id -u コマンドを実行して表示される値が0 のときも含みます。

メモ

- adbinstall コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbinstall（HADB サーバおよび HADB クライアントのインストール)』を参照してください。
- adbinstall コマンドで HADB サーバをインストールしたときのサーバディレクトリの構成については，「付録 A.1　サーバディレクトリ（インストール時）の構成」を参照してください。

(8)　HADB サーバのインストールの確認

HADB サーバが正常にインストールされていることを確認してください。

手順

1. サーバディレクトリ下にadbinstsv.log ファイルがあることを確認する
2. adbinstsv.log ファイルを開いて内容を確認する
   正常にインストールされている場合は，次のどちらかの情報が記載されています。
   - Hitachi Advanced Data Binder *vv-rr* (*yyyymmddhhmmss*)
   - Hitachi Advanced Data Binder *vv-rr-/s* (*yyyymmddhhmmss*)

   vv-rr および *vv-rr-/s* は，HADB サーバのバージョンを示しています。
   yyyymmddhhmmss は，HADB サーバをインストールした日付と時刻を示しています。

メモ

- adbinstsv.log ファイルがすでにある状態のときにadbinstall コマンドを実行すると，adbinstsv.log ファイルの内容が更新されます。また，adbinstall コマンドを複数回実行した場合，最後に実行した日付と時刻が記載されます。
- adbinstall コマンドのリターンコード（KFAA91552-I メッセージに出力されるリターンコード）が0 または4 以外の場合，adbinstsv.log ファイルの内容は更新されません。
- adbinstall コマンドの実行時にKFAA91555-I メッセージが出力された場合，adbinstsv.log ファイルの内容は更新されません。

8.2.3　インストール後に実施する作業

HADB サーバをインストールしたあとで，次の作業を実施してください。

(1)　ウイルス対策ソフトによるスキャン対象の見直し

HADB サーバをインストールしたサーバマシンに，ウイルス対策ソフトがインストールされている場合は，スキャン対象を見直してください。

ウイルス対策ソフトのスキャン対象に，HADB サーバが使用するディレクトリやファイルを設定しておくと，HADB サーバが正常に動作しないなどの問題が発生するおそれがあります。そのため，ウイルス対策ソフトのスキャン対象から，次に示すディレクトリおよびファイルを除いてください。

■スキャン対象から除くディレクトリとファイル

- サーバディレクトリ
- DB ディレクトリ
- 通信情報ファイルを格納するディレクトリ（/dev/HADB/pth）
- サーバ定義adb_core_path オペランドに指定するディレクトリ
- HADB サーバの各コマンドがアクセスするすべてのディレクトリ
- すべてのアーカイブディレクトリ
- 同義語辞書ファイルの格納ディレクトリ（サーバ定義のadb_syndict_storage_path オペランドに指定するディレクトリ）
- 監査証跡の出力先ディレクトリ（サーバ定義のadb_audit_log_path オペランドに指定するディレクトリ）
- マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ（サーバ定義の adb_syndict_node_storage_path オペランドに指定するディレクトリ）
- クライアントディレクトリ
- 統一フォーマット用監査証跡の出力先ディレクトリ（adbconvertaudittrailfile コマンドの-d オプションに指定するディレクトリ）
- ADB_CSVREAD 関数でアクセスするファイル
- ADB_AUDITREAD 関数でアクセスするファイル
- adbconvertaudittrailfile コマンドで指定する変換対象の監査証跡ファイル

(2)　IT Report Utility の設定

HADB サーバをインストールしたサーバマシンに，IT Report Utility（ITRU）がインストールされている場合は，ITRU 経由でトラブルシュート情報を取得できるように設定してください。

重要

HADB サーバに対応する ITRU のバージョンは，サーバマシンの OS のバージョンによって異なります。

- RHEL 6 の場合
  HADB サーバに対応する ITRU のバージョンは，02-00 以降です。
- RHEL 7 の場合
  HADB サーバに対応する ITRU のバージョンは，03-00 以降です。
- RHEL 8 の場合
  HADB サーバに対応する ITRU のバージョンは，03-01 以降です。

ITRU の設定手順を次に示します。

手順

1. サーバマシンにroot でログインする
   HADB 管理者でログインしている場合は，いったんログアウトして，root でログインし直してください。
2. ITRU の採取パターン定義ファイルをコピーする
   HADB サーバのインストールデータを格納するディレクトリ下にある ITRU の採取パターン定義ファイル（!8A9_HADB）を，ITRU の採取パターン定義ファイル格納ディレクトリ（/etc/opt/hitachi/systoru/pattern）にコピーしてください。
   コマンドの実行例

```
cp '/home/adbmanager/install/!8A9_HADB' /etc/opt/hitachi/systoru/pattern
```

   下線部分は，HADB サーバのインストールデータを格納するディレクトリです。
3. ITRU の採取パターン定義ファイルの内容を修正する
   コピーした ITRU の採取パターン定義ファイル（!8A9_HADB）をテキストエディタで開き，次に示す個所を修正してください。
   ■ITRU の採取パターン定義ファイルの内容（修正前）

```
FORMAT_VERSION=1.0

PP_NAME="HADB"
PP_GROUP="HADB"

# HADB
#
BLOCK {
  TARGET="config"
  TARGET="failure"
  TOOL_BLOCK {
    TIMEOUT=300
    COMMAND_LINE="XXX/bin/adbsystoru -s XXX -l YYY -o ¥"%d%¥""
    REDIRECT_PATH="%d%/HADB.txt"
```

```
  }
  DATA_BLOCK {
    DESCRIPTION=”HADB”
    FILE_TYPE=”directory”
    PATH_NAME=”%d%”
  }
}
```

上記のXXX およびYYY の個所を修正してください。

■ITRU の採取パターン定義ファイルの内容（修正後）

```
   :
COMMAND_LINE=”/HADB/server/bin/adbsystoru -s /HADB/server -l UTF8 -o ¥”%d%¥””
   :
```

- XXX の個所

  サーバディレクトリを絶対パスで指定してください。環境変数ADBDIR の指定内容と同じにしてください。この例では，/HADB/server を指定しています。

- YYY の個所

  HADB サーバで使用する文字コードを指定してください。環境変数ADBLANG の指定内容と同じにしてください。この例では，UTF8 を指定しています。

環境変数ADBDIR とADBLANG については，「8.4　環境変数の設定」を参照してください。

4. ITRU の採取パターン定義ファイルを編集した結果を保存する

   手順 3.で編集した内容を保存して，ITRU の採取パターン定義ファイル（!8A9_HADB）を閉じてください。

これで，ITRU の採取パターン定義ファイルの設定が完了します。

(3)　SELinux のモードの確認

HADB サーバをインストールしたサーバマシンで，OS のgetenforce コマンドを実行して SELinux のモードを確認してください。SELinux のモードには，次の 3 種類があります。

- Enforcing

  SELinux が有効な状態であり，強制アクセス制御も有効な状態です。

- Permissive

  SELinux は有効な状態ですが，強制アクセス制御は無効な状態です。リソースへの不正なアクセスが発生した場合，警告メッセージが出力されます。

- Disabled

  SELinux が無効な状態です。

重要

SELinux の影響によって，HADB サーバに関連するプロセスが必要なファイルにアクセスできない場合，予期しない問題が発生するおそれがあります。そのため，HADB サーバの稼働中は，SELinux のモードをPermissive にすることを推奨します。

SELinux のモードをPermissive にするには，/etc/selinux/config ファイルを編集します。次のように，SELINUX オプションにpermissive を指定してください。

```
SELINUX=permissive
```

上記の設定を有効にするためには，OS の再起動が必要です。設定したあとで OS を再起動してください。

メモ

- SELinux の詳細な設定方法については，OS のマニュアルを参照してください。
- 将来，SELinux をEnforcing の状態で運用することを検討している場合は，SELinux をPermissive の状態で HADB サーバを運用してください。その際，Audit ログ（/var/log/audit/audit.log）に出力される警告メッセージを確認して，警告メッセージが出力される原因を取り除いてください。

(4) HADB サーバの自動起動の設定

OS の起動時や再起動時に，HADB サーバを自動的に開始したい場合は，Systemd で HADB サーバの自動起動を設定してください。なお，下記で説明している設定手順は，次のどちらかの条件に該当する場合には対応していません。

- サーバマシンの OS が RHEL 6 の場合
- マルチノード構成の HADB サーバの場合

自動起動の設定手順を次に示します。

手順

1. サーバマシンにroot でログインする
   HADB 管理者でログインしている場合は，いったんログアウトして，root でログインし直してください。
2. ユニットファイルを作成する
   次の表に示すユニットファイルを作成してください。

表 8-9　作成が必要なユニットファイル

項番	作成が必要なユニットファイル	所有者	アクセス権限
1	/etc/systemd/system/hadb.service	スーパユーザ	644（デフォルト）

■ユニットファイルの内容例

```
[Unit]
Description=Hitachi Advanced Data Binder Server
Wants=systemd-tmpfiles-setup-dev.service
After=syslog.target network.target local-fs.target systemd-tmpfiles-setup-dev.service

[Service]
Type=forking
PIDFile=/HADB/server/spool/adbsrvd.pid                                    ...a
Environment=ADBDIR=/HADB/server                                           ...a
Environment=ADBLANG=UTF8                                                  ...b
Environment=LD_LIBRARY_PATH=/HADB/server/lib:/HADB/server/client/lib      ...a
Environment=PATH=/HADB/server/bin:/HADB/server/client/bin:/usr/bin        ...a
User=adbmanager
Group=adbgroup
ExecStart=/HADB/server/bin/adbstart                                       ...a
ExecStop=/HADB/server/bin/adbstop --cancel                                ...a
LimitNOFILE=1048576
LimitMEMLOCK=infinity
LimitNPROC=1048576
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
```

[説明]

a. 下線部分には，環境変数ADBDIR で指定したサーバディレクトリの絶対パスを指定してください。

b. 下線部分には，環境変数ADBLANG で指定した HADB サーバで使用する文字コードを指定してください。

環境変数ADBDIR とADBLANG については，「8.4　環境変数の設定」を参照してください。

また，Environment オプションは，EnvironmentFile オプションを使用して環境変数をまとめることもできます。

3. HADB サーバのサービスユニットを有効化する

次に示すコマンドを実行してください。

```
systemctl daemon-reload
systemctl enable hadb.service
```

重要

HADB サーバのサービスユニットを有効化したら，OS を再起動する前に，HADB サーバをadbstart コマンドで正常に開始できることを確認してください。その際，adbstart コマンドのリターンコードが0 であることを確認してください。adbstart コマンドのリターンコードが0 以外の場合，Systemd はサービスの起動が失敗したと見なします。HADB サーバをadbstart コマンドで正常に開始できない状態で OS を再起動すると，再起動後に HADB サーバのサービスユニットの起動に失敗します。

これで，HADB サーバの自動起動の設定が完了します。

8.3　カーネルパラメタの設定

ここでは，カーネルパラメタの設定について説明します。

HADB サーバをインストールしたサーバマシンに対して，カーネルパラメタを設定してください。カーネルパラメタを設定する流れを次に示します。

カーネルパラメタを設定する流れ

1. 各カーネルパラメタの指定値を求めます。
2. 求めた値を，設定ファイルに指定します。

カーネルパラメタの指定値の求め方，および設定ファイルについては，「6.2　カーネルパラメタの見積もり」を参照してください。カーネルパラメタの設定方法については，OS のマニュアルを参照してください。

8.4 環境変数の設定

ここでは，HADB 管理者，および HADB 管理グループに所属する OS ユーザが設定する環境変数について説明します。

設定した値が，HADB サーバ使用時のシェルで有効になるようにしてください。環境変数の設定方法については，OS のマニュアルを参照してください。

また，マルチノード機能を使用する場合は，全ノードで，それぞれ環境変数を設定してください。

重要

環境変数は，必ず HADB サーバが停止している状態で設定してください。adbstop コマンドを実行することで，HADB サーバを終了できます。

環境変数に設定する値を次の表に示します。

表 8-10 環境変数に設定する値

項番	環境変数	設定する値
1	LANG	OS の文字コードを指定します。※ **日本語環境で使用する場合：** 次のどちらかの文字コードを選択します。 • Unicode（UTF-8） この環境変数にja_JP.UTF-8 を指定してください。 • Shift-JIS この環境変数にja_JP.SJIS を指定してください。 **英語環境で使用する場合：** 文字コードは Unicode（UTF-8）になります。この環境変数にen_US.UTF-8 を指定してください。
2	LD_LIBRARY_PATH	この環境変数に，次に示すディレクトリを追加してください。 • $ADBDIR/lib また，次のどちらかの条件を満たす場合は，この環境変数に$ADBDIR/client/lib も追加してください。 • HADB サーバでadbsql コマンドを実行する場合 • HADB サーバで CLI 関数を使用した AP を開発または実行する場合
3	PATH	この環境変数に，次に示すディレクトリを追加してください。 • $ADBDIR/bin • $ADBDIR/client/bin
4	TZ	HADB サーバをインストールするマシンのタイムゾーンを設定します。 なお，うるう秒対応のタイムゾーンは設定しないでください。

項番	環境変数	設定する値
		[マルチノード機能] マルチノード機能を使用する場合，この環境変数の指定値を，全ノードで一致させてください。
5	ADBDIR	この環境変数には，サーバディレクトリを絶対パスで指定してください。
6	ADBLANG	HADB サーバで使用する文字コードを指定します。※ **日本語環境で使用する場合：** 次のどちらかの文字コードを選択します。 • Unicode（UTF-8） この環境変数にUTF8 を指定してください。 • Shift-JIS この環境変数にSJIS を指定してください。 SJIS を指定する場合，Shift-JIS のマルチバイト文字が syslog に出力されることを防ぐため，次に示すどれかの対処をしてください。 • HADB サーバに拡張 SYSLOG 機能を適用して，syslog の文字コード変換（Shift-JIS から Unicode への変換）を行う 拡張 SYSLOG 機能については，「10.4.6　syslog の文字コード変換と信頼性向上（拡張 SYSLOG 機能の適用）」を参照してください。 • 次に示す項目の指定で，マルチバイト文字を使用しないようにする ・定義系 SQL 中に指定する名前（表名，インデクス名，列名など） ・HADB が使用するディレクトリやファイル名称 • 環境変数ADBSYSLOGLV に0 を指定する ADBSYSLOGLV に0 を指定した場合，syslog に HADB のメッセージは出力されません。ただし，メッセージログファイルには，HADB のすべてのメッセージが出力されます。 **英語環境で使用する場合：** 文字コードは Unicode（UTF-8）になります。この環境変数にUTF8 を指定してください。 [マルチノード機能] マルチノード機能を使用する場合，この環境変数の指定値を，全ノードで一致させてください。
7	ADBMSGLOGSIZE	1 つのサーバメッセージログファイルの最大容量をメガバイト単位で設定できます。設定できる範囲は 1～2,000 です。HADB サーバでは，4 つのサーバメッセージログファイルが作成されます。 この環境変数を省略した場合，1 つのサーバメッセージログファイルの最大容量は 16 メガバイトとなります。
8	ADBSYSLOGLV	この環境変数を設定すると，syslog に出力されるメッセージを出力レベルに従って抑止できます。 syslog に出力されるメッセージを抑止したい場合，この環境変数を設定してください。出力レベルは，0～6 のどれかを指定してください。 syslog に出力されるメッセージの出力レベルの一覧については，「10.4.5　syslog へのメッセージの出力抑止」を参照してください。 なお，出力レベルに0～6 以外を指定した場合，エラーとなります。

項番	環境変数	設定する値
		また，この環境変数を設定しない場合は，出力レベルに6 が仮定されます。
9	ADBCLTDIR	HADB クライアントの各種ファイルが格納されているディレクトリを指定します。この環境変数には，サーバディレクトリを絶対パスで指定してください。 この環境変数は，次のどちらかの条件を満たす場合に設定してください。 • HADB サーバでadbsql コマンドを実行する場合 • HADB サーバで CLI 関数を使用した AP を開発または実行する場合
10	ADBCLTLANG	HADB クライアントで使用する文字コードを指定します。※ **日本語環境で使用する場合：** 次のどちらかの文字コードを選択します。 • Unicode（UTF-8） この環境変数にUTF8 を指定してください。 • Shift-JIS この環境変数にSJIS を指定してください。 **英語環境で使用する場合：** 文字コードは Unicode（UTF-8）になります。この環境変数にUTF8 を指定してください。 この環境変数は，次のどちらかの条件を満たす場合に設定してください。 • HADB サーバでadbsql コマンドを実行する場合 • HADB サーバで CLI 関数を使用した AP を開発または実行する場合
11	ADBSQLNULLCHAR	この環境変数には，adbsql コマンドの検索結果にナル値がある場合に，ナル値を表示する文字列（ナル値表示文字列）を指定します。0～32 バイトの文字列を指定できます。0 バイトの文字列を指定した場合は，ナル値の表示は空白になります。 この環境変数を省略した場合，ナル値は*（アスタリスク）で表示されます。 検索データ中に，アスタリスクがある場合や，ナル値を任意の文字で出力したい場合に，この環境変数を指定してください。 なお，マルチバイト文字を指定すると，検索結果の表示が乱れることがあります。 この環境変数は，HADB サーバでadbsql コマンドを実行する場合に設定を検討してください。
12	CLASSPATH	HADB サーバで Java の AP を開発または実行する場合（JDBC ドライバを使用する場合）に，この環境変数を指定します。この環境変数には，JAR ファイルの絶対パスを指定します。次のパスを指定してください。 • $ADBDIR/client/lib/adbjdbc8.jar JAR ファイルの格納先を上記から変更した場合は，変更先の絶対パスを指定してください。 JDBC ドライバが提供する JDBC 規格外のメソッド（接続情報設定および取得インタフェースのメソッド）など，JDBC ドライバが提供するクラスを直接操作する場合は，AP をコンパイルする前にこの環境変数を指定してください。

注※

環境変数LANG，ADBLANG，およびADBCLTLANG で選択する文字コードを合わせてください。

例 1：日本語環境で Unicode（UTF-8）を使用する場合

```
LANG="ja_JP.UTF-8"
ADBLANG=UTF8
ADBCLTLANG=UTF8
```

例 2：英語環境で Unicode（UTF-8）を使用する場合

```
LANG="en_US.UTF-8"
ADBLANG=UTF8
ADBCLTLANG=UTF8
```

8.5　サーバ定義の作成および変更方法

ここでは，サーバ定義の作成方法および変更方法について説明します。

> **メモ**
>
> クライアント定義の作成方法および変更方法については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』の『クライアント定義の作成』を参照してください。

8.5.1　サーバ定義の作成方法および格納先

HADB サーバの実行環境を，サーバ定義で指定します。

サーバ定義の作成方法，および格納先を次に示します。

手順

1. サーバ定義のひな形ファイルをコピーする

   サーバディレクトリの$ADBDIR/sample/conf ディレクトリに格納されているサーバ定義のひな形ファイル（server.def）をコピーして，サーバディレクトリの$ADBDIR/conf ディレクトリに格納してください。

2. コピーしたサーバ定義のひな形ファイルを編集する

   手順 1.で$ADBDIR/conf ディレクトリにコピーしたサーバ定義のひな形ファイル（server.def）をテキストエディタで開いて，編集してください。

   サーバ定義で指定するオペランドの指定形式，内容，およびサーバ定義の文法規則については，「7. サーバ定義の設計」を参照してください。

3. サーバ定義を上書き保存する

   サーバ定義ファイル（$ADBDIR/conf/server.def）の編集が完了したら，上書き保存してください。$ADBDIR/conf ディレクトリにserver.def として格納することで，HADB サーバ開始時にサーバ定義が読み込まれます。

8.5.2　サーバ定義の変更方法

サーバ定義の定義内容を変更する手順を次に示します。

サーバ定義の定義内容を変更する場合は，HADB サーバへの影響について十分検討してから，実施するようにしてください。

格納先

サーバ定義ファイル：$ADBDIR/conf/server.def

手順

1. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

2. サーバ定義の定義内容を変更する

   サーバ定義ファイル（$ADBDIR/conf/server.def）を，テキストエディタで開き，定義内容を変更してください。

   サーバ定義で指定するオペランドの指定形式，内容，およびサーバ定義の文法規則については，「7. サーバ定義の設計」を参照してください。

3. HADB サーバを開始する

   サーバ定義の内容を変更したら，adbstart コマンドを実行して HADB サーバを再度開始してください。

> **重要**
>
> HADB サーバを再起動しないと，サーバ定義の変更内容が有効になりません。

8.6　HADB サーバのバージョンアップ

ここでは，HADB サーバをバージョンアップする方法について説明します。

なお，HADB サーバのバージョンアップではなく，修正版 HADB サーバとの入れ替えの場合は，「8.8 修正版 HADB サーバとの入れ替え」を参照してください。修正版 HADB サーバについても，「8.8　修正版 HADB サーバとの入れ替え」で説明しています。

また，アプリケーションなどの動作確認テストのために，一時的に HADB サーバをバージョンアップし，動作確認テスト完了後に元のバージョンに戻す場合は，「8.9　一時的に HADB サーバのバージョンアップとバージョンダウンを行う方法（動作確認テスト用）」を参照してください。

8.6.1　バージョンアップ前に実施すること

HADB サーバをバージョンアップする前に実施することを説明します。

重要

HADB サーバをバージョンアップする前に，環境変数ADBLANG およびADBCLTLANG の指定値を変更しないでください。変更した場合，HADB サーバを正しくバージョンアップできないおそれがあります。

環境変数LANG については，次に示すように言語コードおよび国名コードの変更はできます。

- ja_JP.UTF-8 からen_US.UTF-8 への変更
- en_US.UTF-8 からja_JP.UTF-8 への変更

コードセットの変更（文字コード種別の変更）はできません。

なお，環境変数LANG の指定値を変更した場合，HADB クライアントの環境変数LANG の指定値も変更する必要があります。

(1)　ディクショナリ用 DB エリアおよびシステム表用 DB エリアの容量を確認する

HADB サーバをバージョンアップすると，ディクショナリ用 DB エリアおよびシステム表用 DB エリアの容量が増加することがあります。

そのため，DB エリアごとに，次の 2 点を確認してください。

- バージョンアップ時に増加する DB エリアの容量
- DB エリアを構成する DB エリアファイルが格納されているディスクの空き容量

確認した結果を比較して，DB エリアの容量が増加しても問題ないことを確認してください。それぞれの確認方法を次に示します。

■バージョンアップ時に増加する DB エリアの容量を確認する方法

ディクショナリ用 DB エリアの容量については，「5.11　ディクショナリ用 DB エリアの容量見積もり」を参照して，再見積もりしてください。

システム表用 DB エリアの容量については，「5.12　システム表用 DB エリアの容量見積もり」を参照して，再見積もりしてください。

■DB エリアを構成する DB エリアファイルが格納されているディスクの空き容量を確認する方法

DB エリアファイルが格納されているディスクの空き容量を確認する方法は，DB エリアファイルとして使用するファイルの種類によって異なります。

レギュラーファイルの場合

OS のdf コマンドを実行して，DB エリアファイルが格納されているディスクの空き容量を確認してください。実行例を次に示します。

実行例 1（ディクショナリ用 DB エリアファイルの場合）

```
df $DBDIR/ADBDIC
```

実行例 2（システム表用 DB エリアファイルの場合）

```
df $DBDIR/ADBSTBL
```

df コマンドに指定したファイルが格納されているディスクの空き容量が，キロバイト単位で表示されます。

ブロックスペシャルファイルの場合

adbdbstatus コマンドを実行して，「DB エリアのサマリ情報」を出力してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。実行例を次に示します。

実行例 1（ディクショナリ用 DB エリアの場合）

```
adbdbstatus -c dbarea -n ADBDIC -S K
```

実行例 2（システム表用 DB エリアの場合）

```
adbdbstatus -c dbarea -n ADBSTBL -S K
```

「DB エリアのサマリ情報」に出力される項目のうち，次の 2 つの項目を確認してください。

- KB_Total_segments
- KB_Used_segments

確認した値を基に，次に示す計算式でディスクの空き容量を求めてください。

計算式（単位：キロバイト）

```
DBエリアファイルが格納されているディスクの空き容量＝ KB_Total_segments－KB_Used_segments
```

確認した結果，DB エリアの容量増加によってディスクの空き容量が不足する場合は，「15.3.1　DB エリアファイルの容量増加が原因の場合」を参照して対処してください。

(2)　メモリ所要量を確認する

HADB サーバをバージョンアップすると，HADB サーバのメモリ所要量が増加することがあります。「6.3 HADB サーバのメモリ所要量の見積もり」を参照して，メモリ所要量を再見積もりしてください。

(3)　環境変数を確認する

HADB サーバをバージョンアップすると，環境変数の変更が必要になることがあります。「8.4　環境変数の設定」を参照して，環境変数の変更が必要かどうかを確認してください。

(4)　カーネルパラメタを確認する

HADB サーバをバージョンアップすると，カーネルパラメタの変更が必要になることがあります。「6.2 カーネルパラメタの見積もり」を参照して，カーネルパラメタの変更が必要かどうかを確認してください。

(5)　サーバ定義を確認する

- バージョン 02-00 の HADB サーバをバージョンアップする場合，サーバ定義adb_log_usrfile_num オペランドの指定値を確認してください。adb_log_usrfile_num オペランドに1 を指定している場合は，HADB サーバが正常終了しているときに2 以上に変更してください。そのあとで，adbstart コマンドで HADB サーバを開始し，adbstop コマンドで正常終了してください。
- バージョン 04-03 以前の HADB サーバからバージョンアップする場合，サーバ定義 adb_sys_memory_limit オペランドには次の計算式を満たす値を指定してください。次の計算式を満たさない値を指定すると，HADB サーバのバージョンアップに失敗します。その際，KFAA40002-E メッセージが出力されます。

  サーバ定義adb_sys_memory_limitオペランドの指定値
  ≦ カーネルパラメタvm.nr_hugepages × HugePagesの1ページ当たりのページサイズ

(6)　追加される予約語を確認する

HADB サーバをバージョンアップすると，予約語が追加されることがあります。

追加される予約語の一覧を次の表に示します。

表 8-11　追加される予約語の一覧

HADB サーバのバージョン	追加される予約語
02-01	CURRENT_USER_IS_DBA
03-00	MILLISECOND, MILLISECONDS, NANOSECOND, NANOSECONDS, PICOSECOND, PICOSECONDS

予約語を表や列の名称として SQL 文中に指定できません。追加される予約語と同じ文字列を SQL 文中で指定している場合，HADB サーバのバージョンアップ後に SQL 文がエラーとなります。

そのため，追加される予約語と同じ文字列を，SQL 文中で指定していないかどうかを確認してください。指定している場合は，予約語と同じ文字列を二重引用符（”）で囲んでください。詳細については，マニュアル『HADB　SQL リファレンス』の『基本項目』の『予約語』を参照してください。

(7)　スキーマの数を確認する

バージョン 02-00 の HADB サーバで作成したデータベースに，HADB ユーザが定義したスキーマが 1 つもない状態で HADB サーバをバージョンアップしようとすると，バージョンアップに失敗します。そのため，HADB ユーザが定義したスキーマが 1 つ以上あることを確認してください。

また，バージョン 02-00 の HADB サーバで作成したデータベースに，HADB ユーザが定義したスキーマの数が 30,001 以上ある状態で HADB サーバをバージョンアップしようとすると，バージョンアップに失敗します。そのため，DROP SCHEMA 文で不要なスキーマを削除して，スキーマの数を 30,000 以下にしておいてください。

(8)　スキーマ名を確認する

バージョン 02-00 の HADB サーバからバージョンアップした場合，ディクショナリ表（実表）の SQL_SCHEMATA 表に格納されているスキーマ名を基に，HADB ユーザが作成されます。作成される HADB ユーザの認可識別子およびパスワードは，スキーマ名と同じになります。

バージョンアップ後の HADB サーバに接続するために，定義されているスキーマ名を確認してください。スキーマ名を確認する SQL 文の指定例を次に示します。

指定例

```
SELECT ”SCHEMA_NAME” FROM ”MASTER”.”SQL_SCHEMATA”
```

なお，HADB サーバのバージョンアップが完了したら，「8.6.3　バージョンアップ後に実施すること」の「(3)　HADB ユーザのパスワードと権限を見直す（バージョン 02-00）」を参照して，必ず HADB ユーザのパスワードと権限を見直してください。

(9)　AP およびコマンドを停止する

HADB サーバをバージョンアップする前に，実行中の AP およびコマンドがある場合は，すべて終了してください。

(10)　更新不可状態の実表がないことを確認する

HADB サーバをバージョンアップする前に，更新不可状態の実表がないことを確認してください。また，更新不可状態の実表がある場合は，HADB サーバをバージョンアップする前に，更新不可状態を解除してください。

実表が更新不可状態かどうかを確認する方法については，「10.9.2　実表の状態や使用量を確認したい場合」の「(1)　実表が更新不可状態かどうかを確認する場合」を参照してください。

実表の更新不可状態を解除する方法については，「15.8.1　実表が更新不可状態になった場合」を参照してください。

重要

実表の更新不可状態を解除しないでバージョンアップした場合，バージョンアップ後にコマンドを再実行するとエラーになります。中断したコマンドを再実行できないため，実表の更新不可状態は解除されません。このとき，KFAA50243-E メッセージが出力されます。対処方法については，KFAA50243-E メッセージで説明されている対策に従ってください。

なお，次のコマンドを中断して更新不可状態になった実表にインデクスが定義されている場合，KFAA50243-E メッセージの対策を行うと，更新不可状態を解除する実表に定義されているすべてのインデクスが再作成されます。そのため，インデクスのデータ量によっては更新不可状態の解除に時間が掛かります。

- adbimport コマンド
- adbidxrebuild コマンド

(11)　更新行のカラム化機能を無効にする

「(14)　データベースのバックアップを取得する」で，システムデータだけのバックアップを取得する場合は，このタイミングで更新行のカラム化機能を無効にしてください。

更新行のカラム化機能を無効にする手順を次に示します。

1. 更新行のカラム化機能の状態を確認する
   -d オプションを指定してadbcolumnize コマンドを実行してください。
   出力結果のSTATUS 列にACTIVE（有効）が出力されている場合，手順 2.に進んでください。
2. 更新行のカラム化機能を無効にする
   --stop オプションを指定してadbcolumnize コマンドを実行してください。

更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

メモ

adbcolumnize コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbcolumnize（更新行のカラム化機能の管理）』を参照してください。

(12) HADB サーバを正常終了する

HADB サーバをバージョンアップする前に，adbstop コマンドで HADB サーバを正常終了してください。

すでに HADB サーバが終了している場合は，次に示す手順で HADB サーバが正常終了していることを確認してください。

1. adbls -d srv コマンドを実行する

   出力項目STATUS に出力される HADB サーバの状態が，"STOP"であることを確認してください。

   "STOP"である場合は，さらに手順 2.または手順 3.のどちらかに進んでください。

2. サーバメッセージログファイルを確認する

   サーバメッセージログファイル（$ADBDIR/spool/adbmessage*XX*.log※）を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

   注※

   XX は通番であり，01～04 のどれかになります。

3. syslog を確認する

   syslog を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

手順 1.の確認結果が"STOP"であり，かつ手順 2.または手順 3.の確認結果が"normally"である場合は，HADB サーバは正常終了しています。

もし，HADB サーバが正常終了していない場合は，adbstart コマンドで HADB サーバを開始したあとで，adbstop コマンドで正常終了してください。

[マルチノード機能]

マルチノード構成の HADB サーバが正常終了していない場合は，マルチノード構成の HADB サーバをいったん開始し，マルチノード構成の HADB サーバを正常終了させてください。このとき，HA モニタを停止させる必要はありません（monstop コマンドの実行は不要です）。

マルチノード構成の HADB サーバを開始して，正常終了させる方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」と「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照してください。

(13) サーバディレクトリのバックアップを取得する

HADB サーバをバージョンアップする前に，サーバディレクトリのバックアップを取得してください。

HADB サーバが異常終了したまま，誤って HADB サーバをバージョンアップしてしまうと，HADB サーバが開始できなくなります。そのため，サーバディレクトリのバックアップを取得しておいてください。

取得したバックアップは，HADB サーバのバージョンアップが完了したあとで削除します。

バックアップを取得する方法については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

［マルチノード機能］

マルチノード機能を使用している場合は，全ノードの HADB サーバでサーバディレクトリのバックアップを取得してください。

(14)　データベースのバックアップを取得する

HADB サーバをバージョンアップする前に，必ずデータベースのフルバックアップを取得してください。フルバックアップを取得する方法については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

取得したバックアップは，不要であれば HADB サーバのバージョンアップが完了したあとで削除します。

［マルチノード機能］

マルチノード機能を使用している場合は，「16.8.1　バックアップの取得方法」を参照して，バックアップを取得してください。

ただし，フルバックアップではなく，システムデータだけのバックアップでもよいケースがあります。システムデータだけのバックアップの場合，HADB サーバの管理情報など必要最低限のファイルだけをバックアップの対象とします（データ用 DB エリアファイルはバックアップの対象としません）。

■システムデータだけのバックアップでもよいケース

HADB サーバをバージョンアップしたあとに，すぐにバージョンアップ前の状態に戻す場合は，システムデータだけのバックアップでもよいです。例えば，バージョンアップのテストを実施するために，いったんバージョンアップした HADB サーバを，すぐに旧バージョンに戻す場合が該当します。

ただし，バージョンアップ後にデータベースを更新するような操作をした場合，旧バージョンに戻すことができなくなるため注意してください。そのため，バージョンアップ後は，次に示す操作以外は絶対にしないでください。

- SELECT 文，COMMIT 文，またはROLLBACK 文の実行
- adbls コマンドの実行

  adbls コマンド以外にも実行可能なコマンドがあります。実行可能なコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧』にある表『コマンドの一覧（システムデータだけのバックアップを使用したデータベースの回復)』を参照してください。

バージョンアップ後に上記以外の操作をした場合，システムデータだけのバックアップを使用して旧バージョンに戻しても，データベースの整合性が取れません。そのため，HADB サーバが開始できなかったり，データベースが壊れたりすることがあります。

システムデータだけのバックアップを取得する場合は，「表 10-6　バックアップを取得する必要があるファイルの一覧」の項番 1～6 のファイルのバックアップを取得してください。

［マルチノード機能］

システムデータだけのバックアップを取得する場合は，「表 16-9　バックアップを取得するファイルおよびディレクトリの一覧」の項番 1～6 のファイルのバックアップを取得してください。

メモ

- バックアップの取得方法は，フルバックアップも，システムデータだけのバックアップも同じ方法です。バックアップの取得対象が異なるだけです。
- フルバックアップからのデータベースの回復方法も，システムデータだけのバックアップからのデータベースの回復方法も同じ方法になります。

8.6.2　バージョンアップする

「8.6.1　バージョンアップ前に実施すること」で示したことを実施したら，次の手順に従って HADB サーバをバージョンアップしてください。

［マルチノード機能］

マルチノード機能を使用している場合は，全ノードの HADB サーバに対して，『手順 1.　HADB 管理者の OS アカウントで OS にログインする』から『手順 11.　HADB サーバのバージョンを確認する』をそれぞれ実施してください。そのあとで，『手順 12.　データベースをバージョンアップする』を実施してください。

手順

1. HADB 管理者の OS アカウントで OS にログインする

   今まで使用していた HADB 管理者の OS アカウントで，サーバマシンの OS にログインしてください。

   重要

   HADB 管理者の OS アカウントを変更しないでください。HADB 管理者の OS アカウントを変更した場合，HADB サーバを正しくバージョンアップできないおそれがあります。

2. インストールデータを格納するディレクトリを作成する

   HADB サーバのインストールデータを格納するディレクトリを作成します。

   ```
   mkdir /home/adbmanager/install
   ```

   上記の例では，インストールデータを格納するディレクトリとして/home/adbmanager/install を作成しています。

3. インストールデータを格納するディレクトリに書き込み権限を付与する

   インストールデータを格納するディレクトリ（この例では/home/adbmanager/install）に対して，HADB 管理者が書き込みできるように書き込み権限を付与してください。

```
chmod 755 /home/adbmanager/install
```

4. CD-ROM ファイルシステムをマウントする

HADB サーバのインストールデータが格納されている CD-ROM ファイルシステムを自動マウントしてください。

CD-ROM ファイルシステムが自動マウントされない場合，次のコマンドを実行して CD-ROM ファイルシステムをマウントしてください。

```
mount /dev/cdrom /media
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

> **重要**
>
> CD-ROM のディレクトリ名やファイル名は，ご使用のサーバマシンによっては，ここの説明内容と表示が異なることがあります。OS の ls コマンドを実行して，表示されたディレクトリ名をそのまま入力してください。

5. インストールデータをコピーする

マウントした CD-ROM ファイルシステムに格納されている次のファイルを，手順 2.で作成したインストールデータを格納するディレクトリにコピーしてください。

- tar.gz 形式のファイル
- adbinstall コマンドの実行形式ファイル
- !8A9_HADB ファイル

```
cp /media/hitachi_advanced_data_binder_server-$VER.tar.gz /home/adbmanager/install   ...a
cp /media/adbinstall /home/adbmanager/install                                        ...b
cp '/media/!8A9_HADB' /home/adbmanager/install                                       ...c
```

[説明]

a. tar.gz 形式のファイル（HADB サーバのプログラムの圧縮ファイル）をコピーします。

b. adbinstall コマンドの実行形式ファイル（HADB サーバのインストールコマンド）をコピーします。

c. !8A9_HADB ファイルをコピーします。

上記の例では，すべてのファイルを/home/adbmanager/install ディレクトリ下にコピーしています。

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

$VER は，HADB サーバのバージョンとリリース番号です。

> **重要**
>
> 上記の 3 つのファイルは，必ず同じディレクトリにコピーしてください。異なるディレクトリにコピーすると，HADB サーバのインストールが実行できません。

6. コピーしたファイルが破損していないことを確認する

手順 5.でコピーしたファイルが破損していないことを確認します。次のコマンドを実行してください。

```
cd /home/adbmanager/install
sha256sum -c /media/adbhashfile_server_sha256.txt
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

ファイルが破損しているおそれがある（ファイルのハッシュ値が異なる）場合は，エラーを示すメッセージが出力されます。この場合，手順 5.から作業をし直してください。

出力されるメッセージの詳細については，OS のマニュアルを参照してください。

7. 権限を付与する

HADB 管理者がadbinstall コマンド（HADB サーバのインストールコマンド）を実行できるように，adbinstall コマンドに対する実行権限を付与してください。

また，tar.gz 形式のファイルに対する読み取り権限も付与してください。

```
chmod 500 /home/adbmanager/install/adbinstall
chmod 400 /home/adbmanager/install/hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，HADB サーバのバージョンとリリース番号です。

8. HADB サーバをインストールする

adbinstall コマンドを実行して，HADB サーバをインストールしてください。

```
/home/adbmanager/install/adbinstall -s /HADB/server
```

下線部分には，手順 2.で作成したインストールデータを格納するディレクトリを指定します。

-s オプションには，使用中のサーバディレクトリ（この例では/HADB/server）の絶対パスを指定します。

重要

- KFAA91553-E メッセージが出力されて，adbinstall コマンドがエラーになった場合は，「■KFAA91553-E メッセージが出力された場合」を参照してください。
- adbinstall コマンドを実行した際にKFAA91558-W メッセージ（警告メッセージ）が出力された場合は，「■KFAA91558-W メッセージが出力された場合」を参照してください。

9. KFAA91554-Q メッセージに応答する

adbinstall コマンドを実行すると，ディレクトリを上書きしてよいかどうかを確認するKFAA91554-Q メッセージが出力されます。y またはY を入力して，ディレクトリを上書きしてください。

10. HADB サーバが正常にインストールされたことを確認する

サーバディレクトリ下にあるadbinstsv.log ファイルを開いて内容を確認してください。HADB サーバが正常にインストールされている場合は，次のどちらかの情報が記載されています。

- Hitachi Advanced Data Binder *vv-rr* (*yyyymmddhhmmss*)
- Hitachi Advanced Data Binder *vv-rr-/s* (*yyyymmddhhmmss*)

vv-rr および *vv-rr-/s* は，HADB サーバのバージョンを示しています。

yyyymmddhhmmss は，HADB サーバのインストールを実行した日付と時刻を示しています。

メモ

- adbinstsv.log ファイルがすでにある状態のときにadbinstall コマンドを実行すると，adbinstsv.log ファイルの内容が更新されます。また，adbinstall コマンドを複数回実行した場合，最後に実行した日付と時刻が記載されます。
- adbinstall コマンドのリターンコード（KFAA91552-I メッセージに出力されるリターンコード）が0 または4 以外の場合，adbinstsv.log ファイルの内容は更新されません。
- adbinstall コマンドの実行時にKFAA91555-I メッセージが出力された場合，adbinstsv.log ファイルの内容は更新されません。

11. HADB サーバのバージョンを確認する

次のコマンドを実行して，HADB サーバのバージョン情報を確認してください。

```
adbls -d ver
```

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードの HADB サーバで上記のコマンドを実行し，HADB サーバのバージョン情報を確認してください。

adbls -d ver コマンドを実行しても，バージョン情報が表示されない場合は，次に示す環境変数が正しく設定されていないおそれがあります。

- 環境変数ADBDIR にサーバディレクトリが指定されていない
- 環境変数PATH にサーバディレクトリ下のbin ディレクトリが指定されていない

環境変数については，「8.4　環境変数の設定」を参照してください。

12. データベースをバージョンアップする

次に示す HADB のコマンドを実行して，HADB サーバを開始してください。バージョンアップ処理が開始されます。

```
adbstart
```

adbstart コマンドの実行後，データベースをバージョンアップしてよいかを確認するKFAA91107-Q メッセージが出力されます。[Y] キーを押して，データベースをバージョンアップしてください。

adbstart コマンドが正常終了すれば，HADB サーバのバージョンアップは完了です。

[マルチノード機能]

マルチノード機能を使用している場合は，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照して，マルチノード構成の HADB サーバを開始してください。このとき，プライマリノードの HADB サーバを開始すると，データベースをバージョンアップしてよいかを確認するKFAA91107-Q メッセージが出力されます。[Y] キーを押して，データベースをバージョンアップしてください。

全ノードの HADB サーバが正常に開始すれば，HADB サーバのバージョンアップは完了です。

重要

adbstart コマンドが正常終了しないで，HADB サーバのバージョンアップが失敗した場合は，「8.6.4　バージョンアップに失敗した場合の対処方法」を参照してください。

■KFAA91553-E メッセージが出力された場合

次に示す場合は，adbinstall コマンドがエラーになり，KFAA91553-E メッセージが出力されます。この場合，対象のディレクトリに対する書き込み権限を HADB 管理者に付与してください。

- -s オプションに指定したディレクトリに対して，HADB 管理者が書き込み権限を持っていない場合
- インストールデータを格納するディレクトリに対して，HADB 管理者が書き込み権限を持っていない場合

■KFAA91558-W メッセージが出力された場合

HADB 管理者の OS アカウントではなく，root でadbinstall コマンドを実行した場合は，警告メッセージ（KFAA91558-W メッセージ）が出力されます。通常，HADB 管理者の OS アカウントで，adbinstall コマンドを実行します。そのため，KFAA91558-W メッセージが出力された場合は，root で adbinstall コマンドを実行して問題がないかどうかを確認してください。

問題がある場合は，KFAA91558-W メッセージが出力されたあとに出力されるKFAA91559-Q メッセージの入力要求に対して，n またはN を入力してください。そのあとで，HADB 管理者の OS アカウントで adbinstall コマンドを実行してください。

また，root 以外のスーパユーザでadbinstall コマンドを実行した場合は，KFAA91558-W メッセージは出力されません。

メモ

root は，OS のid -u コマンドを実行して表示される値が0 のユーザを指します。また，OS のsu コマンドを使用して，ほかの OS ユーザからroot に切り替えたあとで，OS のid -u コマンドを実行して表示される値が0 のときも含みます。

8.6.3　バージョンアップ後に実施すること

HADB サーバをバージョンアップしたあとで実施することを次に示します。

HADB サーバのバージョンアップ後，旧バージョンに戻したい場合は，「8.7　HADB サーバのバージョンダウン（旧バージョンに戻す方法）」を参照してください。

(1)　データベースのバックアップを取得する

バージョンアップが完了したデータベースのフルバックアップを取得することを推奨します。バージョンアップが完了したデータベースのフルバックアップを取得しておくことで，バックアップから回復する場合に，バージョンアップの完了直後の状態に回復できます。

(2) HADB クライアントをバージョンアップする

HADB サーバのバージョンアップが完了したら，HADB クライアントもバージョンアップしてください。HADB クライアントが，HADB サーバと同じバージョンでない場合，HADB サーバに接続できません。

HADB クライアントのバージョンアップについては，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

(3) HADB ユーザのパスワードと権限を見直す（バージョン 02-00）

バージョン 02-00 の HADB サーバからバージョンアップした場合，ディクショナリ表（実表）の SQL_SCHEMATA 表に格納されているスキーマ名を基に，HADB ユーザが作成されます。作成される HADB ユーザの認可識別子およびパスワードは，スキーマ名と同じになります。また，作成される HADB ユーザには，DBA 権限，CONNECT 権限，およびスキーマ定義権限が付与されます。

そのため，まず，すべての HADB ユーザのパスワードを変更してください。定義系 SQL の ALTER USER 文で変更できます。HADB ユーザのパスワードの変更については，「11.6.2　HADB ユーザのパスワードを変更する方法」を参照してください。

次に，すべての HADB ユーザが持っている権限を必ず確認してください。HADB ユーザが不要な権限を持っている場合は，定義系 SQL の REVOKE 文で権限を取り消してください。権限の取り消しについては，「11.7.3　HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法」を参照してください。

(4) HADB ユーザの権限を見直す（バージョン 03-00 より前のバージョン）

バージョン 03-00 より前のバージョンの HADB サーバからバージョンアップした場合，HADB ユーザに付与されているスキーマ定義権限が必要かどうかを確認してください。

スキーマ定義権限が不要な HADB ユーザ（例えば，自分ではデータベースを作成しないで，ほかの HADB ユーザが作成したデータベースの検索だけを行う HADB ユーザ）が存在する場合は，定義系 SQL の REVOKE 文で権限を取り消してください。権限の取り消しについては，「11.7.3　HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法」を参照してください。

(5) HADB ユーザの権限を見直す（バージョン 03-02 以降のバージョン）

バージョン 03-02 以降のバージョンの HADB サーバにバージョンアップした場合，次に示す HADB ユーザのアクセス権限を見直してください。

■アクセス権限を見直す必要がある HADB ユーザ

- バージョンアップする前に，定義系 SQL の GRANT 文の ALL PRIVILEGES 指定でアクセス権限を付与された HADB ユーザ

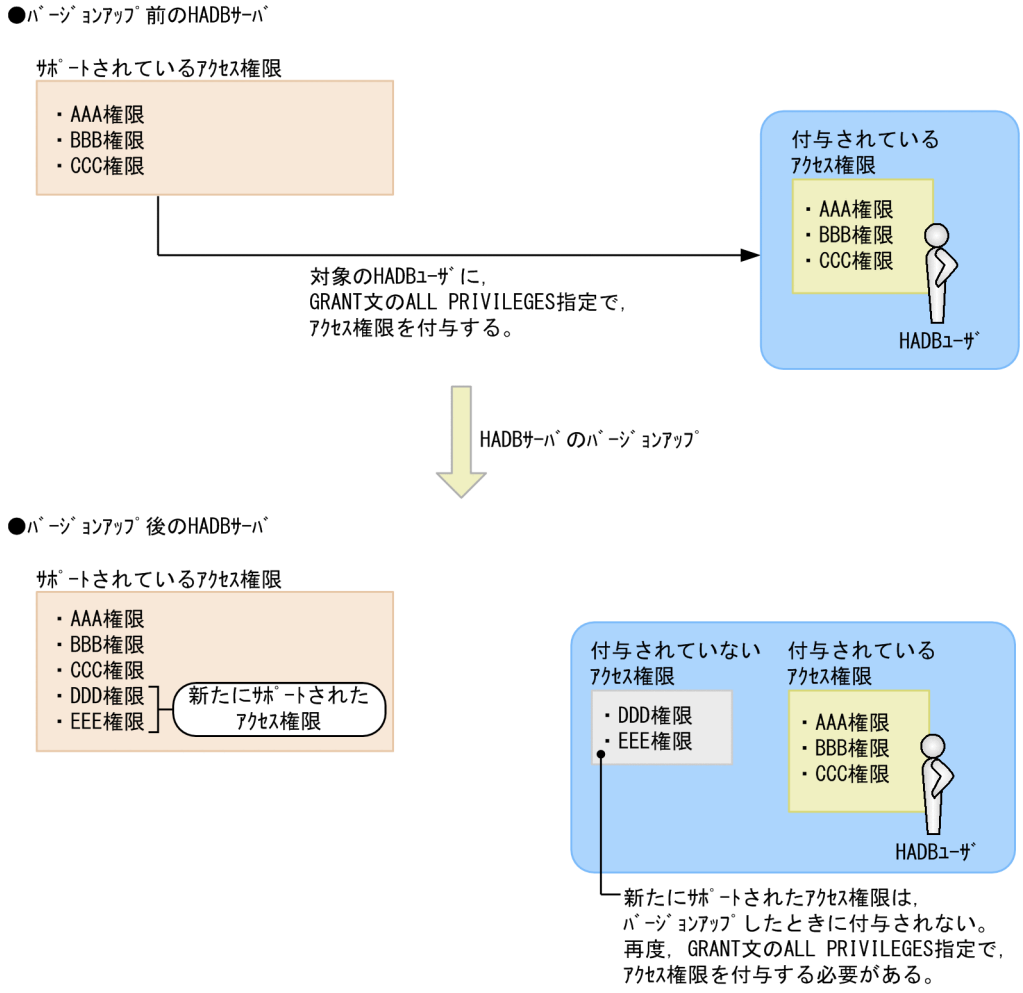
バージョン 03-02 以降のバージョンで，新しい種類のアクセス権限がサポートされた場合，HADB サーバをバージョンアップすると，次に示す HADB ユーザだけに，新しい種類のアクセス権限が自動的に付与されます。

- スキーマオブジェクト（ビュー表を除く）の所有者である HADB ユーザ
- ビュー表の所有者である HADB ユーザ（ただし，ビュー表のすべての基表の所有者である場合に限る）

そのため，アクセス権限を見直す必要がある HADB ユーザには，HADB サーバをバージョンアップしたときに新しいアクセス権限は付与されません。

アクセス権限を見直す必要がある HADB ユーザの概要を，次の図に示します。

図 8-2　アクセス権限を見直す必要がある HADB ユーザの概要



アクセス権限を見直す必要がある HADB ユーザに，新しい種類のアクセス権限も含めたすべてのアクセス権限を付与したい場合は，再度GRANT 文のALL PRIVILEGES 指定でアクセス権限を付与してください。アクセス権限の付与については，「11.8.1　HADB ユーザにアクセス権限を付与する方法」を参照してください。

(6)　ビュー表に対するアクセス権限を見直す（バージョン 03-06 以降のバージョン）

バージョン 03-06 より前の HADB サーバと，バージョン 03-06 以降の HADB サーバとでは，次に示す仕様差があります。

- バージョン 03-06 より前の HADB サーバの場合

基表に対するアクセス権限が新たに付与された場合でも，ビュー表に対するアクセス権限の伝搬が発生しません。

例えば，ビュー表A.V1 の基表X.T1 に対するINSERT 権限が新たに付与されても，ビュー表A.V1 に対するINSERT 権限は新たに付与されません。このようにアクセス権限の伝搬が発生しないため，基表に対して持っているアクセス権限の種類と，ビュー表に対して持っているアクセス権限の種類が一致しないことがあります。

- バージョン 03-06 以降の HADB サーバの場合

  基表に対するアクセス権限が新たに付与された場合，ビュー表に対するアクセス権限の伝搬が発生します。

  例えば，ビュー表A.V1 の基表X.T1 に対するINSERT 権限が新たに付与された場合，ビュー表A.V1 に対するINSERT 権限も新たに付与されます。このようにアクセス権限の伝搬が発生するため，基表に対して持っているアクセス権限の種類と，ビュー表に対して持っているアクセス権限の種類が一致します。

03-06 より前のバージョンから，03-06 以降のバージョンにバージョンアップしても，定義済みのビュー表については，アクセス権限の伝搬は自動的に行われません。そのため，基表に対するアクセス権限の変更を，ビュー表に対するアクセス権限にも反映させたい場合は，基表に対するアクセス権限を付与し直してください。基表に対するアクセス権限をビュー表の所有者に付与し直すと，アクセス権限の伝搬が発生し，基表に対するアクセス権限とビュー表に対するアクセス権限の種類が一致します。付与するアクセス権限は，ビュー表の所有者が元々持っていた種類のアクセス権限でも，持っていなかった種類のアクセス権限でも，どちらでもかまいません。

ビュー表に対するアクセス権限の伝搬については，「2.7.6　ビュー表に対するアクセス権限」を参照してください。

(7)　一意性制約違反状態かどうかを確認する

バージョン 02-02 より前のバージョンの HADB サーバからバージョンアップした場合，ユニークインデクスが一意性制約を満たしているかどうかが不明の状態になります。そのため，ユニークインデクスが一意性制約を満たしているかどうかを，次に示すどちらかの方法で確認してください。

- コマンドを使用して確認する

  1. ユニークインデクスを再作成する

     ユニークインデクスが定義された表に対して，adbidxrebuild コマンドを実行して，ユニークインデクスを再作成してください。adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

  2. 一意性制約違反状態かどうかを確認する

     adbdbstatus コマンドで，対象のユニークインデクスが一意性制約違反状態かどうかを確認してください。詳細については，「10.9.3　B-tree インデクスの状態や使用量を確認したい場合」を参照してください。

- SELECT 文を使用して確認する

対象のユニークインデクスが定義された表に対してSELECT 文を実行して，重複したキー値を検索してください。重複したキー値を検索するSELECT 文の指定例については，「15.9.2　一意性制約に違反した場合の対処方法（KFAA61205-W メッセージが出力された場合）」の「(2)　一意性制約違反状態を解除する方法」を参照してください。

ユニークインデクスが一意性制約を満たしていない場合（一意性制約違反状態の場合）は，「15.9.2　一意性制約に違反した場合の対処方法（KFAA61205-W メッセージが出力された場合）」を参照して，一意性制約違反状態を解除してください。

なお，ユニークインデクスが一意性制約を満たしているかどうかが不明の状態である場合，HADB サーバは，ユニークインデクスが一意性制約を満たしていると仮定して，アクセスパスを構築します。

(8)　ワード検索用のテキストインデクスを再作成する

次の条件をすべて満たす場合，**ワード検索用のテキストインデクス**を，adbidxrebuild コマンドで再作成する必要があります。

- バージョン 03-06 の HADB サーバを，バージョン 04-00 以降の HADB サーバにバージョンアップした場合
- バージョン 03-06 の HADB サーバで，ワード検索用のテキストインデクスを定義した場合（CREATE INDEX 文のINDEXTYPE に，TEXT WORDCONTEXT を指定した場合）

ワード検索用のテキストインデクスを定義したかどうかを確認する場合は，「付録 B.22　ディクショナリ表の検索」の「(34)　テキストインデクスに指定したインデクスオプションを調べる場合」を参照してください。

ワード検索用のテキストインデクスを定義していた場合は，adbidxrebuild コマンドで再作成してください。adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild (インデクスの再作成)』を参照してください。

ワード検索用のテキストインデクスをadbidxrebuild コマンドで再作成しないと，ワード検索用のテキストインデクスを使用できません。次の操作を行う場合に，エラーとなります（KFAA34009-E メッセージが出力されます）。

- ワード検索用のテキストインデクスを使用した検索を行う場合
- ワード検索用のテキストインデクスを更新する場合

(9)　サーバ定義の adbbuff オペランドの指定を見直す（バージョン 04-01 以降のバージョン）

バージョン 04-01 の HADB サーバから，テーブルスキャンバッファを使用するために指定するオプションとして，サーバ定義のadbbuff オペランドに-v オプションを追加しました。そのため，adbbuff オペランドの指定を見直す必要があります。

-v オプションの追加に伴い，テーブルスキャンバッファを使用する場合に指定するオプションは，次のとおり変更になります。

- バージョン 04-00 以前の HADB サーバの場合

  adbbuff オペランドの-k オプションを指定します。

- バージョン 04-01 以降の HADB サーバの場合

  adbbuff オペランドの-v オプションと-k オプション，どちらか 1 つを指定します。

  なお，テーブルスキャンバッファを使用する場合は，-v オプションの指定を推奨します。-v オプションを指定すると，-k オプションを指定したときよりも，テーブルスキャンバッファを効率良く使用できます。

そのため，バージョン 04-00 以前でadbbuff オペランドの-k オプションを指定していた場合，バージョン 04-01 以降では-v オプションに指定を変更してください。adbbuff オペランドの-v オプションについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」を参照してください。

また，-k オプションから-v オプションに指定を変更する場合，次の内容を確認してから-v オプションを指定してください。

■-v オプションの「*テーブルスキャンバッファに使用するメモリサイズ*」に指定する値

*テーブルスキャンバッファに使用するメモリサイズ*には，次の計算式で求めた値を指定してください。

*テーブルスキャンバッファに使用するメモリサイズ*に指定する値＝
　　バージョン04-00以前の-kオプションの指定値×4

上記の計算式で求めた値を指定すると，バージョン 04-00 以前に-k オプションで指定していたテーブルスキャンバッファと同じ大きさになります。

■-v オプションの「*リアルスレッドごとに使用するメモリサイズの上限値*」に指定する値

*リアルスレッドごとに使用するメモリサイズの上限値*には，adbbuff オペランドの-v オプションの説明に従った値を指定してください。adbbuff オペランドの-v オプションについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」を参照してください。

なお，バージョン 04-00 以前で指定していた-k オプションと同様に，各リアルスレッド間で均等にメモリを割り振りたい場合は，次の計算式で求めた値を指定してください。

*リアルスレッドごとに使用するメモリサイズの上限値*に指定する値＝
　　バージョン04-00以前の-kオプションの指定値×4
　　÷adb_sys_rthd_numオペランドの指定値

サーバ定義のadb_sys_rthd_num オペランドは，「7.2.2　性能に関するオペランド（set 形式）」を参照してください。

重要

バージョン 04-00 以前の HADB サーバの運用中に，テーブルスキャンバッファが不足していた場合，バージョン 04-01 以降の HADB サーバには，-v オプションの*リアルスレッド*

*ごとに使用するメモリサイズの上限値*を指定してください。このオプションを指定しないと，複数の SQL 文を同時に実行したときの処理性能が低下するおそれがあります。

(10) サーバ定義およびコマンドオプションの指定値を見直す（バージョン 04-01 以降のバージョン）

バージョン 04-01 の HADB サーバから，コマンドの実行時に使用する処理リアルスレッド数を指定するサーバ定義のadb_cmd_rthd_num オペランドを削除しました。そのため，adb_cmd_rthd_num オペランドを指定したままでadbstart コマンドを実行するとエラーになります（HADB サーバは起動されません）。このとき，KFAA52200-E メッセージが出力されます。サーバ定義にadb_cmd_rthd_num オペランドを指定していた場合は，adb_cmd_rthd_num オペランドの指定を削除してください。

adb_cmd_rthd_num オペランドの削除に伴い，次に示すコマンドの実行時に HADB サーバが使用する処理リアルスレッド数のデフォルト値（コマンドオプションの指定を省略した場合の値）が，adb_cmd_rthd_num オペランドからサーバ定義のadb_sql_exe_max_rthd_num オペランドに変更になります。必要に応じて，サーバ定義のadb_sql_exe_max_rthd_num オペランドや各コマンドオプションに指定するスレッド数を見直す必要があります。

■対象のコマンド

- adbarchivechunk コマンド
- adbexport コマンド
- adbgetcst コマンド
- adbidxrebuild コマンド
- adbimport コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド

■スレッド数を指定する各コマンドオプションの見直し

バージョン 04-01 の HADB サーバから，各コマンドで使用する処理リアルスレッドを指定するコマンドオプションが変更になりました。変更前のコマンドオプションを指定してコマンドを実行するとエラーになります。このとき，KFAA90500-E メッセージが出力されます。そのため，各コマンドのコマンドオプションを見直す必要があります。対象のコマンドとコマンドオプションの一覧を次に示します。

表 8-12　変更対象のコマンドとコマンドオプションの一覧

コマンド名	バージョン 04-00 以前のコマンドオプション名※	バージョン 04-01 以降のコマンドオプション名	バージョン 04-00 以前と同じスレッド数で動くようにするためのコマンドオプションの指定値
adbarchivechunk コマンド	次に示すアーカイブチャンクオプション • adb_arcv_scan_rthd_num	次に示すアーカイブチャンクオプション • adb_arcv_rthd_num	「(a)　adbarchivechunk コマンドの場合（アーカイブチャンクオプション

コマンド名	バージョン 04-00 以前のコマンドオプション名※	バージョン 04-01 以降のコマンドオプション名	バージョン 04-00 以前と同じスレッド数で動くようにするためのコマンドオプションの指定値
			adb_arcv_rthd_num の指定値の求め方)」を参照してください。
adbexport コマンド	次に示すエクスポートオプション • adb_export_scan_rthd_num	次に示すエクスポートオプション • adb_export_rthd_num	「(b)　adbexport コマンドの場合（エクスポートオプション adb_export_rthd_num の指定値の求め方)」を参照してください。
adbgetcst コマンド	次に示すコスト情報収集オプション • adb_getcst_scan_rthd_num	次に示すコスト情報収集オプション • adb_getcst_rthd_num	「(c)　adbgetcst コマンドの場合（コスト情報収集オプション adb_getcst_rthd_num の指定値の求め方)」を参照してください。
adbidxrebuild コマンド	次に示すインデクス再作成オプション • adb_idxrebuild_scan_rthd_num • adb_idxrebuild_sort_rthd_num • adb_idxrebuild_dividx_rthd_num	次に示すインデクス再作成オプション • adb_idxrebuild_rthd_num	「(d)　adbidxrebuild コマンドの場合（インデクス再作成オプション adb_idxrebuild_rthd_num の求め方)」を参照してください。
adbimport コマンド	次に示すインポートオプション • adb_import_dataload_rthd_num • adb_import_sort_rthd_num • adb_import_dividx_rthd_num	次に示すインポートオプション • adb_import_rthd_num	「(e)　adbimport コマンドの場合（インポートオプション adb_import_rthd_num の求め方)」を参照してください。
adbmergechunk コマンド	次に示すマージチャンクオプション • adb_mergechunk_scan_rthd_num • adb_mergechunk_sort_rthd_num • adb_mergechunk_dividx_rthd_num	次に示すマージチャンクオプション • adb_mergechunk_rthd_num	「(f)　adbmergechunk コマンドの場合（マージチャンクオプション adb_mergechunk_rthd_num の求め方)」を参照してください。
adbunarchivechunk コマンド	次に示すアンアーカイブチャンクオプション • adb_unarcv_dataload_rthd_num • adb_unarcv_sort_rthd_num • adb_unarcv_dividx_rthd_num	次に示すアンアーカイブチャンクオプション • adb_unarcv_rthd_num	「(g)　adbunarchivechunk コマンドの場合（アンアーカイブチャンクオプション adb_unarcv_rthd_num の求め方)」を参照してください。

注※

バージョン 04-01 で削除になります。

バージョン 04-00 までと同じスレッド数で動かすためのコマンドオプションの指定値を，コマンドごとに示します。

なお，指定値を求める計算式中の変数*DEFAULT_RTHD_NUM* は，次の表に示す計算式で求めてください。

表 8-13　変数 DEFAULT_RTHD_NUM の求め方

クライアントグループ機能		変数*DEFAULT_RTHD_NUM* の値
クライアントグループ機能を適用している	所属するコマンドグループをサーバ定義adbcltgrp オペランドに設定している	次の値のうち，どちらか小さい方の値 • コマンドグループを設定したサーバ定義adbcltgrp オペランドの-r オプションの指定値 • *A* −*B*
	所属するコマンドグループをサーバ定義adbcltgrp オペランドに設定していない	*A* −*B*
クライアントグループ機能を適用していない		*A*

（凡例）

A：サーバ定義adb_sys_rthd_num オペランドの指定値

B：クライアントグループを設定したサーバ定義adbcltgrp オペランドの-e オプションの指定値の合計

(a)　adbarchivechunk コマンドの場合（アーカイブチャンクオプション adb_arcv_rthd_num の指定値の求め方）

次の表に示す計算式で求めた値をアーカイブチャンクオプションadb_arcv_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

表 8-14　アーカイブチャンクオプション adb_arcv_rthd_num の指定値の求め方

項番	バージョン 04-00 以前のアーカイブチャンクオプションまたはサーバ定義の値		バージョン 04-01 以降のアーカイブチャンクオプション adb_arcv_rthd_num に指定する値
1	アーカイブチャンクオプション adb_arcv_scan_rthd_num の指定を省略した場合	サーバ定義adb_cmd_rthd_num オペランドの指定を省略した場合	次の値のうち，どちらか小さい方の値 • 変数*DEFAULT_RTHD_NUM*※の値 • *サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値*×2 + 1 サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値が 0 の場合は，1 として計算してください。
2		サーバ定義adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	*サーバ定義 adb_cmd_rthd_num オペランドの指定値*×2 + 1

項番	バージョン 04-00 以前のアーカイブチャンクオプションまたはサーバ定義の値	バージョン 04-01 以降のアーカイブチャンクオプション adb_arcv_rthd_num に指定する値
4	アーカイブチャンクオプションadb_arcv_scan_rthd_num に0 を指定した場合	次の値のうち，どちらか小さい方の値 • 変数*DEFAULT_RTHD_NUM*※の値 • *サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値*×2 + 1 サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値が 0 の場合は，1 として計算してください。
5	アーカイブチャンクオプションadb_arcv_scan_rthd_num に1 以上の値を指定した場合	*アーカイブチャンクオプション adb_arcv_scan_rthd_num の指定値*×2 + 1

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(b)　adbexport コマンドの場合（エクスポートオプション adb_export_rthd_num の指定値の求め方）

次の表に示す計算式で求めた値をエクスポートオプションadb_export_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

表 8-15　エクスポートオプション adb_export_rthd_num の指定値の求め方

項番	バージョン 04-00 以前のエクスポートオプションまたはサーバ定義の値		バージョン 04-01 以降のエクスポートオプション adb_export_rthd_num に指定する値
1	次のどちらかの場合 • エクスポートオプションadb_export_scan_rthd_num の指定を省略した場合 • エクスポートオプションadb_export_scan_rthd_num に0 を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定を省略した場合	次の値のうち，どちらか小さい方の値 • 変数*DEFAULT_RTHD_NUM*※の値 • *サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値*×2 + 1 サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値が 0 の場合は，1 として計算してください。
2		サーバ定義adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	*サーバ定義 adb_cmd_rthd_num オペランドの指定値*×2 + 1
4	エクスポートオプションadb_export_scan_rthd_num に1 以上の値を指定した場合		*エクスポートオプション adb_export_scan_rthd_num の指定値*×2 + 1

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(c) adbgetcst コマンドの場合（コスト情報収集オプション adb_getcst_rthd_num の指定値の求め方）

次の表に示す計算式で求めた値をコスト情報収集オプションadb_getcst_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

表 8-16　コスト情報収集オプション adb_getcst_rthd_num の指定値の求め方

項番	バージョン 04-00 以前のコスト情報収集オプションまたはサーバ定義の値		バージョン 04-01 以降のコスト情報収集オプション adb_getcst_rthd_num に指定する値
1	コスト情報収集オプション adb_getcst_scan_rthd_num の指定を省略した場合	サーバ定義adb_cmd_rthd_num オペランドの指定を省略した場合	変数*DEFAULT_RTHD_NUM*※の値
2		サーバ定義adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	*サーバ定義 adb_cmd_rthd_num オペランドの指定値* ＋ 1
4	コスト情報収集オプションadb_getcst_scan_rthd_num に0 を指定した場合		バージョン 04-00 以前と同じスレッド数でコマンドを実行させる指定値はありません。
5	コスト情報収集オプションadb_getcst_scan_rthd_num に1 以上の値を指定した場合		*コスト情報収集オプション adb_getcst_scan_rthd_num の指定値* ＋ 1

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(d) adbidxrebuild コマンドの場合（インデクス再作成オプション adb_idxrebuild_rthd_num の求め方）

次に示す計算式で求めた値をインデクス再作成オプションadb_idxrebuild_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

計算式

```
adb_idxrebuild_rthd_numの指定値
    ＝MAX（IDXREBUILD_SCAN_RTHD_NUM×2,
           IDXREBUILD_SORT_RTHD_NUM,
           IDXREBUILD_DIVIDX_RTHD_NUM）＋1
```

変数の説明

IDXREBUILD_SCAN_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-17　IDXREBUILD_SCAN_RTHD_NUM の求め方

項番	バージョン 04-00 以前のインデクス再作成オプションまたはサーバ定義の値		*IDXREBUILD_SCAN_RTHD_NUM* に代入する値
1	インデクス再作成オプション adb_idxrebuild_scan_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	↓(*変数 DEFAULT_RTHD_NUM の値*※－1）÷2↓
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	インデクス再作成オプション adb_idxrebuild_scan_rthd_num に0 を指定した場合		↓(*変数 DEFAULT_RTHD_NUM の値*※－1）÷2↓
5	インデクス再作成オプション adb_idxrebuild_scan_rthd_num に1 以上の値を指定した場合		インデクス再作成オプション adb_idxrebuild_scan_rthd_num の指定値

注※

変数 *DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

IDXREBUILD_SORT_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-18　IDXREBUILD_SORT_RTHD_NUM の求め方

項番	バージョン 04-00 以前のインデクス再作成オプションまたはサーバ定義の値		*IDXREBUILD_SORT_RTHD_NUM* に代入する値
1	インデクス再作成オプション adb_idxrebuild_sort_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数 *DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	インデクス再作成オプション adb_idxrebuild_sort_rthd_num に0 を指定した場合		変数 *DEFAULT_RTHD_NUM*※の値

項番	バージョン 04-00 以前のインデクス再作成オプションまたはサーバ定義の値	*IDXREBUILD_SORT_RTHD_NUM* に代入する値
5	インデクス再作成オプション adb_idxrebuild_sort_rthd_num に1 以上の値を指定した場合	インデクス再作成オプション adb_idxrebuild_sort_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

IDXREBUILD_DIVIDX_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-19　IDXREBUILD_DIVIDX_RTHD_NUM の求め方

項番	バージョン 04-00 以前のインデクス再作成オプションまたはサーバ定義の値		*IDXREBUILD_DIVIDX_RTHD_NUM* に代入する値
1	インデクス再作成オプション adb_idxrebuild_dividx_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数*DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	インデクス再作成オプション adb_idxrebuild_dividx_rthd_num に0 を指定した場合		変数*DEFAULT_RTHD_NUM*※の値
5	インデクス再作成オプション adb_idxrebuild_dividx_rthd_num に1 以上の値を指定した場合		インデクス再作成オプション adb_idxrebuild_dividx_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(e)　adbimport コマンドの場合（インポートオプション adb_import_rthd_num の求め方）

次に示す計算式で求めた値をインポートオプションadb_import_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

計算式

```
adb_import_rthd_numの指定値
    ＝MAX（IMPORT_DATALOAD_RTHD_NUM,
           IMPORT_SORT_RTHD_NUM,
           IMPORT_DIVIDX_RTHD_NUM）＋1
```

変数の説明

IMPORT_DATALOAD_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-20　IMPORT_DATALOAD_RTHD_NUM の求め方

項番	バージョン 04-00 以前のインポートオプションまたはサーバ定義の値		*IMPORT_DATALOAD_RTHD_NUM* に代入する値
1	インポートオプション adb_import_dataload_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	*変数 DEFAULT_RTHD_NUM の値*※－1
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	インポートオプションadb_import_dataload_rthd_num に0 を指定した場合		*変数 DEFAULT_RTHD_NUM の値*※－1
5	インポートオプションadb_import_dataload_rthd_num に1 以上の値を指定した場合		インポートオプションadb_import_dataload_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

IMPORT_SORT_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-21　IMPORT_SORT_RTHD_NUM の求め方

項番	バージョン 04-00 以前のインポートオプションまたはサーバ定義の値		*IMPORT_SORT_RTHD_NUM* に代入する値
1	インポートオプション adb_import_sort_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数*DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	

項番	バージョン 04-00 以前のインポートオプションまたはサーバ定義の値		*IMPORT_SORT_RTHD_NUM* に代入する値
3		サーバ定義adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	インポートオプションadb_import_sort_rthd_num に0 を指定した場合		変数*DEFAULT_RTHD_NUM*※の値
5	インポートオプションadb_import_sort_rthd_num に1 以上の値を指定した場合		インポートオプションadb_import_sort_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

IMPORT_DIVIDX_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-22　IMPORT_DIVIDX_RTHD_NUM の求め方

項番	バージョン 04-00 以前のインポートオプションまたはサーバ定義の値		*IMPORT_DIVIDX_RTHD_NUM* に代入する値
1	インポートオプションadb_import_dividx_rthd_num の指定を省略した場合	サーバ定義adb_cmd_rthd_num オペランドの指定を省略した場合	変数*DEFAULT_RTHD_NUM*※の値
2		サーバ定義adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	インポートオプションadb_import_dividx_rthd_num に0 を指定した場合		変数*DEFAULT_RTHD_NUM*※の値
5	インポートオプションadb_import_dividx_rthd_num に1 以上の値を指定した場合		インポートオプションadb_import_dividx_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(f) adbmergechunk コマンドの場合（マージチャンクオプション adb_mergechunk_rthd_num の求め方）

次に示す計算式で求めた値をマージチャンクオプションadb_mergechunk_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

計算式

```
adb_mergechunk_rthd_numの指定値
    ＝MAX（MERGECHUNK_SCAN_RTHD_NUM×2,
           MERGECHUNK_SORT_RTHD_NUM,
           MERGECHUNK_DIVIDX_RTHD_NUM）＋1
```

変数の説明

MERGECHUNK_SCAN_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-23　MERGECHUNK_SCAN_RTHD_NUM の求め方

項番	バージョン 04-00 以前のマージチャンクオプションまたはサーバ定義の値		*MERGECHUNK_SCAN_RTHD_NUM* に代入する値
1	マージチャンクオプション adb_mergechunk_scan_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	次の値のうち，どちらか大きい方の値 • ↓（*サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値*－1）÷2↓ • 1
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	マージチャンクオプションadb_mergechunk_scan_rthd_num に0 を指定した場合		次の値のうち，どちらか大きい方の値 • ↓（*サーバ定義 adb_sql_exe_max_rthd_num オペランドの指定値*－1）÷2↓ • 1
5	マージチャンクオプションadb_mergechunk_scan_rthd_num に1 以上の値を指定した場合		マージチャンクオプション adb_mergechunk_scan_rthd_num の指定値

MERGECHUNK_SORT_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-24　MERGECHUNK_SORT_RTHD_NUM の求め方

項番	バージョン 04-00 以前のマージチャンクオプションまたはサーバ定義の値		*MERGECHUNK_SORT_RTHD_NUM に代入する値*
1	マージチャンクオプション adb_mergechunk_sort_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数 *DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	マージチャンクオプションadb_mergechunk_sort_rthd_num に0 を指定した場合		変数 *DEFAULT_RTHD_NUM*※の値
5	マージチャンクオプションadb_mergechunk_sort_rthd_num に1 以上の値を指定した場合		マージチャンクオプション adb_mergechunk_sort_rthd_num の指定値

注※

変数 *DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

MERGECHUNK_DIVIDX_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-25　MERGECHUNK_DIVIDX_RTHD_NUM の求め方

項番	バージョン 04-00 以前のマージチャンクオプションまたはサーバ定義の値		*MERGECHUNK_DIVIDX_RTHD_NUM* に代入する値
1	マージチャンクオプション adb_mergechunk_dividx_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数 *DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	マージチャンクオプション adb_mergechunk_dividx_rthd_num に0 を指定した場合		変数 *DEFAULT_RTHD_NUM*※の値
5	マージチャンクオプション adb_mergechunk_dividx_rthd_num に1 以上の値を指定した場合		マージチャンクオプション adb_mergechunk_dividx_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(g)　adbunarchivechunk コマンドの場合（アンアーカイブチャンクオプション adb_unarcv_rthd_num の求め方）

次に示す計算式で求めた値をアンアーカイブチャンクオプションadb_unarcv_rthd_num に指定すると，バージョン 04-00 以前と同じスレッド数でコマンドを実行できます。

計算式

```
adb_unarcv_rthd_numの指定値
    ＝MAX（UNARCHIVE_DATALOAD_RTHD_NUM,
           UNARCHIVE_SORT_RTHD_NUM,
           UNARCHIVE_DIVIDX_RTHD_NUM） ＋1
```

変数の説明

UNARCHIVE_DATALOAD_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-26　UNARCHIVE_DATALOAD_RTHD_NUM の求め方

項番	バージョン 04-00 以前のアンアーカイブチャンクオプションまたはサーバ定義の値		*UNARCHIVE_DATALOAD_RTHD_NUM* に代入する値
1	アンアーカイブチャンクオプション adb_unarcv_dataload_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	*変数 DEFAULT_RTHD_NUM の値*※－1
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	アンアーカイブチャンクオプション adb_unarcv_dataload_rthd_num に0 を指定した場合		*変数 DEFAULT_RTHD_NUM の値*※－1
5	アンアーカイブチャンクオプション adb_unarcv_dataload_rthd_num に1 以上の値を指定した場合		アンアーカイブチャンクオプション adb_unarcv_dataload_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

UNARCHIVE_SORT_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-27　UNARCHIVE_SORT_RTHD_NUM の求め方

項番	バージョン 04-00 以前のアンアーカイブチャンクオプションまたはサーバ定義の値		*UNARCHIVE_SORT_RTHD_NUM* に代入する値
1	アンアーカイブチャンクオプション adb_unarcv_sort_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数*DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	アンアーカイブチャンクオプション adb_unarcv_sort_rthd_num に0 を指定した場合		変数*DEFAULT_RTHD_NUM*※の値
5	アンアーカイブチャンクオプション adb_unarcv_sort_rthd_num に1 以上の値を指定した場合		アンアーカイブチャンクオプション adb_unarcv_sort_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

UNARCHIVE_DIVIDX_RTHD_NUM

次の表に示す計算式から求めてください。

表 8-28　UNARCHIVE_DIVIDX_RTHD_NUM の求め方

項番	バージョン 04-00 以前のアンアーカイブチャンクオプションまたはサーバ定義の値		*UNARCHIVE_DIVIDX_RTHD_NUM* に代入する値
1	アンアーカイブチャンクオプション adb_unarcv_dividx_rthd_num の指定を省略した場合	サーバ定義 adb_cmd_rthd_num オペランドの指定を省略した場合	変数*DEFAULT_RTHD_NUM*※の値
2		サーバ定義 adb_cmd_rthd_num オペランドに0 を指定した場合	
3		サーバ定義 adb_cmd_rthd_num オペランドに1 以上の値を指定した場合	サーバ定義adb_cmd_rthd_num オペランドの指定値
4	アンアーカイブチャンクオプション adb_unarcv_dividx_rthd_num に0 を指定した場合		変数*DEFAULT_RTHD_NUM*※の値

項番	バージョン 04-00 以前のアンアーカイブチャンクオプションまたはサーバ定義の値	*UNARCHIVE_DIVIDX_RTHD_NUM* に代入する値
5	アンアーカイブチャンクオプション adb_unarcv_dividx_rthd_num に1 以上の値を指定した場合	アンアーカイブチャンクオプション adb_unarcv_dividx_rthd_num の指定値

注※

変数*DEFAULT_RTHD_NUM* は「表 8-13　変数 DEFAULT_RTHD_NUM の求め方」を参照して求めてください。

(11)　更新行のカラム化機能を有効にする

「8.6.1　バージョンアップ前に実施すること」の「(11)　更新行のカラム化機能を無効にする」で，更新行のカラム化機能を無効にした場合は，更新行のカラム化機能を有効にしてください。

--start オプションを指定してadbcolumnize コマンドを実行してください。更新行のカラム化機能が有効になります。

メモ

adbcolumnize コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbcolumnize（更新行のカラム化機能の管理)』を参照してください。

8.6.4　バージョンアップに失敗した場合の対処方法

HADB サーバのバージョンアップに失敗した場合の対処方法について説明します。

対処方法

1. メッセージを確認する

   サーバメッセージログファイルをテキストエディタで開き，出力されているエラーメッセージを確認してください。サーバメッセージログファイルは，サーバディレクトリ下のspool/adbmessage*XX*.log になります。ファイル名の*XX* は通番であり，01～04 のどれかになります。最終更新日時が最も新しいファイルを開いてください。

   次に示すメッセージが出力されていない場合は，手順 2.に進んでください。

   次に示すメッセージが出力されている場合は，参照先に従って対処してください。

   - KFAA30811-E メッセージ

     「(1)　KFAA30811-E メッセージが出力された場合」を参照してください。

   - KFAA41210-E メッセージ

     「(2)　KFAA41210-E メッセージが出力された場合」を参照してください。

   - KFAA50038-E メッセージ

     「(3)　KFAA50038-E メッセージが出力された場合」を参照してください。

- KFAA51400-E メッセージ

  「(4)　KFAA51400-E メッセージが出力された場合」を参照してください。

2. メッセージの対策欄に従って対処する

   出力されているエラーメッセージの ID を基に，マニュアル『HADB　メッセージ』を参照してください。対策欄に従って，対処してください。

   対策欄に「弊社問い合わせ窓口に連絡してください。」と記載されている場合は，手順 3.に進んでください。

3. adbinfoget コマンドを実行する

   対策欄に「弊社問い合わせ窓口に連絡してください。」と記載されている場合は，障害の原因を調査するためのトラブルシュート情報を取得する必要があります。

   この場合，adbinfoget コマンドを実行して，トラブルシュート情報を取得してください。操作手順については，「14.2　トラブルシュート情報の取得（adbinfoget コマンド）」を参照してください。

   取得したトラブルシュート情報は，弊社問い合わせ窓口に連絡する際，あわせて送付してください。

> **メモ**
>
> 障害情報（コアファイル）は，次のどれかのディレクトリ下に出力されます。
>
> - /var/lib/systemd/coredump ディレクトリ
> - サーバ定義のadb_core_path オペランドに指定したディレクトリ
> - $DBDIR/SPOOL ディレクトリ

対処後，再度 HADB サーバを開始してもエラーとなり，バージョンアップに失敗した場合は，データベースをバックアップから回復する必要があります。「(5)　データベースをバックアップから回復する」を参照してください。

(1)　KFAA30811-E メッセージが出力された場合

定義されているスキーマの数が 30,001 以上あるため，バージョンアップに失敗しました。HADB サーバのバージョンを，いったん旧バージョンに戻してから，対策する必要があります。

HADB サーバを旧バージョンに戻して対策する手順を次に示します。

手順

1. サーバディレクトリを削除する

   「8.12.1　アンインストール手順」の「(1)　サーバディレクトリの削除」を参照して，サーバディレクトリを削除してください。

2. 旧バージョンのサーバディレクトリをバックアップから回復する

   あらかじめ取得しておいたバックアップから，旧バージョンのサーバディレクトリをコピーして，手順 1.で削除した格納先に戻してください。

3. HADB サーバを開始する

   adbstart コマンドで，HADB サーバを開始してください。

4. スキーマの数を 30,000 以下に減らす

   DROP SCHEMA 文で不要なスキーマを削除して，定義されているスキーマの数を 30,000 以下にしてください。スキーマの数が 30,001 以上あると，バージョンアップできません。

5. HADB サーバを終了する

   adbstop コマンドで HADB サーバを終了してください。

6. HADB サーバが正常終了したことを確認する

   「8.6.1　バージョンアップ前に実施すること」の「(12)　HADB サーバを正常終了する」を参照して，HADB サーバが正常終了したかどうかを確認してください。

HADB サーバが正常終了したことを確認したら，「8.6.2　バージョンアップする」を参照して，再度バージョンアップをしてください。

(2)　KFAA41210-E メッセージが出力された場合

ユーザログファイル数が不足しているため，バージョンアップに失敗しました。HADB サーバのバージョンを，いったん旧バージョンに戻してから，対策する必要があります。

HADB サーバを旧バージョンに戻して対策する手順を次に示します。

手順

1. サーバディレクトリを削除する

   「8.12.1　アンインストール手順」の「(1)　サーバディレクトリの削除」を参照して，サーバディレクトリを削除してください。

2. 旧バージョンのサーバディレクトリをバックアップから回復する

   あらかじめ取得しておいたバックアップから，旧バージョンのサーバディレクトリをコピーして，手順 1.で削除した格納先に戻してください。

3. サーバ定義adb_log_usrfile_num オペランドの指定値を確認する

   サーバ定義adb_log_usrfile_num オペランドの指定値に，1 が指定されているかどうかを確認してください。1 が指定されている場合は，2 以上に変更してください。

4. HADB サーバを開始・終了する

   adbstart コマンドで，HADB サーバを開始してください。そのあとで，adbstop コマンドで HADB サーバを終了してください。

5. HADB サーバが正常終了したことを確認する

   「8.6.1　バージョンアップ前に実施すること」の「(12)　HADB サーバを正常終了する」を参照して，HADB サーバが正常終了したかどうかを確認してください。

HADB サーバが正常終了したことを確認したら，「8.6.2　バージョンアップする」を参照して，再度バージョンアップをしてください。

(3)　KFAA50038-E メッセージが出力された場合

KFAA50038-E メッセージの変数*aa....aa* に出力された内容によって，対処方法が異なります。

- 変数*aa....aa* がof an abnormal termination the last time it ended の場合
- 変数*aa....aa* がthe adbstart command was never been executed in previous versions の場合
- 変数*aa....aa* がof using the multiple node facility の場合

(a)　変数 aa....aa が of an abnormal termination the last time it ended の場合

HADB サーバが異常終了していたため，バージョンアップに失敗しました。HADB サーバのバージョンを，いったん旧バージョンに戻してから，対策する必要があります。

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードの HADB サーバを，旧バージョンに戻してください。

HADB サーバを旧バージョンに戻して対策する手順を次に示します。

手順

1. サーバディレクトリを削除する

   「8.12.1　アンインストール手順」の「(1)　サーバディレクトリの削除」を参照して，サーバディレクトリを削除してください。

   [マルチノード機能]

   マルチノード機能を使用している場合は，全ノードで実施してください。

2. 旧バージョンのサーバディレクトリをバックアップから回復する

   あらかじめ取得しておいたバックアップから，旧バージョンのサーバディレクトリをコピーして，手順1.で削除した格納先に戻してください。

   [マルチノード機能]

   マルチノード機能を使用している場合は，全ノードで実施してください。

3. HADB サーバを開始・終了する

   adbstart コマンドで，HADB サーバを開始してください。そのあとで，adbstop コマンドで HADB サーバを終了してください。

   [マルチノード機能]

   マルチノード機能を使用している場合は，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照して，マルチノード構成の HADB サーバを開始してください。そのあとで，「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照して，マルチノード構成の HADB サーバを終了してください。

4. HADB サーバが正常終了したことを確認する

   「8.6.1　バージョンアップ前に実施すること」の「(12)　HADB サーバを正常終了する」を参照して，HADB サーバが正常終了したかどうかを確認してください。

HADB サーバが正常終了したことを確認したら，「8.6.2　バージョンアップする」を参照して，再度バージョンアップをしてください。

(b)　変数 aa....aa が the adbstart command was never been executed in previous versions の場合

旧バージョンのときに一度も HADB サーバを開始していないため，バージョンアップに失敗しました。

adbinit コマンドを実行して，データベースの初期設定をする必要があります。データベースの初期設定については，「9.2　データベースを初期設定する（データ用 DB エリアを作成する）」を参照してください。

(c)　変数 aa....aa が of using the multiple node facility の場合

バージョン 03-00 より前のバージョンの HADB サーバで，バージョンアップ，およびマルチノード機能を使用するシステムへの変更を同時に実行しようとしたため，バージョンアップに失敗しました。

バージョンアップを実施する前に，サーバ定義にマルチノード機能のオペランドを指定すると，バージョンアップに失敗します。バージョンアップを実施したあとに，マルチノード機能を使用するシステムへの変更を実施してください。

手順

1. サーバ定義adb_sys_multi_node_info オペランドの指定を削除する

   マルチノード機能を使用しないようにするため，サーバ定義adb_sys_multi_node_info オペランドの指定を削除してください。

   または，サーバ定義adb_sys_multi_node_info オペランドをコメントにしてください。オペランドの指定行の先頭にシャープ（#）を記述すると，その行全体の指定がコメントになります。

2. HADB サーバを開始する

   マルチノード機能を使用しない状態で，adbstart コマンドを実行して，HADB サーバを開始してください。バージョンアップ処理が開始されます。

   adbstart コマンドの実行後，データベースをバージョンアップしてよいかを確認するKFAA91107-Q メッセージが出力されます。[Y] キーを押して，データベースをバージョンアップしてください。

   adbstart コマンドが正常終了すると，HADB サーバのバージョンアップは完了です。

3. HADB サーバを正常終了する

   adbstart コマンドが正常終了し，バージョンアップが完了したら，HADB サーバを正常終了します。adbstop コマンドを実行して，HADB サーバを正常終了してください。

4. サーバ定義adb_sys_multi_node_info オペランドを指定する

   バージョンアップが完了したら，マルチノード機能を使用できます。手順 1.で削除したサーバ定義adb_sys_multi_node_info オペランドを指定してください。

5. マルチノード構成の HADB サーバを開始する

サーバ定義adb_sys_multi_node_info オペランドを指定したら，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照して，マルチノード構成の HADB サーバを開始してください。

(4)　KFAA51400-E メッセージが出力された場合

バージョン 02-00 の HADB サーバで作成したデータベースに，スキーマが 1 つもない状態でバージョンアップしようとしたため，バージョンアップに失敗しました。

adbinit コマンドを実行して，データベースの初期設定をする必要があります。データベースの初期設定については，「9.2　データベースを初期設定する（データ用 DB エリアを作成する）」を参照してください。

(5)　データベースをバックアップから回復する

HADB サーバのバージョンアップに失敗して，出力されたメッセージに従って対処したあとで，再度 HADB サーバを開始してもエラーとなった場合は，データベースをバックアップから回復する必要があります。

データベースをバックアップから回復して，旧バージョンの HADB サーバを再構築する手順を次に示します。

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードの HADB サーバに対して実施してください。

手順

1. サーバディレクトリを削除する

「8.12.1　アンインストール手順」の「(1)　サーバディレクトリの削除」を参照して，サーバディレクトリを削除してください。

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードで実施してください。

2. 旧バージョンのサーバディレクトリをバックアップから回復する

あらかじめ取得しておいたバックアップから，旧バージョンのサーバディレクトリをコピーして，手順 1.で削除した格納先に戻してください。

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードで実施してください。

3. DB ディレクトリをバックアップから回復する

「10.3.2　バックアップからの回復方法」を参照して，DB ディレクトリをバックアップから回復してください。

[マルチノード機能]

マルチノード機能を使用している場合は，「16.8.2　バックアップからの回復方法」を参照して，DB ディレクトリをバックアップから回復してください。

4. サーバ定義adb_db_path オペランドの指定値を確認する

   手順 3.で回復した DB ディレクトリのパス名が，サーバ定義adb_db_path オペランドに指定されているかどうか確認してください。

5. HADB サーバを開始・終了する

   adbstart コマンドで，HADB サーバを開始してください。そのあとで，adbstop コマンドで HADB サーバを終了してください。

   [マルチノード機能]

   マルチノード機能を使用している場合は，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照して，マルチノード構成の HADB サーバを開始してください。そのあとで，「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照して，マルチノード構成の HADB サーバを終了してください。

6. HADB サーバが正常終了したことを確認する

   「8.6.1　バージョンアップ前に実施すること」の「(12)　HADB サーバを正常終了する」を参照して，HADB サーバが正常終了したかどうかを確認してください。

HADB サーバが正常終了していれば，旧バージョンの HADB サーバを再構築できています。HADB サーバのバージョンアップが必要な場合は，「8.6.2　バージョンアップする」を参照してください。

8.6.5　バージョンアップ時の注意事項

ここでは，HADB サーバをバージョンアップする場合の注意事項について説明します。

(1)　コスト情報の再収集

バージョン 04-03 より前の HADB サーバで収集したコスト情報は，バージョン 04-03 以降の HADB サーバでは使用できません。そのため，04-03 より前のバージョンから，04-03 以降のバージョンに HADB サーバをバージョンアップした場合は，adbgetcst コマンドでコスト情報を収集し直してください。

(2)　値式と集合演算の結果

バージョン 03-00 より前のバージョンの HADB サーバをバージョンアップすると，次に示す値式と集合演算の結果のデータ型，データ長および結果の値が，バージョンアップ前とは変わります。バージョンアップの前後で，結果が変わる項目を次の 2 つの表に示します。

表 8-29　結果のデータ型とデータ長が変わる項目

項番	結果のデータ型とデータ長が変わる項目		説明
1	集合関数	PERCENTILE_CONT	結果のデータ型が，被集約引数のデータ型からDOUBLE PRECISION 型に変わります。
2		MEDIAN	
3	0 バイトの定数		CHAR(0)からVARCHAR(1)の実長 0 のデータに変わります。

項番	結果のデータ型とデータ長が変わる項目		説明
4	CASE 式		結果のデータ型が文字データの場合，結果のデータ型がVARCHAR型に変わります。
5	スカラ関数	COALESCE	結果のデータ型が文字データの場合，結果のデータ型がVARCHAR型に変わります。
6		DECODE	
7		GREATEST	
8		LEAST	
9		NULLIF	
10		CEIL	対象データがDECIMAL 型の場合，結果のデータ型がDECIMAL（対象データの精度，対象データの位取り）から，DECIMAL（対象データの精度，0）に変わります。
11		FLOOR	
12		ROUND	対象データがDECIMAL 型（精度が 38 の場合を除く）の場合，結果のデータ型がDECIMAL（対象データの精度，対象データの位取り）から，DECIMAL（対象データの精度＋ 1，対象データの位取り）に変わります。
13	集合演算	UNION	集合演算の結果，導出される列のデータ型が変わります。 各問合せ一次子の結果導出される列で対応する列のデータ型がすべてCHAR 型であり，かつ最大長がすべて等しい場合は，結果のデータ型はCHAR 型になります。それ以外の場合は，結果のデータ型はVARCHAR 型になります。
14		UNION ALL	

表 8-30　結果の値が変わる項目

項番	結果の値が変わる項目		説明
1	スカラ関数	CAST	変換後の長さが，結果のデータ長よりも大きい場合は，エラーとしないで小数部を切り捨てます（最近接偶数への丸め）。
2		CONVERT（数値書式の指定なし，DOUBLE PRECISION 型から文字データへの変換）	
3		SUBSTR	次の項目について，ナル値ではなく，実長 0 バイトのデータを返却します。 • 結果の文字列の長さが 0 の場合 • 抽出元に実長 0 バイト（文字）のデータを指定した場合 • 抽出文字数が 0 の場合 • 開始位置の絶対値が抽出元の文字数より大きい場合 次の項目について，エラーではなく，開始位置を1 として扱います。 • 開始位置に0 を指定した場合 次の項目について，ナル値ではなく，抽出できるデータを返却します。 • 抽出文字数に 32,000 を超える値を指定した場合
4		LEFT	
5		RIGHT	

項番	結果の値が変わる項目		説明
6		TRIM	次の項目について，ナル値ではなく，実長 0 バイトのデータを返却します。 • 対象データに実長 0 バイト（文字）のデータを指定した場合 • 対象データに指定した文字列がすべて取り除かれた場合
7		LTRIM	
8		RTRIM	

(3)　以前のバージョンで定義したビュー表の検索結果

バージョン 03-00 より前のバージョンの HADB サーバで定義したビュー表に，次に示す項目が含まれる場合は，「(2)　値式と集合演算の結果」で示す内容に従って，結果のデータ型およびデータ長が求められます。

そのあとで，バージョン 03-00 より前のバージョンの HADB サーバで求めたデータ型およびデータ長となるように，スカラ関数CONVERT が実行されます。

対象の項目

- 集合関数
  - PERCENTILE_CONT
  - MEDIAN
- 0 バイトの定数
- CASE 式
- スカラ関数
  - COALESCE
  - CEIL
  - DECODE
  - FLOOR
  - GREATEST
  - LEAST
  - NULLIF
  - ROUND（数学関数）

(4)　マルチノード機能を使用するシステムに変更する場合

バージョンアップ時にマルチノード機能を使用するシステムに変更する場合，バージョンアップとマルチノード機能を使用するシステムへの変更は一度にできません。バージョンアップ前に，サーバ定義にマルチノード機能のオペランドを指定すると，バージョンアップに失敗します（KFAA50038-E メッセージが出力されます)。バージョンアップを実施したあとに，マルチノード機能を使用するシステムへの変更を実施してください。

KFAA50038-E メッセージが出力された場合の対処方法については，「8.6.4　バージョンアップに失敗した場合の対処方法」の「(3)　KFAA50038-E メッセージが出力された場合」の「(c)　変数 aa....aa が of using the multiple node facility の場合」を参照してください。

(5)　バージョンアップ時に行われるビュー表の再作成

バージョン 05-05 より前の HADB サーバを，バージョン 05-05 の HADB サーバにバージョンアップする場合，次の条件をすべて満たすビュー表は，バージョンアップ時に再作成されます。また，再作成に失敗したビュー表は無効化されます。

■バージョンアップ時に再作成されるビュー表の条件

次の条件をすべて満たすビュー表が再作成されます。

- バージョンアップで変更されるディクショナリ表とシステム表に依存するビュー表

  ディクショナリ表とシステム表に依存するビュー表を調べる場合は，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

  バージョンアップで変更されるディクショナリ表とシステム表は，「表 8-31　バージョンアップ時に変更されるディクショナリ表とシステム表の一覧（その 1）」，および「表 8-32　バージョンアップ時に変更されるディクショナリ表とシステム表の一覧（その 2）」を参照してください。

- バージョンアップ時点で有効なビュー表（無効化されていないビュー表）

  ビュー表が有効かどうかを調べる場合は，「付録 B.22　ディクショナリ表の検索」の「(22)　ビュー表が無効化されているかどうかを調べる場合」を参照してください。

バージョンアップ時に行われるビュー表の再作成が成功した場合は，KFAA51311-I メッセージが出力されます。

バージョンアップ時に行われるビュー表の再作成が失敗した場合は，KFAA51312-W メッセージが出力されます。そして，再作成に失敗したビュー表が無効化されます。そのあとで，バージョンアップ処理が続行されます。

KFAA51312-W メッセージが出力された場合は，バージョンアップが完了したあとで，KFAA51312-W メッセージの直前に出力されたメッセージを確認してください。そして，ビュー表の再作成が失敗した原因を取り除いてください。そのあとで，ビュー表を再作成してください。詳細については，「11.2.8　ビュー表の無効化を解除する方法」の「(8)　バージョンアップ時に行われるビュー表の再作成が失敗した場合」を参照してください。

バージョン 05-05 の HADB サーバにバージョンアップした場合に変更されるディクショナリ表とシステム表の一覧を，次の 2 つの表に示します。

表 8-31　バージョンアップ時に変更されるディクショナリ表とシステム表の一覧（その 1）

項番	ディクショナリ表とシステム表の名称	バージョンアップする前の HADB サーバのバージョン									
		02-00	02-01	02-02	03-00	03-01	03-02	03-03	03-04	03-05	03-06
1	SQL_TABLES	−	○	○	○	○	○	○	○	○	○

項番	ディクショナリ表とシステム表の名称	バージョンアップする前の HADB サーバのバージョン									
		02-00	02-01	02-02	03-00	03-01	03-02	03-03	03-04	03-05	03-06
2	SQL_COLUMNS	−	○	○	○	○	○	○	○	○	○
3	SQL_INDEXES	−	○	○	○	○	○	○	○	○	○
4	SQL_DIV_TABLE	−	○	○	−	−	−	−	−	−	−
5	SQL_DIV_INDEX	−	○	○	−	−	−	−	−	−	−
6	SQL_DBAREAS	−	−	−	−	−	−	−	−	−	−
7	SQL_SCHEMATA	−	○	○	○	○	○	○	○	○	○
8	SQL_VIEWS	−	○	○	○	○	○	○	○	○	−
9	SQL_VIEW_TABLE_USAGE	−	○	○	○	○	○	○	○	○	○
10	SQL_DEFINE_SOURCE	−	○	○	○	○	○	○	○	○	−
11	SQL_DEFINE_ENVIRONMENT	−	○	○	○	○	○	○	○	○	−
12	SQL_USERS	−	○	○	○	○	○	○	○	○	○
13	SQL_TABLE_CONSTRAINTS	−	−	○	−	−	−	−	−	−	−
14	SQL_INDEX_COLINF	−	−	○	−	−	−	−	−	−	−
15	SQL_KEY_COLUMN_USAGE	−	−	−	−	−	−	−	−	−	−
16	SQL_REFERENTIAL_CONSTRAINTS	−	−	−	−	−	−	−	−	−	−
17	SQL_TABLE_PRIVILEGES	−	−	−	○	○	○	○	○	○	−
18	SQL_AUDITS	−	−	−	−	−	−	−	−	−	−
19	STATUS_TABLES	−	○	○	○	○	○	○	○	○	○
20	STATUS_INDEXES	−	○	○	−	−	−	−	−	−	−
21	STATUS_CHUNKS	−	○	○	○	○	○	○	○	○	○
22	STATUS_SYNONYM_DICTIONARIES	−	−	−	−	−	−	−	−	−	−

項番	ディクショナリ表とシステム表の名称	バージョンアップする前の HADB サーバのバージョン									
		02-00	02-01	02-02	03-00	03-01	03-02	03-03	03-04	03-05	03-06
23	STATUS_COLUMNS	－	－	－	－	－	－	－	○	○	○

（凡例）

○：ディクショナリ表やシステム表をバージョンアップ時に変更します。

－：ディクショナリ表やシステム表をバージョンアップ時に変更しません。

表 8-32　バージョンアップ時に変更されるディクショナリ表とシステム表の一覧（その 2）

項番	ディクショナリ表とシステム表の名称	バージョンアップする前の HADB サーバのバージョン			
		04-00	04-01，04-02	04-03，05-00，05-01，05-02，05-03	05-04
1	SQL_TABLES	○	○	○	－
2	SQL_COLUMNS	○	○	○	－
3	SQL_INDEXES	－	－	－	－
4	SQL_DIV_TABLE	－	－	－	－
5	SQL_DIV_INDEX	－	－	－	－
6	SQL_DBAREAS	－	－	－	－
7	SQL_SCHEMATA	－	－	－	－
8	SQL_VIEWS	－	－	－	－
9	SQL_VIEW_TABLE_USAGE	－	－	－	－
10	SQL_DEFINE_SOURCE	－	－	－	－
11	SQL_DEFINE_ENVIRONMENT	－	－	－	－
12	SQL_USERS	－	－	－	－
13	SQL_TABLE_CONSTRAINTS	－	－	－	－
14	SQL_INDEX_COLINF	－	－	－	－
15	SQL_KEY_COLUMN_USAGE	－	－	－	－
16	SQL_REFERENTIAL_CONSTRAINTS	－	－	－	－
17	SQL_TABLE_PRIVILEGES	－	－	－	－
18	SQL_AUDITS	－	－	－	－
19	STATUS_TABLES	○	○	－	－
20	STATUS_INDEXES	－	－	－	－

項番	ディクショナリ表とシステム表の名称	バージョンアップする前のHADBサーバのバージョン			
		04-00	04-01，04-02	04-03，05-00，05-01，05-02，05-03	05-04
21	STATUS_CHUNKS	○	－	－	－
22	STATUS_SYNONYM_DICTIONARIES	－	－	－	－
23	STATUS_COLUMNS	○	○	○	－

（凡例）

○：ディクショナリ表やシステム表をバージョンアップ時に変更します。

－：ディクショナリ表やシステム表をバージョンアップ時に変更しません。

(6) 再作成できない可能性があるビュー表の確認

バージョン 05-00 より前の HADB サーバで定義したビュー表が，「■ビュー表を再作成できない条件」に示すすべての条件に該当する場合，そのビュー表はALTER VIEW 文で再作成できないことがあります。

■ビュー表を再作成できない条件

1. CREATE VIEW 文中に導出表を指定している
2. 条件 1.の導出表に導出列リストを指定していない
3. 条件 1.の導出表の導出問合せの選択式に，次の各指定をそれぞれ 1 つ以上指定している
   - 列名が「EXP*nnnn*_NO_NAME」である列指定，または列名が「EXP*nnnn*_NO_NAME」となる値式をAS句に指定している

     nnnn は，0001～1000 の符号なし整数です。
   - AS 句を指定していない，列指定以外の値式

ALTER VIEW 文で再作成できないビュー表は，DROP VIEW 文でいったんビュー表を削除し，そのあとにCREATE VIEW 文でビュー表を再定義してください。手順を次に示します。

■ビュー表を再定義する手順

1. 再作成できないビュー表を確認する

   バージョン 05-00 より前の HADB サーバで定義したビュー表の中から，「■ビュー表を再作成できない条件」に該当するビュー表があるかどうかを確認してください。
2. ビュー表のCREATE VIEW 文を修正する

   ビュー表のCREATE VIEW 文は，ディクショナリ表SQL_DEFINE_SOURCE 表のDEFINE_SOURCE 列で確認できます。DEFINE_SOURCE 列を確認する場合は，「付録 B.22　ディクショナリ表の検索」の「(31) ビュー表の定義情報を調べる場合」を参照してください。

   手順 1.で確認したビュー表のCREATE VIEW 文を次のどちらかの方法で修正してください。
   - 導出表の導出問合せの選択式に指定した「EXP*nnnn*_NO_NAME」の列名を変更する

- 導出問合せの選択式にAS句，または導出表に導出列リストを追加して，導出列名が「EXP*nnnn*_NO_NAME」にならないようにする

3. ビュー表を削除する

   手順1.で確認したビュー表をDROP VIEW文で削除してください。

4. ビュー表を再定義する

   手順2.で修正したCREATE VIEW文を実行し，ビュー表を再定義してください。

5. 無効化されたビュー表を再作成する

   手順3.でビュー表を削除した際に，ビュー表が無効化された場合は，ビューレベルの値が小さい順に，無効化されたビュー表をALTER VIEW文で再作成してください。

(7) ビュー表の再作成（05-01より前のバージョンからバージョンアップした場合）

バージョン05-01で，問合せ名に対する内部導出表を展開する際の条件を増やしましたが，バージョン05-01より前のバージョンで定義したビュー表に対しては適用されません。そのため，バージョン05-01で増えた条件をビュー表に適用したい場合は，次の手順でビュー表を再作成してください。

手順

1. 該当するビュー表を確認する

   次の条件をすべて満たすビュー表が，該当するビュー表（再作成対象のビュー表）です。

   - WITH句が指定されているビュー表
   - バージョン05-01より前のバージョンで定義したビュー表

   ビュー表を定義したときのCREATE VIEW文については，ディクショナリ表SQL_DEFINE_SOURCE表のDEFINE_SOURCE列で確認できます。ビュー表を定義したバージョンについては，ディクショナリ表SQL_DEFINE_ENVIRONMENT表のDEFINE_VR列で確認できます。DEFINE_SOURCE列およびDEFINE_VR列の確認方法については，「付録 B.22　ディクショナリ表の検索」の「(31)　ビュー表の定義情報を調べる場合」を参照してください。

2. 該当するビュー表を再作成する

   該当するビュー表をビューレベルの値が小さい順に，ALTER VIEW文で再作成してください。

3. 無効化されたビュー表を再作成する

   手順2.で実行したALTER VIEW文によって無効化されたビュー表がある場合は，ビューレベルの値が小さい順に，無効化されたビュー表をALTER VIEW文で再作成してください。

(8) JDBCドライバに関する注意事項（05-01より前のバージョンからバージョンアップした場合）

バージョン05-01で，JRE6版およびJRE7版のJDBCドライバ（adbjdbc.jar）の提供を中止しました。JRE6版およびJRE7版のJDBCドライバを使用している場合は，JRE8版のJDBCドライバ

（adbjdbc8.jar）を使用してください。使用するJDBCドライバを変更するには，APが参照しているJARファイルをadbjdbc.jarからadbjdbc8.jarに変更してください。adbjdbc.jarからadbjdbc8.jarへの変更による影響はありません。ただし，adbjdbc8.jarを使用するにはJava 8以降が必要です。

(9) マルチノード機能使用時のリアルスレッド数の増加について（05-05より前のバージョンからバージョンアップした場合）

マルチノード機能を使用している場合にバージョン05-05より前のバージョンからバージョンアップすると，HADBサーバが使用するリアルスレッド数が，次に示す計算式の分だけ増加します。

サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定値
×（*サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）×2

なお，サーバ定義のadb_sys_max_parallel_exec_numオペランドの指定を省略した場合，adb_sys_max_parallel_exec_numオペランドの省略値としてサーバ定義のadb_sys_max_usersオペランドの指定値が仮定され，リアルスレッド数は次に示す計算式の分だけ増加します。

サーバ定義のadb_sys_max_usersオペランドの指定値
×（*サーバ定義のadb_sys_multi_node_infoオペランドに指定したホスト名の数*－1）×2

なお，SQLパラレル実行機能を使用しない場合は，adb_sys_max_parallel_exec_numオペランドに0を指定してください。0を指定すると，バージョンアップをしてもここで説明しているリアルスレッド数の増加が発生しません。

また，SQLパラレル実行機能を使用する場合でも，SQLパラレル実行機能を適用した検索系SQLの同時実行数を正確に見積もれるときは，adb_sys_max_parallel_exec_numオペランドに正確に見積もった値を指定すると，リアルスレッド数の増加を抑えることができます。

8.7　HADB サーバのバージョンダウン（旧バージョンに戻す方法）

ここでは，HADB サーバをバージョンアップしたあとで，バージョンアップ前のバージョン（旧バージョン）に戻したい場合について説明します（HADB サーバをバージョンダウンする方法について説明します）。

なお，バージョンアップ後の新機能は，旧バージョンの HADB サーバでは使用できません。

> **重要**
>
> バージョンアップに失敗したために，旧バージョンに戻す必要がある場合は，「8.6.4　バージョンアップに失敗した場合の対処方法」を参照してください。

バージョンダウンの方法は，バージョンアップの前にデータベースのフルバックアップを取得していたかどうかで異なります。データベースのフルバックアップの取得有無によるバージョンダウンの方法について，次の表に示します。

表 8-33　HADB サーバのバージョンダウンの方法

項番	データベースのフルバックアップの取得有無	バージョンダウンの方法
1	○	旧バージョンのデータベースのバックアップを使用して，バージョンダウンを行います。 詳細については，「8.7.1　旧バージョンのデータベースのバックアップを使用する場合」を参照してください。
2	×	旧バージョンのデータベースのバックアップを取得していないため，旧バージョンの HADB サーバでデータベースを再構築します。 詳細については，「8.7.2　旧バージョンでデータベースを再構築する場合」を参照してください。

（凡例）

○：フルバックアップを取得しています。

×：フルバックアップを取得していません。

> **メモ**
>
> データベースのバックアップの詳細については，「10.3　データベースのバックアップ」を参照してください。なお，マルチノード機能を使用している場合は，「16.8　データベースのバックアップ（マルチノード機能の使用時）」を参照してください。また，コールドスタンバイ構成の場合は，「17.6　データベースのバックアップ（コールドスタンバイ構成の場合）」を参照してください。

8.7.1　旧バージョンのデータベースのバックアップを使用する場合

ここでは，旧バージョンのデータベースのバックアップを使用して，HADB サーバをバージョンダウンする方法について説明します。この方法は，バージョンアップ前に，旧バージョンのデータベースのフルバックアップを取得していることを前提としています。

> **メモ**
>
> 旧バージョンのデータベースのフルバックアップを取得していない場合は，旧バージョンでデータベースを再構築して，バージョンダウンする必要があります。詳細については，「8.7.2　旧バージョンでデータベースを再構築する場合」を参照してください。

(1)　バージョンダウン前に実施すること

HADB サーバをバージョンダウンする前に，次に示すことを必ず実施してください。

(a)　HADB サーバの環境設定を確認する

HADB サーバをバージョンダウンする前に，環境変数，カーネルパラメタ，およびサーバ定義の各指定値を，バージョンアップの際に変更していないことを確認してください。指定値を変更している場合は，旧バージョンの指定値に戻す必要があります。

(b)　AP およびコマンドを停止する

HADB サーバをバージョンダウンする前に，実行中の AP およびコマンドがある場合は，すべて終了してください。

(c)　HADB サーバを正常終了する

HADB サーバをバージョンダウンする前に，adbstop コマンドで HADB サーバを正常終了してください。

すでに HADB サーバが終了している場合は，次に示す手順で HADB サーバが正常終了していることを確認してください。

手順

1. adbls -d srv コマンドを実行する

   出力項目STATUS に出力される HADB サーバの状態が，"STOP"であることを確認してください。

   "STOP"である場合は，手順 2.または手順 3.のどちらかに進んでください。

2. サーバメッセージログファイルを確認する

   サーバメッセージログファイル（$ADBDIR/spool/adbmessage*XX*.log※）を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

   注※

   XX は通番であり，01～04 のどれかになります。

3. syslog を確認する

syslog を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

手順 1.の確認結果が"STOP"であり，かつ手順 2.または手順 3.の確認結果が"normally"である場合は，HADB サーバは正常終了しています。

もし，HADB サーバが正常終了していない場合は，adbstart コマンドで HADB サーバを開始したあとで，adbstop コマンドで正常終了してください。

[マルチノード機能]

マルチノード構成の HADB サーバが正常終了していない場合は，マルチノード構成の HADB サーバをいったん開始し，マルチノード構成の HADB サーバを正常終了させてください。このとき，HA モニタを停止させる必要はありません（monstop コマンドの実行は不要です）。

マルチノード構成の HADB サーバを開始して，正常終了させる方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」と「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照してください。

[コールドスタンバイ構成]

コールドスタンバイ構成の HADB サーバが正常終了していない場合は，すべての系の HADB サーバを開始したあとで，すべての系の HADB サーバを正常終了させてください。このとき，HA モニタを停止させる必要はありません（monstop コマンドの実行は不要です）。

コールドスタンバイ構成の HADB サーバを開始して，正常終了させる方法については，「17.4.1　コールドスタンバイ構成の開始方法」と「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

(d)　サーバディレクトリのバックアップを取得する

HADB サーバをバージョンダウンする前に，サーバディレクトリのバックアップを取得してください。

HADB サーバをバージョンダウンする場合は，HADB サーバをアンインストールする必要があるため，サーバディレクトリを削除します。サーバディレクトリを削除すると，定義ファイルやメッセージログファイルなども削除されます。バージョンダウン後に必要になった場合を考慮して，サーバディレクトリのバックアップを取得してください。

サーバディレクトリのバックアップを取得する方法については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

(2)　バージョンダウンする

HADB サーバのバージョンダウンを行う手順を次に示します。

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードの HADB サーバに対して実施してください。

[コールドスタンバイ構成]

コールドスタンバイ構成の場合は，すべての系の HADB サーバに対して実施してください。

手順

1. HADB サーバをアンインストールする

   「8.12　アンインストール」を参照して，HADB サーバをアンインストールしてください。

2. 旧バージョンの HADB サーバをインストールする

   旧バージョンの HADB サーバをインストールしてください。HADB サーバのインストールについては，「8.2　インストール」を参照してください。

3. データベースをバックアップから回復する

   取得したバックアップから，データベースを回復してください。

   データベースをバックアップから回復する方法については，「10.3.2　バックアップからの回復方法」を参照してください。

   なお，マルチノード機能を使用している場合は，「16.8.2　バックアップからの回復方法」を参照してください。また，コールドスタンバイ構成の場合は，「17.6.2　バックアップからの回復方法」を参照してください。

4. HADB サーバの環境設定を行う

   旧バージョンの HADB サーバの環境設定を行います。環境変数，カーネルパラメタ，およびサーバ定義に，旧バージョンの指定値と同じ値を指定してください。

5. HADB サーバを開始する

   `adbstart` コマンドで，HADB サーバを開始してください。

   なお，静止モードでバックアップを取得した場合は，HADB サーバを開始したあとで，HADB サーバの稼働モードを通常モードにする必要があります。この場合，`adbchgsrvmode` コマンドを実行して，通常モードに変更してください。

> **重要**
>
> 旧バージョンのデータベースのバックアップを使用して，HADB サーバをバージョンダウンすると，データベースの内容は，バックアップを取得した時点の内容に戻ります。

(3)　バージョンダウン後に実施すること

HADB サーバをバージョンダウンした場合は，HADB クライアントもバージョンダウンしてください（HADB サーバと HADB クライアントのバージョンを合わせてください）。

HADB クライアントのバージョンダウンについては，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントのバージョンダウン（旧バージョンに戻す方法）』を参照してください。

8.7.2　旧バージョンでデータベースを再構築する場合

ここでは，旧バージョンの HADB サーバでデータベースを再構築することで，HADB サーバをバージョンダウンする方法について，説明します。この方法は，旧バージョンのデータベースのフルバックアップを取得していない場合の説明になります。

> **メモ**
>
> 旧バージョンのデータベースのフルバックアップを取得している場合は，バックアップを使用して，HADB サーバのバージョンダウンを実施できます。詳細については，「8.7.1　旧バージョンのデータベースのバックアップを使用する場合」を参照してください。

(1)　バージョンダウン前に実施すること

HADB サーバをバージョンダウンする前に，次に示すことを必ず実施してください。

(a)　HADB サーバの環境設定を確認する

HADB サーバをバージョンダウンする前に，環境変数，カーネルパラメタ，およびサーバ定義の各指定値を，バージョンアップの際に変更していないことを確認してください。指定値を変更している場合は，旧バージョンの指定値に戻す必要があります。

(b)　DB エリアの状態を確認する（初期設定オプションを確認する）

HADB サーバをバージョンダウンする前に，adbdbstatus コマンドで，DB エリアの状態を確認してください。旧バージョンの HADB サーバでデータベースを再構築する場合に，初期設定オプションを指定して adbinit コマンドを実行するときに必要となります。

adbdbstatus コマンドで，まず「DB エリアのサマリ情報」を確認してください。出力結果のうち，次に示す出力項目の内容を確認します。

- DBarea_name（DB エリア名）
- Page_size（ページサイズ）

そのあとで，adbdbstatus コマンドで「使用量情報」を確認してください。出力結果のうち，DB エリアのサマリ情報で出力された「DBarea_name」の内容に対応する，「DBarea_filename」（DB エリアファイル名）の内容を確認します。

> **メモ**
>
> adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

データベースの再構築時には，adbdbstatus コマンドで確認した出力項目の内容を基に，adbinit コマンドの初期設定オプションを指定してください。adbdbstatus コマンドで確認した出力項目と，adbinit コマンドの初期設定オプションの関係について，次の表に示します。

表 8-34　adbdbstatus コマンドの出力項目とadbinit コマンドの初期設定オプションの関係

項番	adbdbstatus コマンドの出力項目	adbinit コマンドの初期設定オプション
1	DBarea_name（DB エリア名）	adbinitdbarea の-n オプション
2	Page_size（ページサイズ）	adbinitdbarea の-p オプション※
3	DBarea_filename（DB エリアファイル名）	adbinitdbarea の-v オプション

注※

adbinit コマンドの初期設定オプションadbinitdbarea の-p オプションは，キロバイト単位で指定する必要があります。adbdbstatus コマンドで出力されるPage_size（ページサイズ）の値はバイト単位のため，キロバイト単位に変換した値を-p オプションに指定してください。キロバイト単位に変換する計算式を次に示します。

計算式（単位：キロバイト）

```
Page_sizeの値÷1,024
```

(c)　定義系 SQL を確認する

HADB サーバをバージョンダウンする前に，ディクショナリ表を参照して，旧バージョンで実行する必要があるすべての定義系 SQL を確認してください。

ディクショナリ表については，「付録 B　ディクショナリ表」を参照してください。

(d)　データをエクスポートする

HADB サーバをバージョンダウンする前に，adbexport コマンドで実表に格納した全データをエクスポートしてください。

なお，バックグラウンドインポートで格納した実表のデータを，チャンクの構成を維持したままエクスポートしたい場合は，-c オプションを指定したadbexport コマンドを実行してください。

adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。

(e)　AP およびコマンドを停止する

HADB サーバをバージョンダウンする前に，実行中の AP およびコマンドがある場合は，すべて終了してください。

(f)　HADB サーバを正常終了する

HADB サーバをバージョンダウンする前に，adbstop コマンドで HADB サーバを正常終了してください。

すでに HADB サーバが終了している場合は，次に示す手順で HADB サーバが正常終了していることを確認してください。

1. adbls -d srv コマンドを実行する

   出力項目STATUS に出力される HADB サーバの状態が，"STOP"であることを確認してください。

   "STOP"である場合は，手順 2.または手順 3.のどちらかに進んでください。

2. サーバメッセージログファイルを確認する

   サーバメッセージログファイル（$ADBDIR/spool/adbmessage*XX*.log※）を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

   注※

   XX は通番であり，01～04 のどれかになります。

3. syslog を確認する

   syslog を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

手順 1.の確認結果が"STOP"であり，かつ手順 2.または手順 3.の確認結果が"normally"である場合は，HADB サーバは正常終了しています。

もし，HADB サーバが正常終了していない場合は，adbstart コマンドで HADB サーバを開始したあとで，adbstop コマンドで正常終了してください。

[マルチノード機能]

マルチノード構成の HADB サーバが正常終了していない場合は，マルチノード構成の HADB サーバをいったん開始し，マルチノード構成の HADB サーバを正常終了させてください。このとき，HA モニタを停止させる必要はありません（monstop コマンドの実行は不要です）。

マルチノード構成の HADB サーバを開始して，正常終了させる方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」と「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照してください。

[コールドスタンバイ構成]

コールドスタンバイ構成の HADB サーバが正常終了していない場合は，すべての系の HADB サーバを開始したあとで，すべての系の HADB サーバを正常終了させてください。このとき，HA モニタを停止させる必要はありません（monstop コマンドの実行は不要です）。

コールドスタンバイ構成の HADB サーバを開始して，正常終了させる方法については，「17.4.1　コールドスタンバイ構成の開始方法」と「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

(g)　サーバディレクトリのバックアップを取得する

HADB サーバをバージョンダウンする前に，サーバディレクトリのバックアップを取得してください。

HADB サーバをバージョンダウンする場合は，HADB サーバをアンインストールする必要があるため，サーバディレクトリを削除します。サーバディレクトリを削除すると，定義ファイルやメッセージログファイルなども削除されます。バージョンダウン後に必要になった場合を考慮して，サーバディレクトリのバックアップを取得してください。

サーバディレクトリのバックアップを取得する方法については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

(2)　バージョンダウンする

HADB サーバのバージョンダウンを行う手順を次に示します。

[マルチノード機能]

　マルチノード機能を使用している場合は，全ノードの HADB サーバに対して実施してください。

[コールドスタンバイ構成]

　コールドスタンバイ構成の場合は，すべての系の HADB サーバに対して実施してください。

手順

1. HADB サーバをアンインストールする

   「8.12　アンインストール」を参照して，HADB サーバをアンインストールしてください。

2. データベースを構築する

   「(1)　バージョンダウン前に実施すること」で確認した内容を基に，旧バージョンの HADB サーバでデータベースを構築します。構築の手順については，「8.1　システム構築の手順」を参照してください。

> **メモ**
>
> マルチチャンク表を元のチャンクの構成で復元したい場合は，バックグラウンドインポート機能を適用した（-b オプションを指定した）adbimport コマンドを必要な回数分，実行してください。ただし，チャンク ID については，復元できません。
>
> adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport (データのインポート)』を参照してください。

(3)　バージョンダウン後に実施すること

HADB サーバをバージョンダウンした場合は，HADB クライアントもバージョンダウンしてください (HADB サーバと HADB クライアントのバージョンを合わせてください)。

HADB クライアントのバージョンダウンについては，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントのバージョンダウン（旧バージョンに戻す方法)』を参照してください。

8.7.3　バージョンアップに失敗したために，旧バージョンに戻す必要がある場合

バージョンアップに失敗したために，旧バージョンに戻す必要がある場合は，「8.6.4　バージョンアップに失敗した場合の対処方法」を参照してください。

8.8　修正版 HADB サーバとの入れ替え

ここでは，修正版 HADB サーバとの入れ替え方法について説明します。

次に示す条件を満たす場合は，HADB サーバのバージョンアップではなく，修正版 HADB サーバとの入れ替えとなります。

- バージョン番号およびリビジョン番号が同じで，コードだけが異なる HADB サーバを上書きインストールする場合

> **メモ**
>
> バージョン番号，リビジョン番号，コードについて説明します。
>
> （例）HADB サーバのバージョン情報が，03-02-/A の場合
>
> - 03 の部分がバージョン番号になります。
> - 02 の部分がリビジョン番号になります。
> - 下線部分の-/A の部分が，コードになります。

修正版 HADB サーバとの入れ替えになる場合の例と，入れ替えにならない場合の例を次に示します。

■修正版 HADB サーバとの入れ替えになる場合

例えば，次に示すケースは，バージョン番号およびリビジョン番号が同じため，修正版 HADB サーバとの入れ替えになります。

03-02 → 03-02-/A

03-02-/B → 03-02-/D

■修正版 HADB サーバとの入れ替えにならない場合

例えば，次に示すケースは，バージョン番号またはリビジョン番号が異なるため，修正版 HADB サーバとの入れ替えになりません。

03-02 → 03-03-/A

03-02-/B → 03-03-/D

この場合，HADB サーバのバージョンアップになります。HADB サーバのバージョンアップ方法については，「8.6　HADB サーバのバージョンアップ」を参照してください。

8.8.1　修正版 HADB サーバとの入れ替え前に実施すること

修正版 HADB サーバとの入れ替え前に実施することを説明します。

重要

修正版 HADB サーバとの入れ替えを実施する前に，環境変数ADBLANG およびADBCLTLANG の指定値を変更しないでください。変更した場合，入れ替えが正しくできないおそれがあります。

環境変数LANG については，次に示すように言語コードおよび国名コードの変更はできます。

- ja_JP.UTF-8 からen_US.UTF-8 への変更
- en_US.UTF-8 からja_JP.UTF-8 への変更

コードセットの変更（文字コード種別の変更）はできません。

なお，環境変数LANG の指定値を変更した場合，HADB クライアントの環境変数LANG の指定値も変更する必要があります。

(1) メモリ所要量を確認する

修正版 HADB サーバに入れ替えた場合，HADB サーバのメモリ所要量が増加することがあります。詳細については，リリースノートを参照してください。

(2) 環境変数の変更が必要かどうかを確認する

修正版 HADB サーバに入れ替えた場合，環境変数の変更が必要になることがあります。詳細については，リリースノートを参照してください。

(3) カーネルパラメタの変更が必要かどうかを確認する

修正版 HADB サーバに入れ替えた場合，カーネルパラメタの変更が必要になることがあります。詳細については，リリースノートを参照してください。

(4) サーバ定義のオペランドのデフォルト値に変更がないかを確認する

修正版 HADB サーバに入れ替えた場合，サーバ定義のオペランドのデフォルト値が変更されることがあります。詳細については，リリースノートを参照してください。

(5) 予約語が追加されているかどうかを確認する

修正版 HADB サーバに入れ替えた場合，予約語が追加されることがあります。予約語は，表や列などの名称として SQL 文中に指定できません。追加された予約語と同じ文字列を SQL 文中で指定している場合，修正版 HADB サーバに入れ替えたあとに，SQL 文がエラーとなります。そのため，追加された予約語と同じ文字列を，SQL 文中で指定していないかどうかを確認してください。指定している場合は，予約語と同じ文字列を二重引用符（”）で囲んでください。詳細については，マニュアル『HADB　SQL リファレンス』の『基本項目』の『予約語』を参照してください。

(6)　APおよびコマンドを正常終了する

修正版 HADB サーバとの入れ替えを始める前に，AP およびコマンドを正常終了してください。

(7)　HADB サーバを正常終了する

修正版 HADB サーバとの入れ替えを始める前に，adbstop コマンドで HADB サーバを正常終了してください。すでに HADB サーバが終了している場合は，次に示す手順で HADB サーバが正常終了していることを確認してください。

手順

1. adbls -d srv コマンドを実行する

   出力項目STATUS に出力される HADB サーバの状態が，"STOP"であることを確認してください。

   "STOP"である場合は，手順 2.または手順 3.のどちらかに進んでください。

2. サーバメッセージログファイルを確認する

   サーバメッセージログファイル（$ADBDIR/spool/adbmessage*XX*.log※）を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

   注※　*XX* は通番であり，01～04 のどれかになります。

3. syslog を確認する

   syslog を開き，KFAA91154-I メッセージに出力される終了モードが，"normally"であることを確認してください。

手順 1.の確認結果が"STOP"であり，かつ手順 2.または手順 3.の確認結果が"normally"である場合は，HADB サーバは正常終了しています。

HADB サーバが正常終了していない場合は，adbstart コマンドで HADB サーバをいったん開始したあとで，adbstop コマンドで HADB サーバを正常終了してください。

(8)　サーバディレクトリのバックアップを取得する

修正版 HADB サーバとの入れ替えを始める前に，サーバディレクトリのバックアップを取得してください。

HADB サーバが異常終了した状態で，修正版 HADB サーバとの入れ替えを実施した場合，HADB サーバを開始できなくなります。それに備えて，サーバディレクトリのバックアップを取得しておいてください。取得したバックアップは，修正版 HADB サーバとの入れ替えが完了したあとに削除してください。

バックアップを取得する方法については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードでサーバディレクトリのバックアップを取得してください。

8.8.2　修正版 HADB サーバに入れ替える手順

「8.8.1　修正版 HADB サーバとの入れ替え前に実施すること」で説明していることを実施したあとに，次の手順に従って修正版 HADB サーバに入れ替えてください。

手順

1. HADB 管理者の OS アカウントで OS にログインする

   今まで使用していた HADB 管理者の OS アカウントで，サーバマシンの OS にログインしてください。

   > **重要**
   >
   > HADB 管理者の OS アカウントを変更しないでください。HADB 管理者の OS アカウントを変更した場合，修正版 HADB サーバとの入れ替えが正しくできないおそれがあります。

2. インストールデータを格納するディレクトリを作成する

   修正版 HADB サーバのインストールデータを格納するディレクトリを作成します。

   ```
   mkdir /home/adbmanager/install
   ```

   上記の例では，インストールデータを格納するディレクトリとして/home/adbmanager/install を作成しています。

3. インストールデータを格納するディレクトリに書き込み権限を付与する

   インストールデータを格納するディレクトリ（この例では/home/adbmanager/install）に対して，HADB 管理者が書き込みできるように書き込み権限を付与してください。

   ```
   chmod 755 /home/adbmanager/install
   ```

4. CD-ROM ファイルシステムをマウントする

   修正版 HADB サーバのインストールデータが格納されている CD-ROM ファイルシステムを自動マウントしてください。

   CD-ROM ファイルシステムが自動マウントされていない場合，次のコマンドを実行して CD-ROM ファイルシステムをマウントしてください。

   ```
   mount /dev/cdrom /media
   ```

   下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

   > **重要**
   >
   > CD-ROM のディレクトリ名やファイル名は，ご使用のサーバマシンによっては，ここの説明内容と表示が異なることがあります。OS の ls コマンドを実行して，表示されたディレクトリ名をそのまま入力してください。

5. インストールデータをコピーする

   マウントした CD-ROM ファイルシステムに格納されている次のファイルを，手順 2.で作成したインストールデータを格納するディレクトリにコピーしてください。

- tar.gz 形式のファイル
- adbinstall コマンドの実行形式ファイル
- !8A9_HADB ファイル

```
cp /media/hitachi_advanced_data_binder_server-$VER.tar.gz /home/adbmanager/install   ...a
cp /media/adbinstall /home/adbmanager/install                                        ...b
cp '/media/!8A9_HADB' /home/adbmanager/install                                       ...c
```

[説明]

a. tar.gz 形式のファイル（修正版 HADB サーバのプログラムの圧縮ファイル）をコピーします。

b. adbinstall コマンドの実行形式ファイル（修正版 HADB サーバのインストールコマンド）をコピーします。

c. !8A9_HADB ファイルをコピーします。

上記の例では，すべてのファイルを/home/adbmanager/install ディレクトリ下にコピーしています。

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

$VER は，修正版 HADB サーバのバージョンとリリース番号です。

（例）03-02-/A の場合 → *$VER* は「0302a-0」

> **重要**
>
> 上記の 3 つのファイルは，必ず同じディレクトリにコピーしてください。異なるディレクトリにコピーすると，修正版 HADB サーバのインストールが実行できません。

6. コピーしたファイルが破損していないことを確認する

手順 5.でコピーしたファイルが破損していないことを確認します。次のコマンドを実行してください。

```
cd /home/adbmanager/install
sha256sum -c /media/adbhashfile_server_sha256.txt
```

下線部分は，CD-ROM ファイルシステムのマウントディレクトリ名です。環境によって異なります。

ファイルが破損しているおそれがある（ファイルのハッシュ値が異なる）場合は，エラーを示すメッセージが出力されます。この場合，手順 5.から作業をし直してください。

出力されるメッセージの詳細については，OS のマニュアルを参照してください。

7. 権限を付与する

HADB 管理者がadbinstall コマンド（修正版 HADB サーバのインストールコマンド）を実行できるように，adbinstall コマンドに対する実行権限を付与してください。

また，tar.gz 形式のファイルに対する読み取り権限も付与してください。

```
chmod 500 /home/adbmanager/install/adbinstall
chmod 400 /home/adbmanager/install/hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，修正版 HADB サーバのバージョンとリリース番号です。

8. 修正版 HADB サーバをインストールする

adbinstall コマンドを実行して，修正版 HADB サーバをインストールしてください。

```
/home/adbmanager/install/adbinstall -s /HADB/server
```

下線部分には，手順 2.で作成したインストールデータを格納するディレクトリを指定します。

-s オプションには，使用中のサーバディレクトリ（この例では/HADB/server）の絶対パスを指定します。

重要

- KFAA91553-E メッセージが出力されて，adbinstall コマンドがエラーになった場合は，「■KFAA91553-E メッセージが出力された場合」を参照してください。
- adbinstall コマンドを実行した際にKFAA91558-W メッセージ（警告メッセージ）が出力された場合は，「■KFAA91558-W メッセージが出力された場合」を参照してください。

9. KFAA91554-Q メッセージに応答する

adbinstall コマンドを実行すると，ディレクトリを上書きしてよいかどうかを確認するKFAA91554-Q メッセージが出力されます。y またはY を入力して，ディレクトリを上書きしてください。

10. 修正版 HADB サーバが正常にインストールされたことを確認する

サーバディレクトリ下にあるadbinstsv.log ファイルを開いて内容を確認してください。修正版 HADB サーバが正常にインストールされている場合は，次のどちらかの情報が記載されています。

- Hitachi Advanced Data Binder *vv-rr* (*yyyymmddhhmmss*)
- Hitachi Advanced Data Binder *vv-rr-/s* (*yyyymmddhhmmss*)

vv-rr および *vv-rr-/s* は，修正版 HADB サーバのバージョンを示しています。

yyyymmddhhmmss は，修正版 HADB サーバのインストールを実行した日付と時刻を示しています。

メモ

- adbinstsv.log ファイルがすでにある状態のときにadbinstall コマンドを実行すると，adbinstsv.log ファイルの内容が更新されます。また，adbinstall コマンドを複数回実行した場合，最後に実行した日付と時刻が記載されます。
- adbinstall コマンドのリターンコード（KFAA91552-I メッセージに出力されるリターンコード）が0 または4 以外の場合，adbinstsv.log ファイルの内容は更新されません。
- adbinstall コマンドの実行時にKFAA91555-I メッセージが出力された場合，adbinstsv.log ファイルの内容は更新されません。

11. 修正版 HADB サーバのバージョンを確認する

次のコマンドを実行して，修正版 HADB サーバのバージョン情報を確認してください。

```
adbls -d ver
```

[マルチノード機能]

マルチノード機能を使用している場合は，全ノードの HADB サーバで上記のコマンドを実行し，全ノードの修正版 HADB サーバのバージョン情報を確認してください。

adbls -d ver コマンドを実行しても，バージョン情報が表示されない場合は，次に示す環境変数が正しく設定されていない可能性があります。

- 環境変数ADBDIR に，サーバディレクトリが正しく設定されていない
- 環境変数PATH に，サーバディレクトリ下のbin ディレクトリが正しく設定されていない

環境変数については，「8.4　環境変数の設定」を参照してください。

12. 修正版 HADB サーバを正常開始する

adbstart コマンドを実行して，修正版 HADB サーバを正常開始してください。

```
adbstart
```

HADB サーバが正常開始すると，修正版 HADB サーバとの入れ替えが完了します。

■KFAA91553-E メッセージが出力された場合

次に示す場合は，adbinstall コマンドがエラーになり，KFAA91553-E メッセージが出力されます。この場合，対象のディレクトリに対する書き込み権限を HADB 管理者に付与してください。

- -s オプションに指定したディレクトリに対して，HADB 管理者が書き込み権限を持っていない場合
- インストールデータを格納するディレクトリに対して，HADB 管理者が書き込み権限を持っていない場合

■KFAA91558-W メッセージが出力された場合

HADB 管理者の OS アカウントではなく，root でadbinstall コマンドを実行した場合，KFAA91558-W メッセージ（警告メッセージ）が出力されます。通常，HADB 管理者の OS アカウントで，adbinstall コマンドを実行します。そのため，KFAA91558-W メッセージが出力された場合は，root でadbinstall コマンドを実行して問題がないかどうかを確認してください。

問題がある場合は，KFAA91558-W メッセージが出力されたあとに出力されるKFAA91559-Q メッセージの入力要求に対して，n またはN を入力してください。そのあとで，HADB 管理者の OS アカウントで adbinstall コマンドを実行してください。

なお，root 以外のスーパユーザでadbinstall コマンドを実行した場合は，KFAA91558-W メッセージは出力されません。

メモ

root は，OS のid -u コマンドを実行して表示される値が0 のユーザを指します。また，OS のsu コマンドを使用して，ほかの OS ユーザからroot に切り替えたあとで，OS のid -u コマンドを実行して表示される値が0 のときも含みます。

8.9　一時的に HADB サーバのバージョンアップとバージョンダウンを行う方法（動作確認テスト用）

ここでは，アプリケーションなどの動作確認テストのために，一時的に HADB サーバをバージョンアップし，動作確認テスト完了後に元のバージョンに戻す方法について説明します。

動作確認テスト時にデータベースを更新する※かどうかによって，作業手順が異なります。

これ以降の「8.9.1　バージョンアップ前に実施すること（動作確認テスト用）」～「8.9.5　バージョンダウン後に実施すること（動作確認テスト用）」では，動作確認テストにデータベースの更新を含まない場合の手順について説明しています。動作確認テストにデータベースの更新を含まない場合は，各手順に従って，HADB サーバのバージョンアップ，および旧バージョンへのバージョンダウンを実施してください。

動作確認テストにデータベースの更新を含む場合は，次に示す手順に従ってください。

■動作確認テストにデータベースの更新を含む場合

次に示す手順で，HADB サーバのバージョンアップ，および旧バージョンへのバージョンダウンを実施してください。

1. HADB サーバをバージョンアップする

   次の各手順に従ってください。なお，データベースのバックアップを取得する際には，必ずフルバックアップを取得してください。

   - 「8.6.1　バージョンアップ前に実施すること」
   - 「8.6.2　バージョンアップする」
   - 「8.6.3　バージョンアップ後に実施すること」

2. 動作確認テストを実施する
3. HADB サーバを旧バージョンに戻す

   「8.7.1　旧バージョンのデータベースのバックアップを使用する場合」の手順に従ってください。

注※

次に示す SQL 文またはコマンドを実行すると，データベースが更新されます。

- 更新系 SQL
- 定義系 SQL
- adbimport コマンド
- adbidxrebuild コマンド
- adbgetcst コマンド
- adbmodarea コマンド
- adbmergechunk コマンド
- adbchgchunkcomment コマンド

- adbchgchunkstatus コマンド
- adbreorgsystemdata コマンド
- adbarchivechunk コマンド
- adbunarchivechunk コマンド
- adbsyndict コマンド

また，更新行のカラム化機能を有効にしている場合も，データベースが更新されます。更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

8.9.1　バージョンアップ前に実施すること（動作確認テスト用）

「8.6.1　バージョンアップ前に実施すること」の手順に従ってください。

ただし，データベースのバックアップでは，システムデータだけのバックアップを取得してください。

8.9.2　バージョンアップする（動作確認テスト用）

「8.6.2　バージョンアップする」の手順に従って，HADB サーバをバージョンアップしてください。ただし，「8.6.2　バージョンアップする」にある手順『データベースをバージョンアップする』については，次に示す手順に読み替えてください。

データベースをバージョンアップする手順（動作確認テスト用）

次に示す HADB のコマンドを実行して，HADB サーバを開始してください。バージョンアップ処理が開始されます。

```
adbstart --quiescence
```

adbstart コマンドの実行後，データベースをバージョンアップしてよいかを確認するKFAA91107-Q メッセージが出力されます。[Y] キーを押して，データベースをバージョンアップしてください。

adbstart コマンドが正常終了すれば，HADB サーバのバージョンアップは完了です。

[マルチノード機能]

マルチノード機能を使用している場合は，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照して，マルチノード構成の HADB サーバを開始してください。ただし，プライマリノードでは次に示す HADB のコマンドを実行してください。

```
adbstart --quiescence
```

このとき，プライマリノードの HADB サーバを開始すると，データベースをバージョンアップしてよいかを確認するKFAA91107-Q メッセージが出力されます。[Y] キーを押して，データベースをバージョンアップしてください。

全ノードの HADB サーバが正常に開始すれば，HADB サーバのバージョンアップは完了です。

8.9.3　バージョンアップ後に実施すること（動作確認テスト用）

「8.6.3　バージョンアップ後に実施すること」の「(2)　HADB クライアントをバージョンアップする」の手順に従って，HADB クライアントをバージョンアップしてください。

そのあとで，動作確認テストを実施してください。

8.9.4　バージョンダウンする（動作確認テスト用）

動作確認テスト終了後，「8.7.1　旧バージョンのデータベースのバックアップを使用する場合」の「(2) バージョンダウンする」の手順に従って，HADB サーバをバージョンダウンしてください。

ただし，バックアップからデータベースを回復する際は，システムデータだけのバックアップを使用してください。

8.9.5　バージョンダウン後に実施すること（動作確認テスト用）

「8.7.1　旧バージョンのデータベースのバックアップを使用する場合」の「(3)　バージョンダウン後に実施すること」の手順に従って，HADB クライアントをバージョンダウンしてください。

8.10　サーバマシンの OS の時刻変更

ここでは，HADB サーバがインストールされているサーバマシンの OS の時刻を変更する場合の手順について説明します。

サーバマシンの OS の時刻を変更する場合，必ず「8.10.1　注意事項（OS の時刻変更）」を読んだあとで，次のどちらかに従って OS の時刻を変更してください。

- 「8.10.2　サーバマシンの OS の時刻を進める方法」
- 「8.10.3　サーバマシンの OS の時刻を戻す方法」

8.10.1　注意事項（OS の時刻変更）

サーバマシンの OS の時刻を変更する場合の注意事項を，次に示します。

- サーバマシンの OS の時刻を変更する場合，必ず HADB サーバが正常終了しているときに実施してください。HADB サーバの稼働中に，サーバマシンの OS の時刻を変更してしまうと，HADB サーバおよび HADB クライアントが予期しない動作をするおそれがあります。
- サーバマシンの OS の時刻を変更した場合，次に示す時刻表示などは，変更後の OS の時刻になります。そのため，時刻を意識する AP がある場合は，影響を受けるおそれがあります。
  - SQL 文で取得する時刻
  - コマンドを実行した時刻
  - ディクショナリ表に格納されるタイムスタンプ
  - システム表に格納されるタイムスタンプ
- NTP プログラムを使用して OS の時刻を補正する場合は，遅延調整方式（Slew 方式）を選択してください（時刻の戻り，または大幅な時刻変更が発生しないようにしてください）。

8.10.2　サーバマシンの OS の時刻を進める方法

「8.10.1　注意事項（OS の時刻変更）」を読んだあとで，次に示す手順に従って，HADB サーバがインストールされているサーバマシンの OS の時刻を進めてください。

手順

1. HADB サーバを正常終了する

   `adbstop` コマンドを実行して，HADB サーバを正常終了してください。`adbstop` コマンドについては，「10.2.2　HADB サーバの終了方法」を参照してください。

2. サーバマシンの OS の時刻を進める

HADB サーバを正常終了したら，HADB サーバがインストールされているサーバマシンの OS の時刻を進めてください。

3. HADB サーバを開始する

   サーバマシンの OS の時刻を進めたら，adbstart コマンドを実行して，HADB サーバを開始してください。adbstart コマンドについては，「10.2.1　HADB サーバの開始方法」を参照してください。

これで，サーバマシンの OS の時刻を進められます。

8.10.3　サーバマシンの OS の時刻を戻す方法

「8.10.1　注意事項（OS の時刻変更）」を読んだあとで，次に示すどちらかの手順に従って，HADB サーバがインストールされているサーバマシンの OS の時刻を戻してください。

- OS の時刻を数秒から数分ほど戻す場合（戻した時間分，待つことができる場合）
- HADB サーバをすぐに開始したい場合（戻した時間分，待つことができない場合）

(1)　OS の時刻を数秒から数分ほど戻す場合（戻した時間分，待つことができる場合）

手順

1. HADB サーバを正常終了する

   adbstop コマンドを実行して，HADB サーバを正常終了してください。adbstop コマンドについては，「10.2.2　HADB サーバの終了方法」を参照してください。

2. サーバマシンの OS の時刻を戻す

   HADB サーバがインストールされているサーバマシンの OS の時刻を戻してください。

3. 戻した時間以上待ってから HADB サーバを開始する

   サーバマシンの OS の時刻を戻した時間以上待ったあとで，adbstart コマンドを実行して，HADB サーバを開始してください。例えば，OS の時刻を 10 分戻した場合は，10 分以上が経過したあとで，HADB サーバを開始します。

   adbstart コマンドについては，「10.2.1　HADB サーバの開始方法」を参照してください。

これで，サーバマシンの OS の時刻を戻せます。

サーバマシンの OS の時刻を戻す前と，戻したあとの時間差が大きく，待つことができない場合は，「(2) HADB サーバをすぐに開始したい場合（戻した時間分，待つことができない場合）」を参照してください。

(2)　HADB サーバをすぐに開始したい場合（戻した時間分，待つことができない場合）

手順

1. HADB サーバを正常終了する

   adbstop コマンドを実行して，HADB サーバを正常終了してください。adbstop コマンドについては，「10.2.2　HADB サーバの終了方法」を参照してください。

2. サーバディレクトリのspool ディレクトリ下のバックアップを取得する

   HADB サーバを正常終了したら，サーバディレクトリのspool ディレクトリ下（$ADBDIR/spool ディレクトリ下）に格納されているすべてのディレクトリおよびファイルについて，次に示すどちらかの方法でバックアップを取得してください。

   - OS のcp コマンドを実行する
   - adbinfoget コマンドを実行する

   adbinfoget コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbinfoget（トラブルシュート情報の取得)』を参照してください。

3. 統計ログファイルの出力先ディレクトリ下のバックアップを取得する

   サーバ定義adb_sta_log_path オペランドを省略した場合は，この手順を無視して，次の手順に進んでください。

   サーバ定義adb_sta_log_path オペランドを指定した場合は，統計ログファイルのバックアップを取得する必要があります。adb_sta_log_path オペランドに指定した統計ログファイルの出力先ディレクトリ下に格納されているすべてのファイルを，次に示すどちらかの方法でバックアップを取得してください。

   - OS のcp コマンドを実行する
   - adbinfoget コマンドを実行する

   adbinfoget コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbinfoget（トラブルシュート情報の取得)』を参照してください。

   サーバ定義adb_sta_log_path オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式)」の adb_sta_log_path オペランドを参照してください。

4. HADB サーバが出力した情報を削除する

   バックアップを取得したら，adbinfosweep コマンドを実行して，HADB サーバが出力した情報を削除してください。adbinfosweep コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbinfosweep（トラブルシュート情報の削除)』を参照してください。

   > **重要**
   >
   > adbinfosweep コマンドを実行しない場合，サーバマシンの OS の時刻を戻したあとで HADB サーバを開始すると，HADB サーバおよび HADB クライアントが予期しない動作をするおそれがあります。また，adbinfosweep コマンドを実行すると，HADB サーバが出力した情報が削除されるため，事前にバックアップを取得してください。

5. サーバディレクトリのspool ディレクトリを削除する

   OS のコマンドで，サーバディレクトリのspool ディレクトリ（$ADBDIR/spool ディレクトリ）を削除してください。

6. 統計ログファイルを削除する

   サーバ定義adb_sta_log_path オペランドを省略した場合は，この手順を無視して，次の手順に進んでください。

   サーバ定義adb_sta_log_path オペランドを指定した場合は，統計ログファイルを削除する必要があります。adb_sta_log_path オペランドに指定した統計ログファイルの出力先ディレクトリ下に格納されているすべてのファイルを削除してください。統計ログファイルを削除する方法については，「10.10.5 統計ログファイルの運用」の「(4)　統計ログファイルを削除する場合」を参照してください。

7. 空のspool ディレクトリを作成する

   OS のコマンドで，サーバディレクトリ下に，空のspool ディレクトリを作成してください。

8. サーバマシンの OS の時刻を戻す

   HADB サーバが出力した情報を削除したら，HADB サーバがインストールされているサーバマシンの OS の時刻を戻してください。

9. HADB サーバを開始する

   サーバマシンの OS の時刻を戻したら，adbstart コマンドを実行して，HADB サーバを開始してください。adbstart コマンドについては，「10.2.1　HADB サーバの開始方法」を参照してください。

これで，サーバマシンの OS の時刻を戻せます。

8.11　サーバマシンの OS のホスト名・IP アドレスの変更

ここでは，HADB サーバがインストールされているサーバマシンの OS のホスト名または IP アドレスを変更する場合の手順について説明します。

サーバマシンの OS のホスト名または IP アドレスを変更する場合，HADB サーバと HADB クライアントのどちらも設定を変更する必要があります。

■HADB サーバで実施する作業

サーバマシンの OS のホスト名または IP アドレスを変更する場合に，HADB サーバで実施する作業を次に示します。

1. **HADB サーバを正常終了する**

   adbstop コマンドを実行して，HADB サーバを正常終了してください。adbstop コマンドについては，「10.2.2　HADB サーバの終了方法」を参照してください。

2. **サーバマシンの OS のホスト名または IP アドレスを変更する**

   HADB サーバを正常終了したら，HADB サーバがインストールされているサーバマシンの OS のホスト名または IP アドレスを変更してください。

3. **HADB サーバを開始する**

   サーバマシンの OS のホスト名または IP アドレスを変更したら，adbstart コマンドを実行して，HADB サーバを開始してください。adbstart コマンドについては，「10.2.1　HADB サーバの開始方法」を参照してください。

> **メモ**
>
> 次の条件をすべて満たしている場合は，HADB サーバを稼働した状態で，サーバマシンの OS のホスト名または IP アドレスを変更できます。
>
> - マルチノード機能を使用していない
> - 監査証跡機能を使用していない
> - コールドスタンバイ構成の場合は，次の 2 つの条件を満たしている
>   - HA モニタの監視パスとして使用する LAN のホスト名を変更しない
>   - HADB クライアントと HADB サーバ間通信で使用するエイリアス IP アドレスを変更しない
>
> ただし，HADB サーバを稼働した状態でホスト名または IP アドレスを変更した場合，ホスト名または IP アドレスの変更直後から，対象の HADB サーバに接続中の HADB クライアント（AP）が通信エラーとなります。

■HADB クライアントで実施する作業

サーバマシンの OS のホスト名または IP アドレスを変更する場合に，HADB クライアントで実施する作業を次に示します。

1. AP を停止する

HADB サーバに接続する AP をすべて停止してください。

2. クライアント定義に指定している HADB サーバのホスト名または IP アドレスを変更する

AP を停止したら，クライアント定義adb_clt_rpc_srv_host オペランドに指定している HADB サーバのホスト名または IP アドレスを変更してください。

クライアント定義adb_clt_rpc_srv_host オペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』の『クライアント定義のオペランドの内容』を参照してください。

- JDBC ドライバを使用している場合

  クライアント定義adb_clt_rpc_srv_host オペランドに対応するプロパティで指定している HADB サーバのホスト名または IP アドレスを変更してください。JDBC ドライバで使用するプロパティは，システムプロパティ，ユーザプロパティ，または接続用の URL のプロパティがあります。詳細については，マニュアル『HADB　AP 開発ガイド』の『JDBC ドライバの環境設定』を参照してください。

- ODBC ドライバまたは CLI 関数を使用している場合

  クライアント定義adb_clt_rpc_srv_host オペランドに指定している HADB サーバのホスト名または IP アドレスを変更してください。詳細については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

3. AP を開始する

クライアント定義に指定している HADB サーバのホスト名または IP アドレスを変更したら，HADB サーバに接続する AP を開始してください。

重要

- マルチノード機能を使用している場合は，「16.20　サーバマシンの OS のホスト名・IP アドレスの変更（マルチノード機能使用時)」を参照してください。
- コールドスタンバイ構成の場合は，「17.15　サーバマシンの OS のホスト名・IP アドレスの変更（コールドスタンバイ構成の場合)」を参照してください。

8.12　アンインストール

ここでは，HADB サーバのアンインストール手順，およびアンインストール後に実施する作業について説明します。

マルチノード機能を使用している場合は，ここで説明しているアンインストール作業を全ノードで実施してください。

8.12.1　アンインストール手順

HADB サーバのアンインストールは，次で説明する手順に従って実行してください。なお，サーバディレクトリ下に必要なファイルがある場合は，事前にバックアップを取得してください。

> **重要**
>
> HADB サーバをアンインストールする場合，コマンドおよび AP は事前に停止しておいてください。これらを停止しないと，アンインストールに失敗するおそれがあります。

(1)　サーバディレクトリの削除

HADB 管理者で，次に示す OS のコマンドを実行して，サーバディレクトリを削除してください。HADB サーバをアンインストールできます。

```
rm -rf サーバディレクトリのパス名
```

*サーバディレクトリのパス名*を，/HADB/server とした場合の実行例を次に示します。

■コマンドの実行例

```
rm -rf /HADB/server
```

(2)　インストールデータを格納するディレクトリの削除

HADB 管理者で，次に示す OS のコマンドを実行して，HADB サーバのインストールで使用したインストールデータを格納するディレクトリを削除してください。

```
rm -rf インストールデータを格納するディレクトリのパス名
```

*インストールデータを格納するディレクトリのパス名*を，/home/adbmanager/install とした場合の実行例を次に示します。

■コマンドの実行例

```
rm -rf /home/adbmanager/install
```

8.12.2　アンインストール後に実施する作業

HADB サーバのアンインストールが完了したら，次の作業を実施してください。

- 不要な環境変数の削除
- 不要なディレクトリとファイルの削除
- ITRU の採取パターン定義ファイルの削除
- 自動起動設定のユニットファイルの削除
- サーバ定義adb_sta_log_path オペランドで指定した統計ログファイルの出力先ディレクトリの削除
- サーバディレクトリを格納するディレクトリの削除

(1)　不要な環境変数の削除

HADB サーバのインストール作業時に，「8.4　環境変数の設定」を参照して設定した環境変数のうち，不要な環境変数を HADB 管理者で削除してください。

(2)　不要なディレクトリとファイルの削除

不要なディレクトリとファイルを，スーパユーザで削除してください。削除対象のディレクトリとファイルは，サーバマシンの OS のバージョンによって異なります。

- RHEL 6 の場合

  次に示すディレクトリを，スーパユーザで削除してください。

  ■削除が必要なディレクトリ

  - 障害情報（コアファイル）の出力先ディレクトリ
  - 通信情報ファイルを格納するディレクトリ
    - /dev/HADB/pth
    - /lib/udev/devices/HADB/pth

  各ディレクトリ下にファイルが残っている場合，ファイルもあわせて削除してください。

- RHEL 7 以降の場合

  次に示すディレクトリとファイルを，スーパユーザで削除してください。

  ■削除が必要なディレクトリとファイル

  - 障害情報（コアファイル）の出力先ディレクトリ
  - 通信情報ファイルを格納するディレクトリとファイル
    - /dev/HADB/pth
    - /etc/tmpfiles.d/dev-HADB-pth.conf

  各ディレクトリ下にファイルが残っている場合，ファイルもあわせて削除してください。

(3)　ITRU の採取パターン定義ファイルの削除

HADB サーバをアンインストールしたサーバマシンに ITRU がインストールされている場合は，ITRU の採取パターン定義ファイル（!8A9_HADB）を削除する必要があります。

次に示すファイルを，スーパユーザで削除してください。

■削除が必要な ITRU の採取パターン定義ファイル

/etc/opt/hitachi/systoru/pattern/!8A9_HADB

(4)　自動起動設定のユニットファイルの削除

HADB サーバのインストール後に，「8.2.3　インストール後に実施する作業」の「(4)　HADB サーバの自動起動の設定」の手順に従い，OS の起動時や再起動時に HADB サーバを自動的に開始するように設定した場合は，サービスユニット用のユニットファイルを次の手順で削除してください。

手順

1. HADB サーバのサービスユニットを無効化する
   次に示すコマンドをスーパユーザで実行してください。

```
systemctl disable hadb.service
```

2. サービスユニット用のユニットファイルを削除する
   次に示すファイルをスーパユーザで削除してください。
   - /etc/systemd/system/hadb.service

(5)　サーバ定義adb_sta_log_path オペランドで指定した統計ログファイルの出力先ディレクトリの削除

アンインストールした HADB サーバで，サーバ定義adb_sta_log_path オペランドを指定していた場合は，統計ログファイルの出力先ディレクトリを削除してください。この操作は，HADB 管理者で実行してください。

統計ログファイルの出力先ディレクトリ下にファイルがある場合は，ファイルも一緒に削除してください。

(6)　サーバディレクトリを格納するディレクトリの削除

サーバディレクトリを格納するディレクトリ（/HADB）を，スーパユーザで削除してください。次に示す OS のコマンドを入力して，サーバディレクトリを格納するディレクトリ（/HADB）を削除してください。

■コマンドの実行例

```
rm -rf /HADB
```

9 データベースの作成

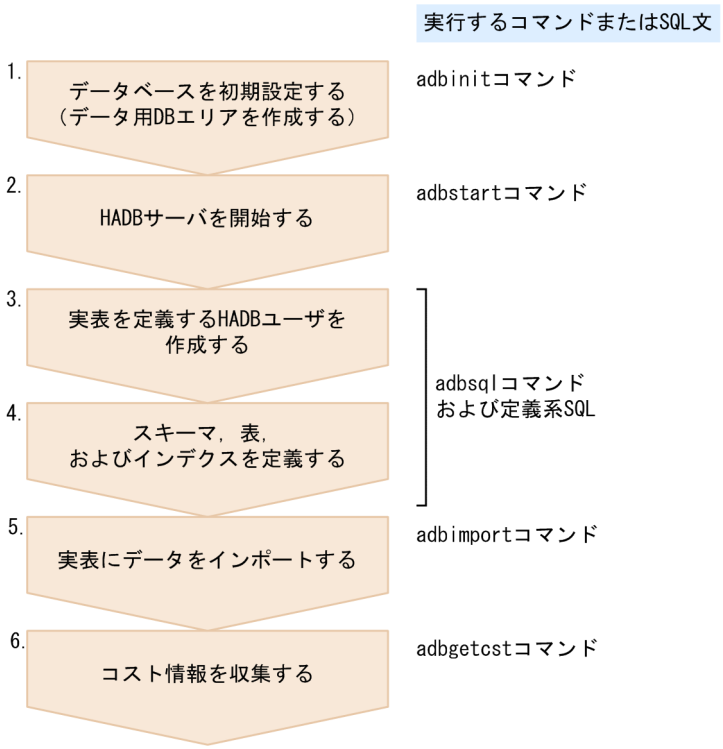
この章では，データベースの作成手順について説明します。

体験環境を構築する手順については，「3.　体験環境構築ナビ」で説明しています。こちらもあわせて読むことを推奨します。

9.1　データベースの作成手順

環境変数の設定およびサーバ定義の作成が完了したあとに，次の図に示す手順でデータベースを作成します。

図 9-1　データベースの作成手順



上記の各手順を次節以降で順次説明します。

9.2　データベースを初期設定する（データ用 DB エリアを作成する）

adbinit コマンドでデータベースの初期設定を実行し，データ用 DB エリアを作成します。

例題

データベースの初期設定を実行して，データ用 DB エリア（DBAREA01，DBAREA02）を作成します。

手順

1. HADB 管理者の OS アカウントで OS にログインする

   HADB 管理者がadbinit コマンドを実行するため，HADB 管理者の OS アカウントで OS にログインする必要があります。

   ログイン済みの場合は，この操作は不要です。

2. 初期設定オプションファイルを作成する

```
set adb_init_wrk_page_size = 32
set adb_init_wrk_blk_path = /dev/mapper/VolG00
adbinitdbarea -n DBAREA01 ¥
    -p 4 ¥
    -v /dev/mapper/VolG01,/dev/mapper/VolG02 ¥
    -q ¥
    -i 500M,2
adbinitdbarea -n DBAREA02 ¥
    -p 4 ¥
    -v /dev/mapper/VolG03,/dev/mapper/VolG04,/dev/mapper/VolG05 ¥
    -q ¥
    -i 1G,3
```

   初期設定オプションを指定した，初期設定オプションファイルを作成します。ここで作成した初期設定オプションファイルは，adbinit コマンドの入力情報になります。

   この例では，作成した初期設定オプションファイルを/HADB/server/conf/adbinit.opt に保存します。

   初期設定オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』の『adbinit コマンドの指定形式』の『初期設定オプションの形式』を参照してください。

3. adbinit コマンドを実行してデータベースを初期設定し，データ用 DB エリアを作成する

```
adbinit -u ADBUSER01 -p '#HelloHADB_01'
        /HADB/server/conf/adbinit.opt
        /HADB/db
```

   -u

   作成する HADB ユーザの認可識別子を指定します。

   -p

   作成する HADB ユーザのパスワードを指定します。

   /HADB/server/conf/adbinit.opt

   手順 2.で作成した初期設定オプションファイルの絶対パス名を指定します。

/HADB/db

DB ディレクトリ名を指定します。

認可識別子およびパスワードの指定規則については，次を参照してください。

- 「9.4.2　認可識別子の指定規則」
- 「9.4.3　パスワードの指定規則」

メモ

-u オプションに指定した認可識別子の HADB ユーザが，1 人目の HADB ユーザとして作成され，HADB サーバに登録されます。この HADB ユーザはDBA 権限とCONNECT 権限を持っています。

9.3 HADB サーバを開始する

データベースの初期設定が完了したあとに，adbstart コマンドで HADB サーバを開始します。

手順

1. adbstart コマンドを実行する

```
adbstart
```

2. HADB サーバが正常に開始したことを確認する

次のメッセージのリターンコードが0 であることを確認してください。

```
KFAA90000-I adbstart processing started.
KFAA91105-I The HADB system was started normally. (HADB server operation mode = ”NORMAL”)
KFAA90001-I adbstart processing ended. (return code = 0)
```

9.4　実表を定義するHADBユーザを作成する

実表を定義するHADBユーザを作成します。

9.4.1　HADBユーザを作成する

定義系SQLでHADBユーザを作成し，CONNECT権限およびスキーマ定義権限を付与します。

例題

実表を定義するHADBユーザADBUSER02を作成し，ADBUSER02にCONNECT権限およびスキーマ定義権限を付与します。

手順

1. adbsqlコマンドを実行する

```
adbsql
  USER-ID ?
ADBUSER01        ←ADBUSER01でHADBサーバに接続します。
  PASSWORD ?
XXXXXXXX         ←入力したパスワードは表示されません。
```

adbinitコマンドの実行時（「9.2　データベースを初期設定する（データ用DBエリアを作成する）」）に作成したHADBユーザの認可識別子とパスワードを指定します。

2. CREATE USER文を実行してHADBユーザを作成する

```
CREATE USER "ADBUSER02" IDENTIFIED BY '#HelloHADB_02'
```

上記のSQL文を実行すると，認可識別子ADBUSER02，パスワード#HelloHADB_02のHADBユーザが作成されます。

認可識別子およびパスワードの指定規則については，次を参照してください。

- 「9.4.2　認可識別子の指定規則」
- 「9.4.3　パスワードの指定規則」

3. GRANT文を実行して，CONNECT権限とスキーマ定義権限を付与する

```
GRANT CONNECT,SCHEMA TO "ADBUSER02"
```

上記のSQL文を実行すると，認可識別子ADBUSER02に，CONNECT権限とスキーマ定義権限が付与されます。

以上で，実表を定義できるHADBユーザADBUSER02の作成が完了しました。

> **メモ**
>
> adbsql コマンドで SQL 文を実行する場合は，末尾に必ず半角のセミコロン（；）を入力してください。

9.4.2 認可識別子の指定規則

認可識別子の指定規則を次に示します。

- 認可識別子に使用できる文字は，半角の英大文字，半角の英小文字，半角の数字，¥（バックスラッシュ），#，および@です。
- 認可識別子に英小文字を使用する場合は，認可識別子を二重引用符（”）で囲んで指定してください。

  （例）CREATE USER ”ADBuser02”…

  二重引用符で囲まないと，英小文字は英大文字と見なされます。例えば，「-u ADBuser02」と指定した場合，「-u ADBUSER02」と指定したと見なされます。
- 認可識別子は名前として指定するため，二重引用符（”）で囲んで指定することを推奨します。
- ALL，HADB，MASTER，およびPUBLIC は認可識別子に指定できません。
- 認可識別子は 100 文字（100 バイト）まで指定できます。
- 認可識別子の指定規則の詳細については，マニュアル『HADB　SQL リファレンス』の『基本項目』の『SQL の記述規則』の『名前の指定』を参照してください。

■コマンドの-u オプションに認可識別子を指定する場合の規則

コマンドの-u オプションに認可識別子を指定する場合，次の規則があります。

- 認可識別子に英小文字が含まれている場合

  英小文字が含まれる認可識別子を指定する場合は，認可識別子を¥”（バックスラッシュと二重引用符）で囲むか，または二重引用符（”）で囲み，さらにアポストロフィ（’）で囲む必要があります。

  （例）

  -u ¥”ADBuser02¥”

  -u ’”ADBuser02”’

  上記のように指定しないと，認可識別子の文字列はすべて英大文字として見なされます。例えば，「-u ADBuser02」と指定した場合，「-u ADBUSER02」と指定したと見なされます。
- 認可識別子に¥が含まれている場合

  ¥はシェルが別の意味で使用している特殊文字のため，認可識別子に¥が含まれている場合は，¥の前にエスケープ文字を指定してください。

  （例）

  ・認可識別子にADBUSER¥01 を指定する場合：-u ADBUSER¥¥01

  ・認可識別子にADBuser¥01 を指定する場合：-u ¥”ADBuser¥¥01¥”

上記の例では，エスケープ文字として¥を指定しています。

9.4.3 パスワードの指定規則

パスワードの指定規則を次に示します。

- パスワードに使用できる文字は，半角の英大文字，半角の英小文字，半角の数字，¥（バックスラッシュ），および次に示す半角の文字です。
  @ ` ! ” # $ % & ’ ( ) * : + ; [ ] { } , = < > | - . ^ ~ / ? _
- JDBC ドライバを利用する場合は，パスワードに次に示す半角文字を使用しないことを推奨します。
  &
- ODBC ドライバを利用する場合は，パスワードに次に示す半角文字を使用しないことを推奨します。
  [ ] { } ( ) , ; ? * = ! @
- パスワードは文字列定数の形式で指定します。そのため，パスワードをアポストロフィで囲む必要があります。
  （例）パスワードにPassword01 を指定する場合
  IDENTIFIED BY ’Password01’
- パスワードの文字列にアポストロフィがある場合，1 個のアポストロフィを表すのに 2 個のアポストロフィを指定する必要があります。
  （例）パスワードにPass’01 を指定する場合
  IDENTIFIED BY ’Pass’’01’
- パスワードに空文字は指定できません（次の指定はできません）。
  （例）IDENTIFIED BY ’’
- パスワードは 255 文字（255 バイト）まで指定できます。
- パスワードの指定形式の詳細（文字列定数の指定形式の詳細）については，マニュアル『HADB　SQL リファレンス』の『基本項目』の『定数』の『定数の記述形式』を参照してください。

■コマンドの-p オプションにパスワードを指定する場合の規則

コマンドの-p オプションにパスワードを指定する場合，次の規則があります。

- パスワードにシェルが別の意味で使用している特殊文字が含まれている場合
  ¥または|などはシェルが別の意味で使用している特殊文字のため，パスワードに特殊文字が含まれている場合は，特殊文字の前にエスケープ文字を指定してください。
  （例）
  ・パスワードにPassword¥01 を指定する場合：-p Password¥¥01
  ・パスワードにPassword|01 を指定する場合：-p Password¥|01
  上記の例では，エスケープ文字として¥を指定しています。

9.5　スキーマ，表，およびインデクスを定義する

定義系 SQL で，スキーマ，表，およびインデクスを定義します。

例題

HADB ユーザADBUSER02 のスキーマを定義し，次の表およびインデクスを定義します。

- 店舗表（SHOPSLIST）
- B-tree インデクス（SHOP_CODE_IDX）
- テキストインデクス（ADDRESS_TIDX）
- レンジインデクス（RGN_CODE_RIDX）

ADBUSER02 は，「9.4.1　HADB ユーザを作成する」で作成した HADB ユーザの認可識別子です。

手順

1. adbsql コマンドを実行する

```
adbsql
  USER-ID ?
ADBUSER02       ←ADBUSER02でHADBサーバに接続します。
  PASSWORD ?
XXXXXXXX        ←入力したパスワードは表示されません。
```

2. CREATE SCHEMA 文を実行してスキーマを定義する

```
CREATE SCHEMA "ADBUSER02"
```

上記の SQL 文を実行すると，ADBUSER02 のスキーマが定義されます。

3. CREATE TABLE 文を実行して店舗表（SHOPSLIST）を定義する

```
CREATE TABLE "SHOPSLIST"
      ("SHOP_CODE" CHAR(8) NOT NULL,
       "RGN_CODE" CHAR(6) NOT NULL,
       "SHOP_NAME" VARCHAR(20) NOT NULL,
       "TEL_NO" CHAR(10) NOT NULL,
       "ADDRESS" VARCHAR(300) NOT NULL BRANCH YES)
    IN "DBAREA01"
    PCTFREE=40
    CHUNK=100
```

4. CREATE INDEX 文を実行して，B-tree インデクス（SHOP_CODE_IDX）を定義する

```
CREATE INDEX "SHOP_CODE_IDX"
      ON "SHOPSLIST" ("SHOP_CODE")
    IN "DBAREA01"
    PCTFREE = 50
    EMPTY
```

上記の SQL 文を実行すると，SHOPSLIST の店舗コード列（SHOP_CODE）に B-tree インデクス（SHOP_CODE_IDX）が定義されます。

5. CREATE INDEX 文を実行して，テキストインデクス（ADDRESS_TIDX）を定義する

```
CREATE INDEX "ADDRESS_TIDX"
      ON "SHOPSLIST"("ADDRESS")
    IN "DBAREA01"
    PCTFREE = 50
    EMPTY
    INDEXTYPE TEXT
```

上記の SQL 文を実行すると，SHOPSLIST の住所列（ADDRESS）にテキストインデクス（ADDRESS_TIDX）が定義されます。

6. CREATE INDEX 文を実行して，レンジインデクス（RGN_CODE_RIDX）を定義する

```
CREATE INDEX "RGN_CODE_RIDX"
      ON "SHOPSLIST"("RGN_CODE")
    IN "DBAREA01"
    EMPTY
    INDEXTYPE RANGE
```

上記の SQL 文を実行すると，SHOPSLIST の店舗エリアコード列（RGN_CODE）にレンジインデクス（RGN_CODE_RIDX）が定義されます。

メモ

adbsql コマンドで SQL 文を実行する場合は，末尾に必ず半角のセミコロン（ ; ）を入力してください。

9.6　実表にデータをインポートする

実表を定義したあとに，adbimport コマンドで実表にデータをインポートします。

例題

店舗表（SHOPSLIST）にデータをインポートします。

手順

1. HADB 管理者の OS アカウントで OS にログインする

   HADB 管理者がadbimport コマンドを実行するため，HADB 管理者の OS アカウントで OS にログインする必要があります。

   ログイン済みの場合は，この操作は不要です。

2. 入力データパスファイルを作成する

```
/home/adbmanager/imp_data/imp1.csv
/home/adbmanager/imp_data/imp2.csv
```

   入力データファイルのパスを指定した，入力データパスファイルを作成します。ここで作成した入力データファイルは，adbimport コマンドの入力情報になります。

   この例では，作成した入力データパスファイルを/home/adbmanager/imp_file/imp_data_path.txt に保存します。

3. インポートオプションファイルを作成する

```
set adb_import_errdata_file_name = ”/home/adbmanager/imp_shopslist/errdata.csv”
set adb_import_errdata_num = 500
set adb_import_rthd_num = 2
set adb_import_sort_buff_size = 2
                :
```

   インポートオプションを指定した，インポートオプションファイルを作成します。ここで作成したインポートオプションファイルは，adbimport コマンドの入力情報になります。

   この例では，作成したインポートオプションファイルを/home/adbmanager/imp_file/imp_opt_file.txt に保存します。

   インポートオプションについては，マニュアル『HADB　コマンドリファレンス』の『adbimport (データのインポート)』の『adbimport コマンドの指定形式』の『インポートオプションの形式』を参照してください。

4. adbimport コマンドを実行して，店舗表（SHOPSLIST）にデータをインポートする

```
adbimport -u ADBUSER02 -p ’#HelloHADB_02’ -k ”’” -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file.txt
          SHOPSLIST
          /home/adbmanager/imp_file/imp_data_path.txt
```

-u

adbimport コマンドを実行する HADB ユーザの認可識別子を指定します。データインポートの対象表を所有している HADB ユーザの認可識別子を指定してください。

-p

adbimport コマンドを実行する HADB ユーザのパスワードを指定します。

-k

入力データファイル中で使用している囲み文字を指定します。

-s

入力データファイル中で使用している区切り文字を指定します。

-g

データインポート処理の経過メッセージ（KFAA80205-I）の出力間隔を指定します。

-w

データインポート処理時に作成される作業用一時ファイルを格納するディレクトリを指定します。

-z

手順 3.で作成したインポートオプションファイルの絶対パス名を指定します。

SHOPSLIST

データインポートの対象表を指定します。

/home/adbmanager/imp_file/imp_data_path.txt

手順 2.で作成した入力データパスファイルの絶対パス名を指定します。

9.7　コスト情報を収集する

実表にデータをインポートしたあとに，adbgetcst コマンドで実表のコスト情報を収集します。

例題

店舗表（SHOPSLIST）のコスト情報を収集します。

手順

1. HADB 管理者の OS アカウントで OS にログインする

   HADB 管理者がadbgetcst コマンドを実行するため，HADB 管理者の OS アカウントで OS にログインする必要があります。

   ログイン済みの場合は，この操作は不要です。

2. コスト情報収集オプションファイルを作成する

```
set adb_getcst_rthd_num = 10
```

   コスト情報収集オプションを指定した，コスト情報収集オプションファイルを作成します。ここで作成したコスト情報収集オプションファイルは，adbgetcst コマンドの入力情報になります。

   この例では，作成したコスト情報収集オプションファイルを/home/adbmanager/cost_file/cost_opt_file.txt に保存します。

   コスト情報収集オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbgetcst (コスト情報の収集)』の『adbgetcst コマンドの指定形式』の『コスト情報収集オプションの形式』を参照してください。

3. adbgetcst コマンドを実行してコスト情報を収集する

```
adbgetcst -u ADBUSER02 -p '#HelloHADB_02' -t SHOPSLIST
          -z /home/adbmanager/cost_file/cost_opt_file.txt
```

   -u

   adbgetcst コマンドを実行する HADB ユーザの認可識別子を指定します。コスト情報を収集する実表を所有している HADB ユーザの認可識別子を指定してください。

   -p

   adbgetcst コマンドを実行する HADB ユーザのパスワードを指定します。

   -t

   コスト情報を収集する実表の表名を指定します。

   -z

   手順 2.で作成したコスト情報収集オプションファイルの絶対パス名を指定します。

> **メモ**
>
> コスト情報を収集するタイミングについては，「11.5　コスト情報の収集」を参照してください。

10 定期運用

この章では，HADB サーバで定期的に行う運用項目について説明します。

10.1　運用項目の一覧

HADB サーバの運用項目の一覧を次の表に示します。

表 10-1　HADB サーバの運用項目の一覧

項番	運用項目			参照先
1	定期運用	HADB サーバの開始		「10.2.1　HADB サーバの開始方法」
2		HADB サーバの終了		「10.2.2　HADB サーバの終了方法」
3		HADB サーバの稼働モードの変更・確認		「10.2.3　HADB サーバの稼働モード」
4		HADB サーバの自動再起動の設定		「10.2.4　HADB サーバの自動再起動の設定」
5		データベースのバックアップ		「10.3　データベースのバックアップ」
6		モニタリング	メッセージの確認	「10.4　メッセージの確認」
7			リソースの使用状況の監視	「10.5　リソースの使用状況の監視（監視対象メッセージの一覧）」
8			メモリの使用状況の確認	「10.6　メモリの使用状況の確認」
9			HADB サーバで実行中のスレッドの確認	「10.7　HADB サーバで実行中のスレッドの確認」
10			トランザクションの処理状況の確認	「10.8　トランザクションの処理状況の確認」
11			データベースの状態や使用量の確認	「10.9　データベースの状態や使用量の確認」
12			HADB サーバの稼働情報の確認	「10.10　統計解析の運用（HADB サーバの稼働情報の確認）」
13			SQL 文の実行履歴情報（SQL トレース情報）の取得および確認	「10.11　SQL トレース機能の運用」
14			システムログファイルの運用	「10.12　システムログファイルの運用」
15	非定期運用	実表の運用		「11.1　実表の運用」
16		ビュー表の定義，検索，削除		「11.2　ビュー表の運用」
17		インデクスの運用		「11.3　インデクスの運用」

項番	運用項目		参照先
18		レギュラーマルチチャンク表およびアーカイブマルチチャンク表の運用	「11.4　マルチチャンク表の運用」
19		コスト情報の収集	「11.5　コスト情報の収集」
20		HADB ユーザの管理	「11.6　HADB ユーザの管理」
21		ユーザ権限とスキーマ操作権限の管理	「11.7　ユーザ権限とスキーマ操作権限の管理」
22		アクセス権限の管理	「11.8　アクセス権限の管理」
23		スキーマの運用	「11.9　スキーマの運用」
24		データ用 DB エリアの運用	「11.10　データ用 DB エリアの運用」
25		作業表用 DB エリアの運用	「11.11　作業表用 DB エリアの運用」
26		バッファの運用	「11.12　バッファの運用」
27		クライアントグループの設定内容の確認	「11.13　クライアントグループの設定内容の確認」
28		クライアント定義の集中管理の運用	「11.14　クライアント定義の集中管理の運用」
29		CSV ファイル中のデータを検索するときの運用	「11.15　CSV ファイル中のデータを検索するときの運用」
30		同義語検索を行う際の運用	「11.16　同義語検索を行う際の運用」
31		システム表の再編成	「11.17　システム表の再編成」
32		更新行のカラム化機能の運用	「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」
33		監査証跡機能の運用	「12.　監査証跡機能の運用」
34	チューニング	処理性能の向上	「13.1　処理性能の向上に関するチューニング」
35		バッファの見直しによる SQL 文の実行時間の短縮	「13.2　バッファの見直しによる SQL 文の実行時間の短縮に関するチューニング」
36		コマンドの実行時間の短縮	「13.3　コマンドの実行時間の短縮に関するチューニング」
37		メモリ使用量の削減	「13.4　メモリ使用量の削減に関するチューニング」
38		ハッシュテーブル領域サイズの見直しによる SQL 文の実行時間の短縮	「13.5　ハッシュテーブル領域サイズの見直しによる SQL 文の実

項番	運用項目		参照先
			行時間の短縮に関するチューニング」
39		ハッシュグループ化領域サイズの見直しによる SQL 文の実行時間の短縮	「13.6　ハッシュグループ化領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング」
40		ハッシュジョインが適用されるINNER JOIN の結合順序の見直しによる SQL 文の実行時間の短縮	「13.7　ハッシュジョインが適用される INNER JOIN の結合順序の見直しによる SQL 文の実行時間の短縮に関するチューニング」
41		ハッシュフィルタ領域サイズの見直しによる SQL 文の実行時間の短縮	「13.8　ハッシュフィルタ領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング」
42		表定義プールサイズの見直しによる SQL 文の実行時間の短縮	「13.9　表定義プールサイズの見直しによる SQL 文の実行時間の短縮に関するチューニング」
43	障害運用	異常終了したかどうかの確認	「14.1　障害発生時の対処の流れ」
44		トラブルシュート情報の取得	「14.2　トラブルシュート情報の取得（adbinfoget コマンド）」
45		取得したトラブルシュート情報の削除	「14.3　トラブルシュート情報の削除（adbinfosweep コマンド）」
46		トラブルシュート	「15.　トラブルシュート」

10.2　HADB サーバの開始・終了と稼働モード

ここでは，HADB サーバの開始方法，終了方法および稼働モードについて説明します。

10.2.1　HADB サーバの開始方法

HADB サーバを開始する場合，adbstart コマンドを実行してください。HADB サーバが開始します。

次のとおりにコマンドを実行してください。

```
adbstart
```

重要

HADB サーバには，複数の稼働モードがあります。稼働モードの詳細については，「10.2.3 HADB サーバの稼働モード」を参照してください。

adbstart コマンド実行時にオプションを指定すると，HADB サーバ開始後の稼働モードを変更できます。オプションを指定しない場合，前回の HADB サーバ終了時の稼働モードが引き継がれます。adbstart コマンドのオプションの指定形式については，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

HADB サーバの開始後，必要に応じて HADB サーバの自動再起動の設定をしてください。詳細については，「10.2.4　HADB サーバの自動再起動の設定」を参照してください。

(1)　HADB サーバの開始モード

HADB サーバには，次に示す開始モードがあります。開始モードによって動作が異なります。

表 10-2　HADB サーバの開始モード

項番	開始モード	実行するコマンド	説明	前回の終了モード
1	正常開始	adbstart	通常の開始モードです。 正常開始では，データベースの回復処理は行いません。	正常終了
2	再開始		前回の終了モードが右記の場合，HADB サーバは（自動的に）再開始となります。 再開始では，システムログを使用してデータベースの回復処理を行います。	• 強制終了 • 異常終了

(2)　開始処理完了の確認

HADB サーバの開始処理の完了を確認する場合，adbstart コマンドのリターンコードを確認してください。

- リターンコードが0 の場合

  開始処理が正常終了しました。AP からのデータベースに対するアクセス要求を受け付けることができます。

- リターンコードが4 の場合

  開始処理は終了しましたが，警告メッセージが出力されています。

  HADB サーバは開始しています。表示されたメッセージ，またはメッセージログファイルに出力されたメッセージを確認して対処してください。

- リターンコードが8 の場合

  開始処理が異常終了しました。HADB サーバの開始処理中にエラーが発生しています。表示されたメッセージ，またはメッセージログファイルに出力されたメッセージを確認して対処してください。

10.2.2　HADB サーバの終了方法

HADB サーバを終了する場合，adbstop コマンドを実行してください。HADB サーバが終了します。

次のとおりにコマンドを実行してください。

```
adbstop※
```

注※

adbstop コマンド実行時にオプションを指定すると，HADB サーバの終了モードを変更できます。詳細については，「(1)　HADB サーバの終了モード」および「(2)　オプションの指定と正常終了の種類」を参照してください。

> **重要**
>
> HADB サーバに接続中の AP またはコマンドがある場合，adbstop コマンドを実行しても HADB サーバを終了できません。

(1)　HADB サーバの終了モード

HADB サーバには，次に示す終了モードがあります。終了モードによって，adbstop コマンドに指定するオプションが異なります。

表 10-3　HADB サーバの終了モード

項番	終了モード	実行するコマンドおよびオプション	説明
1	正常終了	adbstop	通常の終了モードです。 正常終了には 4 種類あり，指定するオプションによって終了処理が異なります。詳細については，「(2)　オプションの指定と正常終了の種類」を参照してください。

項番	終了モード	実行するコマンドおよびオプション	説明
2	強制終了	adbstop --force	処理中のトランザクションの完了を待たないで，HADB サーバが終了します。 adbstop --force コマンドを実行すると，新規のコネクション確立要求を受け付けなくなります。 処理中のトランザクションは，HADB サーバを再開始したときに回復処理の対象となります。
3	異常終了	－	何らかの異常によって，HADB サーバが終了する場合の終了モードです。 処理中のトランザクションの完了を待たないで，HADB サーバは終了します。 処理中のトランザクションは，HADB サーバを再開始したときに回復処理の対象となります。

（凡例）

－：実行するコマンドはありません。

(2) オプションの指定と正常終了の種類

adbstop コマンドに指定するオプションによって，正常終了時の処理が異なります。adbstop コマンドに指定するオプションと正常終了の種類を次の表に示します。

adbstop コマンドのオプションの指定形式については，マニュアル『HADB　コマンドリファレンス』の『adbstop（HADB サーバの終了)』を参照してください。

表 10-4　adbstop コマンドに指定するオプションと正常終了の種類

項番	実行するコマンドおよびオプション	説明
1	adbstop	adbstop コマンド（オプション指定なし）を実行すると，HADB サーバに接続中の AP および実行中のコマンドがない場合に，HADB サーバを終了します。 **adbstop コマンドを実行する場面** HADB サーバに接続中の AP および実行中のコマンドがなく，正常終了したい場合に，このコマンドを実行してください。 **adbstop コマンド実行時の処理** adbstop コマンドを実行して HADB サーバの終了処理が開始されると，新規のコネクション確立要求を受け付けなくなります。 なお，HADB サーバに接続中の AP または実行中のコマンドがある場合は，KFAA91152-E メッセージが出力され，エラーとなります（HADB サーバを終了できません)。その場合は，マニュアル『HADB　メッセージ』を参照し，KFAA91152-E メッセージの対策欄に従って対処してください。
2	adbstop --wait connection	adbstop --wait connection コマンドを実行すると，次の処理を待ってから，HADB サーバを終了します。 • HADB サーバに接続中の AP の，HADB サーバからの切り離し • 実行中のコマンドの終了

項番	実行するコマンドおよびオプション	説明
		adbstop --wait connection コマンドを実行する場面 次のような場合に，このコマンドを実行してください。 • 実行中のコマンド（adbimport コマンドなど）が終了したらすぐに正常終了したい場合 • HADB サーバに接続中の AP による処理が完了し，AP が HADB サーバから切り離されたら，すぐに正常終了したい場合 adbstop --wait connection コマンド実行時の処理 adbstop --wait connection コマンドの実行後，HADB サーバは終了待機状態となり，次の処理を待ちます（終了待機状態の待ち時間は，-t オプションで指定できます）。 • HADB サーバに接続中の AP の，HADB サーバからの切り離し • 実行中のコマンドの終了 このとき，HADB サーバは新規のコネクション確立要求を受け付けなくなります（HADB サーバに接続中の AP または実行中のコマンドから発行される新規のトランザクション開始要求は受け付けます）。 HADB サーバに接続中の AP がすべて切り離され，かつ実行中のコマンドがすべて終了すると，HADB サーバの終了処理が開始されます。
3	adbstop --wait transaction	adbstop --wait transaction コマンドを実行すると，処理中のトランザクションの完了，および実行中のコマンドの終了を待ってから，HADB サーバが終了します。 adbstop --wait transaction コマンドを実行する場面 次のような場合に，このコマンドを実行してください。 • 処理中のトランザクションが完了したらすぐに正常終了したい場合 • 実行中のコマンド（adbimport コマンドなど）が終了したら，すぐに正常終了したい場合 • AP は接続中だが，トランザクションを開始していないときに正常終了したい場合 adbstop --wait transaction コマンド実行時の処理 adbstop --wait transaction コマンドの実行後，HADB サーバは終了待機状態となり，処理中のトランザクションの完了，および実行中のコマンドの終了を待ちます（終了待機状態の待ち時間は，-t オプションで指定できます）。 このとき，HADB サーバは新規のコネクション確立要求および新規のトランザクション開始要求を受け付けなくなります。 処理中のトランザクションがすべて完了し，かつ実行中のコマンドがすべて終了すると，HADB サーバに接続中の AP およびコマンドがすべて切り離されて，HADB サーバの終了処理が開始されます。
4	adbstop --cancel	adbstop --cancel コマンドを実行すると，処理中のトランザクションおよび実行中のコマンドをキャンセルして，HADB サーバを終了します。 adbstop --cancel コマンドを実行する場面 処理をキャンセルしてすぐに正常終了したい場合などに，このコマンドを実行してください。

項番	実行するコマンドおよびオプション	説明
		adbstop --cancel コマンド実行時の処理 adbstop --cancel コマンドの実行後，処理中のトランザクションおよび実行中のコマンドをキャンセルします。 このとき，HADB サーバは新規のコネクション確立要求および新規のトランザクション開始要求を受け付けなくなります。 処理中のトランザクションおよび実行中のコマンドがすべてキャンセルされると，HADB サーバに接続中の AP およびコマンドがすべて強制的に切り離されて，HADB サーバの終了処理が開始されます。

■終了待機状態とは

終了待機状態とは，HADB サーバが次の処理を待っている状態のことです。

- HADB サーバに接続中の AP の，HADB サーバからの切り離し
- 実行中のコマンドの終了
- 処理中のトランザクションの完了

adbstop --wait connection コマンドおよびadbstop --wait transaction コマンドを実行したときだけ，HADB サーバは終了待機状態になります。終了待機状態には待ち時間があり，adbstop コマンドの-t オプションで指定できます。-t オプションを省略した場合は，デフォルトの待ち時間が適用されます。

■終了待機状態の待ち時間

終了待機状態の待ち時間内に，HADB サーバに接続中の AP が HADB サーバから切り離されなかったり，実行中のコマンドが終了しなかったり，処理中のトランザクションが完了しなかったりして，終了待機状態の待ち時間を超えた（タイムアウトした）場合は，HADB サーバは終了しません。この場合は，KFAA91152-E メッセージが出力されるため，メッセージを参照して，エラー要因を取り除いてください。

なお，終了待機状態の待ち時間を超えた（タイムアウトした）場合，HADB サーバは，新規のコネクション確立要求および新規のトランザクション開始要求を受け付けられるようになります。

■終了待機状態の確認方法

HADB サーバが終了待機状態かどうかは，adbls -d srv コマンドで確認できます。出力項目「STATUS」の表示が"STOPWAIT"であれば，HADB サーバは終了待機状態となっています。

終了待機状態をキャンセルしたい場合は，OS のkill コマンドでadbstop コマンドのプロセスを終了させてください。終了待機状態をキャンセルできます。

終了待機状態がキャンセルされた場合，HADB サーバは，新規のコネクション確立要求および新規のトランザクション開始要求を受け付けられるようになります。

adbls -d srv コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d srv (HADB サーバの状態表示)』を参照してください。

(3)　終了処理完了の確認

HADB サーバの終了処理の完了を確認する場合，adbstop コマンドのリターンコードを確認してください。

- リターンコードが0 の場合

  終了処理が正常終了しました。

- リターンコードが4 の場合

  終了処理は終了しましたが，警告メッセージが出力されています。

  HADB サーバは終了しています。表示されたメッセージ，またはメッセージログファイルに出力されたメッセージを確認して対処してください。

- リターンコードが8 の場合

  終了処理が異常終了しました。HADB サーバの終了処理中にエラーが発生しています。表示されたメッセージ，またはメッセージログファイルに出力されたメッセージを確認して対処してください。

10.2.3　HADB サーバの稼働モード

HADB サーバには，次の稼働モードがあります。

- 通常モード
- 静止モード
- オフラインモード
- メンテナンスモード

各稼働モードの詳細について，次の表に示します。

表 10-5　HADB サーバの稼働モードの詳細

項番	稼働モードの種類	説明
1	通常モード	通常の稼働モードです。 通常モードのときは，HADB サーバおよび HADB クライアントで，通常どおり SQL 文およびコマンドを実行できます。 なお，HADB サーバを初めて開始する場合に，オプションを省略してadbstart コマンドを実行すると，通常モードで開始されます。
2	静止モード	データベースの更新を抑止する稼働モードです。 HADB サーバを稼働したまま，データベースのバックアップを取得するときなどに，このモードを使用します。 静止モードのときは，HADB サーバおよび HADB クライアントで，データベースを検索する SQL 文は実行できます。しかし，データベースを更新する SQL 文は実行できません。また，一部のコマンドも実行できません。 静止モードで実行できる SQL 文およびコマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。
3	オフラインモード	HADB サーバがインストールされているサーバマシンとは異なるマシンにインストールされている HADB クライアントからの接続を受け付けない稼働モードです。 コマンド実行時に，HADB サーバの処理性能を最大限に利用したいときなどに，このモードを使用します。

項番	稼働モードの種類	説明
		オフラインモードのときは，HADB サーバと同じサーバマシンにインストールされている HADB クライアントからは，通常どおり SQL 文およびコマンドを実行できます。しかし，HADB サーバがインストールされているサーバマシンとは異なるマシンにインストールされている HADB クライアントからは，SQL 文およびコマンドを実行できません。 オフラインモードで実行できる SQL 文およびコマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。
4	メンテナンスモード	HADB サーバが AP の接続要求を受け付けない稼働モードです。メンテナンスモードの場合は，adbmodarea コマンドなどの特定のコマンドだけが実行できます。メンテナンスモードの場合に実行できるコマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』の『各稼働モードで実行できる SQL 文およびコマンド』を参照してください。 メンテナンスモードは，次に示す場合に使用します。 • ほかのコマンドや AP と同時実行できないコマンドを実行する場合 • マルチノード構成にノードを復帰させる場合 マルチノード構成へのノードの復帰については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

(1)　稼働モードを変更する方法

HADB サーバの稼働モードは，HADB サーバの開始時，または HADB サーバの稼働中に変更できます。稼働モードを変更する方法を次に示します。

■稼働モードを通常モードに変更する方法

- HADB サーバの開始時に変更する場合
  HADB サーバを開始するときに，adbstart --normal コマンドを実行してください。
- HADB サーバの稼働中に変更する場合
  静止モード，オフラインモード，またはメンテナンスモードで稼働しているときに，adbchgsrvmode --normal コマンドを実行してください。

■稼働モードを静止モードに変更する方法

- HADB サーバの開始時に変更する場合
  HADB サーバを開始するときに，adbstart --quiescence コマンドを実行してください。
- HADB サーバの稼働中に変更する場合
  通常モードで稼働しているときに，adbchgsrvmode --quiescence コマンドを実行してください。通常モードから，静止モードに変更できます。

■稼働モードをオフラインモードに変更する方法

- HADB サーバの開始時に変更する場合
  HADB サーバを開始するときに，adbstart --offline コマンドを実行してください。
- HADB サーバの稼働中に変更する場合

通常モードで稼働しているときに，adbchgsrvmode --offline コマンドを実行してください。通常モードから，オフラインモードに変更できます。

■稼働モードをメンテナンスモードに変更する方法

- HADB サーバの開始時に変更する場合
  HADB サーバを開始するときに，adbstart --maintenance コマンドを実行してください。
- HADB サーバの稼働中に変更する場合
  通常モードで稼働しているときに，adbchgsrvmode --maintenance コマンドを実行してください。通常モードから，メンテナンスモードに変更できます。

(2) 稼働モードを確認する方法

HADB サーバの稼働モードは，次に示すコマンドで確認できます。

```
adbls -d srv
```

コマンドを実行すると，HADB サーバの状態が表示されます。実行結果のSTATUS の項目を確認してください。

- STATUS がACTIVE の場合
  稼働モードは，通常モードになります。
- STATUS がQUIESCE の場合
  稼働モードは，静止モードになります。
- STATUS がOFFLINE の場合
  稼働モードは，オフラインモードになります。
- STATUS がMAINTNCE の場合
  稼働モードは，メンテナンスモードになります。

10.2.4 HADB サーバの自動再起動の設定

HADB サーバが異常終了した場合，通常は，HADB サーバは自動的に再起動しません。しかし，adbmonitor コマンドを実行しておくと，HADB サーバが異常終了したときに，HADB サーバの再起動が自動的に行われます。

自動再起動の設定手順を次に示します。

なお，自動再起動の設定ができるのは，マルチノード機能を使用していないときだけです。

手順

1. HADB サーバを開始する

adbstart コマンドを実行してください。

2. HADB サーバの自動再起動の設定を実施する

   adbmonitor -r コマンドを実行してください。

adbmonitor -r コマンドを実行すると，adbmonitor コマンドが HADB サーバの状態を監視します。そして，HADB サーバが異常終了したときに，adbmonitor コマンドのコマンドプロセスから HADB サーバを再起動します。

> **重要**
>
> HADB サーバの異常終了原因によっては，HADB サーバが自動再起動できないことがあります（同じ原因で繰り返し HADB サーバが異常終了することがあります）。この場合は，［Ctrl］＋［C］キーなどで，adbmonitor コマンドのコマンドプロセスを強制終了してください。

HADB サーバが異常終了したかどうかは，メッセージログファイルにKFAA60003-E メッセージが出力されているかどうかで確認できます。HADB サーバが異常終了し，自動再起動によって HADB サーバが正常に再起動した場合でも，adbinfoget コマンドでトラブルシュート情報を取得してください。そして，弊社問い合わせ窓口に連絡してください。

なお，adbstop コマンドで HADB サーバを正常終了した場合，HADB サーバを監視していたadbmonitor コマンドのコマンドプロセスも終了します。そのため，次回の HADB サーバを開始するときにも，自動再起動を設定したい場合は，HADB サーバの開始後にadbmonitor -r コマンドを実行してください。

10.3　データベースのバックアップ

ここでは，データベースのバックアップを取得する方法，およびバックアップから回復する方法について説明します。

10.3.1　バックアップの取得方法

ここでは，バックアップを取得するタイミング，バックアップの種類，およびバックアップを取得する手順について説明します。

マルチノード機能を使用している場合のバックアップの取得方法については，「16.8　データベースのバックアップ（マルチノード機能の使用時）」を参照してください。

(1)　バックアップを取得するタイミング

原則として，障害発生時などにデータベースを回復するために，定期的にバックアップを取得するようにしてください。特に，次に示す運用項目を実施する前には，必ずバックアップを取得してください。

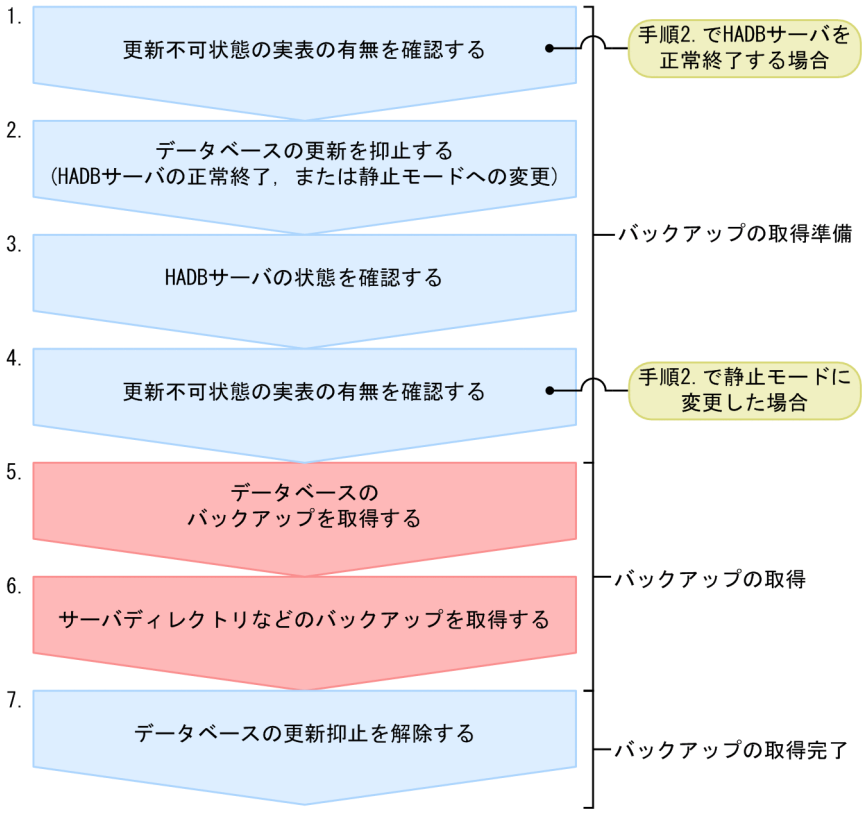
- adbmodarea コマンドを実行する場合
- HADB サーバのバージョンアップを実施する場合

> **重要**
>
> データベースのバックアップを取得する際は，データ用 DB エリアファイルを含む必要なファイルすべてのバックアップを取得してください。このバックアップをフルバックアップといいます。

(2)　バックアップを取得する手順

データベースのバックアップを取得する手順を次に示します。

図 10-1　バックアップを取得する手順



図中の各手順の詳細を次で説明します。なお，図中の番号は，次に示す各手順の番号と対応しています。

なお，手順 2.の『データベースの更新を抑止する』で，『HADB サーバの稼働モードを静止モードに変更する』を選択する場合は，手順 1.を飛ばして手順 2.から開始してください。

1. 更新不可状態の実表の有無を確認する

   データベースのバックアップを取得する前に，更新不可状態の実表がないことを確認する必要があります。adbdbstatus コマンドに，-c table オプションを指定して実行してください。なお，手順 2.で HADB サーバの稼働モードを静止モードに変更する場合は，手順 1.の実施は不要です。手順 2.に進んでください。

   すべての実表の状態を調べる場合の指定例を次に示します。

   ```
   adbdbstatus -c table
   ```

   出力結果のNon-updatable 列に，non-updatable が出力されていないことを確認してください。non-updatable が出力されている場合，実表が更新不可状態になっています。「15.8.1　実表が更新不可状態になった場合」を参照して，更新不可状態を解除してください。

   > **重要**
   >
   > 更新不可状態の実表がある状態でバックアップを取得し，そのバックアップからデータベースを回復してしまうと，実表の更新不可状態を解除できなくなるおそれがあります。実表が更新不可状態の場合，検索処理や更新処理を実行できません。

2. データベースの更新を抑止する

データベースのバックアップを取得する前に，データベースの更新を抑止する必要があります。次に示すどちらかを必ず実施してください。

- **HADB サーバを正常終了する**

  HADB サーバを終了しても問題ない場合は，adbstop コマンドを実行して，HADB サーバを正常終了してください。

- **HADB サーバの稼働モードを静止モードに変更する**

  HADB サーバが稼働中で終了できない場合は，adbchgsrvmode --quiescence コマンドを実行して，稼働モードを静止モードに変更してください。

> **重要**
>
> 稼働モードを静止モードにしてバックアップを取得した場合，データベースの回復後に HADB サーバを開始すると，HADB サーバは再開始します。この場合，バックアップ取得後にサーバ定義を変更していると，取得したバックアップから回復したあとで HADB サーバを再開始できないことがあります。そのため，静止モードにしてバックアップを取得する場合は，手順 6.でサーバ定義ファイルのバックアップをあわせて取得することを推奨します。なお，サーバ定義の変更によって再開始できない場合は，KFAA50024-E メッセージおよびKFAA50025-E メッセージを基に対策してください。

3. HADB サーバの状態を確認する

手順 2.が完了したら，HADB サーバの状態を確認します。次に示すコマンドを実行して，データベースの更新を抑止できていることを確認してください。

```
adbls -d srv
```

実行結果のSTATUS の項目を確認してください。

- **HADB サーバを正常終了した場合**

  出力項目のSTATUS が，STOP になっていることを確認してください。確認したら，手順 5.に進んでください。

- **HADB サーバの稼働モードを静止モードに変更した場合**

  出力項目のSTATUS が，QUIESCE になっていることを確認してください。確認したら，手順 4.に進んでください。

4. 更新不可状態の実表の有無を確認する

データベースのバックアップを取得する前に，更新不可状態の実表がないことを確認する必要があります。adbdbstatus コマンドに，-c table オプションを指定して実行してください。

なお，手順 2.で HADB サーバを正常終了した場合は，手順 5.に進んでください。

すべての実表の状態を調べる場合の指定例を次に示します。

```
adbdbstatus -c table
```

出力結果のNon-updatable 列に，non-updatable が出力されていないことを確認してください。non-updatable が出力されている場合，実表が更新不可状態になっています。「15.8.1　実表が更新不可状態になった場合」を参照して，更新不可状態を解除してください。

5. データベースのバックアップを取得する

データベースの更新を抑止したら，バックアップを取得できます。バックアップを取得する必要があるファイルは，すべてバックアップを取得してください。一部のファイルだけバックアップを取得すると，バックアップから回復するときにデータベースの整合性が取れなくなり，その後の動作を保証できません。

対象のファイルがシンボリックリンクファイルの場合は，リンク先のファイルおよびブロックスペシャルファイルについても，バックアップを取得してください。なお，バックアップを取得するファイルは，スパースファイルにしないでください。

バックアップを取得する必要があるファイルを次の表に示します。

表 10-6　バックアップを取得する必要があるファイルの一覧

項番	バックアップを取得する必要があるファイル	ファイルの格納先	ファイルの種類
1	コマンドステータスファイル	$DBDIR/ADBSYS/ADBUTL	レギュラーファイル
2	システムログファイル	$DBDIR/ADBSYS/ADBSLG	レギュラーファイル
3	ステータスファイル	$DBDIR/ADBSYS/ADBSTS	レギュラーファイル
4	マスタディレクトリ用 DB エリアファイル	$DBDIR/ADBMST	レギュラーファイル，またはブロックスペシャルファイル
5	ディクショナリ用 DB エリアファイル	$DBDIR/ADBDIC	レギュラーファイル，またはブロックスペシャルファイル
6	システム表用 DB エリアファイル	$DBDIR/ADBSTBL	レギュラーファイル，またはブロックスペシャルファイル
7	データ用 DB エリアを構成するすべてのファイル	$DBDIR/DBAREA※1	レギュラーファイル，またはブロックスペシャルファイル
8	アーカイブファイル	アーカイブディレクトリ※2	レギュラーファイル
9	同義語辞書ファイル	同義語辞書ファイルの格納ディレクトリ※3	レギュラーファイル

注※1

adbinit コマンドまたはadbmodarea コマンドで，ユーザが指定した名称になります。

注※2

CREATE TABLE 文のチャンクアーカイブ指定のARCHIVEDIR で指定したディレクトリです。

注※3

サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

同義語辞書ファイルは，同義語一覧定義ファイルから再作成できるため，同義語一覧定義ファイルを保管している場合は，同義語辞書ファイルをフルバックアップの取得対象に含める必要はありま

せん。ただし，この場合，データベースの回復時，データベースと同義語辞書ファイルの同期は取られません。そのため，フルバックアップの取得後に同義語辞書を更新した場合，次の検索結果が異なることがあります。

・フルバックアップ取得時点の同義語検索の検索結果

・データベースをフルバックアップ取得時点に回復したときの同義語検索の検索結果

メモ

次に示すディレクトリ下のファイルについては，バックアップの取得は不要です。

- 作業用ディレクトリ下（$DBDIR/ADBWORK）のファイル
- 障害情報（コアファイル）の出力先ディレクトリ下（$DBDIR/SPOOL）のファイル

  サーバ定義adb_core_path オペランドを指定した場合は，adb_core_path オペランドに指定したディレクトリになります。

なお，これらのファイルも，上記の表に示すファイルと一緒にバックアップを取得して，バックアップから回復しても問題はありません。

メモ

監査証跡の出力先ディレクトリ（サーバ定義のadb_audit_log_path オペランドに指定したディレクトリ）下の監査証跡ファイルのバックアップを取得する必要はありません。監査証跡の出力先ディレクトリ下の監査証跡ファイルは，バックアップを取得するのではなく，監査証跡の保存先ディレクトリに移動してください。運用の詳細については，「12.3.1　監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）」を参照してください。

6. サーバディレクトリなどのバックアップを取得する

ディスク障害などのトラブルが発生し，サーバディレクトリまたはクライアントディレクトリの回復が必要になった場合に備えて，次に示すファイルもあわせてバックアップを取得してください。これらのファイルは，データベースの回復には直接は使用しません。

HADB サーバの場合

- サーバディレクトリ下（$ADBDIR/conf）に格納されているすべての定義ファイル
- サーバディレクトリ下（$ADBDIR）にユーザがファイルを作成している場合，作成したすべてのファイル

HADB クライアントの場合

- クライアントディレクトリ下（Windows 版の場合は%ADBCLTDIR%¥conf，Linux 版の場合は$ADBCLTDIR/conf）に格納されている定義ファイル
- クライアントディレクトリ下（Windows 版の場合は%ADBCLTDIR%，Linux 版の場合は$ADBCLTDIR）にユーザがファイルを作成している場合，作成したすべてのファイル

7. データベースの更新抑止を解除する

バックアップの取得が完了したら，データベースの更新抑止を解除します。

- **手順 2.で HADB サーバを正常終了した場合**

  adbstart コマンドを実行して，HADB サーバを開始してください。

- **手順 2.で HADB サーバの稼働モードを静止モードに変更した場合**

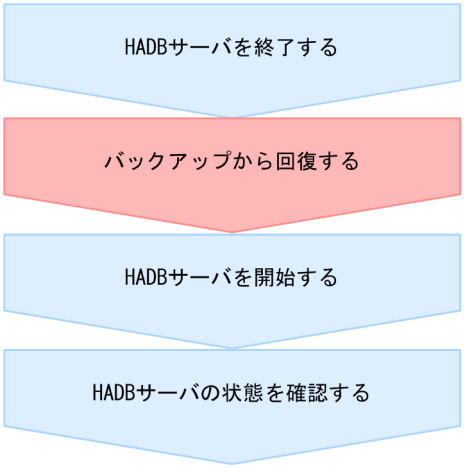
  adbchgsrvmode --normal コマンドを実行して，稼働モードを通常モードに変更してください。

これで，バックアップの取得は完了です。

10.3.2　バックアップからの回復方法

ここでは，取得したバックアップからデータベースを回復する手順について説明します。

図 10-2　バックアップから回復する手順



1. HADB サーバを終了する

   バックアップから回復する場合，HADB サーバを終了する必要があります。次に示すコマンドを実行して，HADB サーバを終了してください。

```
adbstop
```

> **重要**
>
> - バックアップから回復する前に，必ず HADB サーバを終了してください。HADB サーバの稼働中にバックアップから回復した場合，データベースの整合性が取れなくなり，HADB サーバはその後の動作を保証できません。
> - HADB サーバの終了モードが異常終了および強制終了でも，HADB サーバが終了していれば，バックアップから回復できます。

2. バックアップから回復する

   取得したバックアップから，データベースを回復します。「10.3.1　バックアップの取得方法」の「(2) バックアップを取得する手順」の『手順 5.　データベースのバックアップを取得する』で取得したすべてのファイルを，対象のディレクトリ下に格納し直してください。

格納し直すファイルにシンボリックリンクファイルが含まれている場合は，リンク先のファイルおよびブロックスペシャルファイルも回復してください。また，格納し直すファイルはスパースファイルにしないでください。

重要

- バックアップから回復する場合，バックアップを取得したすべてのファイルを必ず格納し直してください。一部のファイルだけを格納し直した場合，データベースの整合性が取れなくなり，その後の動作を保証できません。

  また，バックアップから回復する際，バックアップを取得したファイル以外のファイルを格納しないでください。対象のディレクトリ下のファイルを，バックアップを取得したファイル以外のファイルで上書きした場合，データベースの整合性が取れなくなり，その後の動作を保証できません。

- 作業表用 DB エリアにブロックスペシャルファイルを指定している場合は，DB エリアを回復したあとに，サーバ定義adb_blk_path_wrk オペランドを指定してから HADB サーバを開始してください。サーバ定義adb_blk_path_wrk オペランドについては，「7.2.1 システム構成に関するオペランド（set 形式）」の adb_blk_path_wrk オペランドを参照してください。

3. HADB サーバを開始する

   取得したバックアップからデータベースを回復したら，HADB サーバを開始します。次に示すコマンドを実行して，HADB サーバを開始してください。

```
adbstart
```

4. HADB サーバの状態を確認する

   HADB サーバを開始したら，HADB サーバの稼働モードが通常モードになっているかどうかを確認します。次に示すコマンドを実行して，HADB サーバの状態を確認してください。

```
adbls -d srv
```

   実行結果のSTATUS の項目を確認してください。STATUS がACTIVE になっていれば，稼働モードが通常モードになっています。STATUS がQUIESCE の場合は，稼働モードが静止モードになっています。adbchgsrvmode --normal コマンドを実行して，稼働モードを通常モードにしてください。

なお，ディスク障害などのトラブルが発生し，サーバディレクトリまたはクライアントディレクトリを回復する場合は，「15.4.2　サーバディレクトリの回復」または「15.4.3　クライアントディレクトリの回復」を参照してください。

10.3.3　バックアップの運用例（OS コマンドを使用）

ここでは，OS のコマンドを使用したバックアップの運用例について説明します。

(1) ディレクトリの構成例

バックアップを取得するディレクトリを次に示します。

- DB ディレクトリ（/HADB/db）
- アーカイブディレクトリ（/HADB/archive）
- 同義語辞書ファイルの格納ディレクトリ（/HADB/syndict）

上記のディレクトリの構成例を次の表に示します。

表 10-7　バックアップを取得するディレクトリの構成例

項番	バックアップの取得対象のディレクトリ名およびファイル名	説明	ファイルの種類	バックアップ取得の有無
1	/HADB/db/ADBDIC	ディクショナリ用 DB エリアファイル	レギュラーファイル	○
2	/HADB/db/ADBMST	マスタディレクトリ用 DB エリアファイル	レギュラーファイル	○
3	/HADB/db/ADBSTBL	システム表用 DB エリアファイル	レギュラーファイル	○
4	/HADB/db/ADBSYS	システムディレクトリ	レギュラーファイル群	○
5	/HADB/db/ADBWORK	作業用ディレクトリ	レギュラーファイル	×
6	/HADB/db/ADBWRK	作業表用 DB エリアファイル	レギュラーファイル	×
7	/HADB/db/ADBUTBL01※1	データ用 DB エリアファイル	ブロックスペシャルファイル	○
8	/HADB/db/ADBUIDX01※1	データ用 DB エリアファイル	ブロックスペシャルファイル	○
9	/HADB/db/SPOOL	障害情報（コアファイル）の出力先ディレクトリ	レギュラーファイル群	×
10	/HADB/archive	アーカイブディレクトリ※2	レギュラーファイル	○
11	/HADB/syndict	同義語辞書ファイルの格納ディレクトリ※3	レギュラーファイル	○
12	/HADB/audit	監査証跡の出力先ディレクトリ	レギュラーファイル	×

（凡例）

○：バックアップを取得します。

×：バックアップを取得しません。

注※1

8 個のブロックスペシャルファイルで構成されていることとします。

注※2

アーカイブマルチチャンク表を定義している場合に，アーカイブディレクトリのバックアップを取得する必要があります。

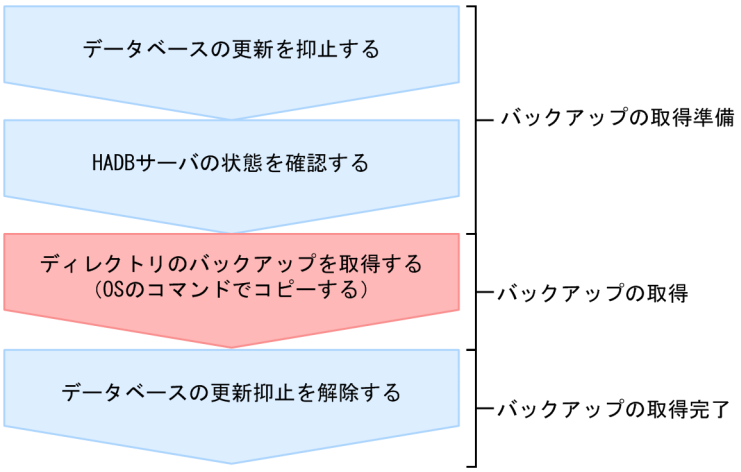
注※3

同義語検索を行っている場合に，同義語辞書ファイルの格納ディレクトリのバックアップを取得する必要があります。

(2)　OS のコマンドでバックアップを取得する

OS のコマンドを使用して，データベースのバックアップを取得する例を次に示します。

図 10-3　OS のコマンドでバックアップを取得する例



この例では，次に示す条件でバックアップを取得することとします。

条件

- データベースの初期設定後，一度以上 HADB サーバを開始している
- 更新不可状態の実表がないことを確認している
- データベースの更新を抑止するために，HADB サーバを正常終了する
- バックアップの種類はフルバックアップとする
- DB ディレクトリ，アーカイブディレクトリ，および同義語辞書ファイルの格納ディレクトリのバックアップを取得する

手順

1. データベースの更新を抑止する

   データベースのバックアップを取得する前に，データベースの更新を抑止します。ここでは，HADB サーバを正常終了します。

```
adbstop
```

2. HADB サーバの状態を確認する

   手順 1.で，HADB サーバが正常終了していることを確認します。

```
adbls -d srv
```

   実行結果のSTATUS の項目が，STOP になっていることを確認します。

3. ディレクトリのバックアップを取得する（OS のコマンドでコピーする）

   OS のコマンドを実行して，「(1)　ディレクトリの構成例」で示したバックアップの取得が必要なファイルをすべてコピーします。

   - DB ディレクトリ下のファイルのコピー先

     /HADB_bkup/db ディレクトリ

   - アーカイブディレクトリ下のファイルのコピー先

     /HADB_bkup/archive ディレクトリ

   - 同義語辞書ファイルの格納ディレクトリ下のファイルのコピー先

     /HADB_bkup/syndict ディレクトリ

   実行する OS のコマンドを次に示します。レギュラーファイルはcp コマンドで，ブロックスペシャルファイルはdd コマンドで取得することを推奨します。

```
cp -r /HADB/db/ADBSYS  /HADB_bkup/db/ADBSYS

cp -H /HADB/db/ADBMST  /HADB_bkup/db/ADBMST

cp -H /HADB/db/ADBDIC  /HADB_bkup/db/ADBDIC

cp -H /HADB/db/ADBSTBL /HADB_bkup/db/ADBSTBL

dd if=/HADB/db/ADBUTBL01
   of=/HADB_bkup/db/ADBUTBL01 bs=524288

dd if=/HADB/db/ADBUTBL01.00001
   of=/HADB_bkup/db/ADBUTBL01.00001 bs=524288

dd if=/HADB/db/ADBUTBL01.00002
   of=/HADB_bkup/db/ADBUTBL01.00002 bs=524288

dd if=/HADB/db/ADBUTBL01.00003
   of=/HADB_bkup/db/ADBUTBL01.00003 bs=524288

dd if=/HADB/db/ADBUTBL01.00004
   of=/HADB_bkup/db/ADBUTBL01.00004 bs=524288

dd if=/HADB/db/ADBUTBL01.00005
   of=/HADB_bkup/db/ADBUTBL01.00005 bs=524288

dd if=/HADB/db/ADBUTBL01.00006
   of=/HADB_bkup/db/ADBUTBL01.00006 bs=524288

dd if=/HADB/db/ADBUTBL01.00007
```

```
   of=/HADB_bkup/db/ADBUTBL01.00007 bs=524288

dd if=/HADB/db/ADBUIDX01
   of=/HADB_bkup/db/ADBUIDX01 bs=524288

dd if=/HADB/db/ADBUIDX01.00001
   of=/HADB_bkup/db/ADBUIDX01.00001 bs=524288

dd if=/HADB/db/ADBUIDX01.00002
   of=/HADB_bkup/db/ADBUIDX01.00002 bs=524288

dd if=/HADB/db/ADBUIDX01.00003
   of=/HADB_bkup/db/ADBUIDX01.00003 bs=524288

dd if=/HADB/db/ADBUIDX01.00004
   of=/HADB_bkup/db/ADBUIDX01.00004 bs=524288

dd if=/HADB/db/ADBUIDX01.00005
   of=/HADB_bkup/db/ADBUIDX01.00005 bs=524288

dd if=/HADB/db/ADBUIDX01.00006
   of=/HADB_bkup/db/ADBUIDX01.00006 bs=524288

dd if=/HADB/db/ADBUIDX01.00007
   of=/HADB_bkup/db/ADBUIDX01.00007 bs=524288

cp -r /HADB/archive  /HADB_bkup/archive

cp -r /HADB/syndict  /HADB_bkup/syndict
```

重要

ブロックスペシャルファイルのデータ量については，ボリューム全体をコピーするため，実際の使用量よりも増えます。

4. データベースの更新抑止を解除する

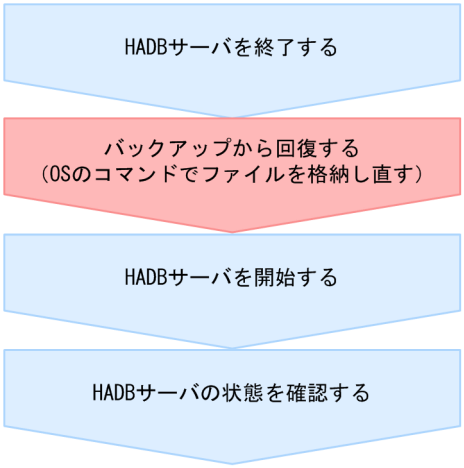
OS のコマンドを使用してバックアップを取得したら，データベースの更新抑止を解除できます。ここでは，次に示すコマンドを実行して，HADB サーバを開始してください。

```
adbstart
```

(3) OS のコマンドでバックアップから回復する

OS のコマンドを使用して，取得したバックアップから回復する例を次に示します。

図 10-4　OS のコマンドでバックアップから回復する例



手順

1. HADB サーバを終了する

   バックアップから回復する場合，HADB サーバを終了する必要があります。次に示すコマンドを実行して，HADB サーバを終了してください。

   ```
   adbstop
   ```

2. バックアップから回復する（OS のコマンドでファイルを格納し直す）

   取得したバックアップから，データベースを回復します。

   まず，バックアップから回復する前のシステムログを削除するため，DB ディレクトリ下にあるシステムディレクトリ（/HADB/db/ADBSYS）を削除してください。次に示す OS のコマンドを実行してください。

   ```
   rm -r /HADB/db/ADBSYS
   ```

   次に，バックアップから回復する前のアーカイブファイルを削除するため，アーカイブディレクトリ下にあるアーカイブファイルを削除してください。次に示す OS のコマンドを実行してください。

   ```
   rm -r /HADB/archive/*
   ```

   次に，バックアップから回復する前の同義語辞書ファイルを削除してください。次に示す OS のコマンドを実行してください。

   ```
   rm -r /HADB/syndict/*
   ```

   そのあとで，「(2)　OS のコマンドでバックアップを取得する」の『手順 3.　ディレクトリのバックアップを取得する（OS のコマンドでコピーする）』で取得したすべてのファイルを格納し直してください。

   - バックアップを取得した DB ディレクトリ下のファイルの格納先

     /HADB/db ディレクトリ

   - バックアップを取得したアーカイブディレクトリ下のファイルの格納先

     /HADB/archive ディレクトリ

- バックアップを取得した同義語辞書ファイルの格納ディレクトリ下のファイルの格納先
  /HADB/syndict ディレクトリ

実行する OS のコマンドを次に示します。

```
cp -r /HADB_bkup/db/ADBSYS  /HADB/db/ADBSYS

cp -H /HADB_bkup/db/ADBMST  /HADB/db/ADBMST

cp -H /HADB_bkup/db/ADBDIC  /HADB/db/ADBDIC

cp -H /HADB_bkup/db/ADBSTBL /HADB/db/ADBSTBL

dd if=/HADB_bkup/db/ADBUTBL01
   of=/HADB/db/ADBUTBL01 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00001
   of=/HADB/db/ADBUTBL01.00001 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00002
   of=/HADB/db/ADBUTBL01.00002 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00003
   of=/HADB/db/ADBUTBL01.00003 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00004
   of=/HADB/db/ADBUTBL01.00004 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00005
   of=/HADB/db/ADBUTBL01.00005 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00006
   of=/HADB/db/ADBUTBL01.00006 bs=524288

dd if=/HADB_bkup/db/ADBUTBL01.00007
   of=/HADB/db/ADBUTBL01.00007 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01
   of=/HADB/db/ADBUIDX01 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01.00001
   of=/HADB/db/ADBUIDX01.00001 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01.00002
   of=/HADB/db/ADBUIDX01.00002 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01.00003
   of=/HADB/db/ADBUIDX01.00003 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01.00004
   of=/HADB/db/ADBUIDX01.00004 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01.00005
   of=/HADB/db/ADBUIDX01.00005 bs=524288

dd if=/HADB_bkup/db/ADBUIDX01.00006
   of=/HADB/db/ADBUIDX01.00006 bs=524288
```

```
dd if=/HADB_bkup/db/ADBUIDX01.00007
   of=/HADB/db/ADBUIDX01.00007 bs=524288

cp -r /HADB_bkup/archive/*  /HADB/archive

cp -r /HADB_bkup/syndict/*  /HADB/syndict
```

重要

DB ディレクトリ下のファイルをバックアップから回復する場合，DB ディレクトリ（/HADB/db）に上書きコピーしてください。空のディレクトリに対して，バックアップ対象のファイルをコピーして格納するだけでは，データベースの整合性が取れなくなり，HADB サーバはその後の動作を保証できません。

3. HADB サーバを開始する

取得したバックアップからデータベースを回復したら，HADB サーバを開始します。次に示すコマンドを実行して，HADB サーバを開始してください。

```
adbstart
```

4. HADB サーバの状態を確認する

HADB サーバを開始したら，HADB サーバの稼働モードが通常モードになっているかどうかを確認します。次に示すコマンドを実行して，HADB サーバの状態を確認してください。

```
adbls -d srv
```

実行結果のSTATUS の項目を確認してください。STATUS がACTIVE になっていれば，稼働モードが通常モードになっています。

10.3.4　バックアップの運用例（ShadowImage を使用）

ここでは，日立ディスクアレイシステムのボリュームレプリケーション機能である ShadowImage を使用したバックアップの運用例について説明します。

ShadowImage を使用すると，バックアップの取得およびバックアップからの回復に掛かる時間を短縮できます。ShadowImage については，マニュアル『ShadowImage　ユーザーズガイド』を参照してください。

(1)　ShadowImage を使用するときの注意事項

隠蔽モード（-m noread）を指定してペアを生成しないでください。

(2)　ディレクトリの構成例

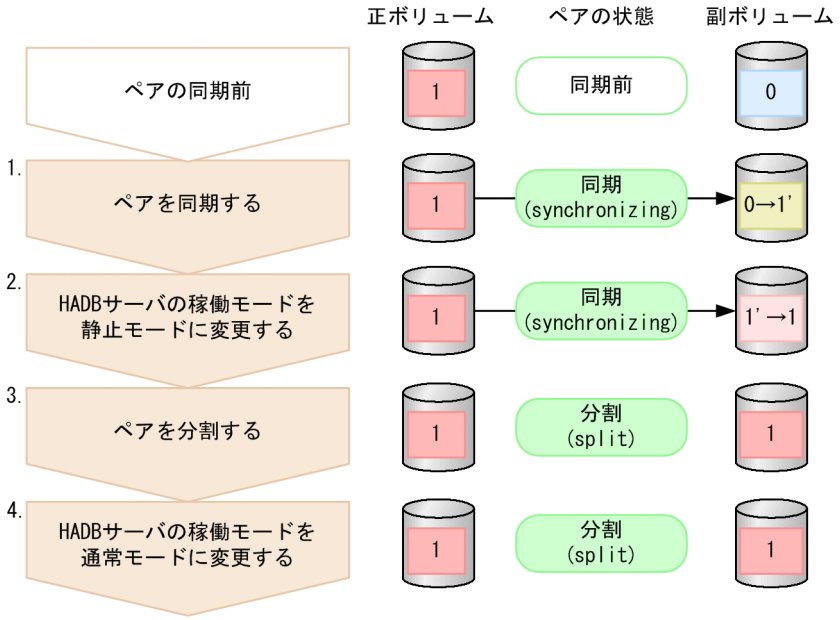
次に示すディレクトリをすべてバックアップの取得対象として，ShadowImage の正ボリュームに格納することとします。

- ブロックスペシャルファイルを含む DB ディレクトリ
- アーカイブディレクトリ
- 同義語辞書ファイルの格納ディレクトリ

(3)　バックアップの取得

ここでは，ShadowImage を使用してバックアップを取得する運用例について説明します。

図 10-5　ShadowImage を使用したバックアップの取得例



条件

- データベースの初期設定後，一度以上 HADB サーバを開始している
- 更新不可状態の実表がないことを確認している
- データベースの更新を抑止するために，HADB サーバの稼働モードを静止モードに変更する
- バックアップの種類はフルバックアップとする
- DB ディレクトリ，アーカイブディレクトリ，および同義語辞書ファイルの格納ディレクトリのバックアップを取得する

手順

1. ペアを同期する

ペアの同期を開始してください。ペアの同期に時間が掛かることがあるため，HADB サーバの稼働モードを静止モードに変更する前に，ペアボリュームの同期を開始しておくことを推奨します。

2. HADB サーバの稼働モードを静止モードに変更する

   `adbchgsrvmode --quiescence` コマンドを実行して，HADB サーバの稼働モードを静止モードに変更してください。静止モードに変更したら，ペアの同期が完了するまで待ってください。

3. ペアを分割する

   ペアの同期が完了したら，バックアップの取得が完了したため，ペアを分割してください。これで，ペアの同期を開始した時点のバックアップが，副ボリュームに保持されます。

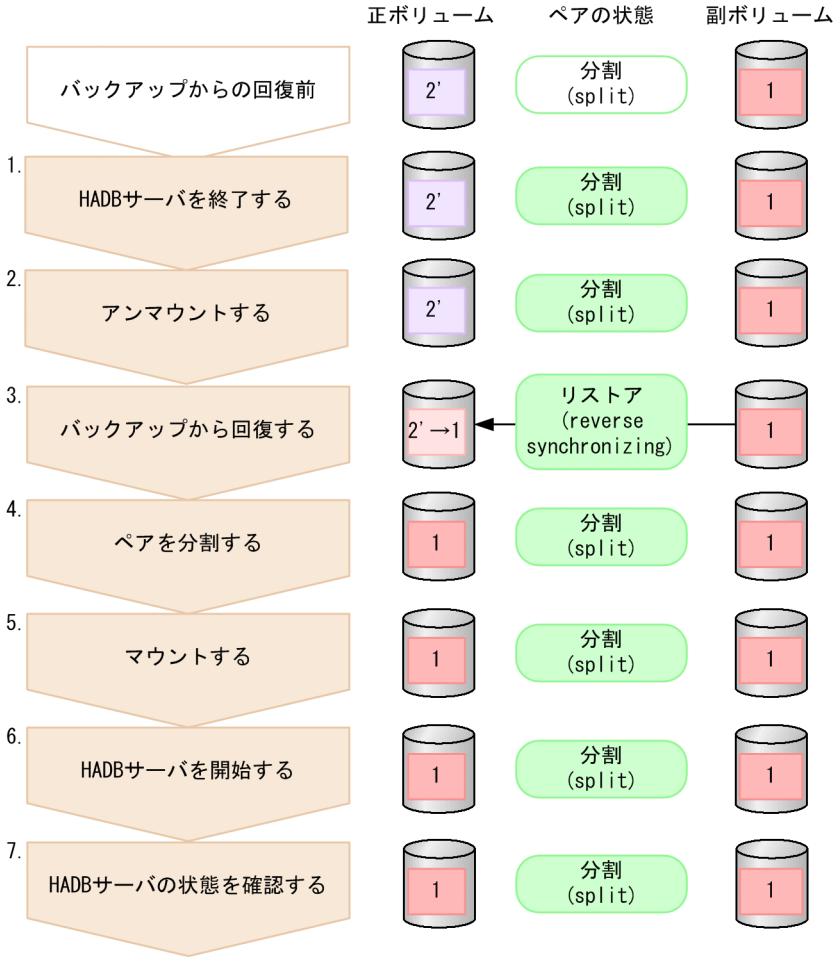
4. HADB サーバの稼働モードを通常モードに変更する

   `adbchgsrvmode --normal` コマンドを実行して，HADB サーバの稼働モードを通常モードに変更してください。これで，ShadowImage を使用したバックアップの取得は完了です。

(4) バックアップからの回復

ここでは，「(3) バックアップの取得」で取得したバックアップからデータベースを回復する運用例について説明します。

図 10-6 ShadowImage を使用したバックアップからの回復例



手順

1. HADB サーバを終了する

   バックアップから回復する前に，HADB サーバを終了します。次に示す HADB のコマンドを実行して，HADB サーバを終了してください。

   ```
   adbstop
   ```

2. アンマウントする

   正ボリュームに格納されたファイルシステムをアンマウントしてください。

3. バックアップから回復する

   ペアを逆同期して，バックアップから回復します。副ボリュームの内容を正ボリュームに同期してください。

   > **重要**
   >
   > 正ボリュームの内容を副ボリュームに同期しないように注意してください。

4. ペアを分割する

   ペアの逆同期が完了したら，バックアップからの回復が完了したため，ペアを分割してください。

5. マウントする

   手順 2.でアンマウントしたファイルシステムをマウントしてください。

6. HADB サーバを開始する

   次に示す HADB のコマンドを実行して，HADB サーバを開始してください。

   ```
   adbstart
   ```

7. HADB サーバの状態を確認する

   HADB サーバを開始したら，HADB サーバの稼働モードが通常モードになっているかどうかを確認します。次に示すコマンドを実行して，HADB サーバの状態を確認してください。

   ```
   adbls -d srv
   ```

   実行結果のSTATUS の項目を確認してください。STATUS がACTIVE になっていれば，稼働モードが通常モードになっています。

10.4　メッセージの確認

HADB の稼働状況は，メッセージで確認します。HADB が出力したメッセージは，履歴情報として保存されます。メッセージの履歴情報をメッセージログといいます。

10.4.1　メッセージの出力先

HADB サーバのメッセージは，標準出力，標準エラー出力，サーバメッセージログファイル，および syslog に出力されます。メッセージの種別によって出力先が異なります。それぞれの出力先と出力されるメッセージの種別について説明します。

(1)　標準出力

コンソールの標準出力にメッセージが出力されます。出力されるメッセージは，インフォメーションメッセージおよび応答メッセージになります。

(2)　標準エラー出力

コンソールの標準エラー出力にメッセージが出力されます。出力されるメッセージは，エラーメッセージおよび警告メッセージになります。

(3)　サーバメッセージログファイル

HADB サーバが出力するメッセージは，サーバディレクトリ下に作成されるファイルにメッセージログとして格納されます。このファイルをサーバメッセージログファイルといいます。

出力されるメッセージは，HADB サーバのすべてのメッセージになります。

サーバメッセージログファイルは，HADB サーバに 4 つ作成されます。作成されるサーバメッセージログファイルのファイルサイズの最大容量は，環境変数ADBMSGLOGSIZE で指定できます。

なお，サーバメッセージログファイルには，HADB クライアントが出力するメッセージは格納されません。

メモ

■クライアントメッセージログファイル

HADB クライアントが出力するメッセージは，クライアントディレクトリ下に作成されるファイルにメッセージログとして格納されます。このファイルをクライアントメッセージログファイルといいます。

クライアントメッセージログファイルは，HADB クライアントに 4 つ作成されます。作成されるクライアントメッセージログファイルのファイルサイズの最大容量は，環境変数 ADBMSGLOGSIZE で指定できます。

(4) syslog

syslog にメッセージが出力されます。syslog に出力されるメッセージの種別は，HADB サーバの環境変数ADBSYSLOGLV で指定できます。

10.4.2 メッセージログの参照方法（メッセージログの出力先）

HADB が出力したメッセージは，メッセージログとしてメッセージログファイルおよび syslog に出力されます。メッセージログは，HADB が出力するメッセージの履歴情報として，次に示す場合に利用します。

- HADB が出力したメッセージを確認する場合
- HADB に障害が発生したときに原因を調べる場合

HADB サーバおよび HADB クライアントのメッセージログファイルに出力されたメッセージログは，テキストエディタを使用して参照できます。syslog に出力されたメッセージログは，OS で決められた手順に従って参照してください。

HADB サーバ，および HADB クライアントに作成されるメッセージログファイルの格納先を次に示します。ファイル名の *XX* は通番であり，01～04 のどれかになります。

- HADB サーバの場合（サーバメッセージログファイル）
  $ADBDIR/spool/adbmessage*XX*.log
- HADB クライアント（Linux 版）の場合（クライアントメッセージログファイル）
  $ADBCLTDIR/spool/adbmessageclt*XX*.log
- HADB クライアント（Windows 版）の場合（クライアントメッセージログファイル）
  %ADBCLTDIR%¥spool¥adbmessageclt*XX*.log

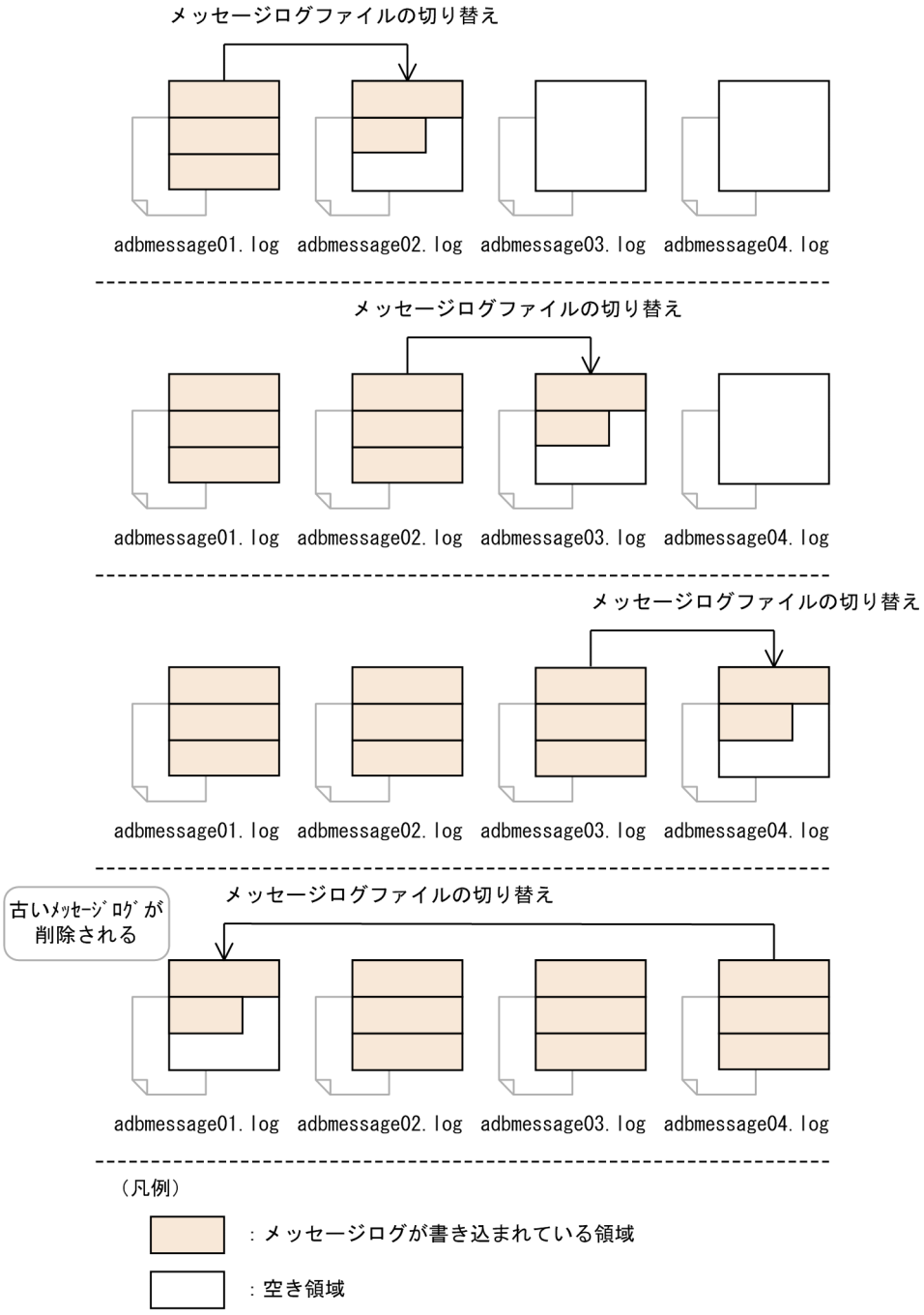
10.4.3 メッセージログファイルの運用

HADB サーバおよび HADB クライアントは，それぞれメッセージログファイルを 4 つ作成して使用します。

1 つのメッセージログファイルの最大容量は，環境変数ADBMSGLOGSIZE に指定した値になります。1 つのメッセージログファイルが最大容量に達すると，HADB はKFAA80099-I メッセージを出力してから，そのメッセージログファイルへのメッセージログの出力をやめて，次のメッセージログファイルにメッセージログを出力します。このとき，切り替え先のメッセージログファイルに格納されている古いメッセージログは削除されます。切り替え前のメッセージログファイルに格納されているメッセージログを保存しておきたい場合は，メッセージログファイルのバックアップを取得してください。

HADB サーバに作成される 4 つのサーバメッセージログファイルを例に，メッセージログファイルの切り替えを次の図に示します。

図 10-7　メッセージログファイルの切り替え



[説明]

メッセージログが出力されてadbmessage01.log が最大容量に達すると，メッセージログの出力先は adbmessage02.log に切り替わります。

adbmessage01.log のときと同じようにadbmessage02.log が最大容量に達すると，メッセージログの出力先はadbmessage03.log に切り替わります。adbmessage03.log が最大容量に達したときも同様に，adbmessage04.log に切り替わります。

adbmessage04.log が最大容量に達すると，メッセージログの出力先はadbmessage01.log に切り替わります。その際，adbmessage01.log に格納されている古いメッセージログが削除されてから，メッセージログが出力されます。

なお，HADB サーバを開始したときには，サーバメッセージログファイルの切り替えは行われません。そのため，前回 HADB サーバを終了したときに出力先となっていたサーバメッセージログファイルに対して出力されます。

メッセージログが出力されているメッセージログファイルを確認したい場合は，OS の ls コマンドなどでファイルの最終更新日時を確認してください。最終更新日時が新しいファイルが，現在使用しているメッセージログファイルです。

ただし，メッセージログファイルをテキストエディタなどで更新してしまい，最終更新日時が変更された場合は，各メッセージログファイルに出力されたメッセージのうち，メッセージの付加情報に示されている出力日時が最も新しいメッセージを格納しているメッセージログファイルに，出力先が変更されます。

メモ

HADB クライアントのクライアントメッセージログファイルも，同じ運用になります。

10.4.4 メッセージログファイルの縮退状態

メッセージログファイルを格納しているディスクの空き容量が少なくなったことを HADB サーバが検知した場合，ディスクの空き容量に応じて出力可能な範囲でメッセージログを出力します。この状態をメッセージログファイルの縮退状態といいます。

メッセージログファイルが縮退状態になった場合，メッセージの出力先が次のメッセージログファイルに切り替わります。このとき，切り替え先のメッセージログファイル中のメッセージが削除されます。メッセージログファイルの切り替えが完了したあとに，警告メッセージKFAA40025-W が切り替え先のメッセージログファイルと syslog に出力されます。ただし，Windows 版の HADB クライアントのメッセージログファイルが縮退状態になった場合は，HADB クライアントのメッセージログファイルにだけKFAA40025-W メッセージが出力されます。

なお，HADB クライアントで SQL 文を実行した延長で，HADB サーバのサーバメッセージログファイルが縮退状態になった場合は，HADB クライアントのメッセージログファイルにもKFAA40025-W メッセージが出力されます。

メモ

- 強制的にメッセージログファイルを切り替える処理は，縮退状態になったときに一度だけ行われます。KFAA40025-W メッセージも，縮退状態になったときに 1 回だけ出力されます。
  メッセージログファイルが縮退状態の間は，切り替え先のメッセージログファイルにメッセージを出力できる場合は出力しますが，出力できない場合はメッセージを出力しません。
  切り替え先のメッセージログファイルの容量が環境変数ADBMSGLOGSIZE で指定したサイズになった場合，それ以上メッセージは出力されません。

- メッセージログファイルが縮退状態となりメッセージログファイルが切り替わっても，切り替え前のメッセージログファイルにKFAA80099-I メッセージ（メッセージログファイルの出力先が変わったときに出力されるメッセージ）は出力されません。

メッセージログファイルを格納しているディスクの空き容量が，環境変数ADBMSGLOGSIZE の指定値×2 以上になったときに SQL 文またはコマンドが実行されると，メッセージログファイルは自動的に正常状態に戻ります。その後，メッセージログファイルを格納しているディスクの空き容量が少なくなったことを HADB サーバが再度検知した場合は，メッセージログファイルは再度縮退状態になります。

メモ

環境変数ADBMSGLOGSIZE には，1 つのメッセージログファイルの最大容量をメガバイト単位で指定します。ADBMSGLOGSIZE の指定を省略した場合は，1 つのメッセージログファイルの最大容量は 16 メガバイトになります。

HADB サーバのメッセージログファイルが縮退状態かどうかをコマンドで確認する方法

adbls -d srv コマンドを実行すると，HADB サーバのメッセージログファイルが縮退状態かどうかを確認できます。実行結果のMSGLOG_STATUS 欄を確認します。

（例）

```
adbls -d srv

SVID   STATUS   START_TIME            MSGLOG_STATUS
5360   ACTIVE   2017/02/02 09:31:04   FALLBACK
```

MSGLOG_STATUS 欄にFALLBACK が表示されている場合は，HADB サーバのメッセージログファイルは縮退状態になっています。正常な状態の場合は，NORMAL が表示されます。

メッセージログファイルの縮退状態を解除する方法については，「15.16.1　メッセージログファイルの縮退状態を解除する方法」を参照してください。

10.4.5　syslog へのメッセージの出力抑止

HADB はシステムの稼働状況を知らせるメッセージとして，障害情報，トランザクション情報，システムファイルの情報などを syslog に出力します。これらのメッセージは，HADB の稼働状況を知るための重要な情報です。

しかし，稼働環境によっては大量のメッセージが出力されることがあります。HADB のメッセージが出力されることによって syslog の容量が増加し続けると，ディスクの空き容量が不足する原因となります。この場合，syslog に出力されるメッセージのうち，不要なメッセージの出力を抑止することで，ディスクの空き容量不足を防止できます。

syslog に出力されるメッセージを抑止する場合は，環境変数ADBSYSLOGLV でメッセージの出力レベルを設定してください。環境変数ADBSYSLOGLV については，「8.4　環境変数の設定」を参照してください。

メッセージの出力レベルと syslog に出力されるメッセージの内容を次の表に示します。

表 10-8　メッセージの出力レベルと syslog に出力されるメッセージの内容

出力レベル	出力されるメッセージの内容
0	syslog にメッセージは出力されません。
1	次に示すメッセージだけが syslog に出力されます。 • メッセージログファイルに異常が発生していることを示す警告メッセージ • プロセスダウンなどの致命的な障害を示すエラーメッセージ • ユーザからの応答が必要な応答メッセージ
2	出力レベル1 のときに出力されるメッセージ，および次に示すメッセージが syslog に出力されます。 • HADB の障害を示すエラーメッセージ • HADB の起動および終了などを示すインフォメーションメッセージ
3	出力レベル1～2 のときに出力されるメッセージ，および次に示すメッセージが syslog に出力されます。 • 業務レベルでの障害などを示すエラーメッセージ
4	出力レベル1～3 のときに出力されるメッセージ，および次に示すメッセージが syslog に出力されます。 • 処理続行に支障はないが，ユーザに状況（障害）を通知する警告メッセージ
5	出力レベル1～4 のときに出力されるメッセージ，および次に示すメッセージが syslog に出力されます。 • スレッド起動，コネクション確立などの情報を示すインフォメーションメッセージ
6	syslog に出力されるメッセージはすべて出力されます。

各メッセージの出力レベルについては，マニュアル『HADB　メッセージ』の『メッセージの出力先一覧』を参照してください。

10.4.6　syslog の文字コード変換と信頼性向上（拡張 SYSLOG 機能の適用）

拡張 SYSLOG 機能を適用すると，次に示す運用を実現できます。

- syslog の文字コード変換（Shift-JIS から Unicode への変換）
- syslog の信頼性向上（メッセージの再出力）

> **メモ**
>
> 拡張 SYSLOG 機能は，日立サポート 360 のログ環境強化サポートオプションで提供される機能です。

拡張 SYSLOG 機能を適用すると実現できる運用について，次で説明します。

(1) syslog の文字コード変換（Shift-JIS から Unicode への変換）

HADB サーバで使用する文字コードが Shift-JIS の場合に，syslog に出力されるメッセージの文字コードを Shift-JIS から Unicode（UTF-8）に変換します。

syslog に出力されるメッセージの文字コードが Unicode（UTF-8）に統一されることで，次のメリットがあります。

- syslog を参照するときに，メッセージの文字化けを防ぐことができる
- メッセージの監視や管理が容易になる

なお，syslog の文字コード変換は，次に示す環境変数に Shift-JIS を適用したときだけ有効です。

■対象の環境変数

- LANG
- ADBLANG

環境変数については，「8.4　環境変数の設定」を参照してください。

(2) syslog の信頼性向上（メッセージの再出力）

syslog へのメッセージの出力に失敗した場合に，再度メッセージの出力を行います。そのため，syslog へのメッセージの出力に失敗しても，メッセージが出力されないまま消失するのを防げます。

(3) 前提ソフトウェアおよび注意事項（拡張 SYSLOG 機能の適用）

拡張 SYSLOG 機能を適用したい場合，HADB サーバを開始する前に，次に示す前提ソフトウェアをインストールする必要があります。

前提ソフトウェア

- HA Logger Kit for Linux（拡張 SYSLOG 機能）：04-00-/A 以降
- 日立コード変換※1：03-03-/A 以降※2

注※1

syslog の文字コード変換（Shift-JIS から Unicode への変換）を実現したい場合にインストールする必要があります。

注※2

RHEL 8 を使用している場合は，04-00 以降の日立コード変換をインストールする必要があります。

> **重要**
>
> マルチノード機能を使用する場合，またはコールドスタンバイ構成の場合は，すべてのサーバマシンに前提ソフトウェアをインストールしてください。

前提ソフトウェアをインストールしている場合は，HADB サーバの開始時に，KFAA50050-I メッセージが，syslog およびサーバメッセージログファイルに出力されます。KFAA50050-I メッセージは，拡張 SYSLOG 機能が適用されたことを示すメッセージです。

なお，HADB サーバが起動している状態で，前提ソフトウェアをインストールしても，拡張 SYSLOG 機能は適用されません。HADB サーバを起動している場合は，次の手順で前提ソフトウェアをインストールしてください。

手順

1. HADB サーバを正常終了する
2. 前提ソフトウェアをインストールする
3. HADB サーバを開始する

また，HADB サーバがインストールされているサーバマシンの環境変数ADBSYSLOGLV に0 を指定している場合も，拡張 SYSLOG 機能は適用されません。syslog にメッセージが出力されないためです。

HA Logger Kit for Linux（拡張 SYSLOG 機能）のインストール方法については，『取扱説明書 HA Logger Kit for Linux 高信頼ログ基盤 拡張 SYSLOG 機能』を参照してください。日立コード変換のインストール方法については，マニュアル『日立コード変換ユーザーズガイド』を参照してください。

10.5　リソースの使用状況の監視（監視対象メッセージの一覧）

ここでは，HADB サーバが使用するリソースの状態を監視する場合に，監視する必要があるメッセージについて説明します。

HADB サーバは，各種リソースを使用して稼働します。そのため，次に示す事象が発生すると，HADB サーバの稼働状況に影響が出るおそれがあります。

- サーバマシンや OS のリソース（ディスク，CPU，メモリなど）が不足したとき
- HADB サーバが管理するリソース（接続数，チャンク数など）が不足したとき
- ディスク障害やネットワーク障害が発生したとき
- 排他制御による処理の待ち状態が発生したとき

このような事象が発生した場合には，発生した事象に対して速やかに対処する必要があります。そのため，HADB 管理者は，HADB サーバが出力するメッセージを確認して，リソースの使用状況を監視してください。

HADB サーバが使用するリソースと監視対象メッセージの一覧を，次の表に示します。各メッセージが出力されたときの対処方法については，マニュアル『HADB　メッセージ』を参照してください。

表 10-9　HADB サーバが使用するリソースと監視対象メッセージの一覧

項番	リソースの種類		監視対象のメッセージ	説明
1	排他資源		• KFAA31371-E • KFAA31663-E • KFAA40005-E • KFAA50290-E	左記のメッセージが出力された場合は，排他資源の確保待ちが発生しています。 対処方法については，各メッセージの対策欄を参照してください。 なお，排他資源の確保待ちの発生状況を確認する方法については，「10.8.3　排他資源の確保待ちの発生状況を確認する方法」を参照してください。 また，排他資源の確保待ちが繰り返し発生する場合は，排他資源を確保した AP やコマンドが長時間実行されている可能性があります。「10.8.1　AP およびコマンドの処理状況を確認する方法」も参照してください。
2	メモリ	HADB サーバが管理するメモリ	• KFAA40021-W • KFAA40024-I	KFAA40021-W メッセージは，HADB サーバのメモリ使用量が，サーバ定義のadb_sys_memory_limit_wrn_pnt オペランド（HADB サーバのメモリ使用量に関する警告メッセージの出力契機）に指定した比率以上になったときに出力される警告メッセージです。KFAA40021-W メッセージが出力された場合は，HADB サーバのメモリ使用状況を確認してください。確認方法については，「10.6.3　リアルスレッドごとのメモリの使用状況を確認したい場合」を参照してください。 KFAA40024-I メッセージは，HADB サーバのメモリ使用量が，サーバ定義のadb_sys_memory_limit_wrn_pnt オペランド（警告メッセージの出力済み状態のリセット契機）に指定した比

項番	リソースの種類		監視対象のメッセージ	説明
				率以下になったときに出力されるインフォメーションメッセージです。 [マルチノード機能] マルチノード機能を使用している場合，左記のメッセージは全ノードで出力される可能性があります。
3			• KFAA30930-E • KFAA40007-E • KFAA40203-E • KFAA40212-E • KFAA51011-W • KFAA96605-E	左記のメッセージが出力された場合は，HADB サーバが管理するメモリが不足しています。 SQL 文またはコマンドを実行した場合に，メモリ不足が発生すると，次に示すメッセージが出力されます。 **SQL 文を実行した場合** • KFAA30930-E • KFAA40007-E • KFAA51011-W **コマンドを実行した場合** • KFAA40007-E • KFAA40203-E • KFAA40212-E • KFAA96605-E 上記のメッセージが出力されたときの対処方法については，KFAA40007-E メッセージの対策欄を参照してください。 また，adbstat コマンドを定期的に実行して，HADB サーバが使用するメモリの使用状況を確認してください（メモリの使用量が上限値に近づいていないかを確認してください）。詳細については，「10.6.1　全メモリの使用状況を確認したい場合」を参照してください。
4		HADB サーバが使用するライブラリで使用するメモリ	• KFAA40207-E • KFAA40208-E • KFAA40293-E • KFAA91000-E • KFAA92000-E • KFAA96211-E • KFAA96457-E	左記のメッセージが出力された場合は，HADB サーバが使用するライブラリで使用するメモリが不足しています。 コマンドを実行した場合に，メモリが不足していると，左記のメッセージが出力されます。 対処方法については，各メッセージの対策欄を参照してください。
5	共有メモリ		KFAA40002-E	左記のメッセージが出力された場合は，共有メモリにアクセスしたときにエラーが発生しています。 対処方法については，メッセージの対策欄を参照してください。
6	CPU		なし	HADB サーバは，CPU 利用率に関するメッセージを出力しません。 HADB サーバの CPU 利用率を監視したい場合は，次に示す OS のコマンドを実行してください。 • sar コマンド • mpstat コマンド • top コマンド

項番	リソースの種類		監視対象のメッセージ	説明
7	プロセス間通信		• KFAA30722-E • KFAA30723-E • KFAA50300-E • KFAA50301-E • KFAA50303-E • KFAA50304-E • KFAA50305-E • KFAA50306-E • KFAA50307-E • KFAA50308-E • KFAA50309-E • KFAA50310-E • KFAA50311-E	左記のメッセージが出力された場合は，HADB サーバのプロセス間通信でエラーが発生しています。 対処方法については，各メッセージの対策欄を参照してください。
8	ディレクトリ	DB ディレクトリ	KFAA30959-E	左記のメッセージが出力された場合は，DB ディレクトリにアクセスしたときにエラーが発生しています。 対処方法については，メッセージの対策欄を参照してください。
9		アーカイブディレクトリ	KFAA31666-E	左記のメッセージが出力された場合は，アーカイブディレクトリにアクセスしたときにエラーが発生しています。 対処方法については，メッセージの対策欄を参照してください。
10			KFAA61212-W	左記のメッセージが出力された場合は，アーカイブディレクトリに対する削除処理でエラーが発生しています。 対処方法については，メッセージの対策欄を参照してください。
11	ファイル	メッセージログファイル	KFAA40025-W	左記のメッセージが出力された場合は，次に示すどちらかのメッセージログファイルが縮退状態になっています。 • サーバメッセージログファイル • クライアントメッセージログファイル 左記のメッセージが出力された場合は，メッセージログファイルの縮退状態を解除してください。縮退状態の解除方法については，「15.16.1　メッセージログファイルの縮退状態を解除する方法」を参照してください。 なお，左記のメッセージは，メッセージログファイルに出力できない場合でも syslog には出力されます。そのため，メッセージログファイルとともに syslog も監視してください。
12			KFAA50057-W	[マルチノード機能] 左記のメッセージが出力された場合は，どこかのノードでメッセージログファイルが縮退状態になっています。この場合，メッセージログファイルの縮退状態を解除してください。縮退状態の解除方法については，「15.16.1　メッセージログファイルの縮退状態を解除する方法」を参照してください。
13			KFAA80099-I	左記のメッセージが出力された場合は，次に示すどちらかのメッセージログファイルの切り替えが発生しています。 • サーバメッセージログファイル

項番	リソースの種類		監視対象のメッセージ	説明
				• クライアントメッセージログファイル 切り替え前のメッセージログファイル中のメッセージログを保存しておきたい場合は，メッセージログファイルのバックアップを取得してください。
14		DB エリアファイル	• KFAA30959-E • KFAA60014-W	左記のメッセージが出力された場合は，各ファイルに対する操作に失敗しています。 対処方法については，各メッセージの対策欄を参照してください。
15		アーカイブファイル		
16		システムファイル		
17		ADB_CSVREAD 関数の入力情報となる CSV ファイル		
18		コマンドの操作対象ファイル	• KFAA40204-E • KFAA40205-E	
19		HADB ダンプファイル	• KFAA40015-E • KFAA40016-E	
20		メッセージカタログファイル	• KFAA50018-E • KFAA50019-E	
21		共有メモリ ID 格納ファイル	KFAA50028-E	
22		共有メモリダンプファイル	KFAA50029-E	
23		セマフォ集合 ID 格納ファイル	KFAA50037-E	
24		コマンドステータスファイル	KFAA50243-E	
25		作業用一時ファイル	KFAA50245-E	
26		インデクスレコードファイル	KFAA50246-E	

項番	リソースの種類		監視対象のメッセージ	説明
27		SQL トレースファイル	KFAA51010-W	
28		同義語辞書ファイル	• KFAA30959-E • KFAA34008-E	
29			KFAA51537-W	[マルチノード機能] 左記のメッセージが出力された場合は，サーバ定義の adb_syndict_node_storage_path オペランドに指定したディレクトリに，最新の同義語辞書ファイルが格納されていません。この場合，同義語検索の性能が低下するおそれがあります。 対処方法については，メッセージの対策欄を参照してください。
30		アンロードファイル	KFAA61400-W	左記のメッセージが出力された場合は，adbreorgsystemdata コマンドを実行したときに一時的に作成されるアンロードファイルの削除に失敗しています。 対処方法については，メッセージの対策欄を参照してください。
31		監査証跡ファイル	KFAA51404-E	左記のメッセージが出力された場合は，監査証跡ファイルに対する操作が失敗しています。 対処方法については，メッセージの対策欄を参照してください。
32			KFAA51411-W	監査証跡ファイルの切り替え時に，現用の監査証跡ファイルが見つからない場合に，左記のメッセージが出力されます。現用の監査証跡ファイルに出力された監査証跡が失われたおそれがあります。
33			KFAA81401-I	左記のメッセージが出力された場合は，監査証跡ファイルの切り替えが行われました。 監査証跡ファイルに対する次の操作を実行する場合に，左記のメッセージを監視してください。 • 監査証跡ファイルを，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリに移動する • 監査証跡ファイルを，統一フォーマット用監査証跡ファイルに変換する
34			KFAA81402-I	左記のメッセージが出力された場合は，監査証跡ファイルのリネームが行われました。 監査証跡ファイルに対する次の操作を実行する場合に，左記のメッセージを監視してください。 • 監査証跡ファイルを，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリに移動する • 監査証跡ファイルを，統一フォーマット用監査証跡ファイルに変換する
35		更新行のカラム化機能で使用するファイル	KFAA41220-W	左記のメッセージが出力された場合は，更新行のカラム化機能で使用するファイルにエラーが発生しています。 対処方法については，メッセージの対策欄を参照してください。

項番	リソースの種類		監視対象のメッセージ	説明
36	ノード	マルチノード機能	• KFAA50053-E • KFAA50153-E • KFAA50154-W • KFAA60008-W • KFAA60009-E • KFAA60010-W	左記のメッセージが出力された場合は，メッセージが出力されたノードで異常が発生しています。 対処方法については，各メッセージの対策欄を参照してください。
37	DB エリア		• KFAA30756-E • KFAA41206-I • KFAA61210-E	左記のメッセージが出力された場合は，ディスクの空き容量が不足しており，対象の DB エリアにデータを格納できません。 対処方法については，各メッセージの対策欄を参照してください。
38			KFAA41230-W	データ用 DB エリアの使用率が，サーバ定義の adb_dbarea_wrn_pnt_use_rate オペランドに指定した使用率に達したときに左記のメッセージが出力されます。必要に応じてデータ用 DB エリアを拡張してください。 マルチノード機能を使用していない場合のデータ用 DB エリアの拡張方法については，「11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を，マルチノード機能を使用している場合のデータ用 DB エリアの拡張方法については，「16.10.3　データ用 DB エリアの追加・削除・拡張（マルチノード機能の使用時）」の「(3)　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照してください。 adb_dbarea_wrn_pnt_use_rate オペランドについては「7.2.4　状態監視に関するオペランド（set 形式）」の adb_dbarea_wrn_pnt_use_rate オペランドを参照してください。
39	ログファイル		• KFAA31711-E • KFAA41210-E	左記のメッセージが出力された場合は，ユーザログファイル数が不足しています。 対処方法については，各メッセージの対策欄を参照してください。 なお，サーバ定義adb_log_usrfile_num オペランドの指定を省略すると，ユーザログファイル数の不足によるエラーは発生しません。サーバ定義adb_log_usrfile_num オペランドについては，「7.2.3　システムログに関するオペランド（set 形式）」の adb_log_usrfile_num オペランドを参照してください。
40			KFAA61211-E	左記のメッセージが出力された場合は，システムログファイルが格納されているディスクの空き容量が不足しています。 対処方法については，メッセージの対策欄を参照してください。
41			KFAA81215-I	左記のメッセージが出力された場合は，データベースの回復処理が行われています。 対処方法については，メッセージの対策欄を参照してください。 左記のメッセージは，データベースの回復処理が完了するまで，サーバ定義のadb_log_rec_msg_interval オペランドに指定された間隔で出力されます。サーバ定義の

項番	リソースの種類	監視対象のメッセージ	説明
			adb_log_rec_msg_interval オペランドについては，「7.2.3 システムログに関するオペランド（set 形式）」の adb_log_rec_msg_interval オペランドを参照してください。
42	バッファ	• KFAA30919-E • KFAA41201-E	左記のメッセージが出力された場合は，バッファ面数が不足しています。 対処方法については，各メッセージの対策欄を参照してください。
43		• KFAA30953-E • KFAA41202-E	左記のメッセージが出力された場合は，バッファの負荷が高い状態になっています。 対処方法については，KFAA41202-E メッセージの対策欄を参照してください。
44	チャンク	KFAA51245-E	左記のメッセージが出力された場合は，チャンク数の最大値を超えるため，新たにチャンクを作成できません。1DB エリア内に作成できるチャンク数の上限を超えようとしています。 対処方法については，メッセージの対策欄を参照してください。
45		KFAA51246-E	左記のメッセージが出力された場合は，チャンク数の最大値を超えるため，新たにチャンクを作成できません。1 表に作成できるチャンク数の上限を超えようとしています。 対処方法については，メッセージの対策欄を参照してください。
46	ハッシュテーブル領域	KFAA51130-W	左記のメッセージが出力された場合は，ハッシュテーブル領域が不足したため，作業表が作成されています。 対処方法については，メッセージの対策欄を参照してください。 また，ハッシュテーブル領域が不足しても問題がないかを確認してください（データベースの設計上，ハッシュテーブル領域が不足することについて，想定されているかを確認してください）。
47	コネクション	• KFAA40020-W • KFAA40023-I	KFAA40020-W メッセージは，HADB サーバへの接続数が，サーバ定義のadb_sys_max_users_wrn_pnt オペランド（最大同時接続数に関する警告メッセージの出力契機）に指定した比率以上になったときに出力される警告メッセージです。KFAA40020-W メッセージが出力された場合は，不当に接続している AP またはコマンドがないかを確認してください。確認方法については，「10.8.1　AP およびコマンドの処理状況を確認する方法」を参照してください。 KFAA40023-I メッセージは，HADB サーバへの接続数が，サーバ定義のadb_sys_max_users_wrn_pnt オペランド（警告メッセージの出力済み状態のリセット契機）に指定した比率以下になったときに出力されるインフォメーションメッセージです。 なお，クライアントグループ機能を使用している場合は，adb_sys_max_users_wrn_pnt オペランドではなく，adbcltgrp オペランドの-w オプションに指定した次の値が適用されます。 • *最大同時接続数に関する警告メッセージの出力契機* • *警告メッセージの出力済み状態のリセット契機*

項番	リソースの種類	監視対象のメッセージ	説明
			[マルチノード機能] マルチノード機能を使用している場合，左記のメッセージはプライマリノードに出力されます。
48		KFAA30932-E	左記のメッセージが出力された場合は，AP またはコマンドが HADB サーバに接続できません。HADB サーバへの接続数が，最大同時接続数を超えています。 対処方法については，メッセージの対策欄を参照してください。 また，サーバ定義adb_sys_max_users オペランドの値が適切かどうかについて確認してください。サーバ定義 adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set 形式）」の adb_sys_max_users オペランドを参照してください。
49	文ハンドル	KFAA30931-E	左記のメッセージが出力された場合は，文ハンドルの数が上限を超えています。 対処方法については，メッセージの対策欄を参照してください。
50	実表	• KFAA30811-E • KFAA30812-E	左記のメッセージが出力された場合は，システム内で定義できる実表やインデクスの数が上限を超えています。または，対象の DB エリア内に格納できる実表やインデクスの数が上限を超えています。 対処方法については，各メッセージの対策欄を参照してください。
51	インデクス		
52	ビュー表	KFAA30811-E	左記のメッセージが出力された場合は，システム内で定義できるビュー表の数が上限を超えています。 対処方法については，メッセージの対策欄を参照してください。
53	システム表（実表）	• KFAA61213-W • KFAA61214-W	左記のメッセージが出力された場合は，システム表用 DB エリアの未使用セグメントが少なくなっています。そのため，adbreorgsystemdata コマンドでシステム表を再編成する必要があります。 対処方法については，各メッセージの対策欄を参照してください。
54	HADB ユーザ	KFAA30811-E	左記のメッセージが出力された場合は，システム内で定義できる HADB ユーザの数が上限を超えています。 対処方法については，メッセージの対策欄を参照してください。

10.6　メモリの使用状況の確認

ここでは，HADB サーバが使用するメモリの使用状況を知る方法を説明します。

10.6.1　全メモリの使用状況を確認したい場合

HADB サーバが使用する全メモリの使用状況を確認したい場合，HADB サーバでadbstat コマンドを実行してください。ここでは，全メモリの使用状況を確認するタイミング，および使用状況の確認方法について説明します。

adbstat コマンドのオプション，および各出力項目についてはマニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』を参照してください。

(1)　全メモリの使用状況を確認するタイミング

HADB サーバが使用している全メモリサイズが，サーバ定義に指定したメモリサイズと合っているかどうかを確認するため，定期的に全メモリの使用状況を確認してください。

(2)　全メモリの使用状況の確認方法

adbstat コマンドを実行して全メモリの使用状況を確認する方法を，次に示します。

■実行するコマンド

```
adbstat
```

adbstat コマンドを実行して出力される「HADB サーバの統計情報」のうち，次に示す情報を確認してください。

- Total_memory_max_size（全メモリの最大使用量）

そのあとで，Total_memory_max_size の値と，サーバ定義adb_sys_memory_limit オペランドの指定値を確認してください。なお，サーバ定義adb_sys_memory_limit オペランドを指定していない場合は，次に示すオペランドの指定値を確認してください。

- adb_sys_rthd_area_max オペランド
- adb_sys_proc_area_max オペランド

もし，Total_memory_max_size の値と，サーバ定義のオペランドに指定した値が大きく異なっている場合は，メモリサイズが適切に設定されていないことが考えられます。

その場合，サーバ定義のオペランドの指定値を，Total_memory_max_size の値より大きい値に変更してください。

10.6.2　共有メモリの使用状況を確認したい場合

共有メモリの使用状況を確認したい場合，HADB サーバでadbls -d shm コマンドを実行してください。HADB サーバが使用している共有メモリの合計値が表示されます。ここでは，共有メモリの使用状況を確認するタイミング，使用状況の確認方法，および空きメモリ量が少ない場合の対処方法について説明します。

adbls -d shm コマンドのオプション，および出力される内容についてはマニュアル『HADB　コマンドリファレンス』の『adbls -d shm（共有メモリの情報表示)』を参照してください。

(1)　共有メモリの使用状況を確認するタイミング

共有メモリの空きメモリ量が不足しないよう，定期的に共有メモリの使用状況を確認してください。

共有メモリの空きメモリ量が不足してメモリが確保できなくなると，エラーメッセージが出力されるため，エラーメッセージに従って対処してください。その際，adbls -d shm コマンドを実行して共有メモリの使用状況を確認してください。

(2)　共有メモリの使用状況の確認方法

adbls -d shm コマンドを実行して出力された共有メモリの使用状況を次に示します。

■共有メモリの使用状況の例

```
SHM_NUMBER    SHM_SIZE
8             1424683947
```

共有メモリの使用状況の例を基に，確認方法について説明します。

確認手順

HADB サーバが使用している共有メモリの量は，SHM_SIZE に表示されている数値（単位：バイト）を確認してください。例の場合，1,424,683,947 バイト（約 1.4 ギガバイト）を使用しています。

この数値と，OS のカーネルパラメタのSHMMAX に指定した値との差が小さい場合，共有メモリの空きメモリ量が不足していると判断できます。その場合は，「(3)　共有メモリの空きメモリ量が少ない場合の対処方法」を参照して対処してください。

差が大きい場合は，空きメモリ量が十分であると判断できるため，特に対応は不要です。

(3)　共有メモリの空きメモリ量が少ない場合の対処方法

SHM_SIZE とSHMMAX に指定した値と差が小さい場合，OS のカーネルパラメタのSHMMAX，SHMMNI，およびSHMALL に指定した値を増やすことを検討してください。

10.6.3　リアルスレッドごとのメモリの使用状況を確認したい場合

リアルスレッドごとのメモリ使用状況を確認したい場合は，HADB サーバでadbls -d mem コマンドを実行してください。

adbls -d mem コマンドのオプション，および出力される内容については，マニュアル『HADB　コマンドリファレンス』の『adbls -d mem（メモリの使用状況の表示)』を参照してください。

■メモリの使用状況を確認するタイミング

定期的に，各リアルスレッドのメモリ使用状況を確認してください。

■メモリの使用状況の確認方法

HADB サーバ，または HADB サーバの各リアルスレッドのメモリ使用状況を知りたい場合は，次のコマンドを実行してください。

```
adbls -d mem
```

コマンドの実行結果の例

```
THREAD-NO   LIMIT        USE          FREE         MAX
0           1073741824   105840224    162660848    105840224
1           1073741824   901024       267600048    901056
2           1073741824   900512       267600560    900512
3           1073741824   900512       267600560    900512
4           1073741824   900512       267600560    900512
5           1073741824   900512       267600560    900512
:               :            :            :            :
```

[説明]

- THREAD-NO が 1 以降の行に，リアルスレッドごとのメモリ使用状況が表示されます。0 の行には，プロセス共通メモリに関する情報が表示されます。
- LIMIT には，HADB サーバが使用できるメモリ容量の上限値（単位：バイト）が表示されます。
- USE には，HADB サーバが使用しているメモリ容量（単位：バイト）が表示されます。
- FREE には，空いているメモリ容量（単位：バイト）が表示されます。
- MAX には，過去に HADB サーバが使用したメモリ容量（単位：バイト）の最大値が表示されます。

各リアルスレッドがどの AP やコマンドで使用されているかを確認したい場合は，「10.7　HADB サーバで実行中のスレッドの確認」を参照してください。

10.7　HADB サーバで実行中のスレッドの確認

次に示すことをしたい場合は，adbls -d thd コマンドを実行してください。

- HADB サーバで実行中のリアルスレッドおよび擬似スレッドを確認したい場合
- リアルスレッドを使用しているコネクションや SQL 文を特定したい場合

adbls -d thd コマンドの実行結果の例を次に示します。

```
adbls -d thd

THREAD-NO       T-ID(OS)        TOTAL_PSEUDO_THREAD  TYPE    CNUMBER    SQL_ID          CLIENT_GROUP        T-ID(LWP)
1               2425685888      130                  SYSTEM                                                 16243
2               1354499840      130                  SYSTEM                                                 18644
3               3501967104      130                  SYSTEM                                                 18506
4               0284952320      130                  SYSTEM                                                 18285
5               2427995904      130                  SYSTEM                                                 18514
6               1983280896      130                  CONNECT 1                          cltgrp01            16500
7               1970689792      130                  CONNECT                                                16527
8               1958098688      130                  CONNECT                                                16584
9               1874847488      130                  CONNECT                                                16624
10              1862256384      130                  CONNECT                                                16735
11              1849665280      130                  CONNECT                                                16790
12              1837074176      130                  CONNECT                                                16807
13              1824483072      130                  CONNECT                                                16836
14              1405085440      130                  PROCESS                                                16863
15              1396692736      130                  PROCESS                                                16890
16              1388300032      130                  PROCESS                                                16908
17              1379907328      130                  PROCESS                                                16933
18              1371514624      130                  PROCESS                                                16958
19              1363121920      130                  PROCESS                                                16988
20              1354729216      130                  PROCESS                                                17018
21              0331114240      130                  PROCESS                                                17034
22              0322721536      130                  PROCESS                                                17065
23              0314328832      130                  PROCESS 1          1               cltgrp01            17092
24              0305936128      130                  PROCESS                                                17118
25              0297543424      130                  PROCESS                                                17142
```

[説明]

HADB サーバで実行中のスレッドの一覧が表示されます。

adbls -d thd コマンドの実行結果の各項目の説明については，『HADB　コマンドリファレンス』の『adbls -d thd（スレッドの状態表示)』の『使用例』を参照してください。

リアルスレッドを使用しているコネクションに対応する AP やコマンドの処理状況を確認したい場合は，「10.8.1　AP およびコマンドの処理状況を確認する方法」を参照してください。

10.8　トランザクションの処理状況の確認

ここでは，AP やコマンドを実行した場合の，トランザクションの処理状況を確認する方法を説明します。

10.8.1　AP およびコマンドの処理状況を確認する方法

AP やコマンドを実行したときにトランザクションの処理時間が予想より長い場合は，トランザクションの処理状況を確認して対処する必要があります。

トランザクションの処理状況を確認する方法を次に示します。なお，adbinit コマンドおよびadbstat コマンドの処理状況は確認できません。

確認手順

1. adbls -d cnct コマンドを実行する

   項目「PROGRAM」に，対象のコネクションが表示されているかどうか確認してください。

   ■AP の処理状況を確認したい場合

   クライアント定義のadb_clt_ap_name オペランドに指定した AP 識別子が表示されます。adbls -d cnct コマンドの実行結果の例を次に示します。

   実行結果の例

```
CID  CNUMBER  CONNECT_TIME         PROGRAM  C-PID  IP-ADDRESS   STATUS   TRN_ISO_LV      ACCESS_MODE
1    2        2016-06-08 10:11:54  AP001    20770  (127.0.0.1)  STARTED  READ_COMMITTED  READ_ONLY

CONNECTION_INFORMATION           NODE_NO  CLIENT_TYPE     CLIENT_GROUP   SQL_ELAPSED_TIME
000010000000002-000050c5df220700          AP(C Library)   GROUP1         3005125
```

   AP 識別子が表示されている場合は手順 3.に進んでください。

   adb_clt_ap_name オペランドを指定していない場合は，********が表示されます。その場合は，手順 2.に進んでください。

> **メモ**
>
> SQL 文が実行中の場合は，SQL_ELAPSED_TIME に，実行開始から最も時間が経過している SQL 文の経過時間（マイクロ秒）が表示されます。

   ■コマンドの処理状況を確認したい場合

   コネクションが確立している場合，対象のコマンド名（adbimport など）が表示されます。adbsql コマンドの場合は，クライアント定義のadb_clt_ap_name オペランドに指定した AP 識別子が表示されます。

   表示されている場合は手順 3.に進んでください。表示されていない場合は手順 2.に進んでください。

2. adbls -d srv コマンドを実行する

   対象のコネクションが表示されていない場合，HADB サーバが稼働中かどうかを確認してください。

   adbls -d srv コマンドの実行結果の例を次に示します。

   実行結果の例

```
SVID  STATUS  START_TIME            MSGLOG_STATUS
6477  ACTIVE  2017/03/02 14:18:34   NORMAL
```

項目「STATUS」が"ACTIVE"の場合，HADB サーバは稼働中のため，手順 6.に進んでください。

項目「STATUS」が"ACTIVE"以外の場合は，adbstart コマンドを実行して HADB サーバを開始してから，AP やコマンドを再度実行してください。

3. コネクション ID を確認する

手順 1.でadbls -d cnct コマンドを実行し対象のコネクションが表示されている場合，項目「CID」にコネクション ID が表示されているかどうか確認してください。コネクション ID が表示されている場合は手順 4.に，表示されていない場合は手順 6.に進んでください。

4. adbstat -c cnct -n コマンドを実行する

トランザクションの処理状況を確認するため，手順 1.で確認した AP 識別子を基に，adbstat -c cnct -n コマンドを実行してください。-n オプションには，AP 識別子を指定してください。

コネクションの稼働情報が出力されるため，次に示す出力項目を確認してトランザクションの処理状況を確認します。

- SQL_execute_wait_total_time（SQL 文の実行待ち状態の合計時間）
- SQL_execute_wait_cnt（SQL 文の実行待ち状態の発生回数）
- 実行している各 SQL 文の実行時間

  SELECT_total_time（SELECT 文の実行時間）

  INSERT_total_time（INSERT 文の実行時間）

  UPDATE_total_time（UPDATE 文の実行時間）

  DELETE_total_time（DELETE 文の実行時間）

  PURGE_CHUNK_total_time（PURGE CHUNK 文の実行時間）

各出力項目の値が，前回adbstat コマンドを実行したときの結果と比較して増加している場合は，処理に時間が掛かっていることが確認できます。もし，処理中のトランザクションを強制終了したい場合は，手順 5.に進んでください。

なお，初めてadbstat コマンドを実行した場合，または前回実行したadbstat コマンドの結果がわからない場合は，処理に時間が掛かっているかどうかを確認できません。

5. adbcancel コマンドを実行する

必要に応じて処理中のトランザクションを強制終了します。

処理中のトランザクションを強制終了する場合は，手順 3.で確認したコネクション ID を-u オプションで指定して，adbcancel コマンドを実行してください。強制終了後，AP やコマンドを再度実行してください。

6. AP を調査する

HADB サーバが稼働中だが，コネクション ID が表示されていない場合，対象の AP が HADB サーバに接続できていません。そのため，AP が HADB サーバに接続できるかどうか調査してください。また，クライアント定義のadb_clt_rpc_srv_host オペランド，およびadb_clt_rpc_srv_port オペランドの指定値を見直してください。

10.8.2　処理リアルスレッドの確保待ちの発生状況を確認する方法

APやコマンドを実行したときに，処理リアルスレッドの確保待ちが発生しているかどうかを確認する方法について説明します。

確認手順

1. adbls -d cnct コマンドを実行する

   項目「STATUS」を確認してください。THREAD_WAITINGが表示されている場合は，処理リアルスレッドの確保待ちが発生しています。

   処理リアルスレッドを使用している，ほかのAPやコマンドの終了を待てない場合は，手順2.に進んでください。

2. adbcancel コマンドを実行する

   処理リアルスレッドを使用している，ほかのAPやコマンドをadbcancel コマンドで強制終了してください。

APやコマンドを実行したときに，処理リアルスレッドの確保待ちの発生を防ぎたい場合は，「6.23.2　コマンドの実行時に使用する処理リアルスレッド数に関する考慮点」を参照してください。

adbls -d cnct コマンド，およびadbcancel コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

10.8.3　排他資源の確保待ちの発生状況を確認する方法

APやコマンドを実行したときに，排他資源の確保待ちが発生しているかどうかを確認する方法について説明します。

確認手順

1. adbls -d cnct コマンドを実行する

   対象のAP，またはコマンドのコネクションIDを確認します。項目「CID」を確認してください。

2. adbls -d lock コマンドを実行する

   手順1.で確認したコネクションIDと，項目「CID」に表示されたコネクションIDを突き合わせてください。そのあとで，項目「STATUS」を確認してください。WAITINGが表示されている場合は，対象のAPまたはコマンドが使用する排他資源について，排他資源の確保待ちが発生しています。

   排他資源を使用している，ほかのAPやコマンドの終了を待てない場合は，手順3.に進んでください。

3. adbcancel コマンドを実行する

   排他資源を使用している，ほかのAPやコマンドをadbcancel コマンドで強制終了してください。

APやコマンドを実行したときに，排他資源の確保待ちの発生を防ぎたい場合は，「6.23.3　コマンド同時実行時の排他制御に関する考慮点」を参照して，排他資源の確保待ちが発生しないように注意してください。

adbls -d cnct コマンド，adbls -d lock コマンドおよびadbcancel コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

メモ

排他資源の種別が表または前処理表の場合，adbls -d lock コマンドで出力される項目「RESOURCEID」で，表 ID を確認できます。表 ID をキーにすると，AP またはコマンドが確保要求を発行した表の表名がわかります。表 ID から表の表名を確認する方法を次に示します。

- 表 ID が0x00020001～0x000200C8 の場合
  「付録 B.1　ディクショナリ表の概要」の「(1)　ディクショナリ表の一覧」を参照すると，表 ID から表の表名を確認できます。
- 表 ID が0x000200C9～0x00020190 の場合
  「付録 C.1　システム表の概要」の「(1)　システム表の一覧」を参照すると，表 ID から表の表名を確認できます。
- 上記以外の表 ID の場合
  「付録 B.22　ディクショナリ表の検索」の「(1)　表 ID から表の表名を調べる場合」を参照してディクショナリ表を検索すると，表 ID から表の表名を確認できます。

10.8.4　トランザクションがロールバックしたときの状況を確認する方法

AP またはコマンドを実行したときに，トランザクションのロールバックが発生しているかどうかを確認する方法について説明します。また，トランザクションのロールバックが発生している場合に，ロールバックの進捗状況を確認する方法についても説明します。

■トランザクションがロールバックしているかどうかを確認する方法

トランザクションがロールバックしているかどうかを確認する場合，adbls -d cnct コマンドを実行してください。出力項目「STATUS」を確認して，"ROLLBACKING"となっている AP およびコマンドがないかどうかを確認してください。"ROLLBACKING"となっている場合は，対象の AP やコマンドはロールバック中です。

■ロールバックの進捗状況を確認する方法

トランザクションのロールバックに想定以上の時間が掛かっている場合は，KFAA81215-I メッセージが出力されているかどうかを確認してください。KFAA81215-I メッセージを確認すると，トランザクションのロールバックに伴う，データベースの回復処理の進捗状況を確認できます。

KFAA81215-I メッセージから，データベースの回復処理が完了して，後続の処理を再開できる時間を見積もってください。

なお，KFAA81215-I メッセージは，サーバ定義のadb_log_rec_msg_interval オペランドの指定値に従って，一定の間隔で出力されます。

■ロールバックが完了したあとに実施すること

トランザクションのロールバックが完了したあとで，KFAA81211-I メッセージが出力されているかどうかを確認してください。KFAA81211-I メッセージには，トランザクションの回復処理に関する情報が出力されます。

KFAA81211-I メッセージを確認して，回復処理に時間が掛かったトランザクションについて，コミット回数を増やすなど，1 トランザクションによるデータベースの更新量を減らすことを検討してください。

なお，KFAA81211-I メッセージは，サーバ定義のadb_log_rec_msg_interval オペランドの指定値よりも，回復処理に時間が掛かったトランザクションが存在する場合に出力されます。

また，KFAA81211-I メッセージに出力されたトランザクション ID から，対象のトランザクションで実行された SQL 文を特定できます。トランザクション ID をキーにして，SQL トレース情報の「SQL 文の実行情報」に出力されたtran_id の値を確認してください。トランザクション ID とtran_id が一致する SQL 文が，対象のトランザクションで実行された SQL 文です。SQL 文の実行情報については，「10.11.2　SQL トレース情報に出力される情報」の「(2)　SQL 文の実行情報」を参照してください。

10.9　データベースの状態や使用量の確認

ここでは，adbdbstatus コマンドを使用して，HADB のデータベースの状態や使用量を確認する方法を説明します。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

10.9.1　データベースの使用量を確認したい場合

DB エリアのサマリ情報を取得すると，HADB のデータベース全体の使用量を確認できます。

実行するコマンドの例

ここでは，DB エリア全体のサマリ情報を出力します。

```
adbdbstatus -d summary -c dbarea -S M
```

コマンドを実行すると，標準出力に DB エリア全体のサマリ情報が出力されます。adbdbstatus コマンドによる出力結果の確認項目を次に示します。

(1)　各 DB エリアの使用率を確認する場合

DBarea_name の出力内容をキーにして，Used_ratio の出力内容を確認してください。Used_ratio の出力内容が 100%に近い場合は，DB エリアファイルの種類によって，次に示す対策を検討してください。

DB エリアファイルがブロックスペシャルファイルのとき

この値が 100%に近いときにデータの格納を続けると，DB エリアファイルが満杯となるおそれがあります。そのため，「15.3.1　DB エリアファイルの容量増加が原因の場合」を参照して，ブロックスペシャルファイルを格納しているディスクの空き容量を確保することを検討してください。

DB エリアファイルがレギュラーファイルのとき

この値が 100%に近いときにデータの格納を続けると，DB エリアの自動増分が行われます。自動増分が行われると，DB エリアの容量が自動的に増加しますが，DB エリアファイル拡張時の更新処理の性能が低下します。もし，DB エリアの自動増分に失敗した場合は，「15.3.2　DB エリアの自動増分の失敗が原因の場合」を参照してください。

(2)　各 DB エリアの使用量を確認する場合

DBarea_name の出力内容をキーにして，MB_Used_segments の出力内容を確認してください。各 DB エリア内でデータを格納するために使用している領域を，メガバイト単位で確認できます。

10.9.2　実表の状態や使用量を確認したい場合

adbdbstatus コマンドで，表のサマリ情報を取得すると，実表の状態や使用量を確認できます。

(1)　実表が更新不可状態かどうかを確認する場合

実表が更新不可状態かどうかを確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，表ADBUSER01.T1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c table -n ADBUSER01.T1
```

adbdbstatus コマンドを実行すると，標準出力に表ADBUSER01.T1 のサマリ情報が出力されます。

表のサマリ情報から，Non-updatable の出力内容を確認してください。non-updatable が出力されている場合，実表は更新不可状態となっています。

更新不可状態を解除するためには，Command_status およびRerun_command の出力内容を確認して，Rerun_command に出力されているコマンドを再実行してください。

(2)　実表の使用量を確認する場合

実表の使用量を確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，表ADBUSER01.T1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c table -n ADBUSER01.T1 -S M --shared-lock
```

adbdbstatus コマンドを実行すると，標準出力に表ADBUSER01.T1 のサマリ情報が出力されます。

表のサマリ情報から，MB_Used_pages の出力内容を確認してください。DB エリア内に格納されている実表のデータが使用している領域を，メガバイト単位で確認できます。

10.9.3　B-tree インデクスの状態や使用量を確認したい場合

adbdbstatus コマンドで，インデクスのサマリ情報を取得すると，B-tree インデクスの状態や使用量を確認できます。

(1)　B-tree インデクスが未完状態かどうかを確認する場合

B-tree インデクスが未完状態かどうかを確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，B-tree インデクスADBUSER01.IDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.IDX1
```

adbdbstatus コマンドを実行すると，標準出力に B-tree インデクスADBUSER01.IDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，Unfinished の出力内容を確認してください。unfinished が出力されている場合，B-tree インデクスは未完状態となっています。

B-tree インデクスの未完状態を解除するためには，「15.9.1　B-tree インデクスが未完状態になったときの対処方法」を参照してください。

(2)　ユニークインデクスが一意性制約違反状態かどうかを確認する場合

ユニークインデクスが一意性制約違反状態かどうかを確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，B-tree インデクスADBUSER01.IDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.IDX1
```

adbdbstatus コマンドを実行すると，標準出力に B-tree インデクスADBUSER01.IDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，Unique_constraint_violated の出力内容を確認してください。

- unique_constraint_violated が出力されている場合，B-tree インデクスは一意性制約違反状態となっています。一意性制約違反状態を解除するためには，「15.9.2　一意性制約に違反した場合の対処方法(KFAA61205-W メッセージが出力された場合)」の「(2)　一意性制約違反状態を解除する方法」を参照してください。
- unknown が出力されている場合は，B-tree インデクスが一意性制約違反状態かどうかを判断できません。adbidxrebuild コマンドで B-tree インデクスを再作成したあとで，再度adbdbstatus コマンドを実行して一意性制約違反状態かどうかを確認してください。

  adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild(インデクスの再作成)』を参照してください。

(3)　B-tree インデクスの使用量を確認する場合

B-tree インデクスの使用量を確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，B-tree インデクスADBUSER01.IDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.IDX1 -S M --shared-lock
```

adbdbstatus コマンドを実行すると，標準出力に B-tree インデクスADBUSER01.IDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，MB_Used_pages の出力内容を確認してください。DB エリア内に格納されている B-tree インデクスのデータが使用している領域を，メガバイト単位で確認できます。

10.9.4　テキストインデクスの状態や使用量を確認したい場合

adbdbstatus コマンドで，インデクスのサマリ情報を取得すると，テキストインデクスの状態や使用量を確認できます。

(1)　テキストインデクスが未完状態かどうかを確認する場合

テキストインデクスが未完状態かどうかを確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，テキストインデクスADBUSER01.TIDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.TIDX1
```

adbdbstatus コマンドを実行すると，標準出力にテキストインデクスADBUSER01.TIDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，Unfinished の出力内容を確認してください。unfinished が出力されている場合，テキストインデクスは未完状態となっています。

テキストインデクスの未完状態を解除するためには，「15.10.1　テキストインデクスが未完状態になったときの対処方法」を参照してください。

(2)　テキストインデクスの使用量を確認する場合

テキストインデクスの使用量を確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，テキストインデクスADBUSER01.TIDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.TIDX1 -S M --shared-lock
```

adbdbstatus コマンドを実行すると，標準出力にテキストインデクスADBUSER01.TIDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，MB_Used_pages の出力内容を確認してください。DB エリア内に格納されているテキストインデクスのデータが使用している領域を，メガバイト単位で確認できます。

10.9.5　レンジインデクスの状態や使用量を確認したい場合

adbdbstatus コマンドで，インデクスのサマリ情報を取得すると，レンジインデクスの状態や使用量を確認できます。

(1)　レンジインデクスが未完状態かどうかを確認する場合

レンジインデクスが未完状態かどうかを確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，レンジインデクスADBUSER01.RIDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.RIDX1
```

adbdbstatus コマンドを実行すると，標準出力にレンジインデクスADBUSER01.RIDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，Unfinished の出力内容を確認してください。unfinished が出力されている場合，レンジインデクスは未完状態となっています。

レンジインデクスの未完状態を解除するためには，「15.11.1　レンジインデクスが未完状態になったときの対処方法」を参照してください。

(2)　レンジインデクスの使用量を確認する場合

レンジインデクスの使用量を確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，レンジインデクスADBUSER01.RIDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.RIDX1 -S M --shared-lock
```

adbdbstatus コマンドを実行すると，標準出力にレンジインデクスADBUSER01.RIDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，MB_Used_pages の出力内容を確認してください。DB エリア内に格納されているレンジインデクスのデータが使用している領域を，メガバイト単位で確認できます。

(3)　レンジインデクスのページグループサイズを確認する場合

レンジインデクスのページグループサイズを確認する場合，次に示すadbdbstatus コマンドを実行してください。

実行するコマンドの例

ここでは，レンジインデクスADBUSER01.RIDX1 のサマリ情報を出力します。

```
adbdbstatus -d summary -c index -n ADBUSER01.RIDX1 -S M --shared-lock
```

adbdbstatus コマンドを実行すると，標準出力にレンジインデクスADBUSER01.RIDX1 のサマリ情報が出力されます。

インデクスのサマリ情報から，Pagegroup_size の出力内容を確認してください。Pagegroup_size の値がページグループサイズ（単位：ページ数）となります。

10.9.6　DB エリアファイルの使用量を確認したい場合

DB エリアの使用量情報を取得すると，DB エリア内の表やインデクスの使用量，および DB エリアファイルごとの使用量を確認できます。

実行するコマンドの例

ここでは，DB エリアDBAREA01 の使用量情報を出力します。

```
adbdbstatus -d used -c dbarea -n DBAREA01 -S M --shared-lock
```

コマンドを実行すると，標準出力に DB エリアDBAREA01 の使用量情報が出力されます。adbdbstatus コマンドによる出力結果の確認項目を次に示します。

(1)　表やインデクスごとの使用量を確認する場合

Object_type，Schema_name およびObject_identifier の出力内容をキーにして，MB_Used_pages の出力内容を確認してください。

1 つの表の使用量を確認するとき

対象表のSchema_name およびObject_identifier を確認して，Object_type がtable であることを確認してください。Schema_name およびObject_identifier をキーにして，MB_Used_pages の出力内容を確認してください。対象表の使用量をメガバイト単位で求められます。

対象表のSchema_name およびObject_identifier で同じものが複数行ある場合は，MB_Used_pages の出力内容をすべて合計してください。合計した結果が，対象表の使用量になります。

1 つのインデクスの使用量を確認するとき

対象インデクスのSchema_name およびObject_identifier を確認して，Object_type がindex であることを確認してください。Schema_name およびObject_identifier をキーにして，MB_Used_pages の出力内容を確認してください。対象インデクスの使用量をメガバイト単位で求められます。

対象インデクスのSchema_name およびObject_identifier で同じものが複数行ある場合は，MB_Used_pages の出力内容をすべて合計してください。合計した結果が，対象インデクスの使用量になります。

メモ

adbdbstatus コマンドに--shared-lock オプションを指定した場合，システム表（実表）およびシステム表（実表）に定義されたインデクスに対応する削除仕掛中のチャンクについて，セグメントに関する内容は出力されません。

(2) 各 DB エリアファイルの使用率を確認する場合

DBarea_filename の出力内容をキーにして，MB_Total_segments およびMB_Used_segments の出力内容を確認してください。次に示す計算式で，各 DB エリアファイルの使用率を求められます。

計算式（単位：パーセント）

（*MB_Used_segments*÷*MB_Total_segments*）×100

計算式で求めた結果が 100%に近い場合は，DB エリアファイルの種類によって，次に示す対策を検討してください。

DB エリアファイルがブロックスペシャルファイルのとき

この値が 100%に近いときにデータの格納を続けると，DB エリアファイルが満杯となるおそれがあります。そのため，「15.3.1　DB エリアファイルの容量増加が原因の場合」を参照して，ブロックスペシャルファイルを格納しているディスクの空き容量を確保することを検討してください。

DB エリアファイルがレギュラーファイルのとき

この値が 100%に近いときにデータの格納を続けると，DB エリアの自動増分が行われます。自動増分が行われると，DB エリアの容量が自動的に増加しますが，DB エリアファイル拡張時の更新処理の性能が低下します。もし，DB エリアの自動増分に失敗した場合は，「15.3.2　DB エリアの自動増分の失敗が原因の場合」を参照してください。

また，システム表用 DB エリアファイルの場合は，上記の対策以外に，adbreorgsystemdata コマンドを実行することも検討してください。adbreorgsystemdata コマンドを実行して，システム表の再編成を行うことで，ディスクの空き容量を確保できます。システム表の再編成については，「11.17　システム表の再編成」を参照してください。

(3) 各 DB エリアファイルの使用量を確認する場合

DBarea_filename の出力内容をキーにして，MB_Used_segments の出力内容を確認してください。各 DB エリアファイル内でデータを格納するために使用している領域をメガバイト単位で求められます。

10.9.7　アーカイブファイルのサイズを確認したい場合

adbdbstatus コマンドを実行して「アーカイブ状態のチャンクの使用量情報」を取得すると，アーカイブファイルのサイズを確認できます。

実行するコマンドの例

アーカイブマルチチャンク表ADBUSER01.T1 に関する「アーカイブ状態のチャンクの使用量情報」を出力します。

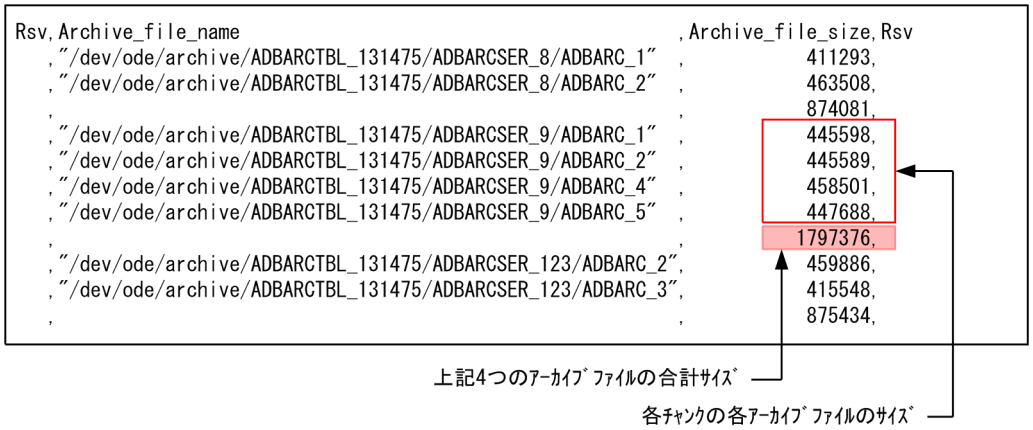
```
adbdbstatus -d used -c archivechunk -n ADBUSER01.T1
```

出力結果の確認項目を次に示します。

■アーカイブファイルのサイズを確認する場合

Archive_file_size に各アーカイブファイルのサイズが出力されます。Archive_file_name に何も出力されていない行に，チャンク内のデータを格納している全アーカイブファイルの合計サイズが出力されます。

(例)



10.10　統計解析の運用（HADB サーバの稼働情報の確認）

adbstat コマンドを実行すると，次に示す HADB サーバの稼働情報を出力できます。

- HADB サーバの統計情報
- コネクションの稼働情報
- グローバルバッファの統計情報
- SQL 文の統計情報

これらの情報を利用して，HADB サーバの稼働情報を確認する方法について説明します。

また，HADB サーバの稼働情報が出力される統計ログファイルの詳細についても説明します。

adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』を参照してください。

10.10.1　HADB サーバの統計情報の運用方法

HADB サーバの統計情報では，HADB サーバ全体のメモリの使用量，および HADB サーバの稼働状況を確認できます。

サーバ定義に指定したオペランドの値を変更したい場合などに，HADB サーバの統計情報を利用できます。

(1)　HADB サーバの統計情報の出力と関連情報の取得

adbstat コマンドを実行すると，HADB サーバの統計情報が出力されます。出力される情報は，前回出力した情報からの差分になります。そのため，HADB サーバの稼働状況を時系列に確認したい場合，定期的に HADB サーバの統計情報を出力することを推奨します。

■HADB サーバの統計情報を出力する方法

マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』の『使用例』の『システム全体の稼働状況を調べる場合』を参照してください。

また，HADB サーバの統計情報を出力するときに，次に示す関連情報もあわせて取得することを推奨します。

- **サーバメッセージログファイル**

  OS のcp コマンドなどでコピーして取得してください。サーバメッセージログファイルの格納先は，「10.4.2　メッセージログの参照方法（メッセージログの出力先)」を参照してください。

(2)　HADB サーバの統計情報と関連情報の利用方法

定期的に，HADB サーバの統計情報と関連情報を取得しておくことで，HADB サーバの稼働状況の変化を確認できます。取得した HADB サーバの統計情報と関連情報は，チューニングを実施するときに利用できます。

■HADB サーバの統計情報と関連情報の利用方法

1. HADB サーバの統計情報の出力日時を確認する

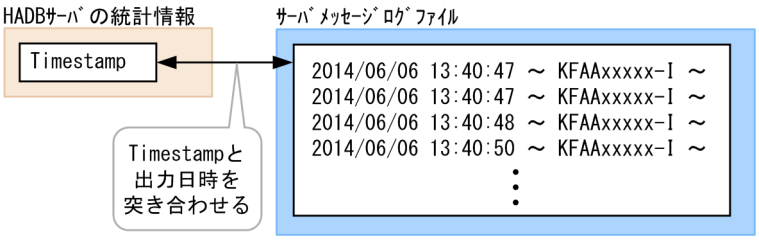
   「HADB サーバの統計情報」に出力されるTimestamp（HADB サーバの統計情報の出力日時）を確認してください。

2. サーバメッセージログファイルの出力日時と突き合わせる

   手順 1.で確認したTimestamp と，サーバメッセージログファイル内のメッセージの出力日時を突き合わせてください。Timestamp と出力日時を突き合わせることで，HADB サーバがどのような処理を行っていたかを確認できます。

HADB サーバの統計情報と関連情報の利用イメージを，次の図に示します。

図 10-8　HADB サーバの統計情報と関連情報の利用イメージ



また，HADB サーバの統計情報と関連情報は，次に示すチューニングを実施するときに利用できます。

- 「13.1.5　SQL 文の実行待ち状態の発生防止」
- 「13.1.11　作業表用 DB エリアのファイル構成の変更」
- 「13.4.1　HugePages を適用した共有メモリの使用量の削減」

10.10.2　コネクションの稼働情報の運用方法

コネクションの稼働情報では，HADB サーバに接続しているコネクションごとの稼働状況を確認できます。

サーバ定義に指定したオペランドの値を変更したり，AP，コマンドおよび SQL 文の実行方法を変更したりする場合などに，コネクションの稼働情報を利用できます。

(1)　コネクションの稼働情報の出力と関連情報の取得

adbstat コマンドを実行すると，コネクションの稼働情報が出力されます。出力される情報は，前回出力した情報からの差分になります。HADB サーバに接続しているコネクションごとの稼働状況を時系列に確

認したい場合，AP，コマンドおよびSQL文の実行中に，定期的にコネクションの稼働情報を出力することを推奨します。

■コネクションの稼働情報を出力する方法

マニュアル『HADB　コマンドリファレンス』の『adbstat（HADBサーバの統計解析）』の『使用例』の『特定のAPの稼働状況を調べる場合』を参照してください。

また，コネクションの稼働情報を出力するときに，次に示す関連情報もあわせて取得することを推奨します。

- **サーバメッセージログファイル**

  OSのcpコマンドなどでコピーして取得してください。サーバメッセージログファイルの格納先は，「10.4.2　メッセージログの参照方法（メッセージログの出力先）」を参照してください。

- **AP識別子**

  実行中のAPやコマンドに設定されたAP識別子は，adbls -d cnctコマンドで確認してください。APの場合，クライアント定義adb_clt_ap_nameオペランドで指定されたAP識別子が出力されます。

  adbls -d cnctコマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示）』を参照してください。

  クライアント定義adb_clt_ap_nameオペランドについては，マニュアル『HADB　AP開発ガイド』の『APの状態監視に関するオペランド』を参照してください。

> **メモ**
>
> 複数のAPがある場合，各APが使用するクライアント定義adb_clt_ap_nameオペランドには，それぞれ異なるAP識別子を指定することを推奨します。

(2)　コネクションの稼働情報と関連情報の利用方法

定期的に，コネクションの稼働情報と関連情報を取得しておくことで，HADBサーバに接続しているコネクションごとの稼働状況を確認できます。取得したコネクションの稼働情報と関連情報は，チューニングを実施するときに利用できます。

■コネクションの稼働情報と関連情報の利用方法

1. コネクションの稼働情報を確認する

   「コネクションの稼働情報」に出力される情報のうち，次に示す情報を確認してください。

   - Timestamp（コネクションの稼働情報の出力日時）
   - AP_name（AP識別子）

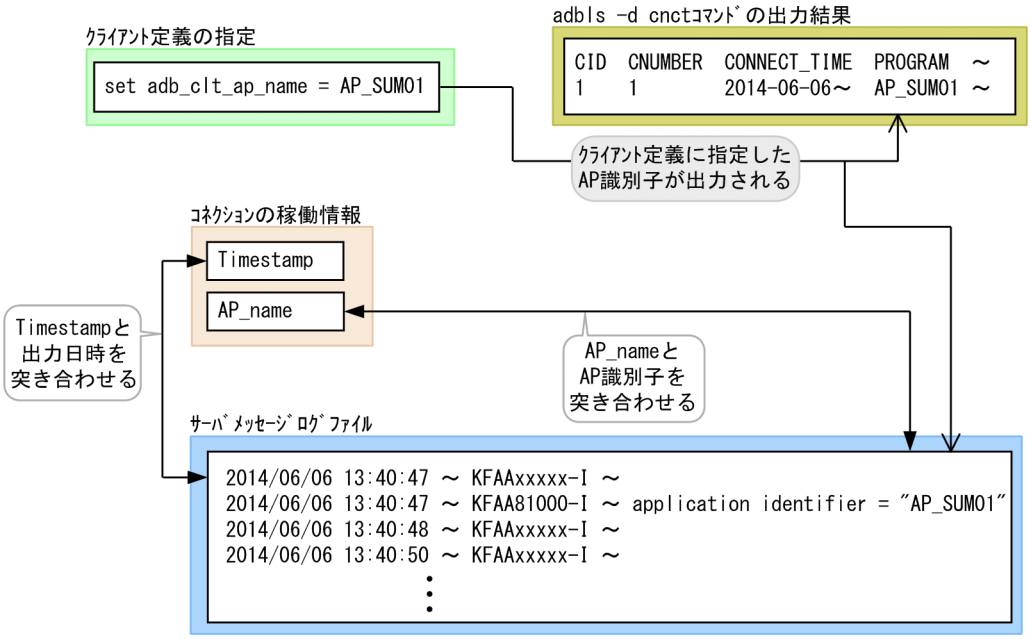
2. サーバメッセージログファイルと突き合わせる

   手順1.で確認したTimestampと，サーバメッセージログファイル内のメッセージの出力日時を突き合わせてください。Timestampと出力日時を突き合わせることで，対象のコネクションがどのような処理を行っていたかを確認できます。

また，手順 1.で確認したAP_name と，サーバメッセージログファイル内のKFAA81000-I メッセージの AP 識別子を突き合わせてください。AP_name と AP 識別子を突き合わせることで，対象の AP やコマンドを確認できます。

コネクションの稼働情報と関連情報の利用イメージを，次の図に示します。

図 10-9　コネクションの稼働情報と関連情報の利用イメージ



また，コネクションの稼働情報と関連情報は，次に示すチューニングを実施するときに利用できます。

- 「13.1.6　SQL 文の処理時間の短縮」
- 「13.2.4　ローカル作業表を作成する SQL 文の実行時間を短縮する方法」の「(1)　コネクションの稼働情報によるローカル作業表を作成する SQL 文の実行時間の短縮」

10.10.3　グローバルバッファの統計情報の運用方法

グローバルバッファの統計情報では，グローバルバッファごとの使用状況を確認できます。

サーバ定義に指定したオペランドの値を変更したり，データ用 DB エリア，表およびインデクスを変更したりする場合などに，グローバルバッファの統計情報を利用できます。

(1)　グローバルバッファの統計情報の出力と関連情報の取得

adbstat コマンドを実行すると，グローバルバッファの統計情報が出力されます。出力される情報は，HADB サーバを開始したときからの情報になります。グローバルバッファの使用状況を確認したい場合，グローバルバッファの統計情報を出力することを推奨します。

■グローバルバッファの統計情報を出力する方法

マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析）』の『使用例』の『特定のグローバルバッファの稼働状況を調べる場合』を参照してください。

また，グローバルバッファの統計情報を出力するときに，次に示す関連情報もあわせて取得することを推奨します。

- サーバメッセージログファイル

  OS のcp コマンドなどでコピーして取得してください。サーバメッセージログファイルの格納先は，「10.4.2　メッセージログの参照方法（メッセージログの出力先）」を参照してください。

- グローバルバッファ名

  グローバルバッファ名を，adbls -d gbuf コマンドで確認してください。サーバ定義adbbuff オペランドの-g オプションで指定されたグローバルバッファ名が出力されます。

  adbls -d gbuf コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d gbuf (グローバルバッファの情報表示)』を参照してください。サーバ定義adbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」を参照してください。

> **メモ**
>
> 次に示す方法でもグローバルバッファ名を確認できます。
>
> - HADB サーバ開始時のメッセージログを確認し，出力されているKFAA50027-I メッセージのサーバ定義情報から，グローバルバッファ名を調べます。
> - HADB サーバの開始に使用したサーバ定義ファイル（$ADBDIR/conf/server.def）を参照し，グローバルバッファ名を調べます。

(2)　グローバルバッファの統計情報と関連情報の利用方法

グローバルバッファの統計情報と関連情報を取得しておくことで，グローバルバッファごとの使用状況を確認できます。取得したグローバルバッファの統計情報と関連情報は，チューニングを実施するときに利用できます。

■グローバルバッファの統計情報と関連情報の利用方法

1. グローバルバッファの統計情報を確認する

   「グローバルバッファの統計情報」に出力される情報のうち，次に示す情報を確認してください。

   - Timestamp（グローバルバッファの統計情報の出力日時）
   - DBbuff_name（グローバルバッファ名）

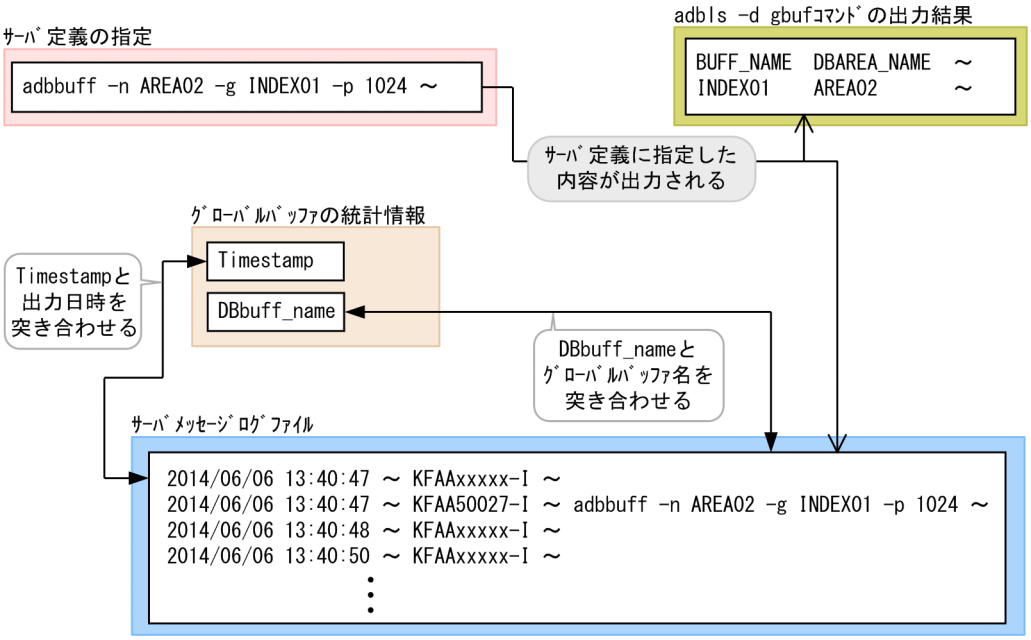
2. サーバメッセージログファイルと突き合わせる

手順 1.で確認したTimestamp と，サーバメッセージログファイル内のメッセージの出力日時を突き合わせてください。Timestamp と出力日時を突き合わせることで，対象のグローバルバッファを使用して，どのような処理を行っていたかを確認できます。

また，手順 1.で確認したDBbuff_name と，サーバメッセージログファイル内のKFAA50027-I メッセージのグローバルバッファ名を突き合わせてください。DBbuff_name とグローバルバッファ名を突き合わせることで，対象のグローバルバッファ名を確認できます。

グローバルバッファの統計情報と関連情報の利用イメージを，次の図に示します。

図 10-10　グローバルバッファの統計情報と関連情報の利用イメージ



また，グローバルバッファの統計情報と関連情報は，次に示すチューニングを実施するときに利用できます。

- 「13.2.2　SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報による SQL 文の実行時間の短縮」
- 「13.2.3　グローバル作業表を作成する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるグローバル作業表を作成する SQL 文の実行時間の短縮」
- 「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」
- 「13.3.2　adbidxrebuild コマンドの実行時間の短縮」
- 「13.3.3　adbgetcst コマンドの実行時間の短縮」
- 「13.3.4　adbexport コマンドの実行時間の短縮」
- 「13.3.5　adbmergechunk コマンドの実行時間の短縮」
- 「13.3.6　adbarchivechunk コマンドの実行時間の短縮」

10.10.4　SQL 文の統計情報の運用方法

SQL 文の統計情報では，各 SQL 文の動作状況を確認できます。

サーバ定義に指定したオペランドの値を変更したり，実行する SQL 文を変更したり，表およびインデクスを変更したりする場合などに，SQL 文の統計情報を利用できます。

> **メモ**
>
> SQL 文の実行状況を確認する場合，adbstat コマンドで出力した SQL 文の統計情報を確認する以外に，SQL トレースファイルに出力された SQL トレース情報を確認する方法もあります。SQL トレース情報については，「10.11　SQL トレース機能の運用」を参照してください。

(1)　SQL 文の統計情報の出力と関連情報の取得

SQL 文の実行が終了したときに，SQL 文の統計情報が統計ログファイルに出力されます。adbstat コマンドを実行すると，統計ログファイルに出力された SQL 文の統計情報を表示できます。なお，エラーとなった SQL 文は，SQL 文の統計情報が出力されないことがあります。

定期的に SQL 文の統計情報を出力することを推奨します。

■SQL 文の統計情報を出力する方法

マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析）』の『使用例』の『特定の日時範囲で SQL 文の統計情報を調べる場合』を参照してください。

また，SQL 文の統計情報を出力するときに，次に示す関連情報もあわせて取得することを推奨します。

- **サーバメッセージログファイル**

  OS のcp コマンドなどでコピーして取得してください。サーバメッセージログファイルの格納先は，「10.4.2　メッセージログの参照方法（メッセージログの出力先）」を参照してください。

  > **ヒント**
  >
  > サーバ定義adb_sql_text_out オペランドにY を指定すると，HADB サーバが受け付けたすべての SQL 文がサーバメッセージログファイルに出力されます。SQL 文の統計情報とサーバメッセージログファイルを確認する際に，対象の SQL 文を特定しやすくなるため，サーバ定義adb_sql_text_out オペランドにY を指定することを検討してください。サーバ定義adb_sql_text_out オペランドは，「7.2.5　SQL 文に関するオペランド（set 形式）」を参照してください。

- **AP 識別子**

  実行中の AP やコマンドに設定された AP 識別子は，adbls -d cnct コマンドで確認してください。AP の場合，クライアント定義adb_clt_ap_name オペランドで指定された AP 識別子が出力されます。

  adbls -d cnct コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示）』を参照してください。

クライアント定義adb_clt_ap_name オペランドについては，マニュアル『HADB　AP 開発ガイド』の『AP の状態監視に関するオペランド』を参照してください。

> **メモ**
>
> 複数の AP がある場合，各 AP が使用するクライアント定義adb_clt_ap_name オペランドには，それぞれ異なる AP 識別子を指定することを推奨します。

- **AP のコネクション情報**

  AP のコネクション情報は，adbls -d cnct コマンドで確認してください。

  adbls -d cnct コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct (コネクションの状態表示)』を参照してください。

- **SQL トレースファイル**

  OS のcp コマンドなどでコピーして取得してください。SQL トレースファイルの格納先は，「10.11.1 SQL トレース機能とは」を参照してください。

> **メモ**
>
> SQL トレースファイルに SQL トレース情報を出力する方法は，「10.11.5　SQL トレース情報を出力するための準備」を参照してください。

(2)　SQL 文の統計情報と関連情報の利用方法

定期的に，SQL 文の統計情報と関連情報を取得しておくことで，各 SQL 文の動作状況を確認できます。取得した SQL 文の統計情報と関連情報は，チューニングを実施するときに利用できます。

■SQL 文の統計情報と関連情報の利用方法

1. SQL 文の統計情報を確認する

   「SQL 文の統計情報」に出力される情報のうち，次に示す情報を確認してください。

   - Timestamp（統計ログファイルの情報の出力日時）
   - AP_name（AP 識別子）
   - Connection_information（コネクション情報）
   - SQL_serial_number（コネクションが確立してからの SQL 文通番）

2. サーバメッセージログファイルと突き合わせる

   手順 1.で確認した情報と，サーバメッセージログファイル内の情報を突き合わせてください。

   - **Timestamp の場合**

     手順 1.で確認したTimestamp と，サーバメッセージログファイル内のメッセージの出力日時を突き合わせてください。Timestamp と出力日時を突き合わせることで，対象の SQL 文が，どのような処理を行っていたかを確認できます。

   - **AP_name の場合**

手順 1.で確認したAP_name と，サーバメッセージログファイル内のKFAA81000-I メッセージの AP 識別子を突き合わせてください。AP_name と AP 識別子を突き合わせることで，対象の AP やコマンドを確認できます。

- Connection_information の場合

  手順 1.で確認したConnection_information と，サーバメッセージログファイル内のKFAA81000-I メッセージのコネクション情報を突き合わせてください。Connection_information とコネクション情報を突き合わせることで，対象の AP のコネクションを確認できます。

- SQL_serial_number の場合

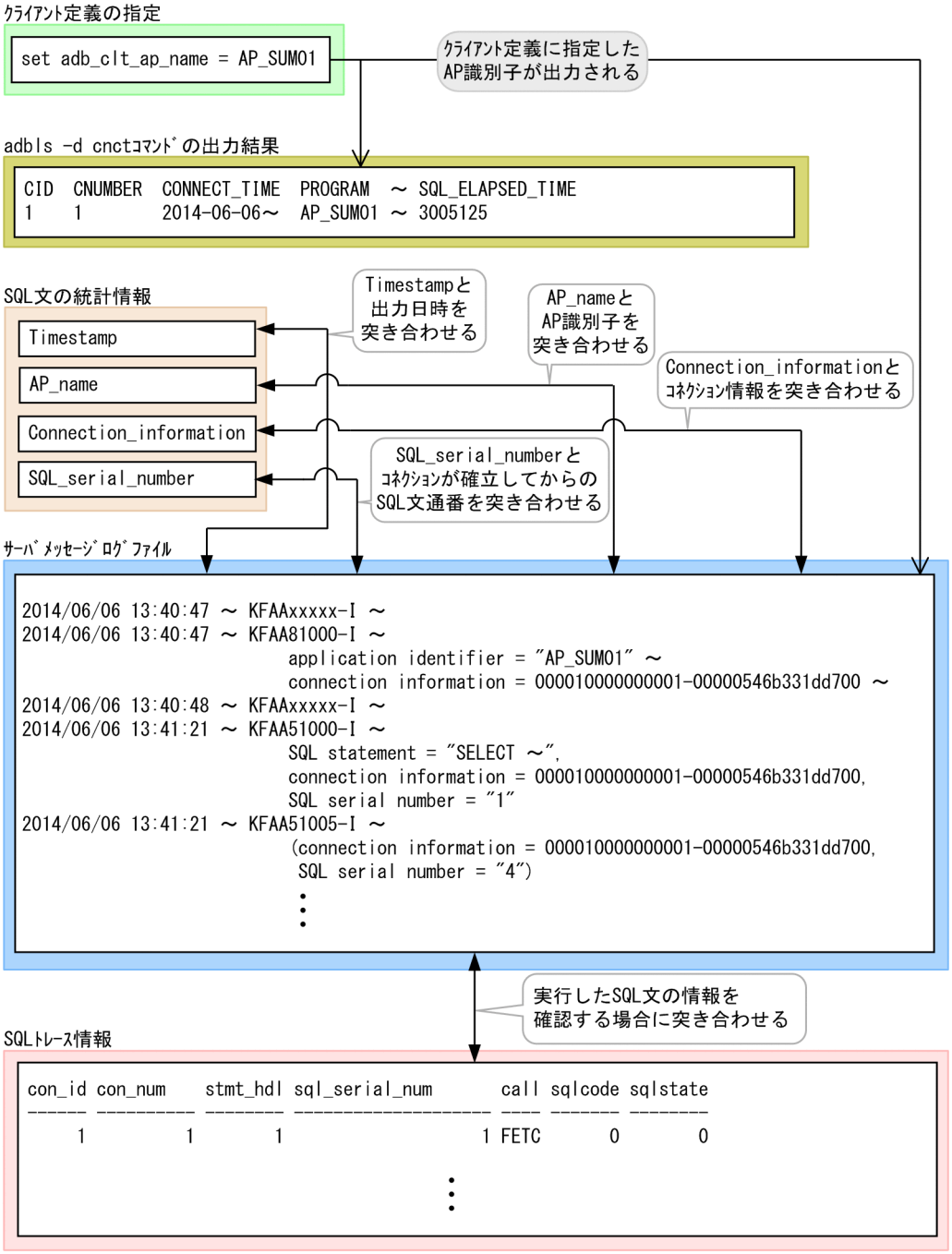
  サーバ定義adb_sql_text_out オペランドにY を指定している場合，手順 1.で確認した SQL_serial_number と，サーバメッセージログファイル内のKFAA51000-I メッセージのコネクションが確立してからの SQL 文通番を突き合わせてください。SQL_serial_number とコネクションが確立してからの SQL 文通番を突き合わせることで，対象の SQL 文を確認できます。

  また，エラーになった SQL 文の場合は，手順 1.で確認したSQL_serial_number と，サーバメッセージログファイル内のKFAA51005-I メッセージのコネクションが確立してからの SQL 文通番を突き合わせてください。

また，実行した SQL 文の情報を確認したい場合は，サーバメッセージログファイル内の情報と，SQL トレース情報を突き合わせてください。

SQL 文の統計情報と関連情報の利用イメージを，次の図に示します。

図 10-11　SQL 文の統計情報と関連情報の利用イメージ



また，SQL 文の統計情報と関連情報は，次に示すチューニングを実施するときに利用できます。

- 「13.2.2　SQL 文の実行時間を短縮する方法」の「(2)　SQL 文の統計情報による SQL 文の実行時間の短縮」
- 「13.2.3　グローバル作業表を作成する SQL 文の実行時間を短縮する方法」の「(2)　SQL 文の統計情報によるグローバル作業表を作成する SQL 文の実行時間の短縮」
- 「13.2.4　ローカル作業表を作成する SQL 文の実行時間を短縮する方法」の「(2)　SQL 文の統計情報によるローカル作業表を作成する SQL 文の実行時間の短縮」

- 「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(2)　SQL 文の統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」

10.10.5　統計ログファイルの運用

(1)　統計ログファイルに出力される情報

統計ログファイルには，次に示す HADB サーバの稼働情報が出力されます。

- SQL 文の統計情報

統計ログファイルに出力された情報は，adbstat コマンドを実行することで確認できます。

HADB サーバは，統計ログファイルを自動的に 4 つ作成して使用します。

重要

作成された統計ログファイルを手動で更新しないでください。統計ログファイルに出力される情報が不正になるおそれがあります。

1 つの統計ログファイルの最大容量は，サーバ定義adb_sta_log_max_size オペランド，および adb_sta_log_size_unit オペランドで指定できます。サーバ定義adb_sta_log_max_size オペランド，およびadb_sta_log_size_unit オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式）」の次の個所を参照してください。

- adb_sta_log_max_size オペランド
- adb_sta_log_size_unit オペランド

1 つの統計ログファイルが最大容量に達すると，HADB サーバはその統計ログファイルへの出力をやめて，次の統計ログファイルに出力します。このとき，切り替え先の統計ログファイルに格納されている古い情報は，上書きされます。そのため，古い情報が上書きされる前に，定期的にadbstat コマンドを実行することで，統計ログファイルに出力された情報を出力しておく必要があります。

メモ

- 使用される統計ログファイルは，HADB サーバを開始するときにサーバメッセージログファイルに出力されるKFAA80295-I メッセージで確認できます。
- 1 つの統計ログファイルが最大容量に達して，統計ログファイルが切り替わる場合，サーバメッセージログファイルにKFAA80296-I メッセージが出力されます。
- 統計ログファイルの切り替えテストを実施する場合，サーバ定義adb_sta_log_size_unit オペランドを指定すると，統計ログファイルの最大容量の単位を小さくできます（メガバイト単位に変更できます）。

(2) 統計情報が上書きされない期間の目安

統計ログファイル中の統計情報が上書きされない期間の目安は，次の計算式で求められます。

計算式（単位：日）

統計情報が上書きされない期間の目安＝（*log_max*×4,294,967,296÷*output_info*）÷1,440

変数の説明

- *log_max*：サーバ定義のadb_sta_log_max_size オペランドの指定値（オペランド省略時は 4）
- *output_info*：1 分当たりに出力される統計情報量の目安（単位：バイト）
  次の計算式から求めてください。

  690＋432×*adbbuff_num*＋960×*connect_num*＋4,376×*sql_num*

- *adbbuff_num*：サーバ定義のadbbuff オペランドで定義したグローバルバッファ数（HADB サーバによって自動的に定義されるグローバルバッファ数も含みます）

  メモ

  HADB サーバによって自動的に定義されるグローバルバッファについては，「7.2.11 グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドの説明を参照してください。

- *connect_num*：HADB サーバへの同時接続数
- *sql_num*：1 分当たりの SQL 文の実行回数

計算例

統計ログファイル中の統計情報が上書きされない期間の目安の計算例を示します。条件は次のとおりとします。

- adb_sta_log_max_size オペランドの指定値：4
- adbbuff オペランドで定義したグローバルバッファ数：128
- HADB サーバへの同時接続数：128
- 1 分当たりの SQL 文の実行回数：120

1 分当たりに出力される統計情報量の目安（*output_info*）
＝ 690 ＋ 432×128 ＋ 960×128 ＋ 4,376×120
＝ 703,986（バイト）

統計情報が上書きされない期間の目安
＝（4×4,294,967,296÷703,986）÷1,440
≒約 17 日

上記の条件の場合，統計情報が上書きされない期間は約 17 日となります。

(3)　統計ログファイルの出力先を変更する場合

統計ログファイルは，$ADBDIR/spool ディレクトリ下に出力されますが，サーバ定義adb_sta_log_path オペランドを指定して，統計ログファイルの出力先ディレクトリを変更できます。サーバ定義adb_sta_log_path オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式）」を参照してください。

> **重要**
>
> 統計ログファイルの出力先ディレクトリを変更した場合，変更前の統計ログファイルに格納されている統計情報を参照できなくなります。そのため，統計ログファイルの出力先ディレクトリを変更する前に，adbstat コマンドを実行して統計情報を出力しておいてください。

(4)　統計ログファイルを削除する場合

統計ログファイルを削除する手順を次に示します。

> **重要**
>
> 統計ログファイルを削除する場合は，ここで説明している手順に必ず従ってください。手順に従わないと，統計ログファイルに出力される情報が不正になるおそれがあります。

> **メモ**
>
> HADB サーバの運用テストを実施したあとに，統計ログファイルが不要になり削除することがあります。その場合は，ここで説明する手順に従って統計ログファイルを削除してください。

手順

1. adbstop コマンドで HADB サーバを正常終了する

   マルチノード機能を使用している場合は，統計ログファイルを削除するノードの HADB サーバを正常終了してください。

   コールドスタンバイ構成の場合は，コールドスタンバイ構成を正常終了してください。

2. OS のrm コマンドなどで，統計ログファイルの出力先ディレクトリ下のファイルを削除する

   次のファイルをすべて削除してください。

   - adbstatlog*XX*

     ファイル名の *XX* は通番であり，01～04 のどれかになります。

   - .adbstatlog

     隠しファイルです。

3. adbstart コマンドで HADB サーバを正常開始する

10.10.6　adbstat コマンド実行時に統計ログファイルから出力される情報の容量見積もり

統計ログファイルに出力された情報を adbstat コマンドで出力する場合に，adbstat コマンドで出力される情報の容量を見積もる方法について説明します。

SQL 文の統計情報として出力される CSV 形式のファイルサイズ（変数 *SQL_CSV_SIZE*）は，次に示す計算式から求めてください。

計算式（単位：キロバイト）

adbstat コマンドで，統計ログファイル内のすべての情報を出力すると仮定した容量を求めます。

$$SQL_CSV_SIZE = \frac{STLOG_SUM}{(0.81 + 0.22 \times (DBAREA_NUM - 1)) \times 2} \times (1.37 + 0.69 \times (DBAREA_NUM - 1))$$

変数の説明

STLOG_SUM

すべての統計ログファイルのファイルサイズの合計値（単位：キロバイト）

DBAREA_NUM

HADB サーバ全体の DB エリアの数

10.11　SQL トレース機能の運用

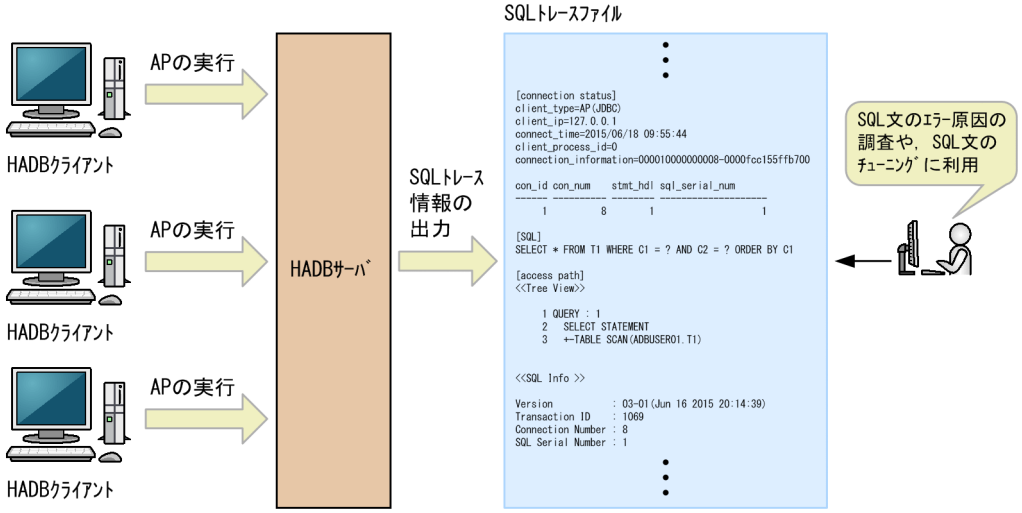
ここでは，SQL トレース機能の運用方法と，SQL トレース情報に出力される情報について説明します。

10.11.1　SQL トレース機能とは

SQL トレース機能とは，SQL 文の実行履歴情報（SQL トレース情報）をファイル（SQL トレースファイル）に出力する機能です。SQL トレースファイルには，実行された SQL 文，SQL 文の実行時間，およびアクセスパスなどの情報が出力されます。

SQL トレース機能の概要を次の図に示します。

図 10-12　SQL トレース機能の概要



[説明]

SQL トレース情報は，HADB サーバの SQL トレースファイルに出力されます。テキストエディタを使用して，SQL トレースファイルに出力された SQL トレース情報を参照します。

SQL トレースファイルを参照できるのは，HADB 管理者および HADB 管理者グループに所属する OS ユーザだけです。

出力された SQL トレース情報は，次のことに使用します。

- SQL 文のエラー原因の調査

  SQL トレース情報に出力される**SQLCODE** を確認し，**SQLCODE** に対応するメッセージから SQL 文のエラー原因を調査します。

- SQL 文のチューニング

  SQL トレース情報に出力される SQL 文の実行時間を確認し，実行時間が長い SQL 文を特定します。SQL トレース情報に出力されるコネクション情報と SQL 文通番を基に，SQL 文の統計情報を使って SQL 文のチューニングを行います。

■SQL トレース情報の出力先

- 出力先ディレクトリ名

  $ADBDIR/spool

- SQL トレースファイル名

  adbsqltrc01.log～adbsqltrc08.log

SQL トレースファイルは全部で 8 つ作成されます。1 つの SQL トレースファイルの容量が満杯になった場合，次の番号の SQL トレースファイルに出力先が順次変更されます。adbsqltrc08.log が満杯になった場合は，adbsqltrc01.log に出力先が変更されます。

メモ

adbinfoget コマンドで取得されるトラブルシュート情報に，SQL トレースファイルも含まれています。

10.11.2 SQL トレース情報に出力される情報

SQL トレース情報には，次の情報が出力されます。

- HADB サーバのバージョン情報
- SQL 文の実行情報
- クライアント定義情報
- コネクション状態に関する情報
- 実行した SQL 文とアクセスパス情報
- ？パラメタの情報
- SQL パラレル実行機能に関する情報
- トランザクションの情報
- 認可識別子の情報
- SQL 文の統計情報
- アクセスパスの統計情報
- コネクションの確立または終了の区切り線
- SQL 文の基本情報

メモ

エラーが発生して SQL 文の処理が中断した場合，その時点で取得できなかった SQL トレース情報は出力されません。

各情報の出力項目について説明します。

(1) HADB サーバのバージョン情報

HADB サーバのバージョン情報の出力例を次に示します。

```
Hitachi Advanced Data Binder 03-01 Aug  5 2015 18:32:45
```

[説明]

- 下線部分には，HADB サーバのバージョンが出力されます。
- 下線部分より後ろの部分には，HADB サーバのバージョン付加情報が次の形式で出力されます。
  MMM DD YYYY hh:*mm*:*ss*
  MMM：月が英略語の形式で出力されます。
  DD：日が出力されます。
  YYYY：西暦が出力されます。
  hh:*mm*:*ss*：時刻が出力されます。

> **メモ**
>
> HADB サーバのバージョン情報は，SQL トレースファイルの先頭に出力されます。

(2) SQL 文の実行情報

SQL 文の実行情報の出力例と出力される項目について説明します。

(a) SQL 文の実行情報の出力例

SQL 文の実行情報の出力例を次に示します。

```
con_id con_num    stmt_hdl sql_serial_num       call sqlcode sqlstate
------ ---------- -------- -------------------- ---- ------- --------
     1          1        1                    1 FETC       0 00000
start_time                 end_time                   exe_time(us)         rows
-------------------------- -------------------------- -------------------- --------------------
2015/07/31 21:48:46.736799 2015/07/31 21:48:46.737725                  926                    1
tran_id             trn_iso_lv      trn_access_mode sql_order_mode cursor_holdability       message_log_info
------------------- --------------- --------------- -------------- ----------------------- --------------------
               4342 READ_COMMITTED  READ_WRITE      ISO            CLOSE_CURSORS_AT_COMMIT  00000054070000000001
```

(b) SQL 文の実行情報に出力される項目

SQL 文の実行情報に出力される項目を次の表に示します。

表 10-10　SQL 文の実行情報に出力される項目

項番	項目名	説明
1	con_id	コネクション ID が出力されます。
2	con_num	HADB サーバの開始後に割り振られるコネクション通番が出力されます。
3	stmt_hdl	SQL 文に割り当てられた文ハンドルが出力されます。
4	sql_serial_num	コネクション確立後に割り振られる SQL 文通番が出力されます。
5	call	SQL 文の実行情報の対象となるコール種別が出力されます。出力されるコール種別と実行された処理の対応を「表 10-11　コール種別と実行された処理の対応」に示します。 なお，**コール**とは，コネクションの確立または終了，トランザクションの正常終了または取り消し，および SQL 文を使用したデータベースに対する操作の単位のことです。 SQL 文単位で SQL トレース情報を出力している場合，SQL 文の実行が完了したときに（コール単位のCLOS，EXEC，およびEXDI の実行が完了したときに），SQL 文の実行情報が出力されます。このとき，call 欄にはSQL と出力されます。
6	sqlcode	SQLCODE が出力されます。 SQLCODE とメッセージの対応付けの方法については，マニュアル『HADB　メッセージ』の『SQLCODE の見方』を参照してください。
7	sqlstate	SQLSTATE が出力されます。 SQLSTATE については，マニュアル『HADB　メッセージ』の『SQLSTATE の一覧』を参照してください。
8	start_time	処理の開始時刻が出力されます。 コール単位で SQL トレース情報を出力している場合は，各コールの開始時刻が出力されます。 SQL 文単位で SQL トレース情報を出力している場合は，「表 10-12　コール種別が SQL の場合に実行されるコール」で説明している最初のコールの開始時刻が出力されます。
9	end_time	処理の終了時刻が出力されます。 コール単位で SQL トレース情報を出力している場合は，各コールの終了時刻が出力されます。 SQL 文単位で SQL トレース情報を出力している場合は，最後に実行したコールの終了時刻が出力されます。
10	exe_time(us)	処理の実行時間がマイクロ秒単位で出力されます。 • コール単位で SQL トレース情報を出力している場合 各コールの終了時刻と開始時刻の差が出力されます。 コールの延長で暗黙的なコミットやロールバックが発生したときは，その処理時間も実行時間に含まれます。また，2 回目以降のフェッチに対応するFETC の実行時間は，同一 SQL 文の実行情報に出力される終了時刻と開始時刻の差ではなく，各フェッチの実行時間の合計になります。 • SQL 文単位で SQL トレース情報を出力している場合 各コールの実行時間の合計が出力されます。
11	rows	処理対象の行数が出力されます。 コール種別がFETC，EXEC，EXDI，またはSQL の場合に行数が出力されます。 • コール種別がFETC の場合 1 回目のフェッチのときは，0 または1 が出力されます。そのあとのフェッチのときは，2 回目から最後のフェッチの処理行数が出力されます。

項番	項目名	説明
		• コール種別がEXEC またはEXDI の場合 実行した SQL 文がDELETE，INSERT，またはUPDATE の場合は，更新した行数が出力されます。それ以外の SQL 文の場合は，0 が出力されます。 • コール種別がSQL の場合 実行した SQL 文がSELECT，DELETE，INSERT，またはUPDATE の場合は，処理対象の行数が出力されます。それ以外の SQL 文の場合は，0 が出力されます。
12	tran_id	データベースを初期化したときからの通番であるトランザクション ID が出力されます。
13	trn_iso_lv	トランザクション隔離性水準が出力されます。 • READ_COMMITTED：トランザクション隔離性水準がREAD COMMITTED の場合に出力されます。 • REPEATABLE_READ：トランザクション隔離性水準がREPEATABLE READ の場合に出力されます。
14	trn_access_mode	トランザクションアクセスモードが出力されます。 • READ_WRITE：トランザクションアクセスモードが読み書き可能モードの場合に出力されます。 • READ_ONLY：トランザクションアクセスモードが読み取り専用モードの場合に出力されます。
15	sql_order_mode	文字データの並び替え順序が出力されます。 • BYTE：文字データをバイトコード順に並び替えた場合に出力されます。 • ISO：文字データをソートコード順に並び替えた場合に出力されます。
16	cursor_holdability	カーソルの動作が出力されます。 • CLOSE_CURSORS_AT_COMMIT：コミット時にカーソルをクローズした場合に出力されます。 • HOLD_CURSORS_OVER_COMMIT：コミットしてもカーソルをクローズしなかった場合に出力されます。 次に示す場合にカーソルの動作が出力されます。 • コール種別がOPEN，FETC，またはCLOS の場合 • コール種別がSQL で，かつ実行したSQL 文がSELECT 文の場合
17	message_log_info	メッセージログ情報が出力されます。 メッセージログファイルに出力されるメッセージのクライアントプロセス ID およびコネクション通番が出力されます。 メッセージログ情報をメッセージに出力されているクライアントプロセス ID およびコネクション通番と突き合わせることで，SQL 文のエラー発生時の原因調査に利用できます。利用方法の詳細については，「10.11.7　SQL トレース情報を使用した SQL 文のエラー原因調査」を参照してください。

注

- コール種別がCNCT，DISC，またはDISP の場合，stmt_hdl，sql_serial_num，rows，tran_id，trn_iso_lv，trn_access_mode，およびsql_order_mode の値は出力されません。
- コール種別がCMIT またはRLBK の場合，stmt_hdl，sql_serial_num，およびrows の値は出力されません。
- トランザクションの決着によって解放される文ハンドル（SQL 文のエラーによってトランザクションが決着した場合は，エラーが発生した文ハンドルを除く）が，次のどれかの条件を満たすときは，call，sqlcode，sqlstate，start_time，end_time，exe_time，およびrows の値は出力されません。

・上記の文ハンドルの直前のコールがエラーとなった場合

・上記の文ハンドルの直前のコール種別がOPEN の場合（SQL トレース情報の出力単位がコール単位の場合）

・上記の文ハンドルの直前のコール種別がFETC で，かつフェッチが 1 回だけ実行された場合（SQL トレース情報の出力単位がコール単位の場合）

表 10-11　コール種別と実行された処理の対応

項番	コール種別※1（call に出力された情報）	実行された処理	出力単位※2	出力される追加の情報※3
1	PREP	SQL 文の前処理	コール単位	2，4
2	OPEN	カーソルのオープン		3，4
3	CLOS	カーソルのクローズ		4，6
4	FETC	1 回目のフェッチ処理，または 2 回目から最後のフェッチ処理※4		4
5	EXEC	SQL 文の実行※5		3，4，6
6	EXDI	SQL 文の前処理および実行		2，4，6
7	CMIT	コミット処理	コール単位または SQL 文単位	4
8	RLBK	ロールバック処理		4
9	CNCT	コネクションの確立（AP の接続またはコマンドの接続）		1，4，5，7
10	DISC	コネクションの終了		4，5，7
11	DISP	コネクションプール機能を使用した際のコネクションの終了		4
12	SQL	SQL 文単位での出力※6，※7		2，3，4，6

注※1

コール種別と実行した JDBC のメソッドの対応については，「(d)　実行した JDBC のメソッドと出力されるコール種別の対応」を参照してください。

コール種別と実行した ODBC 関数の対応については，「(e)　実行した ODBC 関数と出力されるコール種別の対応」を参照してください。

コール種別と実行した CLI 関数の対応については，「(f)　実行した CLI 関数と出力されるコール種別の対応」を参照してください。

注※2

サーバ定義のadb_sql_trc_level オペランド，またはadbchgsqltrc コマンドで指定した SQL トレース情報を出力する単位を意味しています。

注※3

SQL 文の実行情報以外に出力される情報です。表中の番号は，次の番号と対応しています。

1：クライアント定義情報およびコネクション状態に関する情報

2：実行した SQL 文およびアクセスパス情報

3：？パラメタの情報

4：トランザクションの情報

5：認可識別子の情報

6：SQL 文の統計情報およびアクセスパスの統計情報

7：コネクションの確立または終了の区切り線

注※4

コール種別がFETC の場合，フェッチの回数分の SQL トレース情報が出力されるわけではありません。フェッチの回数が 2 回以上の場合は，フェッチを繰り返し実行しても，SQL 文の実行情報は 2 つ（1 回目のフェッチ処理の SQL 文の実行情報と，それ以降の全フェッチの実行結果をまとめた SQL 文の実行情報）しか出力されません。フェッチの回数が 1 回の場合は，SQL 文の実行情報は 1 つだけ出力されます。

注※5

？パラメタの値の一括転送を実施した場合，EXEC の SQL 文の実行情報は 1 つだけ出力されます。？パラメタの値の一括転送については，マニュアル『HADB　AP 開発ガイド』の『？パラメタの値の一括転送』を参照してください。

注※6

SQL 記述ファイルを指定してadbexport コマンドを実行した場合，出力レベルがCALL でも，1 つの SQL 文の実行情報にまとめられて出力されます。その際，コール種別にはSQL が出力されます。

注※7

SQL 文単位で SQL トレース情報を出力する場合，「表 10-12　コール種別が SQL の場合に実行されるコール」に示す『実行されるコール』に記載されている各コールの SQL 文の実行情報が，1 つの SQL 文の実行情報にまとめられて出力されます。その際，コール種別にはSQL が出力されます。

表 10-12　コール種別が SQL の場合に実行されるコール

項番	SQL 文の種類	実行されるコール※1
1	更新系 SQL	PREP → EXEC
2		EXEC（項番 1 のコールが実行されたあとに，連続してEXEC が実行される場合）
3		EXDI
4	定義系 SQL	PREP → EXEC
5		EXDI
6	検索系 SQL	PREP → OPEN → FETC → 〔FETC〕 → CLOS（明示的にカーソルをクローズした場合）
7		PREP → OPEN → FETC → 〔FETC〕（明示的にカーソルをクローズしない場合※2）

注

〔　〕で囲まれたコール種別は実行されないことがあります。

注※1

SQL 文の実行時にエラーが発生した場合，「(c)　SQL 文の実行時にエラーが発生した場合の SQL 文の実行情報の出力（コール種別が SQL の場合）」に示すとおりに，SQL 文の実行情報が出力されます。

注※2

カーソルをクローズしないで文ハンドルを解放したり，トランザクションを決着したりした場合が該当します。ただし，「表 10-10　SQL 文の実行情報に出力される項目」で説明している「cursor_holdability（カーソル動作）」がHOLD_CURSORS_OVER_COMMIT の場合は，コミットしてもカーソルがクローズしないため，該当しません。

(c)　SQL 文の実行時にエラーが発生した場合の SQL 文の実行情報の出力（コール種別が SQL の場合）

SQL 文の実行時にエラーが発生した場合，次に示すコールの SQL 文の実行情報が，1 つにまとめて出力されます。

- エラーとなったコール
- 「表 10-12　コール種別が SQL の場合に実行されるコール」で示すコールのうちエラーとなったコールの前に実行されたすべてのコール

ただし，エラーが発生したコール種別がFETC だけの場合は，そのエラーの延長で暗黙的なロールバックが発生したかどうかで処理が異なります。

- エラー時に暗黙的なロールバックが発生した場合

  エラーとなったFETC と，「表 10-12　コール種別が SQL の場合に実行されるコール」で示すコールのうちエラーとなったFETC の前に実行されたコールの SQL 文の実行情報が 1 つにまとめて出力されます。
- 暗黙的なロールバックが発生しなかった場合

  FETC がエラーとなっても，検索系 SQL の実行が完了したとは見なされません。そのあと，CLOS の実行が完了した時点で，検索系 SQL の実行が完了したと見なされます。その際，「表 10-12　コール種別が SQL の場合に実行されるコール」の「検索系 SQL」の各コールの SQL 文の実行情報が 1 つにまとめて出力されます。

SQL 文がエラーになった場合の例を次に示します。

- 検索系 SQL の実行時にOPEN でエラーになった場合

  PREP とOPEN の SQL 文の実行情報を 1 つにまとめて，SQL 文単位の SQL 文の実行情報として出力されます。
- 検索系 SQL の実行時にFETC でエラーになり，暗黙的なロールバックが発生した場合

  PREP，OPEN，〔FETC〕，FETC の SQL 文の実行情報を 1 つにまとめて，SQL 文単位の SQL 文の実行情報として出力されます。
- 更新系 SQL の実行時にPREP でエラーになった場合

  PREP の SQL 文の実行情報が SQL 文単位の SQL 文の実行情報として出力されます。

(d) 実行した JDBC のメソッドと出力されるコール種別の対応

実行した JDBC のメソッドと出力されるコール種別の対応を次の表に示します。

表 10-13 実行した JDBC のメソッドと出力されるコール種別の対応

項番	実行した JDBC のメソッド	出力されるコール種別
1	Driver インタフェースのconnect(String url, Properties info)	CNCT
2	Connection インタフェースのclose()	CMIT, DISC, DISP
3	Connection インタフェースのcommit()	CMIT
4	Connection インタフェースのprepareStatement(String sql)	PREP
5	Connection インタフェースのprepareStatement(String sql, int resultSetType, int resultSetConcurrency)	
6	Connection インタフェースのprepareStatement(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)	
7	Connection インタフェースのrollback()	RLBK
8	Statement インタフェースのclose()	CLOS, CMIT
9	Statement インタフェースのexecute(String sql)	PREP, OPEN, EXEC, EXDI, CLOS, CMIT, RLBK
10	Statement インタフェースのexecuteBatch()またはexecuteLargeBatch()	PREP, EXEC, EXDI, CMIT, RLBK
11	Statement インタフェースのexecuteQuery(String sql)	PREP, OPEN, CLOS, CMIT
12	Statement インタフェースのexecuteUpdate(String sql)または executeLargeUpdate(String sql)	PREP, EXEC, EXDI, CMIT, RLBK
13	Statement インタフェースのgetMoreResults()	CLOS
14	PreparedStatement インタフェースのexecute()	PREP, OPEN, EXEC, CLOS, CMIT, RLBK
15	PreparedStatement インタフェースのexecuteQuery()	PREP, OPEN, CLOS, CMIT, RLBK
16	PreparedStatement インタフェースのexecuteUpdate()または executeLargeUpdate()	PREP, EXEC, CMIT, RLBK
17	ResultSet インタフェースのabsolute(int row)	FETC
18	ResultSet インタフェースのafterLast()	
19	ResultSet インタフェースのbeforeFirst()	
20	ResultSet インタフェースのfirst()	
21	ResultSet インタフェースのisAfterLast()	
22	ResultSet インタフェースのisBeforeFirst()	
23	ResultSet インタフェースのilast()	
24	ResultSet インタフェースのirelative(int rows)	

項番	実行した JDBC のメソッド	出力されるコール種別
25	ResultSet インタフェースのisLast()	
26	ResultSet インタフェースのnext()	
27	ResultSet インタフェースのclose()	CLOS, CMIT
28	DatabaseMetaData インタフェースのgetColumns(String catalog, String schemaPattern, String tableNamePattern, String columnNamePattern)	PREP, OPEN
29	DatabaseMetaData インタフェースのgetCrossReference(String primaryCatalog, String primarySchema, String primaryTable, String foreignCatalog, String foreignSchema, String foreignTable)	
30	DatabaseMetaData インタフェースのgetExportedKeys(String catalog, String schema, String table)	
31	DatabaseMetaData インタフェースのgetImportedKeys(String catalog, String schema, String table)	
32	DatabaseMetaData インタフェースのgetIndexInfo(String catalog, String schema, String table, boolean unique, boolean approximate)	
33	DatabaseMetaData インタフェースのgetPrimaryKeys(String catalog, String schema, String table)	
34	DatabaseMetaData インタフェースのgetSchemas(String catalog, String schemaPattern)	
35	DatabaseMetaData インタフェースのgetTablePrivileges(String catalog, String schemaPattern, String tableNamePattern)	
36	DatabaseMetaData インタフェースのgetTables(String catalog, String schemaPattern, String tableNamePattern, String[] types)	
37	DatabaseMetaData インタフェースのgetSchemas()	
38	DataSource インタフェースのgetConnection()	CNCT
39	DataSource インタフェースのgetConnection(String username, String password)	
40	PooledConnection インタフェースのclose()	DISC
41	PooledConnection インタフェースのgetConnection()	CNCT

(e) 実行した ODBC 関数と出力されるコール種別の対応

実行した ODBC 関数と出力されるコール種別の対応を次の表に示します。

表 10-14 実行した ODBC 関数と出力されるコール種別の対応

項番	実行した ODBC 関数	出力されるコール種別
1	SQLPrepare(W)	PREP
2	SQLExecute	PREP, EXEC, OPEN, CMIT

項番	実行した ODBC 関数	出力されるコール種別
3	SQLExecDirect(W)	PREP, EXEC, CMIT
4	SQLParamData	EXEC, OPEN, CMIT
5	SQLFetch	FETC
6	SQLColumns(W)	PREP, OPEN
7	SQLForeignKeys(W)	
8	SQLPrimaryKeys(W)	
9	SQLStatistics(W)	
10	SQLTablePrivileges(W)	
11	SQLTables(W)	
12	SQLFreeStmt	CLOS, CMIT
13	SQLCloseCursor	
14	SQLEndTran	CMIT, RLBK
15	SQLDisconnect	CMIT, DISC
16	SQLFreeHandle	CMIT
17	SQLSetConnectAttr(W)	
18	SQLConnect(W)	CNCT, DISC
19	SQLDriverConnect(W)	
20	SQLBrowseConnect(W)	

(f)　実行した CLI 関数と出力されるコール種別の対応

実行した CLI 関数と出力されるコール種別の対応を次の表に示します。

表 10-15　実行した CLI 関数と出力されるコール種別の対応

項番	実行した CLI 関数	出力されるコール種別
1	a_rdb_SQLPrepare()	PREP
2	a_rdb_SQLExecute()	OPEN, EXEC
3	a_rdb_SQLCloseCursor()	CLOS
4	a_rdb_SQLFetch()	FETC
5	a_rdb_SQLExecDirect()	EXDI
6	a_rdb_SQLEndTran()	CMIT, RLBK
7	a_rdb_SQLConnect()	CNCT
8	a_rdb_SQLDisconnect()	DISC

(3) クライアント定義情報

クライアント定義情報の出力例を次に示します。

■クライアント定義情報の出力例

```
[client definitions]
adb_clt_ap_name="SAMPLE"
adb_clt_fetch_size=1024
adb_clt_group_name="test_group"
adb_clt_sql_order_mode="ISO"
adb_clt_sql_parallel_exec="N"
adb_clt_sql_text_out="Y"
adb_clt_trn_access_mode="READ_WRITE"
adb_clt_trn_iso_lv="READ_COMMITTED"
adb_dbbuff_wrktbl_clt_blk_num=256
adb_sql_exe_hashflt_area_size=200
adb_sql_exe_hashgrp_area_size=4800
adb_sql_exe_hashtbl_area_size=2000
adb_sql_exe_max_rthd_num=0
adb_sql_prep_dec_div_rs_prior="FRACTIONAL_PART"
adb_sql_prep_delrsvd_use_srvdef="Y"
```

AP の接続の場合は，その AP に適用されているクライアント定義の情報が出力されます。コマンドの接続の場合は，クライアント定義情報は出力されません。

なお，クライアント定義情報に出力される各オペランドの値は，クライアント定義に指定した値ではなく，AP に適用された値が出力されます。例えば，クライアント定義のadb_sql_exe_max_rthd_num オペランドの指定値が不正なため，HADB サーバが仮定する値に変更されている場合，クライアント定義の指定値ではなく，HADB サーバが仮定した値が出力されます。

メモ

次に示す場合にクライアント定義情報が出力されます。

- コール種別がCNCT で，かつCNCT が正常終了したとき
- adbchgsqltrc -s コマンドを実行して，SQL トレース情報を出力したあとの最初のトランザクションを開始したとき（トランザクションを開始したコネクションのクライアント定義情報が出力されます）

(4) コネクション状態に関する情報

コネクション状態に関する情報の出力例を次に示します。

■コネクション状態に関する情報の出力例（AP の接続の場合）

```
[connection status]
client_type=AP(JDBC)
client_ip=127.0.0.1
```

```
connect_time=2015/07/31 21:54:41
client_process_id=0
connection_information=000010000000001-0000181584fb6700
```

■コネクション状態に関する情報の出力例（コマンドの接続の場合）

```
[connection status]
client_type=command(adbimport)
client_ip=127.0.0.1
connect_time=2015/07/31 16:10:37
client_process_id=30422
connection_information=000010000000003-000073b601e46700
```

コネクション状態に関する情報に出力される項目を次の表に示します。

表 10-16　コネクション状態に関する情報に出力される項目

項番	項目名	説明
1	client_type	HADB クライアントの種類が出力されます。 • JDBC ドライバからのアクセスの場合：AP(JDBC) • JDBC ドライバ以外からのアクセスの場合：AP(C library) • コマンドの場合：command(*コマンド名*) （例）command(adbimport)
2	client_ip	HADB クライアントの IP アドレスが出力されます。
3	connect_time	コネクションを確立した時刻が出力されます。
4	client_process_id	HADB クライアントのプロセス ID が出力されます。 JDBC ドライバを使用している場合は0 が出力されます。
5	connection_information	コネクション情報（KFAA81000-I メッセージのconnection information に出力される情報）が出力されます。

メモ

次に示す場合にコネクション状態に関する情報が出力されます。

- コール種別がCNCT で，かつCNCT が正常終了したとき
- adbchgsqltrc -s コマンドを実行して，SQL トレース情報を出力したあとの最初のトランザクションを開始したとき（トランザクションを開始したコネクションのコネクション状態に関する情報が出力されます）

(5)　実行した SQL 文とアクセスパス情報

実行した SQL 文とアクセスパス情報の出力例を次に示します。

■実行した SQL 文とアクセスパス情報の出力例

```
[SQL]
SELECT * FROM "T1" WHERE "C1"=? AND "C2"=? AND "C3"=?

[access path]
<<Tree View>>

     1 QUERY : 1
     2   SELECT STATEMENT
     3   +-TABLE SCAN(ADBUSER01.T1)


<<SQL Info >>

Version           : 03-01(Jul 31 2015 21:31:44)
Transaction ID    : 4342
Connection Number : 1
SQL Serial Number : 1
```

[説明]

- [SQL]

  実行したSQL文が出力されます。

  ALTER USER文またはCREATE USER文のパスワード部分については，*が出力されます。

  SQL文中に印字できない特殊文字が含まれている場合でも，そのまま出力されます。

- [access path]

  アクセスパス情報が出力されます。アクセスパス情報に出力される項目については，マニュアル『HADB　AP開発ガイド』の『アクセスパスの見方（SQL文の実行計画の見方)』を参照してください。

メモ

次に示す場合に，実行したSQL文とアクセスパス情報が出力されます。

- コール種別がPREPの場合※
- コール種別がEXDIの場合
- コール種別がSQLの場合

注※

アクセスパス情報は，PREPが正常終了したときに出力されます。PREPが正常終了しなかったときは，実行したSQL文だけが出力されます。

(6)　?パラメタの情報

?パラメタの情報の出力例を次に示します。

■?パラメタの情報の出力例

```
[param]
param_no type       len1  len2  data
-------- ---------- ----- ----- ----
       1 int            8       1
       2 char           2       'ab'
       3 dec            7     3 1234.567
```

？パラメタの情報に出力される項目を次の表に示します。

表 10-17　？パラメタの情報に出力される項目

項番	項目名	説明
1	param_no	SQL 文中の？パラメタの通番が出力されます。
2	type	？パラメタのデータ型が出力されます。
3	len1	パラメタ長が出力されます。
4	len2	パラメタ長属性が出力されます。パラメタ長属性がない場合は出力されません。 パラメタ長およびパラメタ長属性については，マニュアル『HADB　AP 開発ガイド』の『a_rdb_SQLDescribeParams()（？パラメタの情報取得)』を参照してください。
5	data	？パラメタにバインドされたデータが出力されます。

？パラメタの情報の出力形式を次の表に示します。

表 10-18　？パラメタの情報の出力形式

項番	データ型	type	len1	len2	出力形式
1	INTEGER	int	8	なし	10 進数形式
2	DECIMAL(*m*, *n*)※1	dec	*m*	*n*	10 進数形式
3	NUMERIC(*m*, *n*)※1				
4	SMALLINT	smallint	4	なし	10 進数形式
5	DOUBLE PRECISION	double	8	なし	*仮数*E*指数* 仮数および指数は 10 進数形式で出力されます。
6	FLOAT				
7	CHARACTER(*n*)	char	*n*	なし	'*文字列形式*'
8	VARCHAR(*n*)	varchar	*n*	なし	'*文字列形式*' 実長 0 の場合は，'' と出力されます。
9	DATE	date	4	なし	日付データの既定の出力表現※2 で出力されます。
10	TIME(*p*)※3	time	3 ＋ ↑*p*÷2↑	*p*	時刻データの既定の出力表現※2 で出力されます。

項番	データ型	type	len1	len2	出力形式
11	TIMESTAMP(*p*)※3	timestamp	7 ＋ ↑*p*÷2↑	*p*	時刻印データの既定の出力表現※2 で出力されます。
12	BINARY(*n*)	binary	*n*	なし	X'*16 進数値*'
13	VARBINARY(*n*)	varbinary	*n*	なし	X'*16 進数値*' 実長 0 の場合は，X'' と出力されます。
14	ROW	row	行長	なし	X'*16 進数値*'

（凡例）

m，*n*，*p*：正の整数

注※1

データ型がDECIMAL 型またはNUMERIC 型の場合，len1 として精度を取得します。また，len2 として位取りを取得します。

注※2

既定の出力表現については，マニュアル『HADB　SQL リファレンス』の『既定の文字列表現』を参照してください。

注※3

データ型がTIME 型またはTIMESTAMP 型の場合，len1 としてデータ長を取得します。また，len2 として小数秒の桁数を取得します。

■？パラメタの情報の出力規則

- バインドされたデータがナル値の場合は，NULL が出力されます。
- ？パラメタに印字できない特殊文字が含まれていても，そのまま出力されます。
- NULL 述語に？パラメタを単独で指定した場合，ナル値以外のデータをバインドしてもdata 欄には何も出力されません。ナル値がバインドされた場合は，data 欄にNULL が出力されます。
- ？パラメタの値の一括転送を実施した場合，？パラメタセットごとにヘッダ（param_no，type，len1，len2，data，および区切り線）が出力されて，データも出力されます。ただし，[param]は，1 回の一括転送で 1 回だけ出力されます。

  ？パラメタの値の一括転送については，マニュアル『HADB　AP 開発ガイド』の『？パラメタの値の一括転送』を参照してください。

メモ

次に示す条件をすべて満たす場合に？パラメタの情報が出力されます。

- SQL 文中に？パラメタが指定されている場合
- コール種別がOPEN，EXEC，またはSQL の場合

(7) SQL パラレル実行機能に関する情報

SQL パラレル実行機能に関する情報の出力例を次に示します。

■SQL パラレル実行機能に関する情報の出力例

```
[parallel]
node_no sql_node_type
------- -------------
      3 MAIN
      1 SUB
      4 SUB
      6 SUB
```

SQL パラレル実行機能に関する情報に出力される項目を次の表に示します。

表 10-19 SQL パラレル実行機能に関する情報に出力される項目

項番	項目名	説明
1	node_no	SQL パラレル実行機能を適用した検索系 SQL が実行されたノードのノード番号が出力されます。
2	sql_node_type	SQL パラレル実行機能を適用した検索系 SQL が実行されたノードの SQL ノード種別が出力されます。 • MAIN：SQL メインノード • SUB：SQL サブノード

> **メモ**
>
> SQL 文の実行時に SQL パラレル実行機能が適用され，かつ次に示すどちらかの条件を満たす場合に，SQL パラレル実行機能に関する情報が出力されます。
>
> - コール種別がOPEN で，かつOPEN が正常終了した場合
> - コール種別がSQL の場合

(8) トランザクションの情報

トランザクションの情報の出力例を次に示します。

■トランザクションの情報の出力例

```
tran_id              start_time                 end_time                   exe_time(us)
-------------------- -------------------------- -------------------------- --------------------
                4342 2015/07/31 21:48:40.537485 2015/07/31 21:48:49.711265              9173780
trn_iso_lv      trn_access_mode sql_order_mode
--------------- --------------- --------------
READ_COMMITTED  READ_WRITE      ISO
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

トランザクションの情報に出力される項目を次の表に示します。

表 10-20　トランザクションの情報に出力される項目

項番	項目名	説明
1	tran_id	データベースを初期化したときからの通番であるトランザクション ID が出力されます。
2	start_time	トランザクションの開始時刻が出力されます。
3	end_time	トランザクションの終了時刻が出力されます。
4	exe_time(us)	トランザクションの実行時間がマイクロ秒単位で出力されます。トランザクションの終了時刻と開始時刻の差が出力されます。
5	trn_iso_lv	トランザクション隔離性水準が出力されます。 • READ_COMMITTED：トランザクション隔離性水準がREAD COMMITTED の場合に出力されます。 • REPEATABLE_READ：トランザクション隔離性水準がREPEATABLE READ の場合に出力されます。
6	trn_access_mode	トランザクションアクセスモードが出力されます。 • READ_WRITE：トランザクションアクセスモードが読み書き可能モードの場合に出力されます。 • READ_ONLY：トランザクションアクセスモードが読み取り専用モードの場合に出力されます。
7	sql_order_mode	文字データの並び替え順序が出力されます。 • BYTE：文字データをバイトコード順に並び替えた場合に出力されます。 • ISO：文字データをソートコード順に並び替えた場合に出力されます。
8	~	トランザクション情報の区切り線として，「~」が出力されます。

メモ

次に示す場合にトランザクションの情報が出力されます。

- コール種別がCMIT またはRLBK の場合
- 暗黙的なトランザクションの決着が発生した場合※

注※

定義系 SQL，TRUNCATE TABLE 文，またはPURGE CHUNK 文の実行が正常終了したときに暗黙的なコミットが実行されます。

また，SQL 文のエラーが発生したときに，暗黙的なロールバックが実行されることがあります。その場合，メッセージログファイルにKFAA51001-E メッセージが出力されます。

トランザクション実行中にコネクションを終了した場合も，暗黙的なコミットが実行されます。

(9)　認可識別子の情報

認可識別子の情報の出力例を次に示します。

■認可識別子の情報の出力例

コール種別がCNCT の場合

```
ADBUSER01 has been connected.
```

コール種別がDISC の場合

```
ADBUSER01 has been disconnected.
```

[説明]

下線部分には，AP のコネクション確立時に指定した認可識別子が出力されます。

コマンドのコネクションの場合は，コマンドの-u オプションに指定した認可識別子が出力されます。HADB サーバに接続するコマンドであっても，-u オプションを指定しないコマンドの場合は，認可識別子の情報は出力されません。

メモ

次に示す場合に認可識別子の情報が出力されます。

- コール種別がCNCT で，かつCNCT が正常終了した場合
- コール種別がDISC で，かつDISC が正常終了した場合

(10) SQL 文の統計情報

SQL 文の統計情報の出力例を次に示します。情報量が多いため，2 つに分けています。

■SQL 文の統計情報の出力例（その 1）

```
[SQL statistics]
Type HADB_system_version Timestamp                  AP_name                        Connection_information
---- ------------------- ------------------------- ------------------------------ ------------------------------
SQL  03-01               2015-07-31 21:48:46.738029 SAMPLE                         000010000000001-000014f70d8e1700
SQL  03-01               2015-07-31 21:48:46.738029 SAMPLE                         000010000000001-000014f70d8e1700
SQL  03-01               2015-07-31 21:48:46.738029 SAMPLE                         000010000000001-000014f70d8e1700
Connect_time               SQL_serial_number    SQL_type       SQL_total_time       Fetch_row_cnt        Update_row_cnt
-------------------------- -------------------- -------------- -------------------- -------------------- --------------------
2015-07-31 21:48:37.590351                    1 SELECT                         1007                    1                    0
2015-07-31 21:48:37.590351                    1 SELECT
2015-07-31 21:48:37.590351                    1 SELECT
Hashgrp_area_max_size Hashgrp_area_get_cnt Hashtbl_area_max_size Lock_dbarea_request_cnt Lock_dbarea_wait_cnt Lock_dbarea_wait_time
--------------------- --------------------- --------------------- ------------------------ --------------------- ---------------------
                    0                     0                     0                        0                     0                     0


Lock_table_request_cnt Lock_table_wait_cnt  Lock_table_wait_time DBbuff_dbarea_information_num DBbuff_information_num
---------------------- -------------------- -------------------- ------------------------------ ----------------------
                    13                    0                    0                              1                      1
                                                                                              2                      1
                                                                                              3                      2
DBbuff_dbarea_name             DBbuff_name                                DBbuff_page_request_cnt DBbuff_page_hit_cnt
------------------------------ ------------------------------------------ ----------------------- --------------------
ADBDIC                         ##ADBOTHER#0000004096                                           38                   23
ADBSTBL                        ##ADBOTHER#0000004096                                            8                    4
ADBUTBL01                      TBLBUF01                                                         5                    3
DBbuff_page_hit_rate DBbuff_page_put_cnt  DBbuff_page_read_cnt DBbuff_page_write_cnt DBbuff_page_rng_request_cnt
-------------------- -------------------- -------------------- -------------------- ---------------------------
                  61                    0                   15                    0                           0
                  50                    0                    4                    0                           0
                  60                    0                    2                    0                           0
DBbuff_page_rng_hit_cnt DBbuff_page_rng_hit_rate DBbuff_page_rng_put_cnt DBbuff_page_rng_read_cnt DBbuff_page_rng_write_cnt
----------------------- ------------------------ ------------------------ ------------------------ -------------------------
                      0                        0                        0                        0                         0
                      0                        0                        0                        0                         0
                      0                        0                        0                        0                         0
DBbuff_tblscan_request_cnt DBbuff_tblscan_hit_cnt DBbuff_tblscan_hit_rate DBbuff_tblscan_read_cnt DBbuff_tblscan_failed_cnt
-------------------------- ---------------------- ------------------------ ------------------------ -------------------------
                         0                      0                        0                        0                         0
                         0                      0                        0                        0                         0
                         0                      0                        0                        0                         0
DBbuff_tblscan_insufficient_buff_num DBbuff_tblscan_reset_cnt DBbuff_wrktbl_clt_request_cnt DBbuff_wrktbl_clt_hit_cnt
------------------------------------ ------------------------ ------------------------------ ------------------------
                                   0                        0                              0                        0
                                   0                        0
                                   0                        0
DBbuff_wrktbl_clt_hit_rate DBbuff_wrktbl_clt_put_cnt DBbuff_wrktbl_clt_read_cnt DBbuff_wrktbl_clt_write_cnt DBbuff_wrktbl_clt_tbl_cnt
-------------------------- ------------------------- ------------------------- -------------------------- -------------------------
                         0                         0                         0                          0                         0


DBbuff_wrktbl_clt_sort_merge_cnt DBbuff_wrktbl_clt_sort_page_max_cnt Log_usrbuf_out_cnt   Log_usrfile_max_size DBarea_extension_cnt
-------------------------------- ---------------------------------- -------------------- -------------------- --------------------
                               0                                  0                    0                    0                    0


Log_usrfile_write_cnt DBbuff_wrktbl_clt_request_fix_cnt DBbuff_wrktbl_clt_pagein_fix_cnt DBbuff_wrktbl_clt_pageout_fix_cnt
--------------------- ---------------------------------- ---------------------------------- -----------------------------------
                    0                                  0                                  0                                   0


DBbuff_wrktbl_clt_max_used_blk_num Directory_send_num   Directory_recv_num   Log_send_num         Log_recv_num
---------------------------------- -------------------- -------------------- -------------------- --------------------
                                 0                    0                    0                    0                    0
```

■SQL 文の統計情報の出力例（その 2）

```
DBbuff_send_num      DBbuff_recv_num      Node_com_num         Node_com_time        Bidx_page_split_cnt  Tidx_page_split_cnt
-------------------- -------------------- -------------------- -------------------- -------------------- --------------------
                   0                    0                    0                    0                    0                    0

Bidx_validation_check_cnt Ridx_sgmt_skip_cnt   Ridx_chunk_skip_cnt  Ridx_chunk_read_cnt  Ridx_chunk_judge_cnt Ridx_sgmt_read_cnt
------------------------- -------------------- -------------------- -------------------- -------------------- --------------------
                       21                    0                    0                    0                    0                    0

Ridx_sgmt_judge_cnt  SegmentRel_rthd_max_num Csvread_file_cnt     Csvread_file_read_size Hashgrp_area_shortage
-------------------- ------------------------ -------------------- ---------------------- --------------------
                   0                        0                    0                      0 N

Hashgrp_area_sufficient_size Hashtbl_area_shortage Syndict_file_access_time Auditread_file_cnt   Auditread_file_read_size
---------------------------- -------------------- ------------------------ -------------------- ------------------------
                           0 N                                            0                    0                        0

DBbuff_tblscan_read_size DBbuff_tblscan_rthd_min_size DBbuff_tblscan_rthd_max_size DBbuff_tblscan_use_size
------------------------ ---------------------------- ---------------------------- ------------------------
                       0                            0                            0                        0
                       0                            0                            0                        0
                       0                            0                            0                        0
DBbuff_tblscan_insufficient_buff_size Hashflt_disabled Tbldef_req_cnt       Tbldef_access_cnt    Tbldef_cache_access_cnt
------------------------------------- ---------------- -------------------- -------------------- ----------------------
                                    0 N                                  1                    1                      0
                                    0
                                    0
Tbldef_cache_register_cnt Tbldef_cache_sweep_cnt DBbuff_page_wait_cnt Max_sql_rthd_num     Hashtbl_area_size    Hashflt_area_size
------------------------- ------------------------ -------------------- -------------------- -------------------- --------------------
                        1                        0                    0                    4                 2000                  200
                                                                      0
                                                                      0
Parallel_com_send_num Parallel_com_time_sum Parallel_com_time_max Parallel_com_recv_num
--------------------- --------------------- --------------------- --------------------
                    0                     0                     0                    0
```

[説明]

SQL 文の統計情報に出力される項目は，adbstat コマンドの SQL 文の統計情報に出力される項目と同じです。出力される項目については，マニュアル『HADB　コマンドリファレンス』の『SQL 文の統計情報で出力される項目』を参照してください。

メモ

次に示す場合に SQL 文の統計情報が出力されます。

- コール種別がCLOS の場合（カーソルをクローズしないで文ハンドルを解放したり，トランザクションを決着したりした場合を含む）
- コール種別がEXEC，EXDI，およびSQL の場合

(11)　アクセスパスの統計情報

アクセスパスの統計情報の出力例および出力項目については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」を参照してください。

(12)　コネクションの確立または終了の区切り線

コネクションの確立または終了の区切り線として，「*」を 255 個出力します。

■コネクションの確立の区切り線の出力例（コール種別がCNCT の場合）

```
********************************************************************************************************************************
********************************************************************************************************************************
******************************
con_id con_num    stmt_hdl sql_serial_num       call sqlcode sqlstate
------ ---------- -------- -------------------- ---- ------- --------
     1          1                              CNCT       0 00000
start_time                end_time                  exe_time(us)         rows
------------------------- ------------------------- -------------------- --------------------
2015/07/31 21:48:37.501670 2015/07/31 21:48:37.612007               110337
tran_id              trn_iso_lv      trn_access_mode sql_order_mode cursor_holdability       message_log_info
-------------------- --------------- --------------- -------------- ------------------------ --------------------
                                                                                              00000054070000000001
```

■コネクションの終了の区切り線の出力例（コール種別がDISC の場合）

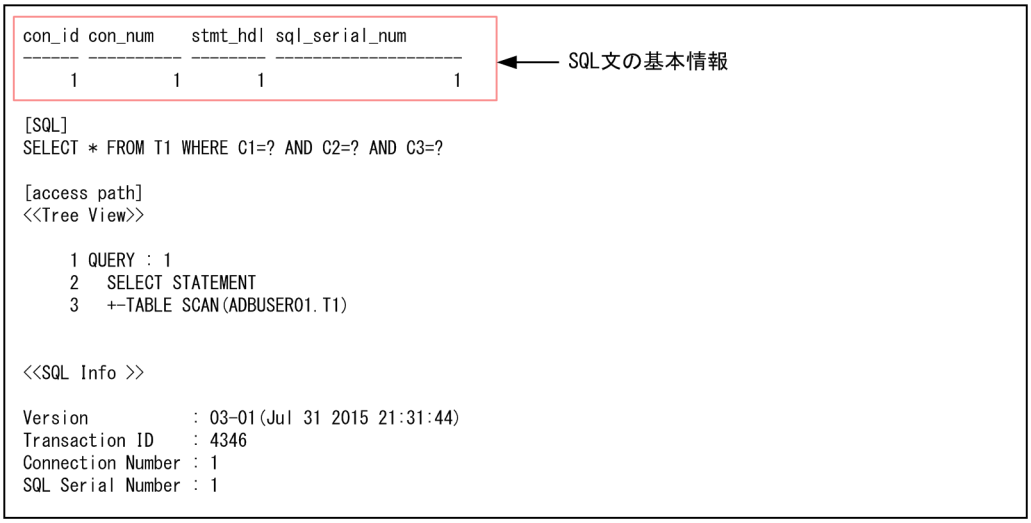
```
con_id con_num    stmt_hdl sql_serial_num       call sqlcode sqlstate
------ ---------- -------- -------------------- ---- ------- --------
     1          1                              DISC       0 00000
start_time                end_time                  exe_time(us)         rows
------------------------- ------------------------- -------------------- --------------------
2015/07/31 21:48:49.688406 2015/07/31 21:48:49.713685                25279
tran_id              trn_iso_lv      trn_access_mode sql_order_mode cursor_holdability       message_log_info
-------------------- --------------- --------------- -------------- ------------------------ --------------------
                                                                                              00000054070000000001

********************************************************************************************************************************
********************************************************************************************************************************
******************************
```

> **メモ**
>
> 次に示す場合に，コネクションの確立または終了の区切り線が出力されます。
>
> - コール種別がCNCT の場合
> - コール種別がDISC の場合で，かつDISC が正常終了した場合

(13)　SQL 文の基本情報

SQL 文単位で SQL トレース情報を出力した場合，SQL 文の基本情報が出力されます。SQL 文の基本情報の出力例を次に示します。

■SQL 文の基本情報の出力例



[説明]

- con_id
  コネクション ID が出力されます。
- con_num
  HADB サーバの開始後に割り振られるコネクション通番が出力されます。
- stmt_hdl
  SQL 文に割り当てられた文ハンドルが出力されます。
- sql_serial_num
  コネクション確立後に割り振られる SQL 文通番が出力されます。

メモ

次に示す情報が出力されるときに SQL 文の基本情報が出力されます。

- 実行した SQL 文とアクセスパス情報
- ？パラメタの情報
- SQL パラレル実行機能に関する情報

上記の出力例は，実行した SQL 文とアクセスパス情報と，SQL 文の基本情報が出力された例です。

10.11.3 アクセスパスの統計情報の出力例と出力項目

ここでは，アクセスパスの統計情報の出力例，出力項目および注意事項について説明します。

アクセスパスの統計情報は，SQL トレース情報の一部として，実行した SQL 文の終了時に出力されます。なお，エラーとなった SQL 文については，アクセスパスの統計情報が出力されないことがあります。

アクセスパスの統計情報に出力される『対応するアクセスパスのツリー行番号』をキーにして，アクセスパス情報と突き合わせることで，SQL トレース情報を確認できます。アクセスパス情報については，「10.11.2　SQL トレース情報に出力される情報」の「(5)　実行した SQL 文とアクセスパス情報」を参照してください。

メモ

次に示す場合にアクセスパスの統計情報が出力されます。

- コール種別がCLOS の場合（カーソルをクローズしないで文ハンドルを解放したり，トランザクションを決着したりした場合を含む）
- コール種別がEXEC，EXDI，またはSQL の場合

(1)　アクセスパスの統計情報の出力例

アクセスパスの統計情報の出力例を次に示します。情報量が多いため，3 つに分けています。

■アクセスパスの統計情報の出力例（その 1）

```
 [access path statistics]
<<Data access information>>
Data_path#          Data_dbbuff_page_request_cnt(table) Data_dbbuff_page_hit_cnt(table)
--------------- ----------------------------------- ---------------------------
              3                                   5                             3
Data_dbbuff_page_read_cnt(table) Data_dbbuff_page_write_cnt(table) Data_dbbuff_tblscan_request_cnt
-------------------------------- --------------------------------- ---------------------------
                               2                                 0                           0
Data_dbbuff_tblscan_hit_cnt Data_dbbuff_tblscan_read_cnt Data_dbbuff_tblscan_read_size Data_dbbuff_tblscan_failed_cnt
--------------------------- -------------------------- --------------------------- ------------------------------
                          0                          0                           0                              0
Data_dbbuff_page_request_cnt(btree) Data_dbbuff_page_hit_cnt(btree) Data_dbbuff_page_read_cnt(btree)
----------------------------------- ------------------------------- -------------------------------
                                  0                               0                             0
Data_dbbuff_page_write_cnt(btree) Data_dbbuff_page_request_cnt(text) Data_dbbuff_page_hit_cnt(text)
--------------------------------- ---------------------------------- ---------------------------
                                0                                  0                           0
Data_dbbuff_page_read_cnt(text) Data_dbbuff_page_write_cnt(text) Data_dbbuff_page_request_cnt(range)
------------------------------- -------------------------------- ----------------------------------
                              0                                0                                  0
Data_dbbuff_page_hit_cnt(range) Data_dbbuff_page_read_cnt(range) Data_dbbuff_page_write_cnt(range)
------------------------------- -------------------------------- ----------------------------------
                              0                                0                                  0
Data_dbbuff_wrktbl_clt_tbl_cnt Data_dbbuff_wrktbl_clt_request_cnt Data_dbbuff_wrktbl_clt_page_hit_cnt
------------------------------ ------------------------------- ------------------------------------
                             0                               0                                    0
Data_dbbuff_wrktbl_clt_read_cnt Data_dbbuff_wrktbl_clt_write_cnt Data_dbbuff_page_request_cnt(glb_wrk)
------------------------------- -------------------------------- ----------------------------------
                              0                                0                                  0
Data_dbbuff_page_hit_cnt(glb_wrk) Data_dbbuff_page_read_cnt(glb_wrk) Data_dbbuff_page_write_cnt(glb_wrk)
--------------------------------- -------------------------------- ----------------------------------
                                0                                0                                  0
Data_bidx_page_split_cnt Data_bidx_validation_check_cnt Data_ridx_chunk_judge_cnt Data_ridx_chunk_skip_cnt
------------------------ ------------------------------ ------------------------ ------------------------
                       0                              0                        0                        0
Data_ridx_sgmt_judge_cnt Data_ridx_sgmt_skip_cnt Data_tidx_page_split_cnt Data_wrktbl_create_cnt
------------------------ ----------------------- ------------------------ ----------------------
                       0                       0                        0                      0
Data_wrktbl_drop_cnt Data_wrktbl_page_assign_num Data_wrktbl_page_free_num Data_wrktbl_page_use_max
-------------------- ------------------------- ------------------------ ------------------------
                   0                         0                        0                        0
Data_wrktbl_sort_merge_cnt Data_wrktbl_sort_page_max_cnt Data_dbarea_extension_cnt Data_segmentrel_rthd_max_num
-------------------------- ---------------------------- ---------------------------- ---------------------------
                         0                            0                            0                           0
Data_log_usrfile_write_cnt Data_log_usrbuf_out_cnt Data_log_usrfile_max_size Data_directory_send_num
-------------------------- ----------------------- ------------------------ ----------------------
                         0                       0                        0                      0
Data_access_info_type    Data_csvread_file_cnt Data_csvread_file_read_size Data_deleted_rows_cnt Data_tidx_all_search_cnt
------------------------ --------------------- ---------------------------- ---------------------- ------------------------
Access                                       0                            0                      0                        0
Data_auditread_file_cnt Data_auditread_file_read_size Data_grpagg_mem_short_cnt Data_grpagg_mem_exceed_limit_cnt
----------------------- ------------------------------ ------------------------------ ------------------------------
                      0                              0                              0                            0
Data_grpagg_mem_try_get_cnt Data_grpagg_mem_max_size Data_grpagg_preagg_row_cnt Data_grpagg_rslttbl_reuse_cnt
--------------------------- ------------------------ ------------------------- ---------------------------
                          0                        0                         0                           0
Data_cs_column_invalid_rows_cnt Data_cs_rowstore_fetch_rows_cnt Data_cs_invalid_info_page_request_cnt
------------------------------- ------------------------------- ------------------------------------
                              0                               0                                    0
Data_cs_invalid_false_positive_cnt Data_scan_info_save_sum_cnt Data_scan_info_save_max_cnt Data_scan_info_save_min_cnt
---------------------------------- ------------------------- ------------------------- -------------------------
                                 0                         0                         0                         0
Data_scan_info_restore_sum_cnt Data_scan_info_restore_max_cnt Data_scan_info_restore_min_cnt Data_dbbuff_page_request_max_cnt(table)
------------------------------ ------------------------------ ------------------------------ ------------------------------------
                             0                              0                              0                                    5
Data_dbbuff_page_request_min_cnt(table) Data_dbbuff_page_read_max_cnt(table) Data_dbbuff_page_read_min_cnt(table)
--------------------------------------- --------------------------------------- ---------------------------------------
                                      0                                       2                                       0
Data_dbbuff_tblscan_request_max_cnt Data_dbbuff_tblscan_request_min_cnt Data_dbbuff_tblscan_read_max_cnt
----------------------------------- ----------------------------------- -----------------------------------
                                  0                                   0                                   0
Data_dbbuff_tblscan_read_min_cnt Data_dbbuff_tblscan_read_max_size Data_dbbuff_tblscan_read_min_size
-------------------------------- -------------------------------- --------------------------------
                               0                                0                                0
```

■アクセスパスの統計情報の出力例（その 2）

```
Data_dbbuff_page_request_max_cnt(btree) Data_dbbuff_page_request_min_cnt(btree) Data_dbbuff_page_read_max_cnt(btree)
--------------------------------------- --------------------------------------- --------------------------------------
                                      0                                       0                                      0
Data_dbbuff_page_read_min_cnt(btree) Data_dbbuff_page_request_max_cnt(text) Data_dbbuff_page_request_min_cnt(text)
------------------------------------ -------------------------------------- --------------------------------------
                                     0                                      0                                     0
Data_dbbuff_page_read_max_cnt(text) Data_dbbuff_page_read_min_cnt(text) Data_dbbuff_page_request_max_cnt(range)
----------------------------------- ---------------------------------- -------------------------------------
                                    0                                  0                                    0
Data_dbbuff_page_request_min_cnt(range) Data_dbbuff_page_read_max_cnt(range) Data_dbbuff_page_read_min_cnt(range)
--------------------------------------- --------------------------------------- --------------------------------------
                                      0                                       0                                      0
Data_dbbuff_wrktbl_clt_request_max_cnt Data_dbbuff_wrktbl_clt_request_min_cnt Data_dbbuff_wrktbl_clt_read_max_cnt
-------------------------------------- ------------------------------------- -------------------------------------
                                     0                                     0                                     0
Data_dbbuff_wrktbl_clt_read_min_cnt Data_dbbuff_wrktbl_clt_write_max_cnt Data_dbbuff_wrktbl_clt_write_min_cnt
----------------------------------- ------------------------------------ ------------------------------------
                                    0                                    0                                    0
Data_dbbuff_page_request_max_cnt(glb_wrk) Data_dbbuff_page_request_min_cnt(glb_wrk) Data_dbbuff_page_read_max_cnt(glb_wrk)
----------------------------------------- ----------------------------------------- -----------------------------------------
                                        0                                         0                                        0
Data_dbbuff_page_read_min_cnt(glb_wrk) Data_dbbuff_page_write_max_cnt(glb_wrk) Data_dbbuff_page_write_min_cnt(glb_wrk)
-------------------------------------- -------------------------------------- --------------------------------------
                                     0                                      0                                     0
Data_wrktbl_page_branch_num
---------------------------
                          0

<<Scan information>>
Scan_path#           Scan_row_cnt         Scan_start_time            Scan_end_time
-------------------- -------------------- -------------------------- --------------------------
                   3                    1 2015-07-31 21:48:46.736068 2015-07-31 21:48:46.737075
Scan_info_data_max_num Scan_info_data_max_free_num Scan_info_data_size  Scan_info_inpred_max_num
---------------------- ------------------------- ---------------------- -----------------------
                     0                         0                      0                       0
Scan_info_inpred_max_free_num Scan_info_inpred_size Scan_info_quantpred_max_num Scan_info_quantpred_max_free_num
----------------------------- ---------------------- ------------------------- -----------------------------
                            0                      0                         0                             0
Scan_info_quantpred_size Scan_info_drvtbl_max_num Scan_info_drvtbl_max_free_num Scan_info_drvtbl_size Scan_info_hash_max_num
------------------------ ------------------------ ----------------------------- ---------------------- ---------------------
                       0                        0                             0                      0                     0
Scan_info_hash_max_free_num Scan_info_hash_size  Scan_info_hashbckt_max_num Scan_info_hashbckt_size
--------------------------- ---------------------- -------------------------- ------------------------
                          0                      0                          0                        0
Scan_outerjoin_info_max_num Scan_outerjoin_info_size Scan_info_csvread_file_max_num Scan_info_csvread_file_max_free_num
--------------------------- ------------------------ ------------------------------ ----------------------------------
                          0                        0                              0                                  0
Scan_info_csvread_file_size Scan_info_csvread_data_max_num Scan_info_csvread_data_max_free_num Scan_info_csvread_data_size
--------------------------- ------------------------------ ------------------------------------ ----------------------------
                          0                              0                                    0                            0
Scan_max_rt_row_cnt  Scan_min_rt_row_cnt  Scan_sum_rt_aggr_row_cnt Scan_max_rt_aggr_row_cnt Scan_min_rt_aggr_row_cnt
-------------------- -------------------- ------------------------ ------------------------ ------------------------
                   1                    0                        0                        0                        0
Scan_sum_rt_array_any_eval_cnt Scan_max_rt_array_any_eval_cnt Scan_min_rt_array_any_eval_cnt Scan_mn_sum_rt_send_rdata_size
------------------------------ ------------------------------ ------------------------------ ------------------------------
                             0                              0                              0                              0
Scan_mn_sum_rt_send_rdata_cnt Scan_mn_sum_rt_recv_rdata_size Scan_mn_sum_rt_recv_rdata_cnt Scan_mn_sum_rt_send_odata_size
----------------------------- ------------------------------ ------------------------------ ------------------------------
                            0                              0                              0                              0
Scan_mn_sum_rt_send_odata_cnt Scan_mn_sum_rt_recv_odata_size Scan_mn_sum_rt_recv_odata_cnt
----------------------------- ------------------------------ ------------------------------
                            0                              0                              0
```

■アクセスパスの統計情報の出力例（その 3）

```
<<Set operation information>>

<<Hash grouping area information>>

<<Hash table area information>>

<<Correlated subqueries information>>

<<Correlated subqueries result cache information>>

<<Recursive queries information>>

<<SQL exec information>>
SQLexec_result_wrk_size SQLexec_result_wrk_max_num SQLexec_result_wrk_init_num SQLexec_mem_size     SQLexec_mem_max_num
----------------------- ------------------------- ------------------------- ------------------- -------------------
                     48                         0                         0                   0                   0
SQLexec_mem_init_num SQLexec_dynamic_mem_alloc_unit SQLexec_delayed_free_mem_max_size SQLexec_stack_size
-------------------- ------------------------------ ------------------------------- ------------------
                   0                              0                               0             163840
SQLexec_result_ext_wrk_max_num
------------------------------
                             0
```

(2)　アクセスパスの統計情報に出力される項目

アクセスパスの統計情報に出力される項目，および出力対象となるアクセスパスの種類を，次の表に示します。

表 10-21　アクセスパスの統計情報に出力される項目の一覧

項番	出力項目	出力対象となるアクセスパスの種類		出力される内容の詳細
1	データアクセス情報	作業表の作成情報	CREATE GLOBAL WORK TABLE	「(a)　データアクセス情報」を参照してください。
2			CREATE LOCAL WORK TABLE	
3		表の検索方式の情報	TABLE SCAN	
4			KEY SCAN	
5			INDEX SCAN	
6		作業表の検索情報	WORK TABLE SCAN	
7		導出表の情報	DERIVED TABLE	
8		表関数導出表の情報	TABLE FUNCTION DERIVED TABLE	
9		結合方式の情報	HASH JOIN	
10			NESTED LOOP JOIN	
11		副問合せ種別の情報	SUBQUERY HASH	
12		グループ化集合の情報	GROUPING SET	
13	検索処理に関する情報 ※1，※2，※3	副問合せ種別の情報	SUBQUERY HASH	「(b)　検索処理に関する情報」を参照してください。
14		導出表の情報	DERIVED TABLE	
15		表関数導出表の情報	TABLE FUNCTION DERIVED TABLE	

項番	出力項目	出力対象となるアクセスパスの種類		出力される内容の詳細
16		表値構成子の検索情報	TABLE VALUE CONSTRUCTOR SCAN	
17		グループ化の処理方式の情報	GLOBAL HASH GROUPING	
18		重複排除方式の情報	GLOBAL HASH UNIQUE	
19		表の検索方式の情報	TABLE SCAN	
20			KEY SCAN	
21			INDEX SCAN	
22		作業表の検索情報	WORK TABLE SCAN	
23		結合方式の情報	NESTED LOOP JOIN	
24			HASH JOIN	
25	集合演算に関する情報※4	集合演算種別の情報	SET OPERATION	「(c)　集合演算に関する情報」を参照してください。
26	ハッシュグループ化領域に関する情報※5	グループ化の処理方式の情報	LOCAL HASH GROUPING	「(d)　ハッシュグループ化領域に関する情報」を参照してください。
27	ハッシュテーブル領域に関する情報	重複排除方式の情報	GLOBAL HASH UNIQUE	「(e)　ハッシュテーブル領域に関する情報」を参照してください。
28		副問合せ種別の情報	SUBQUERY HASH	
29		グループ化の処理方式の情報	GLOBAL HASH GROUPING	
30		結合方式の情報	HASH JOIN	
31	外への参照列を含む副問合せに関する情報※6, ※7	副問合せ種別の情報	SUBQUERY LOOP	「(f)　外への参照列を含む副問合せに関する情報」を参照してください。
32	副問合せの結果を格納するキャッシュに関する情報	副問合せのキャッシュの情報	USING CACHE	「(g)　副問合せの結果を格納するキャッシュに関する情報」を参照してください。
33	再帰的問合せに関する情報	再帰的問合せの情報	RECURSIVE	「(h)　再帰的問合せに関する情報」を参照してください。
34	SQL 文の実行に関する情報	－	－	「(i)　SQL 文の実行に関する情報」を参照してください。

（凡例）

－：アクセスパスの統計情報の出力対象となる SQL 文を実行すると，アクセスパスの種類に関係なく，情報が出力されます。

注※1

副問合せの処理方式が，ネストループ作業表実行またはネストループ行値実行となる，外への参照列を含む副問合せの中で実行される検索処理の場合，対象のアクセスパスは，アクセスパスの統計情報に出力されません。

ただし，外への参照列を含む副問合せの中で実行される次の処理のアクセスパスについては，アクセスパスの統計情報に出力されます。

- 外への参照列を含まない副問合せの中で実行される検索処理

注※2

副問合せの処理方式が，作業表実行，作業表行値実行またはネストループ作業表実行となる副問合せについて，その結果を格納した作業表の検索処理の場合，対象のアクセスパスは，アクセスパスの統計情報に出力されません。

注※3

ネストループジョインが適用される表の結合処理で，ネストループジョインの外表および内表に関する検索処理の場合，対象のアクセスパスは，アクセスパスの統計情報に出力されません。ただし，ネストループジョインの最も外側の表が導出表の場合，その導出問合せに関する検索処理については，対象のアクセスパスがアクセスパスの統計情報に出力されます。

注※4

副問合せの処理方式が，ネストループ作業表実行またはネストループ行値実行となる，外への参照列を含む副問合せの中で実行される集合演算の場合，対象のアクセスパスは，アクセスパスの統計情報に出力されません。

ただし，外への参照列を含む副問合せの中で実行される次の処理のアクセスパスについては，アクセスパスの統計情報に出力されます。

- 外への参照列を含まない副問合せの中で実行される集合演算

注※5

副問合せの処理方式が，ネストループ作業表実行またはネストループ行値実行となる，外への参照列を含む副問合せの中で実行されるローカルハッシュグループ化の場合，対象のアクセスパスは，アクセスパスの統計情報に出力されません。

ただし，外への参照列を含む副問合せの中で実行される次の処理のアクセスパスについては，アクセスパスの統計情報に出力されます。

- 外への参照列を含まない副問合せの中で実行されるローカルハッシュグループ化の処理

注※6

副問合せの結果を評価する個所で，同じSUBQUERY LOOP が複数ある場合は，最初の 1 行目に出力されます。

注※7

アクセスパス情報のツリー表示でSUBQUERY LOOP がネストした場合，ネストしたSUBQUERY LOOP に対してアクセスパスの統計情報が出力されないことがあります。

(a) データアクセス情報

アクセスパスの統計情報に出力されるデータアクセス情報について，次の表に示します。

なお，更新系 SQL に関するアクセスパスの統計情報の場合，更新対象となる表については，更新対象データまたは削除対象データを検索するときの情報だけが出力されます。データの追加処理，更新処理および削除処理を行うときの情報は出力されません。

また，データアクセス情報は，アクセスパスに対して複数行出力されることがあります。詳細については，「表 10-23　各アクセスパスのデータアクセス情報の出力行数と出力情報」を参照してください。

表 10-22　アクセスパスの統計情報に出力される項目（データアクセス情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Data_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Data_dbbuff_page_request_cnt(table)	実表へのアクセス時に使用するグローバルバッファのページ要求回数	UINT8	最大値
3	Data_dbbuff_page_hit_cnt(table)	実表へのアクセス時に使用するグローバルバッファのヒット回数	UINT8	最大値
4	Data_dbbuff_page_read_cnt(table)	実表へのアクセス時に使用するグローバルバッファに対するファイル読み込み回数	UINT8	最大値
5	Data_dbbuff_page_write_cnt(table)	実表へのアクセス時に使用するグローバルバッファからのファイル書き込み回数	UINT8	最大値
6	Data_dbbuff_tblscan_request_cnt	実表へのアクセス時に使用するテーブルスキャンバッファのページ要求回数	UINT8	最大値
7	Data_dbbuff_tblscan_hit_cnt	実表へのアクセス時に使用するテーブルスキャンバッファのページのヒット回数	UINT8	最大値
8	Data_dbbuff_tblscan_read_cnt	実表へのアクセス時に使用するテーブルスキャンバッファブロックへのファイル読み込み回数	UINT8	最大値
9	Data_dbbuff_tblscan_read_size	実表へのアクセス時に使用するテーブルスキャンバッファのファイル読み込みサイズ（単位：キロバイト）	UINT8	最大値
10	Data_dbbuff_tblscan_failed_cnt	実表へのアクセス時に使用するテーブルスキャンバッファブロックの割り当てに失敗した回数 割り当てに失敗した場合は，グローバルバッファを使用します。	UINT8	最大値
11	Data_dbbuff_page_request_cnt(btree)	B-tree インデクスへのアクセス時に使用するグローバルバッファのページ要求回数	UINT8	最大値
12	Data_dbbuff_page_hit_cnt(btree)	B-tree インデクスへのアクセス時に使用するグローバルバッファのヒット回数	UINT8	最大値
13	Data_dbbuff_page_read_cnt(btree)	B-tree インデクスへのアクセス時に使用するグローバルバッファへのファイル読み込み回数	UINT8	最大値

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
14	Data_dbbuff_page_write_cnt(btree)	B-tree インデクスへのアクセス時に使用するグローバルバッファからのファイル書き込み回数	UINT8	最大値
15	Data_dbbuff_page_request_cnt(text)	テキストインデクスへのアクセス時に使用するグローバルバッファのページ要求回数	UINT8	最大値
16	Data_dbbuff_page_hit_cnt(text)	テキストインデクスへのアクセス時に使用するグローバルバッファのヒット回数	UINT8	最大値
17	Data_dbbuff_page_read_cnt(text)	テキストインデクスへのアクセス時に使用するグローバルバッファに対するファイル読み込み回数	UINT8	最大値
18	Data_dbbuff_page_write_cnt(text)	テキストインデクスへのアクセス時に使用するグローバルバッファからのファイル書き込み回数	UINT8	最大値
19	Data_dbbuff_page_request_cnt(range)	レンジインデクスへのアクセス時に使用するグローバルバッファのページ要求回数	UINT8	最大値
20	Data_dbbuff_page_hit_cnt(range)	レンジインデクスへのアクセス時に使用するグローバルバッファのヒット回数	UINT8	最大値
21	Data_dbbuff_page_read_cnt(range)	レンジインデクスへのアクセス時に使用するグローバルバッファに対するファイル読み込み回数	UINT8	最大値
22	Data_dbbuff_page_write_cnt(range)	レンジインデクスへのアクセス時に使用するグローバルバッファからのファイル書き込み回数	UINT8	最大値
23	Data_dbbuff_wrktbl_clt_tbl_cnt	ローカル作業表用バッファの生成回数	UINT8	最大値
24	Data_dbbuff_wrktbl_clt_request_cnt	ローカル作業表用バッファのページ要求回数	UINT8	最大値
25	Data_dbbuff_wrktbl_clt_page_hit_cnt	ローカル作業表用バッファのヒット回数	UINT8	最大値
26	Data_dbbuff_wrktbl_clt_read_cnt	ローカル作業表用バッファへのファイルの読み込み回数	UINT8	最大値
27	Data_dbbuff_wrktbl_clt_write_cnt	ローカル作業表用バッファからファイルへの書き込み回数	UINT8	最大値
28	Data_dbbuff_page_request_cnt(glb_wrk)	グローバル作業表用のグローバルバッファのページ要求回数	UINT8	最大値
29	Data_dbbuff_page_hit_cnt(glb_wrk)	グローバル作業表用のグローバルバッファのヒット回数	UINT8	最大値
30	Data_dbbuff_page_read_cnt(glb_wrk)	グローバル作業表用のグローバルバッファへのファイルの読み込み回数	UINT8	最大値
31	Data_dbbuff_page_write_cnt(glb_wrk)	グローバル作業表用のグローバルバッファからファイルへの書き込み回数	UINT8	最大値

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
32	Data_bidx_page_split_cnt	システムで使用する値 常に0が出力されます。	UINT8	最大値
33	Data_bidx_validation_check_cnt	B-treeインデクスを使用した検索で，行が有効かどうかを判定した回数	UINT8	最大値
34	Data_ridx_chunk_judge_cnt	レンジインデクスを使用した検索で，表のチャンクを判定した回数	UINT8	最大値
35	Data_ridx_chunk_skip_cnt	レンジインデクスを使用した検索で，表のチャンクをスキップした回数	UINT8	最大値
36	Data_ridx_sgmt_judge_cnt	レンジインデクスを使用した検索で，表のセグメントを判定した回数	UINT8	最大値
37	Data_ridx_sgmt_skip_cnt	レンジインデクスを使用した検索で，表のセグメントをスキップした回数	UINT8	最大値
38	Data_tidx_page_split_cnt	システムで使用する値 常に0が出力されます。	UINT8	最大値
39	Data_wrktbl_create_cnt	作業表を作成した回数	UINT8	最大値
40	Data_wrktbl_drop_cnt	作業表を削除した回数	UINT8	最大値
41	Data_wrktbl_page_assign_num	割り当てた作業表ページ数	UINT8	最大値
42	Data_wrktbl_page_free_num	解放された作業表ページ数	UINT8	最大値
43	Data_wrktbl_page_use_max	割り当てた作業表のページ数のうち，最も大きいページ数	UINT8	最大値
44	Data_wrktbl_sort_merge_cnt	作業表のソートでのソート段数の最大値	UINT8	最大値
45	Data_wrktbl_sort_page_max_cnt	作業表のソート処理で割り当てた最も大きい作業表のページ数	UINT8	最大値
46	Data_dbarea_extension_cnt	システムで使用する値 常に0が出力されます。	UINT8	最大値
47	Data_segmentrel_rthd_max_num	システムで使用する値 常に0が出力されます。	UINT8	最大値
48	Data_log_usrfile_write_cnt	ユーザログファイルへの書き込み回数	UINT8	最大値
49	Data_log_usrbuf_out_cnt	ユーザログバッファの満杯による掃き出し回数	UINT8	最大値
50	Data_log_usrfile_max_size	ユーザログファイルの使用量の最大値	UINT8	最大値
51	Data_directory_send_num	システムで使用する値 常に0が出力されます。	UINT8	最大値
52	Data_access_info_type	データアクセスの種別 • Access	CHAR	－

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
		アクセスパスが対象とするスキーマオブジェクトにアクセスする場合のデータアクセス情報であることを示しています。 • Create_hash_wrktbl ハッシュテーブル領域不足によって作業表が作成される場合のデータアクセス情報であることを示しています。 • Scan_hash_wrktbl ハッシュテーブル領域不足によって作成される作業表を検索する場合のデータアクセス情報であることを示しています。 • Create_split_hash_wrktbl ハッシュテーブル領域不足によって作業表※が新たに作成される場合のデータアクセス情報であることを示しています。 • Scan_split_hash_wrktbl ハッシュテーブル領域不足によって新たに作成される作業表※を検索する場合のデータアクセス情報であることを示しています。 • Rowid_fetch 行 ID を使用してデータページからデータを取り出すための検索が実行される場合のデータアクセス情報であることを示しています。 • Create_grpset_wrktbl グループ化処理を複数回実行する際に使用される作業表を作成するときのデータアクセス情報であることを示しています。 • Scan_grpset_wrktbl グループ化処理を複数回実行する際に使用される作業表を検索するときのデータアクセス情報であることを示しています。 データアクセスの種別と，アクセスパスの関係については，「表 10-23　各アクセスパスのデータアクセス情報の出力行数と出力情報」を参照してください。		
53	Data_csvread_file_cnt	ADB_CSVREAD 関数を使用してオープンされたファイルの数 アーカイブマルチチャンク表にアクセスする SQL 文の等価変換によって生成された ADB_CSVREAD 関数を使用してオープンされたファイルも含まれます。	UINT8	最大値

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
54	Data_csvread_file_read_size	ADB_CSVREAD 関数を使用してファイルから読み込んだデータサイズの合計値（単位：バイト） アーカイブマルチチャンク表の等価変換によって生成されたADB_CSVREAD 関数を使用してファイルから読み込んだデータサイズも含まれます。	UINT8	最大値
55	Data_deleted_rows_cnt	検索時に無効行と判定された行数 無効行については，次の個所を参照してください。 • 「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」 • 「11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法」 • 「11.17.1　システム表を再編成する理由」	UINT8	最大値
56	Data_tidx_all_search_cnt	テキストインデクスを使用してチャンク内のデータを検索した際，チャンク内のデータを全件検索した回数	UINT8	最大値
57	Data_auditread_file_cnt	ADB_AUDITREAD 関数を使用してオープンされたファイルの数	UINT8	最大値
58	Data_auditread_file_read_size	ADB_AUDITREAD 関数を使用してファイルから読み込んだデータサイズの合計値（単位：バイト）	UINT8	最大値
59	Data_cs_column_invalid_rows_cnt	カラムストア表のカラムデータ用セグメント検索時に，無効行と判定された行数	UINT8	最大値
60	Data_cs_rowstore_fetch_rows_cnt	カラムストア表の検索時に，ローストア形式のデータ（INSERT 文またはUPDATE 文で追加した行）にアクセスしてヒットした行数	UINT8	最大値
61	Data_cs_invalid_info_page_request_cnt	カラムストア表の無効情報管理用ページのページ要求回数	UINT8	最大値
62	Data_dbbuff_page_request_max_cnt(table)	実表へのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値
63	Data_dbbuff_page_request_min_cnt(table)	実表へのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値
64	Data_dbbuff_tblscan_request_max_cnt	実表へのアクセス時に使用するテーブルスキャンバッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値
65	Data_dbbuff_tblscan_request_min_cnt	実表へのアクセス時に使用するテーブルスキャンバッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値
66	Data_dbbuff_page_request_max_cnt(btree)	B-tree インデクスへのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
67	Data_dbbuff_page_request_min_cnt(btree)	B-tree インデクスへのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値
68	Data_dbbuff_page_request_max_cnt(text)	テキストインデクスへのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値
69	Data_dbbuff_page_request_min_cnt(text)	テキストインデクスへのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値
70	Data_dbbuff_page_request_max_cnt(range)	レンジインデクスへのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値
71	Data_dbbuff_page_request_min_cnt(range)	レンジインデクスへのアクセス時に使用するグローバルバッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値
72	Data_dbbuff_wrktbl_clt_request_max_cnt	ローカル作業表用バッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値
73	Data_dbbuff_wrktbl_clt_request_min_cnt	ローカル作業表用バッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値
74	Data_dbbuff_page_request_max_cnt(glb_wrk)	グローバル作業表用のグローバルバッファのページ要求回数（リアルスレッド間の最大値）	UINT8	最大値
75	Data_dbbuff_page_request_min_cnt(glb_wrk)	グローバル作業表用のグローバルバッファのページ要求回数（リアルスレッド間の最小値）	UINT8	最大値

（凡例）

UINT8：8バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0または実行途中の値が出力されることがあります。

CHAR：『出力される内容』で示した文字列が出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，空白が出力されることがあります。

最大値：オーバフローした場合，18,446,744,073,709,551,615が出力されます。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

注※

ハッシュテーブルの作成中にハッシュテーブル領域が不足した場合，複数の作業表が作成され，ハッシュテーブルに格納されるデータが，各作業表に分割されて格納されます。各作業表に対するハッシュテーブルの作成中にハッシュテーブル領域不足が再度発生した場合，作業表が新たに作成されます。この新たに作成された作業表を示しています。

各アクセスパスのデータアクセス情報の出力行数と出力情報について，次の表に示します。

表 10-23　各アクセスパスのデータアクセス情報の出力行数と出力情報

項番	アクセスパスの種類	データアクセス情報の出力行数と出力情報	Data_access_info_type に出力される内容
1	CREATE GLOBAL WORK TABLE	出力行数 アクセスパスに対して 1 行出力されます。 出力情報 グローバル作業表を作成する場合（ソート処理を含む）のデータアクセス情報が出力されます。この情報には，グローバル作業表を作成する前段階に当たるローカル作業表を作成するときの情報も含まれます。	Access
2	CREATE LOCAL WORK TABLE	出力行数 アクセスパスに対して 1 行出力されます。 出力情報 ローカル作業表を作成する場合（ソート処理を含む）のデータアクセス情報が出力されます。	Access
3	TABLE SCAN	出力情報（1 行出力される場合） 通常は，アクセスパスに対して 1 行出力されます。 対象の実表，作業表，または表関数導出表を検索する場合のデータアクセス情報が出力されます。 出力情報（3 行出力される場合） アクセスパスが次のどれかに該当する場合は，3 行出力されます。 • ハッシュジョインの内表の場合 • ハッシュ実行が適用される副問合せで，外側の問合せが対象である場合 1 行目には，対象の実表，作業表，または表関数導出表を検索する場合のデータアクセス情報が出力されます。 2 行目には，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 3 行目には，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。 出力情報（5 行出力される場合） アクセスパスが次のどれかに該当する場合は，5 行出力されます。 • ハッシュジョインの外表の場合	1 行出力される場合 Access 3 行出力される場合 • Access（1 行目） • Create_hash_wrktbl（2 行目） • Scan_hash_wrktbl（3 行目） 5 行出力される場合 • Access（1 行目） • Create_hash_wrktbl（2 行目） • Scan_hash_wrktbl（3 行目） • Create_split_hash_wrktbl（4 行目） • Scan_split_hash_wrktbl（5 行目）
4	INDEX SCAN		
5	KEY SCAN		

項番	アクセスパスの種類	データアクセス情報の出力行数と出力情報	Data_access_info_type に出力される内容
		• ハッシュ実行が適用される副問合せの対象の場合 • グローバルハッシュグループ化の対象の場合 • ハッシュ実行が適用されるSELECT DISTINCT の対象の場合 1 行目には，対象の実表，作業表，または表関数導出表を検索する場合のデータアクセス情報が出力されます。 2 行目には，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 3 行目には，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。 4 行目には，作業表に分割されたあとに，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 5 行目には，作業表に分割されたあとに，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。	
6	WORK TABLE SCAN	出力情報（1 行出力される場合） 通常は，アクセスパスに対して 1 行出力されます。 対象の作業表を検索する場合のデータアクセス情報が出力されます。 出力情報（2 行出力される場合） アクセスパスが次に該当する場合は，2 行出力されます。 • ORDER BY 句を指定した SQL 文で作成された作業表に対して，FROM 句に指定した表の行 ID を格納する場合 1 行目には，対象の作業表を検索する場合のデータアクセス情報が出力されます。 2 行目には，行 ID を使用してデータページからデータを取り出すために検索した場合のデータアクセス情報が出力されます。データを取出す対象表が複数ある場合は，複数の表にアクセスしたときの値の合計となります。 出力情報（3 行出力される場合） アクセスパスが次のどれかに該当する場合は，3 行出力されます。 • ハッシュジョインの内表の場合	1 行出力される場合 Access 2 行出力される場合 • Access（1 行目） • Rowid_fetch（2 行目） 3 行出力される場合 • Access（1 行目） • Create_hash_wrktbl（2 行目） • Scan_hash_wrktbl（3 行目） 5 行出力される場合 • Access（1 行目） • Create_hash_wrktbl（2 行目） • Scan_hash_wrktbl（3 行目） • Create_split_hash_wrktbl（4 行目） • Scan_split_hash_wrktbl（5 行目）

項番	アクセスパスの種類	データアクセス情報の出力行数と出力情報	Data_access_info_type に出力される内容
		• ハッシュ実行が適用される副問合せで，外側の問合せが対象である場合 1 行目には，対象の作業表を検索する場合のデータアクセス情報が出力されます。 2 行目には，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 3 行目には，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。 出力情報（5 行出力される場合） アクセスパスが次のどれかに該当する場合は，5 行出力されます。 • ハッシュジョインの外表の場合 • ハッシュ実行が適用される副問合せの対象の場合 • グローバルハッシュグループ化の対象の場合 • ハッシュ実行が適用されるSELECT DISTINCT の対象の場合 1 行目には，対象の作業表を検索する場合のデータアクセス情報が出力されます。 2 行目には，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 3 行目には，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。 4 行目には，作業表に分割されたあとに，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 5 行目には，作業表に分割されたあとに，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。	
7	DERIVED TABLE	出力情報（2 行出力される場合） アクセスパスが次のどれかに該当する場合は，2 行出力されます。 • ハッシュジョインの内表の場合 • ハッシュ実行が適用される副問合せで，外側の問合せが対象である場合	2 行出力される場合 • Create_hash_wrktbl（1 行目） • Scan_hash_wrktbl（2 行目） 4 行出力される場合 • Create_hash_wrktbl（1 行目） • Scan_hash_wrktbl（2 行目）
8	TABLE FUNCTION DERIVED TABLE		
9	HASH JOIN		
10	NESTED LOOP JOIN		
11	SUBQUERY HASH		

項番	アクセスパスの種類	データアクセス情報の出力行数と出力情報	Data_access_info_type に出力される内容
		1 行目には，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 2 行目には，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。 出力情報（4 行出力される場合） アクセスパスが次のどれかに該当する場合は，4 行出力されます。 • ハッシュジョインの外表の場合 • ハッシュ実行が適用される副問合せの対象の場合 • グローバルハッシュグループ化の対象の場合 • ハッシュ実行が適用されるSELECT DISTINCT の対象の場合 • ハッシュ実行が適用される集合演算について作成される導出表の場合 1 行目には，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 2 行目には，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。 3 行目には，作業表に分割されたあとに，ハッシュテーブル領域で領域不足が発生して作業表が作成された場合のデータアクセス情報が出力されます。 4 行目には，作業表に分割されたあとに，ハッシュテーブル領域で領域不足が発生して作業表が検索された場合のデータアクセス情報が出力されます。	• Create_split_hash_wrktbl（3 行目） • Scan_split_hash_wrktbl（4 行目）
12	GROUPING SET	出力行数 アクセスパスに対して 2 行出力されます。 出力情報 1 行目には，グループ化処理を複数回実行する際に使用される作業表を作成するときのデータアクセス情報が出力されます。 2 行目には，グループ化処理を複数回実行する際に使用される作業表を検索するときのデータアクセス情報が出力されます。	• Create_grpset_wrktbl（1 行目） • Scan_grpset_wrktbl（2 行目）

(b) 検索処理に関する情報

アクセスパスの統計情報に出力される検索処理に関する情報について，次の表に示します。

表 10-24　アクセスパスの統計情報の出力項目（検索処理に関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Scan_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Scan_row_cnt	検索処理した行数※1 各リアルスレッドによって実行された検索処理の行数の合計値が出力されます。	UINT8	WRAP
3	Scan_start_time	検索処理の開始日時※2 検索処理を開始した日時が，次の形式で出力されます。 *YYYY-MM-DD*△*hh*:*mm*:*ss*.*nnnnnn*※4	CHAR	－
4	Scan_end_time	検索処理の終了日時※3 検索処理を終了した日時が，次の形式で出力されます。 *YYYY-MM-DD*△*hh*:*mm*:*ss*.*nnnnnn*※4	CHAR	－
5	Scan_max_rt_row_cnt	検索処理した行数※1 各リアルスレッドによって実行された検索処理の行数のうち，最大値が出力されます。	UINT8	WRAP
6	Scan_min_rt_row_cnt	検索処理した行数※1 各リアルスレッドによって実行された検索処理の行数のうち，最小値が出力されます。	UINT8	WRAP

（凡例）

UINT8：

８バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0 または実行途中の値が出力されることがあります。

CHAR

：『出力される内容』で示した文字列が出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，空白が出力されることがあります。

WRAP：

18,446,744,073,709,551,615 を超えると，ラップアラウンドして最小値に戻ります。

－：

オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

注※1

次のように検索処理が複数回実行される場合，複数回の検索処理の値を合計した値が出力されます。

- 再帰的問合せ内にある，再帰的問合せ名を参照している検索処理

- グループ化処理で実行される検索処理（同一問合せ指定中に引数が異なるDISTINCT集合関数を複数指定した場合，各DISTINCT集合関数の結果を求めるためにグループ化処理が複数回実行されます）
- ハッシュジョインが分割実行される場合，各ハッシュジョインで実行される検索処理

注※2

次のように検索処理が複数回実行される場合，1回目の検索処理の開始時刻が出力されます。

- 再帰的問合せ内にある，再帰的問合せ名を参照している検索処理
- グループ化処理で実行される検索処理（同一問合せ指定中に引数が異なるDISTINCT集合関数を複数指定した場合，各DISTINCT集合関数の結果を求めるためにグループ化処理が複数回実行されます）
- ハッシュジョインが分割実行される場合，各ハッシュジョインで実行される検索処理

注※3

次のように検索処理が複数回実行される場合，最後の検索処理の終了時刻が出力されます。

- 再帰的問合せ内にある，再帰的問合せ名を参照している検索処理
- グループ化処理で実行される検索処理（同一問合せ指定中に引数が異なるDISTINCT集合関数を複数指定した場合，各DISTINCT集合関数の結果を求めるためにグループ化処理が複数回実行されます）
- ハッシュジョインが分割実行される場合，各ハッシュジョインで実行される検索処理

注※4

年（*YYYY*）は4桁で出力されます。月（*MM*），日（*DD*），時（*hh*），分（*mm*），および秒（*ss*）は2桁で出力されます。マイクロ秒（*nnnnnn*）は6桁で出力されます。桁数が足りない場合は，足りない分，左側に0が補われます。△は半角空白を表します。

(c) 集合演算に関する情報

アクセスパスの統計情報に出力される集合演算に関する情報について，次の表に示します。

表 10-25 アクセスパスの統計情報の出力項目（集合演算に関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Setop_path#	対応するアクセスパスのツリー行番号	UINT8	−
2	Setop_type	集合演算種別 • UNION ALL • UNION DISTINCT • EXCEPT ALL • EXCEPT DISTINCT • INTERSECT ALL • INTERSECT DISTINCT なお，SQL文に指定された集合演算種別と異なることがあります。	CHAR	−
3	Setop_sort	集合演算時のソート処理の有無※1	CHAR	−

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
		• Y：ソート処理がある • N：ソート処理がない		
4	Setop_row_cnt	集合演算による処理行数	UINT8	WRAP
5	Setop_start_time	集合演算による処理の開始日時 集合演算による処理を開始した日時が，次の形式で出力されます。※2 *YYYY-MM-DD*△*hh*:*mm*:*ss*.*nnnnnn*※4	CHAR	－
6	Setop_end_time	集合演算による処理の終了日時 集合演算による処理を終了した日時が，次の形式で出力されます。※3 *YYYY-MM-DD*△*hh*:*mm*:*ss*.*nnnnnn*※4	CHAR	－

（凡例）

UINT8：8バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0または実行途中の値が出力されることがあります。

CHAR：『出力される内容』で示した文字列が出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，空白が出力されることがあります。

WRAP：18,446,744,073,709,551,615を超えると，ラップアラウンドして最小値に戻ります。

－：オーバフローは発生しません。または，該当しません。

注※1

再帰的問合せ内にある，再帰的問合せ名を参照している集合演算による処理の場合，再帰ごとの値を合計した値が出力されます。

注※2

再帰的問合せ内にある，再帰的問合せ名を参照している集合演算による処理の場合，1回目の再帰の開始時刻が出力されます。

注※3

再帰的問合せ内にある，再帰的問合せ名を参照している集合演算による処理の場合，最後の再帰の終了時刻が出力されます。

注※4

年（*YYYY*）は4桁で出力されます。月（*MM*），日（*DD*），時（*hh*），分（*mm*），および秒（*ss*）は2桁で出力されます。マイクロ秒（*nnnnnn*）は6桁で出力されます。桁数が足りない場合は，足りない分，左側に0が補われます。△は半角空白を表します。

(d) ハッシュグループ化領域に関する情報

アクセスパスの統計情報に出力されるハッシュグループ化領域に関する情報について，次の表に示します。

表 10-26　アクセスパスの統計情報の出力項目（ハッシュグループ化領域に関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Hashgrp_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Hashgrp_area_shortage	ハッシュグループ化領域の領域不足の発生有無 • Y：領域不足が発生している • N：領域不足が発生していない	CHAR	－
3	Hashgrp_area_sufficient_size	ハッシュグループ化領域で領域不足が発生しない十分な領域サイズ（単位：キロバイト） なお，十分な領域サイズが，サーバ定義またはクライアント定義の adb_sql_exe_hashgrp_area_size オペランドに指定できる最大値を超える場合は，各オペランドに指定できる最大値が表示されます。	UINT8	－
4	Hashgrp_group_num	グループ化による処理行数（HAVING 句の評価前）	UINT8	WRAP
5	Hashgrp_wrktbl_max_row_cnt	作業表への挿入行数の最大値（リアルスレッドごとの最大値）	UINT8	WRAP

（凡例）

UINT8：8 バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0 または実行途中の値が出力されることがあります。

CHAR：『出力される内容』で示した文字列が出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，空白が出力されることがあります。

WRAP：18,446,744,073,709,551,615 を超えると，ラップアラウンドして最小値に戻ります。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

(e)　ハッシュテーブル領域に関する情報

アクセスパスの統計情報に出力されるハッシュテーブル領域に関する情報について，次の表に示します。

ハッシュテーブル領域に関する情報には，ハッシュフィルタの情報も出力されます。

表 10-27　アクセスパスの統計情報の出力項目（ハッシュテーブル領域に関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Hashtbl_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Hashtbl_area_shortage	ハッシュテーブル領域の領域不足の発生有無※1 • Y：領域不足が発生している • N：領域不足が発生していない	CHAR	－

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
3	Hashtbl_wrktbl_bckt_max_row_cnt	ハッシュテーブル領域が不足した場合の作業表への挿入行数の最大値（リアルスレッドごとの最大値）※2	UINT8	WRAP
4	Hashtbl_filter_num	ハッシュフィルタ数※3	UINT8	－
5	Hashtbl_filter_disabled_num	無効化されたハッシュフィルタの合計数※4，※5	UINT8	－
6	Hashtbl_sum_filter_check_cnt	ハッシュフィルタでハッシュ値をチェックした回数の合計値※4，※5 全ハッシュフィルタの合計値が出力されます。	UINT8	－
7	Hashtbl_max_filter_check_cnt	ハッシュフィルタでハッシュ値をチェックした回数の最大値※4，※5 ハッシュフィルタごとにチェックした回数のうち，最も大きな値が出力されます。	UINT8	－
8	Hashtbl_min_filter_check_cnt	ハッシュフィルタでハッシュ値をチェックした回数の最小値※4，※5 ハッシュフィルタごとにチェックした回数のうち，最も小さな値が出力されます。	UINT8	－
9	Hashtbl_sum_filtering_cnt	ハッシュ値がハッシュフィルタで除外された回数の合計値※4，※5 全ハッシュフィルタの合計値が出力されます。	UINT8	－
10	Hashtbl_max_filtering_cnt	ハッシュ値がハッシュフィルタで除外された回数の最大値※4，※5 ハッシュフィルタごとの除外された回数のうち，最も大きな値が出力されます。	UINT8	－
11	Hashtbl_min_filtering_cnt	ハッシュ値がハッシュフィルタで除外された回数の最小値※4，※5 ハッシュフィルタごとの除外された回数のうち，最も小さな値が出力されます。	UINT8	－

（凡例）

UINT8：8バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0または実行途中の値が出力されることがあります。

CHAR：『出力される内容』で示した文字列が出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，空白が出力されることがあります。

WRAP：18,446,744,073,709,551,615を超えると，ラップアラウンドして最小値に戻ります。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

注※1

ハッシュ検索が複数回実行される次の処理の場合，すべてのハッシュ検索を通した情報が出力されます。

- 再帰的問合せ内にある，再帰的問合せ名を参照している検索処理で実行されるハッシュ検索
- 同一問合せ指定中に引数が異なるDISTINCT 集合関数を複数指定した場合に，各DISTINCT 集合関数の結果を求めるために実行される複数回のグローバルハッシュグループ化
- 分割実行されるハッシュジョイン，または分割実行されるハッシュジョインに含まれるハッシュジョイン

注※2

ハッシュ検索が複数回実行される次の処理の場合，ハッシュ検索ごとの値のうち，最も大きい値が出力されます。

- 再帰的問合せ内にある，再帰的問合せ名を参照している検索処理で実行されるハッシュ検索
- 同一問合せ指定中に引数が異なるDISTINCT 集合関数を複数指定した場合に，各DISTINCT 集合関数の結果を求めるために実行される複数回のグローバルハッシュグループ化
- 分割実行されるハッシュジョイン，または分割実行されるハッシュジョインに含まれるハッシュジョイン

注※3

ハッシュフィルタが適用されない SQL 文の場合，0 が出力されます。adb_sql_exe_hashflt_area_size オペランドが0 の場合も，0 が出力されます。

注※4

ハッシュフィルタが適用されない SQL 文の場合，0 が出力されます。adb_sql_exe_hashflt_area_size オペランドが0 の場合，およびハッシュフィルタサイズが十分でない場合も，0 が出力されます。

注※5

ハッシュ検索が複数回実行される次の処理の場合，すべてのハッシュ検索を通して合計した値に関する情報が出力されます。

- 再帰的問合せ内にある，再帰的問合せ名を参照している検索処理で実行されるハッシュ検索
- 分割実行されるハッシュジョイン，または分割実行されるハッシュジョインに含まれるハッシュジョイン

なお，ハッシュフィルタに関する項目については，ハッシュフィルタごとに合計した値に関する情報が出力されます。

(f) 外への参照列を含む副問合せに関する情報

アクセスパスの統計情報に出力される外への参照列を含む副問合せに関する情報について，次の表に示します。

表 10-28　アクセスパスの統計情報の出力項目（外への参照列を含む副問合せに関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Corsubq_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Corsubq_request_cnt	外への参照列を含む副問合せを要求した回数※	UINT8	WRAP

（凡例）

UINT8：8 バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0 または実行途中の値が出力されることがあります。

WRAP：18,446,744,073,709,551,615 を超えると，ラップアラウンドして最小値に戻ります。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

注※

再帰的問合せ内にある，再帰的問合せ名を参照している検索処理中の，外への参照列を含む副問合せである場合は，再帰ごとの値を合計した値が出力されます。

(g)　副問合せの結果を格納するキャッシュに関する情報

アクセスパスの統計情報に出力される副問合せの結果を格納するキャッシュに関する情報について，次の表に示します。

表 10-29　アクセスパスの統計情報の出力項目（副問合せの結果を格納するキャッシュに関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Corsubqc_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Corsubqc_request_cnt	キャッシュを要求した回数※	UINT8	WRAP
3	Corsubqc_exec_cnt	キャッシュにヒットしないで副問合せを実行した回数※	UINT8	WRAP

（凡例）

UINT8：8 バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0 または実行途中の値が出力されることがあります。

WRAP：18,446,744,073,709,551,615 を超えると，ラップアラウンドして最小値に戻ります。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

注※

再帰的問合せ内にある，再帰的問合せ名を参照している検索処理中の，副問合せである場合，再帰ごとの値を合計した値が出力されます。

(h) 再帰的問合せに関する情報

アクセスパスの統計情報に出力される再帰的問合せに関する情報について，次の表に示します。

表 10-30 アクセスパスの統計情報の出力項目（再帰的問合せに関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	Recurq_path#	対応するアクセスパスのツリー行番号	UINT8	－
2	Recurq_row_cnt	再帰的問合せの処理行数	UINT8	WRAP
3	Recurq_start_time	再帰的問合せによる処理の開始日時 再帰的問合せによる処理を開始した日時が，次の形式で出力されます。 *YYYY-MM-DD*△*hh*:*mm*:*ss*.*nnnnnn*※	CHAR	－
4	Recurq_end_time	再帰的問合せによる処理の終了日時 再帰的問合せによる処理を終了した日時が，次の形式で出力されます。 *YYYY-MM-DD*△*hh*:*mm*:*ss*.*nnnnnn*※	CHAR	－
5	Recurq_max_recursion	最大再帰数指定の値	UINT8	－
6	Recurq_recursion_cnt	再帰した数	UINT8	WRAP

（凡例）

UINT8：8 バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0 または実行途中の値が出力されることがあります。

CHAR：『出力される内容』で示した文字列が出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，空白が出力されることがあります。

WRAP：18,446,744,073,709,551,615 を超えると，ラップアラウンドして最小値に戻ります。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

注※

年（*YYYY*）は 4 桁で出力されます。月（*MM*），日（*DD*），時（*hh*），分（*mm*），および秒（*ss*）は 2 桁で出力されます。マイクロ秒（*nnnnnn*）は 6 桁で出力されます。桁数が足りない場合は，足りない分，左側に 0 が補われます。△は半角空白を表します。

(i)　SQL 文の実行に関する情報

アクセスパスの統計情報に出力される SQL 文の実行に関する情報について，次の表に示します。

表 10-31　アクセスパスの統計情報の出力項目（SQL 文の実行に関する情報）

項番	列のタイトル	出力される内容	出力形式	オーバフローが発生した場合
1	SQLexec_result_wrk_size	SQL 文の実行結果を格納する領域のサイズ（単位：バイト）	UINT8	－

（凡例）

UINT8：８バイトの符号なし整数で出力されます。なお，対象のアクセスパスを実行しなかったり，対象のアクセスパスが実行途中だったりした場合は，0 または実行途中の値が出力されることがあります。

－：オーバフローは発生しません。または，該当しません。

注

表に記載されていない出力項目は，すべて保守情報です。

(3)　アクセスパスの統計情報に関する注意事項

- すべてのアクセスパスが，アクセスパスの統計情報の出力対象となるわけではありません。そのため，アクセスパスの統計情報と SQL 文の統計情報を突き合わせた場合，出力内容が不一致となることがあります。アクセスパスの統計情報に出力される項目については，「(2)　アクセスパスの統計情報に出力される項目」を参照してください。
- アクセスパスの統計情報の出力対象となる SQL 文については，マニュアル『HADB　AP 開発ガイド』の『AP のチューニング』の『アクセスパスの見方（SQL 文の実行計画の見方)』の『アクセスパスとは』にある『アクセスパスの出力対象となる SQL 文』を参照してください。
- ？パラメタの値の一括転送を使用して実行した SQL 文の場合，アクセスパスの統計情報は出力されません。ただし，？パラメタの値の一括転送を使用した場合でも，？パラメタの組を 1 個しか指定していないときは，アクセスパスの統計情報が出力されます。？パラメタの値の一括転送については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『？パラメタの値の一括転送』を参照してください。

10.11.4　SQL トレース情報の出力例と見方

複数のコネクションが存在する場合，各コネクションの SQL トレース情報が混在して出力されます。複数のコネクションが存在する場合の SQL トレース情報の出力例と見方を説明します。

前提条件

複数のコネクションから次の SQL 文が実行され，各 SQL 文の SQL トレース情報が出力されています。

- SELECT * FROM T2

コネクション ID が1 のコネクションで実行した SQL 文です。
出力例中では，『[1] の SQL 文』と説明しています。

- INSERT INTO ADBUSER01.T2 VALUES (?,?)
  コネクション ID が2 のコネクションで実行した SQL 文です。
  出力例中では，『[2] の SQL 文』と説明しています。

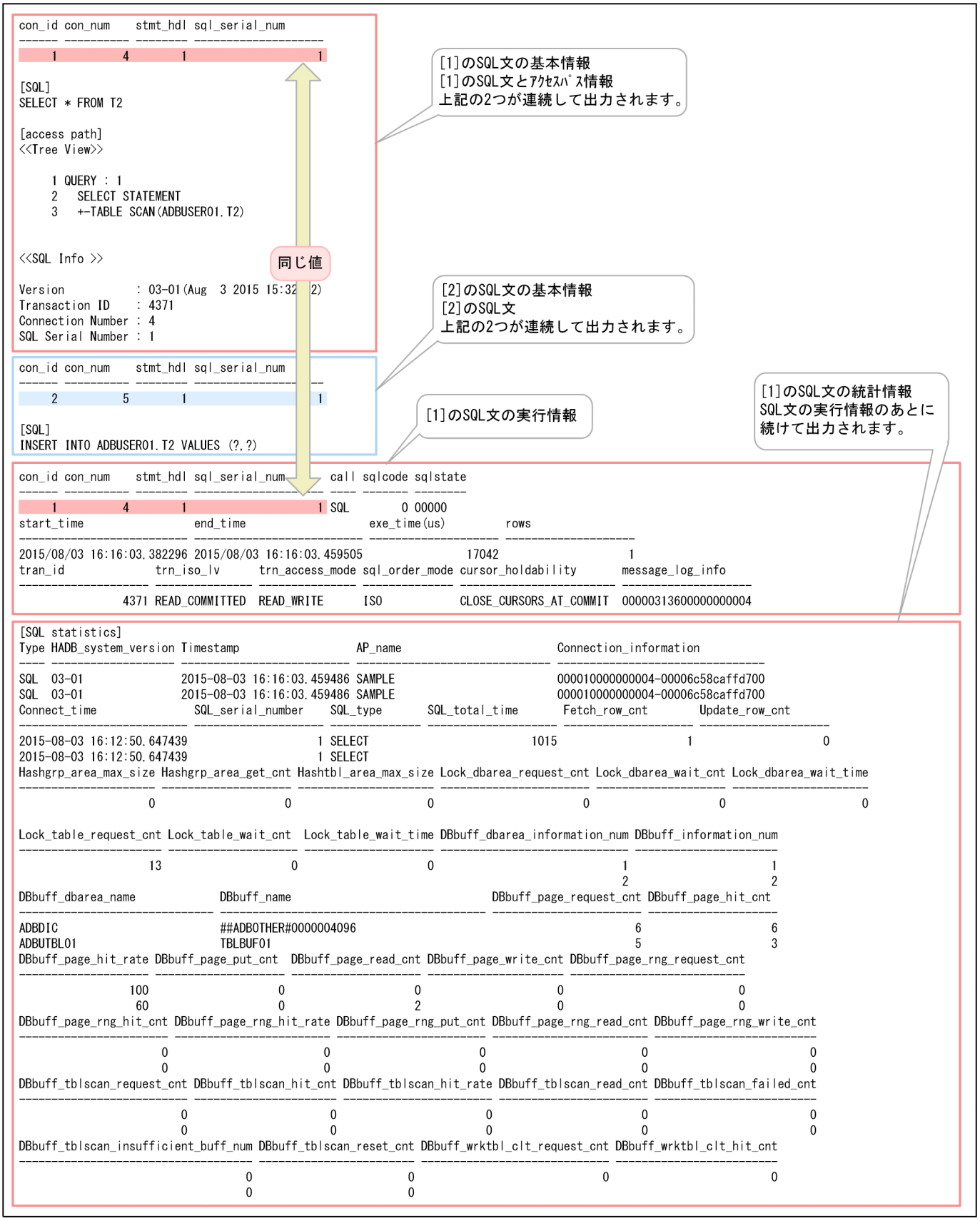
メモ

1 コネクションの SQL 文の基本情報または SQL 文の実行情報と，その関連情報※は連続して出力されます。これらの情報の間に，ほかのコネクションの情報が出力されることはありません。したがって，SQL 文の基本情報または SQL 文の実行情報に出力されるcon_id，con_num，stmt_hdl，およびsql_serial_num を使って，実行した SQL 文に対応する関連情報を探すことができます。

注※

関連情報とは，「10.11.2　SQL トレース情報に出力される情報」の「(3)　クライアント定義情報」～「(11)　アクセスパスの統計情報」で説明している情報のことです。

■SQL トレース情報の出力例（その 1）



■SQL トレース情報の出力例（その 2）

```
DBbuff_wrktbl_clt_hit_rate DBbuff_wrktbl_clt_put_cnt DBbuff_wrktbl_clt_read_cnt DBbuff_wrktbl_clt_write_cnt DBbuff_wrktbl_clt_tbl_cnt
-------------------------- ------------------------- -------------------------- --------------------------- --------------------------
                         0                         0                          0                           0                          0

DBbuff_wrktbl_clt_sort_merge_cnt DBbuff_wrktbl_clt_sort_page_max_cnt Log_usrbuf_out_cnt   Log_usrfile_max_size DBarea_extension_cnt
-------------------------------- ------------------------------------ ------------------ -------------------- ---------------------
                               0                                    0                  0                    0                     0

Log_usrfile_write_cnt DBbuff_wrktbl_clt_request_fix_cnt DBbuff_wrktbl_clt_pagein_fix_cnt DBbuff_wrktbl_clt_pageout_fix_cnt
--------------------- ------------------------------------ ------------------------------------ -------------------------------------
                    0                                    0                                    0                                     0

DBbuff_wrktbl_clt_max_used_blk_num Directory_send_num   Directory_recv_num   Log_send_num         Log_recv_num
---------------------------------- -------------------- -------------------- -------------------- --------------------
                                 0                    0                    0                    0                    0

DBbuff_send_num      DBbuff_recv_num      Node_com_num         Node_com_time        Bidx_page_split_cnt  Tidx_page_split_cnt
-------------------- -------------------- -------------------- -------------------- -------------------- --------------------
                   0                    0                    0                    0                    0                    0

Bidx_validation_check_cnt Ridx_sgmt_skip_cnt   Ridx_chunk_skip_cnt  Ridx_chunk_read_cnt  Ridx_chunk_judge_cnt Ridx_sgmt_read_cnt
------------------------- -------------------- -------------------- -------------------- -------------------- --------------------
                        1                    0                    0                    0                    0                    0

Ridx_sgmt_judge_cnt  SegmentRel_rthd_max_num Csvread_file_cnt     Csvread_file_read_size Hashgrp_area_shortage
-------------------- ---------------------- -------------------- ---------------------- --------------------
                   0                      0                    0                      0 N

Hashgrp_area_sufficient_size Hashtbl_area_shortage Syndict_file_access_time Auditread_file_cnt   Auditread_file_read_size
---------------------------- --------------------- ------------------------ -------------------- ------------------------
                           0 N                                             0                    0                        0

DBbuff_tblscan_read_size DBbuff_tblscan_rthd_min_size DBbuff_tblscan_rthd_max_size DBbuff_tblscan_use_size
------------------------ ----------------------------- ----------------------------- ------------------------
                       0                             0                             0                        0
                       0                             0                             0                        0
DBbuff_tblscan_insufficient_buff_size Hashflt_disabled Tbldef_req_cnt       Tbldef_access_cnt    Tbldef_cache_access_cnt
------------------------------------- ---------------- -------------------- -------------------- -----------------------
                                    0 N                                    1                    1                       0
                                    0
Tbldef_cache_register_cnt Tbldef_cache_sweep_cnt DBbuff_page_wait_cnt Max_sql_rthd_num     Hashtbl_area_size    Hashflt_area_size
------------------------- ------------------------ ------------------------ -------------------- -------------------- --------------------
                        1                        0                        0                    4                 2000                  200
                                                                          0
Parallel_com_send_num Parallel_com_time_sum Parallel_com_time_max Parallel_com_recv_num
--------------------- --------------------- --------------------- ---------------------
                    0                     0                     0                     0
```

■SQL トレース情報の出力例（その 3）

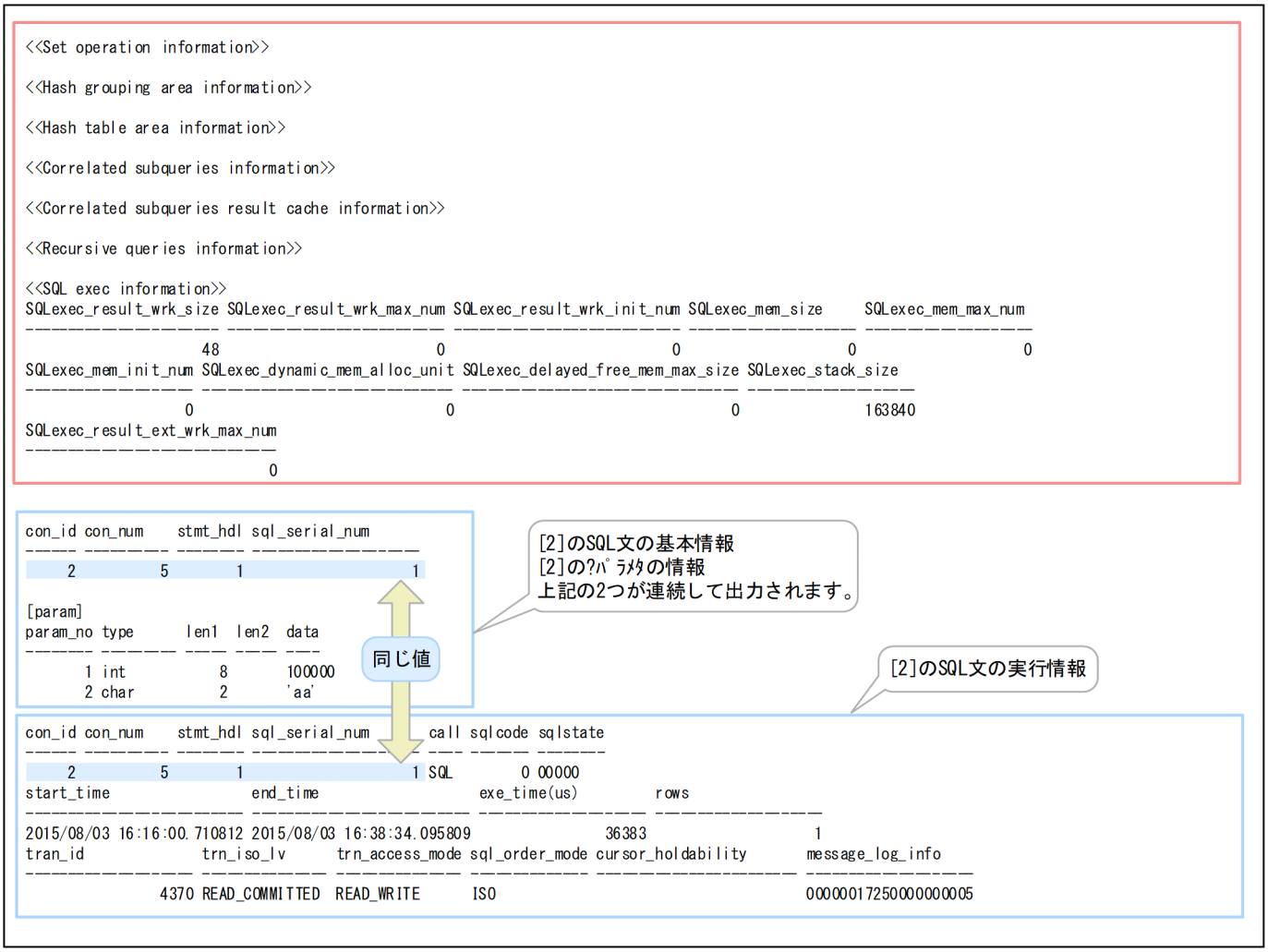
```
[access path statistics]
<<Data access information>>
Data_path#          Data_dbbuff_page_request_cnt(table) Data_dbbuff_page_hit_cnt(table)
------------------ ----------------------------------- ---------------------------
                 3                                   5                           3
Data_dbbuff_page_read_cnt(table) Data_dbbuff_page_write_cnt(table) Data_dbbuff_tblscan_request_cnt
-------------------------------- --------------------------------- ---------------------------
                               2                                 0                           0
Data_dbbuff_tblscan_hit_cnt Data_dbbuff_tblscan_read_cnt Data_dbbuff_tblscan_read_size Data_dbbuff_tblscan_failed_cnt
--------------------------- --------------------------- --------------------------- ------------------------------
                          0                           0                           0                              0
Data_dbbuff_page_request_cnt(btree) Data_dbbuff_page_hit_cnt(btree) Data_dbbuff_page_read_cnt(btree)
----------------------------------- ------------------------------- ------------------------------
                                  0                               0                              0
Data_dbbuff_page_write_cnt(btree) Data_dbbuff_page_request_cnt(text) Data_dbbuff_page_hit_cnt(text)
--------------------------------- ---------------------------------- ---------------------------
                                0                                  0                           0
Data_dbbuff_page_read_cnt(text) Data_dbbuff_page_write_cnt(text) Data_dbbuff_page_request_cnt(range)
------------------------------- -------------------------------- ----------------------------------
                              0                                0                                  0
Data_dbbuff_page_hit_cnt(range) Data_dbbuff_page_read_cnt(range) Data_dbbuff_page_write_cnt(range)
------------------------------- -------------------------------- ---------------------------------
                              0                                0                                 0
Data_dbbuff_wrktbl_clt_tbl_cnt Data_dbbuff_wrktbl_clt_request_cnt Data_dbbuff_wrktbl_clt_page_hit_cnt
------------------------------ ------------------------------- ------------------------------
                             0                               0                              0
Data_dbbuff_wrktbl_clt_read_cnt Data_dbbuff_wrktbl_clt_write_cnt Data_dbbuff_page_request_cnt(glb_wrk)
------------------------------- -------------------------------- ----------------------------------
                              0                                0                                  0
Data_dbbuff_page_hit_cnt(glb_wrk) Data_dbbuff_page_read_cnt(glb_wrk) Data_dbbuff_page_write_cnt(glb_wrk)
--------------------------------- -------------------------------- ---------------------------------
                                0                                0                                 0
Data_bidx_page_split_cnt Data_bidx_validation_check_cnt Data_ridx_chunk_judge_cnt Data_ridx_chunk_skip_cnt
------------------------ ------------------------------ ------------------------ ------------------------
                       0                              0                        0                        0
Data_ridx_sgmt_judge_cnt Data_ridx_sgmt_skip_cnt Data_tidx_page_split_cnt Data_wrktbl_create_cnt
------------------------ ----------------------- ------------------------ ------------------------
                       0                       0                        0                       0
Data_wrktbl_drop_cnt Data_wrktbl_page_assign_num Data_wrktbl_page_free_num Data_wrktbl_page_use_max
-------------------- ------------------------- ------------------------- -----------------------
                   0                         0                         0                       0
Data_wrktbl_sort_merge_cnt Data_wrktbl_sort_page_max_cnt Data_dbarea_extension_cnt Data_segmentrel_rthd_max_num
-------------------------- ---------------------------- ------------------------- ----------------------------
                         0                            0                         0                            0
Data_log_usrfile_write_cnt Data_log_usrbuf_out_cnt Data_log_usrfile_max_size Data_directory_send_num
-------------------------- ----------------------- ------------------------- ----------------------
                         0                       0                         0                      0
Data_access_info_type    Data_csvread_file_cnt Data_csvread_file_read_size Data_deleted_rows_cnt Data_tidx_all_search_cnt
------------------------ --------------------- ---------------------------- --------------------- ------------------------
Access                                       0                            0                     0                        0
Data_auditread_file_cnt Data_auditread_file_read_size Data_grpagg_mem_short_cnt Data_grpagg_mem_exceed_limit_cnt
----------------------- ------------------------------ ------------------------ ------------------------------
                      0                              0                        0                              0
Data_grpagg_mem_try_get_cnt Data_grpagg_mem_max_size Data_grpagg_preagg_row_cnt Data_grpagg_rslttbl_reuse_cnt
--------------------------- ------------------------ ------------------------- ---------------------------
                          0                        0                         0                           0
Data_cs_column_invalid_rows_cnt Data_cs_rowstore_fetch_rows_cnt Data_cs_invalid_info_page_request_cnt
------------------------------- ------------------------------- ---------------------------------------
                              0                               0                                       0
Data_cs_invalid_false_positive_cnt Data_scan_info_save_sum_cnt Data_scan_info_save_max_cnt Data_scan_info_save_min_cnt
---------------------------------- ------------------------- ------------------------- -------------------------
                                 0                         0                         0                         0
Data_scan_info_restore_sum_cnt Data_scan_info_restore_max_cnt Data_scan_info_restore_min_cnt Data_dbbuff_page_request_max_cnt(table)
------------------------------ ------------------------------ ------------------------------ ---------------------------------------
                             0                              0                              0                                       5
Data_dbbuff_page_request_min_cnt(table) Data_dbbuff_page_read_max_cnt(table) Data_dbbuff_page_read_min_cnt(table)
--------------------------------------- -------------------------------------- --------------------------------------
                                      0                                      2                                      0
Data_dbbuff_tblscan_request_max_cnt Data_dbbuff_tblscan_request_min_cnt Data_dbbuff_tblscan_read_max_cnt
----------------------------------- ----------------------------------- -----------------------------------
                                  0                                   0                                   0
Data_dbbuff_tblscan_read_min_cnt Data_dbbuff_tblscan_read_max_size Data_dbbuff_tblscan_read_min_size
-------------------------------- -------------------------------- --------------------------------
                               0                                0                                0
Data_dbbuff_page_request_max_cnt(btree) Data_dbbuff_page_request_min_cnt(btree) Data_dbbuff_page_read_max_cnt(btree)
--------------------------------------- -------------------------------------- --------------------------------------
                                      0                                      0                                      0
```

[1]のｱｸｾｽﾊﾟｽの統計情報

■SQL トレース情報の出力例（その 4）

```
Data_dbbuff_page_read_min_cnt(btree) Data_dbbuff_page_request_max_cnt(text) Data_dbbuff_page_request_min_cnt(text)
------------------------------------ ----------------------------------- -----------------------------------
                                   0                                   0                                   0
Data_dbbuff_page_read_max_cnt(text) Data_dbbuff_page_read_min_cnt(text) Data_dbbuff_page_request_max_cnt(range)
----------------------------------- ----------------------------------- ------------------------------------
                                  0                                   0                                   0
Data_dbbuff_page_request_min_cnt(range) Data_dbbuff_page_read_max_cnt(range) Data_dbbuff_page_read_min_cnt(range)
--------------------------------------- -------------------------------------- --------------------------------------
                                      0                                      0                                      0
Data_dbbuff_wrktbl_clt_request_max_cnt Data_dbbuff_wrktbl_clt_request_min_cnt Data_dbbuff_wrktbl_clt_read_max_cnt
-------------------------------------- -------------------------------------- -------------------------------------
                                     0                                      0                                      0
Data_dbbuff_wrktbl_clt_read_min_cnt Data_dbbuff_wrktbl_clt_write_max_cnt Data_dbbuff_wrktbl_clt_write_min_cnt
------------------------------------ ------------------------------------- -------------------------------------
                                   0                                     0                                     0
Data_dbbuff_page_request_max_cnt(glb_wrk) Data_dbbuff_page_request_min_cnt(glb_wrk) Data_dbbuff_page_read_max_cnt(glb_wrk)
----------------------------------------- ----------------------------------------- ----------------------------------------
                                        0                                         0                                         0
Data_dbbuff_page_read_min_cnt(glb_wrk) Data_dbbuff_page_write_max_cnt(glb_wrk) Data_dbbuff_page_write_min_cnt(glb_wrk)
-------------------------------------- --------------------------------------- ---------------------------------------
                                     0                                       0                                       0
Data_wrktbl_page_branch_num
---------------------------
                          0

<<Scan information>>
Scan_path#          Scan_row_cnt        Scan_start_time            Scan_end_time
------------------- ------------------- -------------------------- --------------------------
                 3                   1 2015-08-03 16:16:03.458122 2015-08-03 16:16:03.459137
Scan_info_data_max_num Scan_info_data_max_free_num Scan_info_data_size  Scan_info_inpred_max_num
---------------------- ---------------------------- ------------------- -----------------------
                     0                            0                   0                       0
Scan_info_inpred_max_free_num Scan_info_inpred_size Scan_info_quantpred_max_num Scan_info_quantpred_max_free_num
----------------------------- ---------------------- ------------------------- -----------------------------
                            0                      0                         0                             0
Scan_info_quantpred_size Scan_info_drvtbl_max_num Scan_info_drvtbl_max_free_num Scan_info_drvtbl_size Scan_info_hash_max_num
------------------------ ------------------------ ----------------------------- ---------------------- ---------------------
                       0                        0                             0                      0                     0
Scan_info_hash_max_free_num Scan_info_hash_size  Scan_info_hashbckt_max_num Scan_info_hashbckt_size
--------------------------- ---------------------- ---------------------------- ------------------------
                          0                      0                            0                        0
Scan_outerjoin_info_max_num Scan_outerjoin_info_size Scan_info_csvread_file_max_num Scan_info_csvread_file_max_free_num
--------------------------- ------------------------ ------------------------------ ----------------------------------
                          0                        0                              0                                  0
Scan_info_csvread_file_size Scan_info_csvread_data_max_num Scan_info_csvread_data_max_free_num Scan_info_csvread_data_size
--------------------------- ------------------------------ ------------------------------------ ----------------------------
                          0                              0                                    0                            0
Scan_max_rt_row_cnt  Scan_min_rt_row_cnt  Scan_sum_rt_aggr_row_cnt Scan_max_rt_aggr_row_cnt Scan_min_rt_aggr_row_cnt
-------------------- -------------------- ------------------------ ------------------------ ------------------------
                   1                    0                        0                        0                        0
Scan_sum_rt_array_any_eval_cnt Scan_max_rt_array_any_eval_cnt Scan_min_rt_array_any_eval_cnt Scan_mn_sum_rt_send_rdata_size
------------------------------ ------------------------------ ------------------------------ ------------------------------
                             0                              0                              0                              0
Scan_mn_sum_rt_send_rdata_cnt Scan_mn_sum_rt_recv_rdata_size Scan_mn_sum_rt_recv_rdata_cnt Scan_mn_sum_rt_send_odata_size
----------------------------- ------------------------------ ------------------------------ -----------------------------
                            0                              0                              0                             0
Scan_mn_sum_rt_send_odata_cnt Scan_mn_sum_rt_recv_odata_size Scan_mn_sum_rt_recv_odata_cnt
----------------------------- ------------------------------ ------------------------------
                            0                              0                              0
```

■SQL トレース情報の出力例（その 5）



■SQL トレース情報の出力例（その 6）

[2]のSQL文の統計情報
SQL文の実行情報のあとに
続けて出力されます。

```
[SQL statistics]
Type HADB_system_version Timestamp                  AP_name                        Connection_information
---- ------------------- ------------------------- ------------------------------ ------------------------------
SQL  03-01               2015-08-03 16:38:34.095803 PERFORMACEMEASURE              000020000000005-00006c58c95fa700
SQL  03-01               2015-08-03 16:38:34.095803 PERFORMACEMEASURE              000020000000005-00006c58c95fa700
SQL  03-01               2015-08-03 16:38:34.095803 PERFORMACEMEASURE              000020000000005-00006c58c95fa700
Connect_time              SQL_serial_number    SQL_type       SQL_total_time       Fetch_row_cnt        Update_row_cnt
------------------------- -------------------- -------------- -------------------- -------------------- --------------------
2015-08-03 16:13:16.011371                    1 INSERT                       18982                    0                    1
2015-08-03 16:13:16.011371                    1 INSERT
2015-08-03 16:13:16.011371                    1 INSERT
Hashgrp_area_max_size Hashgrp_area_get_cnt Hashtbl_area_max_size Lock_dbarea_request_cnt Lock_dbarea_wait_cnt Lock_dbarea_wait_time
--------------------- -------------------- --------------------- ----------------------- --------------------- ---------------------
                    0                    0                     0                       1                     0                     0


Lock_table_request_cnt Lock_table_wait_cnt  Lock_table_wait_time DBbuff_dbarea_information_num DBbuff_information_num
---------------------- -------------------- -------------------- ---------------------------- ----------------------
                    13                    0                    0                            1                      1
                                                                                            2                      1
                                                                                            3                      2
DBbuff_dbarea_name             DBbuff_name                                 DBbuff_page_request_cnt DBbuff_page_hit_cnt
------------------------------ ------------------------------------------- ----------------------- --------------------
ADBDIC                         ##ADBOTHER#0000004096                                            31                   22
ADBSTBL                        ##ADBOTHER#0000004096                                             6                    2
ADBUTBL01                      TBLBUF01                                                          3                    3
DBbuff_page_hit_rate DBbuff_page_put_cnt  DBbuff_page_read_cnt DBbuff_page_write_cnt DBbuff_page_rng_request_cnt
-------------------- -------------------- -------------------- --------------------- ---------------------------
                  71                    0                    9                     0                           0
                  34                    0                    4                     0                           0
                 100                    1                    0                     1                           0
DBbuff_page_rng_hit_cnt DBbuff_page_rng_hit_rate DBbuff_page_rng_put_cnt DBbuff_page_rng_read_cnt DBbuff_page_rng_write_cnt
----------------------- ------------------------ ------------------------ ------------------------ -------------------------
                      0                        0                        0                        0                         0
                      0                        0                        0                        0                         0
                      0                        0                        0                        0                         0
DBbuff_tblscan_request_cnt DBbuff_tblscan_hit_cnt DBbuff_tblscan_hit_rate DBbuff_tblscan_read_cnt DBbuff_tblscan_failed_cnt
-------------------------- ---------------------- ------------------------ ------------------------ -------------------------
                         0                      0                        0                        0                         0
                         0                      0                        0                        0                         0
                         0                      0                        0                        0                         0
DBbuff_tblscan_insufficient_buff_num DBbuff_tblscan_reset_cnt DBbuff_wrktbl_clt_request_cnt DBbuff_wrktbl_clt_hit_cnt
------------------------------------ ------------------------ ------------------------------ ------------------------
                                   0                        0                              0                        0
                                   0                        0
                                   0                        0
DBbuff_wrktbl_clt_hit_rate DBbuff_wrktbl_clt_put_cnt DBbuff_wrktbl_clt_read_cnt DBbuff_wrktbl_clt_write_cnt DBbuff_wrktbl_clt_tbl_cnt
-------------------------- ------------------------- -------------------------- --------------------------- -------------------------
                         0                         0                          0                           0                         0


DBbuff_wrktbl_clt_sort_merge_cnt DBbuff_wrktbl_clt_sort_page_max_cnt Log_usrbuf_out_cnt   Log_usrfile_max_size DBarea_extension_cnt
-------------------------------- ---------------------------------- -------------------- -------------------- --------------------
                               0                                  0                    0                    0                    0
```

■SQL トレース情報の出力例（その 7）

```
Log_usrfile_write_cnt DBbuff_wrktbl_clt_request_fix_cnt DBbuff_wrktbl_clt_pagein_fix_cnt DBbuff_wrktbl_clt_pageout_fix_cnt
--------------------- ------------------------------- ------------------------------ -------------------------------
                    1                               0                              0                               0

DBbuff_wrktbl_clt_max_used_blk_num Directory_send_num   Directory_recv_num   Log_send_num         Log_recv_num
---------------------------------- ------------------ ------------------- -------------------- --------------------
                                 0                  0                   0                    0                    0

DBbuff_send_num      DBbuff_recv_num      Node_com_num         Node_com_time        Bidx_page_split_cnt  Tidx_page_split_cnt
-------------------- -------------------- -------------------- -------------------- -------------------- --------------------
                   0                    0                    0                    0                    0                    0

Bidx_validation_check_cnt Ridx_sgmt_skip_cnt   Ridx_chunk_skip_cnt  Ridx_chunk_read_cnt  Ridx_chunk_judge_cnt Ridx_sgmt_read_cnt
------------------------- -------------------- -------------------- -------------------- -------------------- --------------------
                       15                    0                    0                    0                    0                    0

Ridx_sgmt_judge_cnt  SegmentRel_rthd_max_num Csvread_file_cnt     Csvread_file_read_size Hashgrp_area_shortage
-------------------- ---------------------- -------------------- ---------------------- --------------------
                   0                      0                    0                      0 N

Hashgrp_area_sufficient_size Hashtbl_area_shortage Syndict_file_access_time Auditread_file_cnt   Auditread_file_read_size
---------------------------- -------------------- ------------------------ -------------------- ------------------------
                           0 N                                            0                    0                        0

DBbuff_tblscan_read_size DBbuff_tblscan_rthd_min_size DBbuff_tblscan_rthd_max_size DBbuff_tblscan_use_size
------------------------ ---------------------------- ---------------------------- ------------------------
                       0                            0                            0                        0
                       0                            0                            0                        0
                       0                            0                            0                        0
DBbuff_tblscan_insufficient_buff_size Hashflt_disabled Tbldef_req_cnt       Tbldef_access_cnt    Tbldef_cache_access_cnt
------------------------------------- ---------------- -------------------- -------------------- --------------------
                                    0 N                                  1                    0                    1
                                    0
                                    0
Tbldef_cache_register_cnt Tbldef_cache_sweep_cnt DBbuff_page_wait_cnt Max_sql_rthd_num     Hashtbl_area_size    Hashflt_area_size
------------------------- ------------------------ ------------------------ -------------------- -------------------- --------------------
                        0                        0                        0                    4                 2000                  200
                                                                          0
                                                                          0
Parallel_com_send_num Parallel_com_time_sum Parallel_com_time_max Parallel_com_recv_num
--------------------- --------------------- --------------------- ---------------------
                    0                     0                     0                     0
```

メモ

- SQL 文の実行情報は，SQL 文またはコールの終了時刻順に出力されていない可能性があります。ただし，各コネクション内では時刻順に並んでいます。
- SQL トレース情報の出力先ファイルが切り替わった場合，SQL 文の基本情報または SQL 文の実行情報と関連情報は，2 つのファイルに分かれて出力されることがあります。その場合，関連情報が出力された SQL トレースファイルの 1 つ前の番号（番号が最小値だった場合は最後の番号）の SQL トレースファイルに，対応する SQL 文の基本情報または SQL 文の実行情報が出力されます。

10.11.5　SQL トレース情報を出力するための準備

SQL トレース情報を出力する前に，次のことを実施してください。

(1)　ディスク容量を確認する

SQL トレース情報は，SQL トレースファイルに出力されます。1 個当たりの SQL トレースファイルの最大容量は，サーバ定義のadb_sql_trc_txtfile_size オペランドで指定します（デフォルト値は 256 メガバイト）。SQL トレースファイルは最大 8 個作成されるので，ディスク容量に問題がないかを確認してください。

(2)　サーバ定義を指定する

SQL トレース情報を出力するかどうかは，サーバ定義のadb_sql_trc_out オペランドで指定します。デフォルト値はY（SQL トレース情報を出力する）です。

SQL トレース情報は，トラブルシュートやチューニングの際に有用となる情報のため，基本的にはすべての SQL トレース情報を取得することを推奨します。ただし，SQL トレース情報を出力する場合，SQL トレース情報がファイルに出力される分，SQL 文の処理性能に影響を与えることがあります。また，出力される情報量が多いほど，処理性能に与える影響も大きくなる可能性があります。SQL 文の処理性能に与える影響が大きい場合は，次のサーバ定義のオペランドで，SQL トレース情報の出力量を削減することができます。

- adb_sql_trc_param

  ？パラメタの情報を出力するかどうかを選択するオペランドです。デフォルト値はY（？パラメタの情報を出力する）です。
- adb_sql_trc_accesspath

  アクセスパス情報およびアクセスパスの統計情報を出力するかどうかを選択するオペランドです。デフォルト値はY（アクセスパス情報およびアクセスパスの統計情報を出力する）です。
- adb_sql_trc_level

  SQL トレース情報を出力する単位を指定するオペランドです。デフォルト値はSQL（SQL トレース情報を SQL 文単位で出力する）です。

> **メモ**
>
> HADB サーバの開始後に SQL トレース情報の出力を停止したり，出力する SQL トレース情報を変更したりすることがadbchgsqltrc コマンドでできます。

10.11.6　SQL トレース情報を出力しているときの運用

SQL トレース情報を出力しているときの運用について説明します。

「(1)　SQL トレースファイルのバックアップを取得する」は，SQL トレース情報を出力しているときに行う定期的な運用を説明しています。「(2)　SQL トレース情報の出力情報を確認したい場合」以降は，必要時に行う運用を説明しています。

(1) SQL トレースファイルのバックアップを取得する

SQL トレース情報が出力された結果，SQL トレースファイルの容量が満杯になった場合，別の SQL トレースファイルに出力先を切り替えます。SQL トレースファイルは全部で８つあり，すべての SQL トレースファイルが満杯になった場合は，いちばん古いファイルから順次上書きされていきます。そのため，SQL トレース情報を保存しておきたい場合は，SQL トレースファイルのバックアップを取得してください。

SQL トレースファイルの出力先が切り替わったときに，KFAA81004-I メッセージがメッセージログファイルに出力されます。このメッセージが出力されたときに，切り替え元の SQL トレースファイルのバックアップを取得するようにしてください。

(2) SQL トレース情報の出力情報を確認したい場合

SQL トレース情報に何の情報が出力されているかを確認したい場合は，adbchgsqltrc コマンドを実行してください。

■コマンドの実行例

```
adbchgsqltrc -d

sqltrace param    accesspath level
ACTIVE   ACTIVE   INACTIVE   SQL
```

[説明]

- sqltrace

  SQL トレース情報が出力されているかどうかが表示されます。この例では，ACTIVE（SQL トレース情報が出力されている）が表示されています。

- param

  ？パラメタの情報が出力されているかどうかが表示されます。この例では，ACTIVE（？パラメタの情報が出力されている）が表示されています。

- accesspath

  アクセスパス情報およびアクセスパスの統計情報が出力されているかどうかが表示されます。この例では，INACTIVE（アクセスパス情報およびアクセスパスの統計情報が出力されていない）が表示されています。

- level

  SQL トレース情報の出力単位が表示されます。この例では，SQL（SQL 文単位で SQL トレース情報を出力する）が表示されています。

adbchgsqltrc コマンドの実行結果の詳細については，マニュアル『HADB　コマンドリファレンス』の『adbchgsqltrc（SQL トレース情報の出力開始・停止)』を参照してください。

(3) SQL トレース情報の出力情報を変更したい場合

SQL トレース情報の出力情報を変更したい場合は，adbchgsqltrc コマンドを実行してください。

■コマンドの実行例

（例 1）？パラメタの情報の出力を停止します。

```
adbchgsqltrc -n param
```

[説明]

- -n param
  ？パラメタの情報の出力を停止する指定です。

（例 2）アクセスパス情報およびアクセスパスの統計情報の出力を開始します。出力単位はコール単位にします。

```
adbchgsqltrc -y accesspath -l call
```

[説明]

- -y accesspath
  アクセスパス情報およびアクセスパスの統計情報の出力を開始する指定です。
- -l call
  出力単位をコール単位にする指定です。

adbchgsqltrc コマンドを実行する際のオプションの指定については，マニュアル『HADB　コマンドリファレンス』の『adbchgsqltrc（SQL トレース情報の出力開始・停止)』を参照してください。

なお，adbchgsqltrc コマンドによる SQL トレース情報の出力情報の変更は，次回の HADB サーバの開始時には適用されません。次回の HADB サーバの開始時には，サーバ定義の指定が適用されます。

(4) SQL トレース情報の出力を停止したい場合

SQL トレース情報の出力を停止したい場合は，adbchgsqltrc コマンドを実行してください。

■コマンドの実行例

```
adbchgsqltrc -e
```

adbchgsqltrc コマンドを実行する際のオプションの指定については，マニュアル『HADB　コマンドリファレンス』の『adbchgsqltrc（SQL トレース情報の出力開始・停止)』を参照してください。

なお，adbchgsqltrc コマンドによる SQL トレース情報の出力停止は，次回の HADB サーバの開始時には適用されません。次回の HADB サーバの開始時には，サーバ定義の指定が適用されます。

(5)　停止していたSQLトレース情報の出力を開始したい場合

停止していたSQLトレース情報の出力を開始したい場合は，adbchgsqltrcコマンドを実行してください。

■コマンドの実行例

```
adbchgsqltrc -s
```

adbchgsqltrcコマンドを実行する際のオプションの指定については，マニュアル『HADB　コマンドリファレンス』の『adbchgsqltrc（SQLトレース情報の出力開始・停止)』を参照してください。

なお，adbchgsqltrcコマンドによるSQLトレース情報の出力開始は，次回のHADBサーバの開始時には適用されません。次回のHADBサーバの開始時には，サーバ定義の指定が適用されます。

(6)　SQLトレース情報の出力先ファイルを確認したい場合

SQLトレース情報がどのSQLトレースファイルに出力されているかは，SQLトレースファイルの更新日時で判断してください。ファイルの更新日時が最も新しいSQLトレースファイルが，現在のSQLトレース情報の出力先になります。OSのlsコマンドなどを使用して，SQLトレースファイルの更新日時を確認してください。

なお，SQLトレースファイルをテキストエディタなどで更新し，ファイルの更新日時を変更してしまった場合は，更新日時から現在の出力先を判断することはできません。この場合，各SQLトレースファイル内に出力されているSQL文の終了日時を確認してください。SQL文の終了日時が最も新しいファイルが出力先となります。

10.11.7　SQLトレース情報を使用したSQL文のエラー原因調査

(1)　SQL文のエラー原因調査の手順

SQLトレース情報を使用したSQL文のエラー原因調査の手順を次に示します。

手順

1. SQL文の実行情報に出力されているSQLCODEを確認する

   SQLCODEがsqlcode欄に出力されます。負の値になっているSQLCODEを確認してください。

   (例)

```
con_id con_num    stmt_hdl sql_serial_num       call sqlcode sqlstate
------ ---------- -------- -------------------- ---- ------- --------
     1          1        1                    1 SQL    -559 42K07
start_time                 end_time                   exe_time(us)         rows
-------------------------- -------------------------- -------------------- --------------------
2015/08/03 19:02:01.185562 2015/08/03 19:02:01.246807                61245                    0
tran_id             trn_iso_lv      trn_access_mode sql_order_mode cursor_holdability       message_log_info
------------------- --------------- --------------- -------------- ------------------------ --------------------
               4375 READ_COMMITTED  READ_WRITE      ISO                                     00000692530000000001
```

2. 出力されたSQLCODE に対応するメッセージを確認する

SQLCODE とメッセージの対応付けの方法については，マニュアル『HADB　メッセージ』の『SQLCODE の見方』を参照してください。

この例の場合，SQLCODE が-559 となっています。そのため，対応するメッセージは，KFAA30559-E メッセージとなります。

3. message_log_info に出力されたメッセージログ情報を記録する

(例)

```
con_id con_num    stmt_hdl sql_serial_num       call sqlcode sqlstate
------ ---------- -------- -------------------- ---- ------- --------
     1          1        1                    1 SQL     -559 42K07
start_time                 end_time                   exe_time(us)         rows
-------------------------- -------------------------- -------------------- --------------------
2015/08/03 19:02:01.185562 2015/08/03 19:02:01.246807                61245                    0
tran_id              trn_iso_lv      trn_access_mode sql_order_mode cursor_holdability       message_log_info
-------------------- --------------- --------------- -------------- ------------------------ --------------------
                4375 READ_COMMITTED  READ_WRITE      ISO                                     00000692530000000001
```

この情報を記録する

4. メッセージログファイルに出力されているメッセージを検索する

メッセージログファイルに出力されている手順 2.で確認したメッセージを検索してください。この例の場合，メッセージログファイルに出力されているKFAA30559-E メッセージを検索します。

また，手順 3.で記録したメッセージログ情報と，メッセージに出力されているメッセージログ情報が一致しているかを確認してください。一致しているKFAA30559-E メッセージが該当するメッセージになります。

(例)

SQLﾄﾚｰｽﾌｧｲﾙ

```
con_id con_num    stmt_hdl sql_serial_num       call sqlcode sqlstate
------ ---------- -------- -------------------- ---- ------- --------
     1          1        1                    1 SQL     -559 42K07
start_time                 end_time                   exe_time(us)         rows
-------------------------- -------------------------- -------------------- --------------------
2015/08/03 19:02:01.185562 2015/08/03 19:02:01.246807                61245                    0
tran_id              trn_iso_lv      trn_access_mode sql_order_mode cursor_holdability       message_log_info
-------------------- --------------- --------------- -------------- ------------------------ --------------------
                4375 READ_COMMITTED  READ_WRITE      ISO                                     00000692530000000001
```

ﾒｯｾｰｼﾞﾛｸﾞﾌｧｲﾙ

```
2015/08/03 19:02:01 0000069011 2113939200 00000692530000000001 KFAA30559-E The operation "ALTER USER" cannot be
executed using an authorization identifier "USER2" that does not exist.
```

5. メッセージに記載されている対処方法を参照する

メッセージに記載されている対処方法を参照して，SQL 文のエラー原因を特定してください。

この例の場合，KFAA30559-E メッセージの対策欄に記載されている対処方法に従ってください。

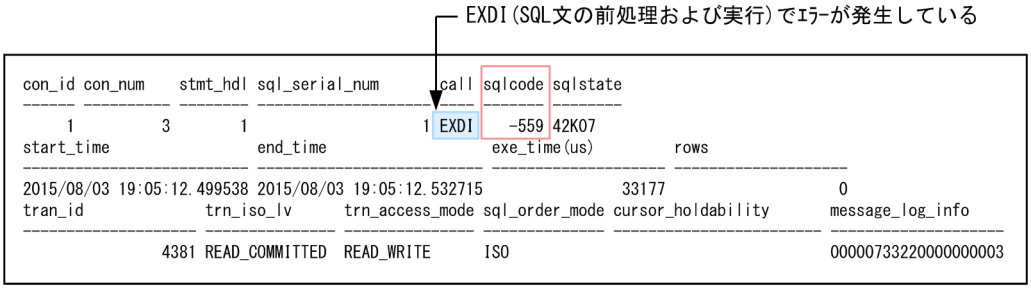
(2)　SQL 文がどこまで実行されたかを特定する方法

SQL トレース情報の出力単位によって方法が異なります。

(a)　コール単位で SQL トレース情報を出力している場合

sqlcode 欄に出力されているSQLCODE が負の値になっているコールでエラーが発生しています。

（例）

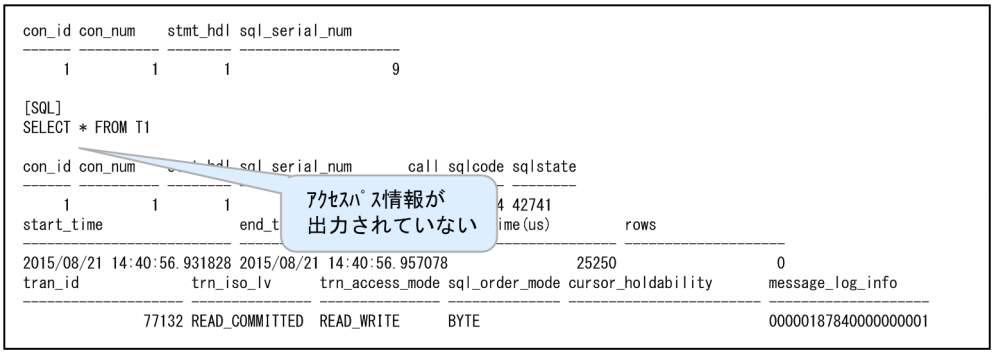


(b)　SQL 文単位で SQL トレース情報を出力している場合

出力された項目と出力されなかった項目を確認して，どのコールでエラーが発生したかを判断します。例を次に示します。

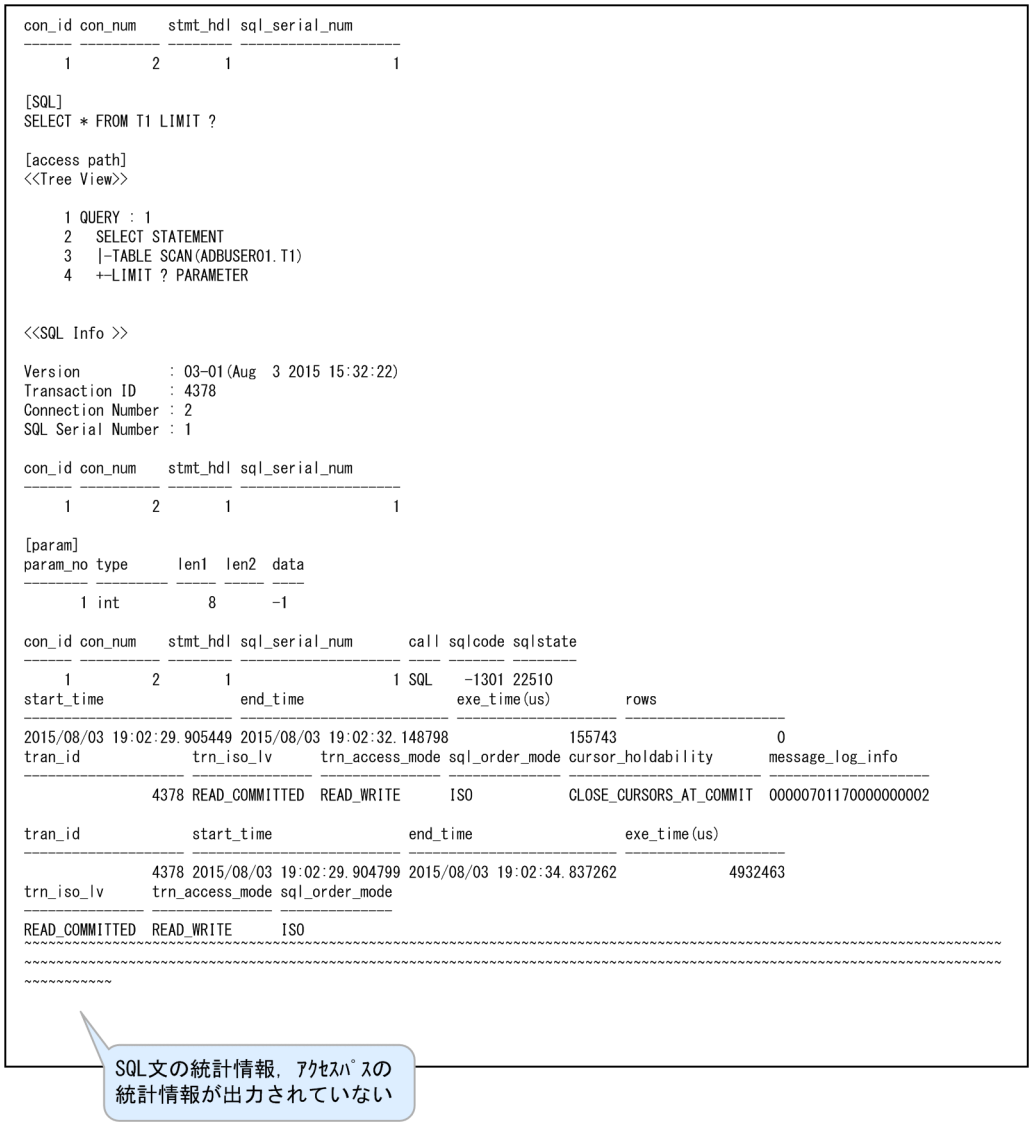
- 例 1：実行した SQL 文は出力されているが，アクセスパス情報は出力されなかった場合

  この場合，PREP（SQL 文の前処理）またはEXDI（SQL 文の前処理および実行）の処理時にエラーが発生しています。



- 例 2：実行した SQL 文，アクセスパス情報，および？パラメタの情報は出力されているが，SQL 文の統計情報およびアクセスパスの統計情報は出力されなかった場合

  この場合，OPEN（カーソルのオープン）の処理時にエラーが発生しています。



10.11.8　SQL トレース情報を利用した SQL 文のチューニング

ここでは，SQL トレース情報を利用した SQL 文のチューニング手順について説明します。

(1)　SQL トレース情報を SQL 文単位で出力している場合

SQL トレース情報を SQL 文単位で出力している場合に，SQL 文をチューニングする手順について次に示します。

手順

1. チューニングに利用する情報を準備する
   SQL トレース情報を参照できるように準備してください。
2. 実行時間の長い SQL 文を特定する

SQL トレース情報の SQL 文の実行情報のexe_time に出力されている SQL 文の実行時間を確認して，実行時間の長い SQL 文を特定してください。

3. SQL 文の実行情報の出力値を記録する

手順 2.で特定した SQL 文の実行情報の次の値を記録してください。

- con_id（コネクション ID）
- con_num（コネクション通番）
- stmt_hdl（SQL 文に割り当てられた文ハンドル）
- sql_serial_num（SQL 文通番）

4. SQL 文の基本情報を確認する

次の条件を満たす SQL 文の基本情報を探してください。

- con_id，con_num，stmt_hdl，およびsql_serial_num の値が，手順 3.で記録した値と同じである

複数の SQL 文の基本情報の中から，該当する SQL 文の基本情報を探す方法については，「10.11.4 SQL トレース情報の出力例と見方」を参照してください。

5. 実行した SQL 文とアクセスパス情報，および？パラメタの情報を確認する

手順 4.で特定した SQL 文の基本情報の直後に出力されている『実行した SQL 文とアクセスパス情報』，および『？パラメタの情報』を確認してください。

6. 統計情報を確認する

手順 2.で特定した SQL 文の実行情報の直後に出力されている次の統計情報を参照して，SQL 文のチューニングを実施してください。

- SQL 文の統計情報
- アクセスパスの統計情報

(2) SQL トレース情報をコール単位で出力している場合

SQL トレース情報をコール単位で出力している場合に，SQL 文をチューニングする手順について次に示します。

手順

1. チューニングに利用する情報を準備する

SQL トレース情報を参照できるように準備してください。

2. 実行時間の長い SQL 文を特定する

SQL トレース情報の SQL 文の実行情報のexe_time に出力されている SQL 文の実行時間を確認して，実行時間の長い SQL 文を特定してください。

3. SQL 文の実行情報の出力値を記録する

手順 2.で特定した SQL 文の実行情報の次の値を記録してください。

- con_id（コネクション ID）

- con_num（コネクション通番）
- stmt_hdl（SQL 文に割り当てられた文ハンドル）
- sql_serial_num（SQL 文通番）

4. SQL 文の実行情報を確認する
   次の条件を満たす SQL 文の実行情報を探してください。
   - con_id，con_num，stmt_hdl，およびsql_serial_num の値が，手順 3.で記録した値と同じである
   - 検索系 SQL を実行した場合は，コール種別がPREP またはOPEN である
   - 更新系 SQL を実行した場合は，コール種別がPREP，EXEC またはEXDI である
5. 実行した SQL 文とアクセスパス情報，および？パラメタの情報を確認する
   手順 4.で特定した SQL 文の実行情報の直後に出力されている次の情報を確認してください。
   - コール種別がPREP またはEXDI の場合は，『実行した SQL 文とアクセスパス情報』を確認する
   - コール種別がOPEN またはEXEC の場合は，『？パラメタの情報』を確認する
6. SQL 文の実行情報を確認する
   次の条件をすべて満たす SQL 文の実行情報を探してください。
   - con_id，con_num，stmt_hdl，およびsql_serial_num の値が，手順 3.で記録した値と同じである
   - コール種別がCLOS，EXEC，またはEXDI である
7. 統計情報を確認する
   手順 6.で特定した SQL 文の実行情報の直後に出力されている次の統計情報を参照して，SQL 文のチューニングを実施してください。
   - SQL 文の統計情報
   - アクセスパスの統計情報

10.11.9　障害によって SQL トレース情報の出力が停止した場合の対処方法

次のような障害が発生した場合，SQL トレース情報の出力が停止されます。

- ディスク領域不足（KFAA51010-W メッセージが出力）
- SQL トレースファイルに対する権限がない（KFAA51010-W メッセージが出力）
- ファイルディスクリプタ不足（KFAA51010-W メッセージが出力）
- メモリ不足（KFAA51011-W メッセージが出力）

SQL トレース情報の出力を再開する手順を次に示します。

手順

1. SQL トレース情報の出力状態を確認する

adbchgsqltrc コマンドを実行して確認します。

```
adbchgsqltrc -d

sqltrace param    accesspath level
INACTIVE ACTIVE   ACTIVE     CALL
```

SQL トレース情報の出力が停止している場合は，sqltrace 欄にINACTIVE が表示されます。

2. エラー要因を取り除く

メッセージログファイルに出力されたエラーメッセージを確認し，エラー要因を取り除いてください。

3. SQL トレース情報の出力を再開する

adbchgsqltrc コマンドを実行して再開します。

```
adbchgsqltrc -s
```

メモ

SQL トレースファイルを格納しているディスクが満杯になった場合，警告メッセージがメッセージログファイルに出力されて，SQL トレース情報の出力が停止されます。SQL トレース情報の出力が停止されても，SQL 文の処理は続行されます。

10.11.10　SQL トレース機能の使用をやめる場合

SQL トレース機能の使用をやめる場合は，サーバ定義のadb_sql_trc_out オペランドにN を指定してください。不要になった SQL トレースファイルは，HADB サーバが終了したあとに手動で削除してください。

メモ

HADB サーバの稼働中に SQL トレース機能の使用をやめたい場合は，adbchgsqltrc コマンドを実行してください。

```
adbchgsqltrc -e
```

10.11.11　マルチノード機能を使用している場合の留意事項

マルチノード機能を使用している場合，ノードごとに SQL トレースファイルが作成されます。SQL トレース情報は，トランザクションが実行されたノードの SQL トレースファイルに出力されます。そのため，全トランザクションの SQL トレース情報を出力したい場合は，マルチノード構成の全 HADB サーバで SQL トレース情報を出力する必要があります。

> **メモ**
>
> クライアント定義情報に出力されるオペランドの値は，各ノードで異なることがあります。そのため，全ノードでクライアント定義情報とそのクライアント定義を使用した AP のコネクション状態に関する情報が出力されます。

■SQL パラレル実行機能を使用している場合

SQL パラレル実行機能を適用した検索系 SQL を実行した場合，SQL トレース情報は次のように出力されます。SQL トレース情報に出力される各情報については，「10.11.2　SQL トレース情報に出力される情報」を参照してください。

- SQL 文の実行情報

1. SQL 文の実行情報の出力内容がノードによって異なります。
   - SQL メインノードの場合

     SQL トレースの出力レベルに従って，コール種別ごとに出力されます。通常の SQL 文の実行時と同じです。そのため，call には，SQL トレースの出力レベルに従ったコール種別が出力されます。
   - SQL サブノードの場合

     SQL パラレル実行機能を適用した 1SQL 文につき，1 つの SQL 文の実行情報だけが出力されます。そのため，call には，常にSQL が出力されます。
2. SQL 文の実行情報の次の出力項目には，SQL メインノードと SQL サブノードで同じ情報が出力されます。
   - con_id
   - con_num
   - stmt_hdl
   - sql_serial_num
   - tran_id
   - trn_iso_lv
   - trn_access_mode
   - sql_order_mode
   - cursor_holdability
   - message_log_info
3. SQL 文の実行情報の次の出力項目には，SQL メインノードと SQL サブノードで異なる情報が出力されます。
   - sqlcode

     SQL メインノードでは，HADB クライアントに返却されるSQLCODE の値が出力されます。

SQL サブノードでは，SQL サブノードでエラーが発生した場合に限り，エラー要因を示すSQLCODE の値が出力されます。

- sqlstate

  SQL メインノードでは，HADB クライアントに返却されるSQLSTATE の値が出力されます。

  SQL サブノードでは，SQL サブノードでエラーが発生した場合に限り，エラー要因を示すSQLSTATE の値が出力されます。

- start_time

  各ノードでの SQL 文の実行結果に基づいた情報が出力されます。

- end_time

  各ノードでの SQL 文の実行結果に基づいた情報が出力されます。

- exe_time

  各ノードでの SQL 文の実行結果に基づいた情報が出力されます。

- rows

  SQL メインノードでは，SQL 文の実行結果に基づいた情報が出力されます。SQL サブノードでは，常に 0 が出力されます。

- **実行した SQL 文とアクセスパス情報**

  SQL メインノードと SQL サブノードで同じ情報が出力されます。

- **？パラメタの情報**

  SQL メインノードと SQL サブノードで同じ情報が出力されます。

- **SQL パラレル実行機能に関する情報**

  SQL メインノードと SQL サブノードで同じ情報が出力されます。

- **アクセスパスの統計情報**

  SQL メインノードと SQL サブノードで異なる情報が出力されます。それぞれのノードでの SQL 文の実行結果に基づいた情報が出力されます。

メモ

- SQL パラレル実行機能を適用した検索系 SQL のトラブルシュートやチューニングを行う場合は，SQL メインノードおよび SQL サブノードに出力された SQL トレース情報をすべて確認する必要があります。各ノードに出力された SQL トレース情報で，次の出力項目が一致する SQL トレース情報が，SQL パラレル実行機能を適用した検索系 SQL の SQL トレース情報になります。

  ・con_id

  ・con_num

  ・stmt_hdl

  ・sql_serial_num

- SQL メインノード，SQL サブノードがどのノードなのかを知りたい場合は，SQL トレース情報の SQL パラレル実行機能に関する情報を確認してください。

10.12　システムログファイルの運用

ここでは，システムログファイルの運用方法について説明します。

10.12.1　システムログファイルの構成とサーバ定義の指定

システムログファイルは，マスタログファイルとユーザログファイルの 2 種類のファイルで構成されています。マスタログファイルには，トランザクションの開始および終了など，システム制御に関する履歴情報（イベントログ）と，ユーザログファイルの管理情報が出力されます。マスタログファイルは，HADB サーバを開始したときに自動的に 1 つだけ作成されます。

ユーザログファイルには，データベースの操作履歴情報（ユーザログ）が出力されます。ユーザログファイルは，データベースを更新するトランザクションおよびリアルスレッドごとに作成されます。ユーザログファイルは，HADB サーバを開始したときに自動的に作成されます。サーバ定義の指定によって，作成されるユーザログファイルの個数や初期化容量が決まります。

ユーザログファイルに関するサーバ定義のオペランド

- adb_log_usrfile_num

  HADB サーバの開始時に作成されるユーザログファイル数を指定します。

- adb_log_usrfile_size

  ユーザログファイルの初期化容量とユーザログファイルの容量の縮小契機を指定します。

  ユーザログファイルの容量が，ここで指定したユーザログファイルの初期化容量を超えた場合，ユーザログファイルの容量が自動的に拡張されます。ただし，ユーザログファイルの拡張処理は，更新性能への影響が大きいため，注意が必要です。詳細については，「13.1.8　ユーザログファイルの初期化容量の拡張」を参照してください。

- adb_log_usrbuf_num

  ユーザログバッファ面数を指定します。

各オペランドの詳細については，「7.2.3　システムログに関するオペランド（set 形式）」を参照してください。

メモ

- システムログファイルの詳細については，「2.11.1　再開始による回復処理の流れ」の「(2) システムログ」を参照してください。
- ユーザログファイル数の求め方については，「6.12.15　ユーザログファイル数の求め方」を参照してください。

10.12.2　システムログファイルの容量の縮小契機

データベースの更新処理が行われるとシステムログが出力されるため，データベースの更新量に比例してシステムログファイルの容量が増加していきます。特に，非定期運用などでデータベースの大量更新を行った場合，一部のユーザログファイルの容量が非常に大きくなることがあります。その結果，DB ディレクトリの容量を圧迫するなどの問題が発生するおそれがあります。DB ディレクトリの容量圧迫を防ぐために，HADB サーバは次に示す契機でシステムログファイルの容量を自動的に縮小します。

マスタログファイルの容量の縮小契機

データベースを更新したトランザクションが決着した際に，マスタログファイルの容量が 2 メガバイトを超えている場合，マスタログファイルの容量が縮小されます。ただし，次に示すどちらかの条件を満たす場合は，マスタログファイル内に使用中のシステムログが存在するため，マスタログファイルの容量を縮小する処理がスキップされます。

- データベースを更新するトランザクションがほかにも存在する場合
- マスタログファイルの容量の縮小契機となったトランザクションが，アーカイブディレクトリ下のデータを削除する処理を行っている場合

> **重要**
>
> データベースを更新するトランザクションを複数実行している状態が続くと，マスタログファイルの縮小処理が行われません。そのため，定期的（例えば 1 日に 1 回）に更新トランザクションを複数同時に実行しない時間帯を意図的に設定することを推奨します。

> **メモ**
>
> アーカイブマルチチャンク表に対して次に示す SQL 文またはコマンドを実行した場合，アーカイブディレクトリ下のデータを削除する処理が行われます。
>
> - DROP SCHEMA 文
> - REVOKE 文のSCHEMA
> - DROP TABLE 文
> - TRUNCATE TABLE 文
> - ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表へ変更する場合）
> - PURGE CHUNK 文
> - adbunarchivechunk コマンド

ユーザログファイルの容量の縮小契機

更新トランザクションの決着時，その更新トランザクションが使用していたユーザログファイルの容量が，サーバ定義のadb_log_usrfile_size オペランドのユーザログファイルの容量の縮小契機に指定した容量を超えている場合，ユーザログファイルの容量が初期化容量※に縮小されます。

注※

adb_log_usrfile_size オペランドのユーザログファイルの初期化容量に指定した容量です。

ユーザログファイルの縮小に関して次のことを考慮してください。

- ユーザログファイルの縮小処理に伴う更新性能への影響をなくしたい場合

  ディスクに十分な空き容量がある場合は，adb_log_usrfile_size オペランドのユーザログファイルの容量の縮小契機に0 を指定してください。0 を指定した場合，ユーザログファイルの容量の縮小処理は行われません。

- ユーザログファイルの容量増加に伴うディスク容量不足を回避したい場合

  adb_log_usrfile_size オペランドを省略するか，またはadb_log_usrfile_size オペランドのユーザログファイルの容量の縮小契機に0 以外を指定してください。この場合，ユーザログファイルの容量の縮小処理が行われます。ただし，定常運用中にユーザログファイルの容量の縮小処理の回数が増加すると，更新性能に影響を及ぼすおそれがあります。そのため，ユーザログファイルの容量の縮小処理が実行された回数を定期的に確認してください。ユーザログファイルの容量の縮小処理が実行された回数を調べる方法については，「13.1.9　ユーザログファイルの縮小化容量の見直し」を参照してください。

11 非定期運用

この章では，HADB サーバで非定期に実施する運用項目について説明します。

11.1　実表の運用

ここでは，実表の運用について説明します。

実表のうち，マルチチャンク表の運用については「11.4　マルチチャンク表の運用」を参照してください。

11.1.1　実表を定義する方法

実表を定義するには，定義系 SQL のCREATE TABLE 文を実行します。

CREATE TABLE 文については，マニュアル『HADB　SQL リファレンス』の『CREATE TABLE（表の定義)』を参照してください。

なお，実表をローストア表として定義するか，またはカラムストア表として定義するかによって，CREATE TABLE 文のSTORAGE FORMAT の指定内容が異なります。

- **実表をローストア表として定義する場合**

  CREATE TABLE 文のSTORAGE FORMAT に，ROW を指定してください。または，STORAGE FORMAT の指定を省略してください。
- **実表をカラムストア表として定義する場合**

  CREATE TABLE 文のSTORAGE FORMAT に，COLUMN を指定してください。

また，実表をマルチチャンク表として定義する場合は，「11.4.1　マルチチャンク表を定義する方法」を参照してください。

> **重要**
>
> - HADB では，自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有するスキーマに対してだけ，実表を定義できます。
> - 実表の設計を行ってから，実表を定義してください。実表の設計については，「5.2　表の設計」を参照してください。

11.1.2　実表に列を追加する方法

実表に列を追加する場合，定義系 SQL のALTER TABLE 文を実行します。

実表を定義した HADB ユーザが，ALTER TABLE 文で列を追加できます。

なお，追加できる列は 1 列だけであり，実表の最後の列に追加されます。追加した列にはナル値が格納されます。

実表に列を追加する指定例を次に示します。

指定例

店舗表（SHOPSLIST）に，列名「EMAIL_ADDRESS」を追加します。列名「EMAIL_ADDRESS」のデータ型はVARCHAR(100)とします。

```
ALTER TABLE "SHOPSLIST"
    ADD COLUMN "EMAIL_ADDRESS" VARCHAR(100)
```

SHOPSLIST

SHOP_CODE	AREA_CODE	SHOP_NAME	TEL_NO	ADDRESS	EMAIL_ADDRESS
S0000001	P00002	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL
S0000002	P00001	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL
S0000003	P00002	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL

追加した列

ALTER TABLE 文を実行する際，列定義にNOT NULL やBRANCH も指定できます。ALTER TABLE 文の指定形式および規則については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『ALTER TABLE（表定義の変更)』を参照してください。

なお，次に示す場合はALTER TABLE 文による列追加はできません。

- 操作対象の実表が更新不可状態の場合
- 操作対象の実表が，行を格納するセグメントが割り当てられている状態であり，かつ FIX 表の場合
- 操作対象の実表が，行を格納するセグメントが割り当てられている状態であり，かつCREATE TABLE 文を実行した際にBRANCH ALL を指定した実表の場合
- 行を格納するセグメントが割り当てられている状態の実表に対して，ALTER TABLE 文の列定義でNOT NULL を指定した場合

上記のどれかに該当して列を追加できない場合は，「15.8.2　実表に列を追加できない場合」を参照してください。

行を格納するセグメントが割り当てられている状態については，「5.3.1　B-tree インデクスを定義する場合の注意点（B-tree インデクスの未完状態)」を参照してください。

11.1.3　実表の列名を変更する方法

実表の列名を変更する場合，定義系 SQL のALTER TABLE 文を実行します。

実表を定義した HADB ユーザが，ALTER TABLE 文で列名を変更できます。

実表の列名を変更する指定例を次に示します。

指定例

店舗表（SHOPSLIST）のEMAIL_ADDRESS 列の列名を，EMAIL に変更します。

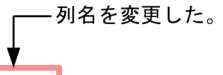
```
ALTER TABLE "SHOPSLIST"
    RENAME COLUMN FROM "EMAIL_ADDRESS" TO "EMAIL"
```

■列名の変更前

SHOPSLIST

SHOP_CODE	AREA_CODE	SHOP_NAME	TEL_NO	ADDRESS	EMAIL_ADDRESS
S0000001	P00002	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL
S0000002	P00001	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL
S0000003	P00002	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL

■列名の変更後



SHOPSLIST

SHOP_CODE	AREA_CODE	SHOP_NAME	TEL_NO	ADDRESS	EMAIL
S0000001	P00002	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL
S0000002	P00001	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL
S0000003	P00002	XXXXXXX	XXXXXXXXXX	XXXXXXXXXX	NULL

ALTER TABLE 文の指定形式および規則については，マニュアル『HADB　SQL リファレンス』の『ALTER TABLE（表定義の変更）』を参照してください。

なお，実表の列名を変更する場合，次に示す個所に影響があります。影響がないかどうかを確認してください。

- 列名を変更した実表に依存するビュー表がある場合

  列名を変更した実表に依存するビュー表は無効化されます。ビュー表の定義時に，変更前の列名を明示的に指定しているかどうかに関係なく，ビュー表は無効化されます。

  無効化されるビュー表（依存するビュー表）を調べる場合は，「11.2.11　依存するビュー表を確認する方法」を参照してください。無効化されるビュー表がある場合は，ALTER TABLE 文で列名を変更したあとで，ビュー表の無効化を解除する必要があります。

  ビュー表の無効化を解除する方法については，「11.2.8　ビュー表の無効化を解除する方法」の「(2) ALTER TABLE 文で表の列名を変更したことによってビュー表が無効化された場合」を参照してください。

- SQL 文に変更前の列名を明示的に指定している場合

  AP などに指定している SQL 文に，変更前の列名を指定している場合は，変更後の列名に修正する必要があります。

- adbimport コマンドに-r オプションで列構成情報ファイルを指定している場合

  列構成情報ファイルに対して，列構成情報オプションで変更前の列名を指定している場合は，変更後の列名に修正する必要があります。

また，次に示す場合は，ALTER TABLE 文で実表の列名を変更できません。

- 操作対象の実表が更新不可状態になっている場合

実表の更新不可状態を解除したあとで，ALTER TABLE 文で列名を変更してください。実表の更新不可状態を解除する方法については，「15.8.1　実表が更新不可状態になった場合」の「(1)　実表の更新不可状態を解除する方法」を参照してください。

11.1.4　実表の表名を変更する方法

実表の表名を変更する場合，定義系 SQL のALTER TABLE 文を実行します。

実表を定義した HADB ユーザが，ALTER TABLE 文で表名を変更できます。

実表の表名を変更する指定例を次に示します。

指定例

店舗表（SHOPSLIST）の表名をSTORELIST に変更します。

```
ALTER TABLE "SHOPSLIST"
    RENAME TABLE TO "STORELIST"
```

ALTER TABLE 文の指定形式および規則については，マニュアル『HADB　SQL リファレンス』の『ALTER TABLE（表定義の変更)』を参照してください。

なお，実表の表名を変更する場合，次に示す個所に影響があります。影響がないかどうかを確認してください。

- 表名を変更した実表に依存するビュー表がある場合

  表名を変更した実表に依存するビュー表は無効化されます。無効化されるビュー表（依存するビュー表）を調べる場合は，「11.2.11　依存するビュー表を確認する方法」を参照してください。無効化されるビュー表がある場合は，ALTER TABLE 文で表名を変更したあとで，ビュー表の無効化を解除する必要があります。

  ビュー表の無効化を解除する方法については，「11.2.8　ビュー表の無効化を解除する方法」の「(3) ALTER TABLE 文で表の表名を変更したことによってビュー表が無効化された場合」を参照してください。

- SQL 文に変更前の表名を明示的に指定している場合

  AP などに指定している SQL 文に，変更前の表名を指定している場合は，変更後の表名に修正する必要があります。

また，次に示す場合は，ALTER TABLE 文で実表の表名を変更できません。

- 操作対象の実表が更新不可状態になっている場合

  実表の更新不可状態を解除したあとで，ALTER TABLE 文で表名を変更してください。実表の更新不可状態を解除する方法については，「15.8.1　実表が更新不可状態になった場合」の「(1)　実表の更新不可状態を解除する方法」を参照してください。

11.1.5　実表のデータを検索・更新する方法

SQL 文で，実表のデータを検索・更新する方法について説明します。

実表のデータを検索する方法

実表に格納されたデータを検索する場合，操作系 SQL のSELECT 文を使用します。

SELECT 文については，マニュアル『HADB　SQL リファレンス』の『SELECT（行の検索)』を参照してください。

実表にデータを追加する方法（行を挿入する方法）

実表にデータを追加する場合，操作系 SQL のINSERT 文を使用します。

INSERT 文については，マニュアル『HADB　SQL リファレンス』の『INSERT（行の挿入)』を参照してください。

> **重要**
>
> カラムストア表に大量のデータを追加する場合は，INSERT 文ではなく，adbimport コマンドを使用してください。カラムストア表にINSERT 文でデータを追加した場合，追加したデータはローストア形式で格納され，それが原因でカラムストア表の検索性能が低下するおそれがあります。したがって，カラムストア表に大量のデータを追加する場合は，adbimport コマンドを使用してください。adbimport コマンドを使用したデータの追加方法については，「11.1.8　実表にデータを格納する方法（データインポート)」を参照してください。

実表のデータを更新する方法（行を更新する方法）

実表に格納されたデータを更新する場合，操作系 SQL のUPDATE 文を使用します。

UPDATE 文については，マニュアル『HADB　SQL リファレンス』の『UPDATE（行の更新)』を参照してください。

実表のデータを削除する方法（行を削除する方法）

実表に格納されたデータを削除する場合，操作系 SQL のDELETE 文を使用します。

DELETE 文については，マニュアル『HADB　SQL リファレンス』の『DELETE（行の削除)』を参照してください。

> **メモ**
>
> 実表に格納されたすべてのデータを削除する場合，DELETE 文ではなく，TRUNCATE TABLE 文を使用することを推奨します。
>
> TRUNCATE TABLE 文については，「11.1.6　実表の全行を削除する方法」を参照してください。

重要

カラムストア表に対して，SQL 文（INSERT 文，UPDATE 文，DELETE 文）によるデータの追加，更新，または削除を繰り返し実行すると，カラムストア表の検索性能が低下するおそれがあります。検索性能の低下を防ぐために，次のことを実施することを推奨します。

- 更新行のカラム化機能を有効にする
- 表またはチャンクの**再編成要否の情報**を定期的に取得し，表またはチャンクの再編成が必要かどうかを確認する

更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。

表またはチャンクの再編成が必要かどうかを確認する方法については，「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」または「11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法」を参照してください。

11.1.6　実表の全行を削除する方法

実表の全行を削除する場合，基本的には，操作系 SQL のTRUNCATE TABLE 文を実行することを推奨します。

TRUNCATE TABLE 文を実行すると，DELETE 文（WHERE 句を省略）よりも短時間で全行を削除できます。また，セグメントが解放されるため，削除した行データが格納されていた領域を再利用できます。

TRUNCATE TABLE 文とDELETE 文の違いについて，次の表に示します。

表 11-1　TRUNCATE TABLE 文と DELETE 文の違い

項番	比較項目	TRUNCATE TABLE 文	DELETE 文（WHERE 句を省略）
1	処理性能	高速 DELETE 文よりも短時間で全行を削除できます。	低速 TRUNCATE TABLE 文と比べると，全行の削除に時間が掛かります。
2	セグメントの解放	解放される 削除した行データが格納されていた領域を再利用できます。	解放されない 削除した行は無効になりますが，ディスクからは削除されません。そのため，削除した行データが格納されている領域は再利用できません。 また，カラムストア表の場合は，無効情報管理用ページを確保するため，使用するセグメントが増加することがあります。

項番	比較項目	TRUNCATE TABLE 文	DELETE 文（WHERE 句を省略）
3	COMMIT 文の実行要否	自動的にコミットされる 処理が正常終了した場合，処理完了と同時にコミットされます。COMMIT 文の実行は不要です。	自動的にコミットされない 処理完了後，COMMIT 文を実行する必要があります。
4	排他制御（同一表に対する SELECT 文との同時実行の可否）※	同時実行できない TRUNCATE TABLE 文を実行すると，処理対象表の排他を占有モードで確保します。そのため，同一表に対しては，SELECT 文とTRUNCATE TABLE 文は同時実行できません。	同時実行できる 同一表に対して，SELECT 文とDELETE 文は同時実行できます。
5	推奨する実行タイミング	• 短時間で全行を削除したいとき • 削除した行データが確保していた領域を再利用したいとき	• 同一表に対して，削除処理と検索処理を同時実行したいとき • トランザクションの 1 要素として使用したいとき

注※

TRUNCATE TABLE 文およびDELETE 文を実行したときに確保される排他資源と排他制御モードの詳細については，「2.10.4　確保される排他資源と排他制御モード」を参照してください。

実表の全行を削除する指定例を次に示します。

指定例

販売履歴表（SALESLIST）内の全行を削除します。

```
TRUNCATE TABLE "SALESLIST"
```

TRUNCATE TABLE 文の指定形式および規則については，マニュアル『HADB　SQL リファレンス』の『操作系 SQL』の『TRUNCATE TABLE（実表の全行削除）』を参照してください。

重要

全行を削除する前に，削除する行が定義されている実表に対して，コマンドの再実行が必要かどうか（実表が更新不可状態かどうか）を確認してください。確認する方法については，「10.9.2　実表の状態や使用量を確認したい場合」の「(1)　実表が更新不可状態かどうかを確認する場合」を参照してください。

コマンドの再実行が必要な場合

コマンドを再実行してください。そのあとで，全行を削除してください。

コマンドの再実行が必要な状態のままで（実表が更新不可状態のままで）全行を削除した場合

中断したコマンドによって作成された作業用一時ファイルが残ることがあります。作業用一時ファイルを削除するには，すべての実表の更新不可状態を解除する必要があります。

作業用一時ファイルの削除については，「15.2.6　コマンド実行時の作業用一時ファイルを格納するディスクの空き容量が不足した場合の対処方法」の「(2)　ディスクに不要な作業用一時ファイルが残っている場合」を参照してください。

11.1.7　シングルチャンク表をマルチチャンク表に変更する方法

シングルチャンク表をマルチチャンク表に変更する場合は，シングルチャンク表をいったん削除してください。そのあとで，次のどちらかのマルチチャンク表として再定義してください。

- レギュラーマルチチャンク表
- アーカイブマルチチャンク表

重要

- 変更対象の表がカラムストア表の場合，シングルチャンク表からアーカイブマルチチャンク表への変更はできません。
- シングルチャンク表をマルチチャンク表に変更する前に，「5.2.4　マルチチャンク表を定義する場合の考慮点」および「5.2.5　アーカイブマルチチャンク表を定義する場合の考慮点【ローストア表】」を参照して，留意事項などを確認してください。

シングルチャンク表をマルチチャンク表に変更する手順を次に示します。

手順

1. マルチチャンク表に変更するシングルチャンク表の表定義を確認する

   シングルチャンク表を定義したときのCREATE TABLE 文の指定内容を確認してください。手順 4.でマルチチャンク表を定義する際に使用します。

   なお，シングルチャンク表を定義したときのCREATE TABLE 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(28)　実表の定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，シングルチャンク表を定義したときのCREATE TABLE 文の指定内容が確認できます。

2. シングルチャンク表の全データを出力する

   adbexport コマンドで，シングルチャンク表の全データを出力データファイルに出力してください。

3. シングルチャンク表を削除する

   *削除動作*の指定を省略したDROP TABLE 文で，シングルチャンク表を削除してください。

4. マルチチャンク表を定義する

   CREATE TABLE 文で，マルチチャンク表を定義してください。CREATE TABLE 文の指定内容を作成する際の留意事項を次に示します。

- 手順 1.で確認したシングルチャンク表を定義したときのCREATE TABLE 文の指定内容を使用して，CREATE TABLE 文を作成してください。
- CREATE TABLE 文にチャンク指定を追加してください。なお，レギュラーマルチチャンク表とアーカイブマルチチャンク表で，チャンク指定の指定内容が異なります。チャンク指定の指定内容については，「11.4.1　マルチチャンク表を定義する方法」を参照してください。

5. マルチチャンク表にインデクスを定義する

   削除したシングルチャンク表にインデクスを定義していた場合は，手順 4.で定義したマルチチャンク表に，CREATE INDEX 文でインデクスを定義してください。CREATE INDEX 文を作成する際の留意事項を次に示します。

   - シングルチャンク表のインデクスを定義したときのCREATE INDEX 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，シングルチャンク表のインデクスを定義したときのCREATE INDEX 文の指定内容が確認できます。
   - マルチチャンク表には，ユニークインデクスは定義できません。

6. マルチチャンク表にデータを格納する

   adbimport コマンドで，手順 4.で定義したマルチチャンク表に，手順 2.で出力したデータを格納してください。その際，adbimport コマンドには-d オプションおよび-b オプションは指定しないでください。

7. ビュー表の無効化を解除する

   手順 3.で削除したシングルチャンク表を基表とするビュー表が無効化されます。そのため，マルチチャンク表の定義後に，ビュー表の無効化を解除する必要があります。「11.2.8　ビュー表の無効化を解除する方法」の「(6)　表を誤って削除してしまったことによってビュー表が無効化された場合」を参照して，ビュー表の無効化を解除してください。

メモ

adbexport コマンドおよびadbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

DROP TABLE 文，CREATE TABLE 文およびCREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』を参照してください。

11.1.8　実表にデータを格納する方法（データインポート）

実表にデータを格納する場合，adbimport コマンドでデータインポートを行ってください。adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

なお，adbimport コマンドに指定するオプションによって，データインポートの方法が異なります。adbimport コマンドによるデータインポートの方法を次に示します。

- 作成モードによるデータインポート

  作成モードによるデータインポートを実行すると，実表に格納されている既存のデータをすべて削除したあとで，データを新規にインポートします。マルチチャンク表の場合は，既存のチャンクをすべて削除したあとで，新しいチャンクを作成し，そのチャンクにデータをインポートします。そのため，実表中のデータをすべて削除して，新規にデータを格納し直したい場合に，作成モードによるデータインポートを実行してください。

  作成モードによるデータインポートを実行する場合は，実表に対して，-d オプションを指定したadbimport コマンドを実行してください。

  > **メモ**
  >
  > 新規に定義した表の場合，-d オプションの指定有無に関係なく，作成モードによるデータインポートが行われます。

- バックグラウンドインポート

  マルチチャンク表に対して実行できるデータインポートの方法です。バックグラウンドインポートを実行すると，新しいチャンクを作成し，そのチャンクにデータをインポートします。データインポートの実行中に，既存のチャンク内のデータを検索することができます。そのため，データインポートとデータ検索を同時に実行したい場合に，バックグラウンドインポートを実行してください。

  バックグラウンドインポートを実行する場合は，マルチチャンク表に対して，-b オプションを指定したadbimport コマンドを実行してください。バックグラウンドインポートの実行方法については，「11.4.2　マルチチャンク表にデータを格納する方法（バックグラウンドインポート）」を参照してください。

- 追加モードによるデータインポート

  追加モードによるデータインポートを実行すると，実表中のデータはそのままの状態で，データを追加でインポートします。マルチチャンク表の場合は，既存のチャンク内のデータはそのままの状態で，カレントチャンクに新たなデータを追加でインポートします。その際，空きページが存在するセグメントがある場合は，そのセグメントに優先的にデータを追加します。ただし，カラムストア表の場合は，空きページが存在するセグメントがあっても，必ず新しいセグメントに対してデータを格納します。なお，追加モードによるデータインポートの実行中は，インポート対象表のデータを検索することはできません。

  追加モードによるデータインポートを実行する場合は，実表に対して，-d オプションおよび-b オプションの指定を省略してadbimport コマンドを実行してください。

  > **メモ**
  >
  > 少量のデータをインポートする場合，追加モードによるデータインポートの方が，バックグラウンドインポートよりも，データの格納効率が良くなります。そのため，業務時間外の表のメンテナンスなどの際に，少量のデータをインポートするようなことがある場合は，追加モードによるデータインポートの実行を検討してください。

-d オプションおよび-b オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』の『adbimport コマンドの指定形式』の『指定形式およびオプションの説明』を参照してください。

11.1.9　実表のデータをファイルに出力する方法（データをエクスポートする方法）

実表に格納されているデータをファイルに出力する場合，次に示す 2 つの方法があります。

- adbexport コマンドを実行する

  adbexport コマンドを実行すると，実表に格納されているデータをファイルに出力できます。

  - 実表に格納されている全データを出力したい場合は，-n オプションに*処理対象表*を指定してadbexport コマンドを実行してください。

    -n オプションに指定した処理対象表に格納されているデータを出力できます。

  - 実表に格納されているデータのうち，特定の条件を満たすデータだけを出力したい場合は，-q オプションを指定してadbexport コマンドを実行してください。

    -q オプションの SQL 記述ファイルに指定した SQL 文の検索結果を出力できます。

  adbexport コマンドを使用すると，adbsql コマンドよりも短時間でデータを出力できます。ただし，adbexport コマンドは，HADB サーバで実行できますが，HADB クライアントでは実行できません。

  adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。

- adbsql コマンドを実行する

  adbsql コマンドで実表のデータを検索して，検索結果をリダイレクションすると，実表のデータをファイルに出力できます。

  なお，adbsql コマンドは，HADB サーバおよび HADB クライアント（Linux 版）で実行できます。

  adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』を参照してください。

基本的には，adbexport コマンドで実表に格納されているデータを出力することを推奨します。HADB クライアント（Linux 版）で，実表のデータをファイルに出力する必要があるときだけ，adbsql コマンドを使用してください。

重要

待機状態のチャンク内のデータを出力したい場合は，次のどちらかを実施してください。

- チャンク ID を指定して待機状態のチャンク内のデータを出力する

チャンク ID を指定してadbexport コマンドやadbsql コマンドを実行すると，待機状態のチャンク内のデータも出力できます。詳細については，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

- 待機状態のチャンクを通常状態のチャンクに変更する

  通常状態のチャンク内のデータは，上記 2 つの方法で出力できます。そのため，adbchgchunkstatus コマンドで，待機状態のチャンクを通常状態のチャンクに変更してください。詳細については，「11.4.12　チャンクの状態を変更する方法」を参照してください。

11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法

シングルチャンク表の再編成が必要かどうかを確認する方法について説明します。

メモ

マルチチャンク表の再編成が必要かどうかを確認する方法については，「11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法」を参照してください。

(1)　表の再編成が必要な理由

シングルチャンク表に対して，SQL 文による行の挿入，更新，削除を繰り返し実行すると，表の検索性能やデータの格納効率が低下します。この場合，表を再編成すると，データ用 DB エリア内のむだな空き領域が解放されて，表の検索性能やデータの格納効率が向上します。

シングルチャンク表の検索性能やデータの格納効率が低下する理由を次に示します。

シングルチャンク表がローストア表の場合

DELETE 文による行の削除を実行すると，削除対象の行データは無効となります。また，UPDATE 文による行の更新を実行すると，更新後の行データが新たな行として追加されて，更新前の行データは無効となります。無効となった行データは，ディスクからは削除されません。

行の削除および行の更新を繰り返し実行した場合，表内に無効な行データが増加します。無効な行データが増加すると，次のような影響があります。

- 無効な行データが増加した影響で，データの格納効率が悪くなる
- 検索時に参照するページ数が増えるため，検索性能が低下する

シングルチャンク表がカラムストア表の場合

- INSERT 文による行の挿入，またはUPDATE 文による行の更新を実行すると，データがローストア形式で格納されます。行の挿入および行の更新を繰り返し実行した場合，ローストア形式で格納されるデータが増加します。ローストア形式で格納されるデータが増加すると，カラムストア表に適した検索（特定の列データ全体に対する検索や，特定の年や月などの範囲内の，特定の列データに対する検索など）の性能が低下します。

- DELETE 文による行の削除を実行すると，削除対象の行に関する情報が無効情報管理用ページに格納されます。また，UPDATE 文による行の更新を実行すると，更新前の行に関する情報が無効情報管理用ページに格納されます。

  行の削除および行の更新を繰り返し実行した場合，無効情報管理用ページが増加し，検索時に参照するページ数が増えるため，検索性能が低下します。

シングルチャンク表がローストア表の場合は，「(2)　シングルチャンク表がローストア表の場合」を参照して，表の再編成が必要かどうかを確認してください。

シングルチャンク表がカラムストア表の場合は，「(3)　シングルチャンク表がカラムストア表の場合」を参照して，表の再編成が必要かどうかを確認してください。

(2)　シングルチャンク表がローストア表の場合

データの格納効率を計算し，データの格納効率が悪い場合は，表の再編成を実行してください。データの格納効率は，次の計算式で求めてください。

計算式

```
データの格納効率＝ A ÷（A+B）
```

変数の説明

A：表の行数

B：無効行数

計算式で求めた結果が 1 に近いほど，データの格納効率が良いと判断できます。逆に，0 に近いほど，データの格納効率が悪いと判断できます。0 に近い場合は，「11.1.11　シングルチャンク表を再編成する方法」を参照して，表のデータを再編成してください。

変数*A* および*B* については，次に示す手順で求めてください。

■手順（変数*A* および*B* の求め方）

1. SQL トレース情報が出力される設定になっていることを確認する

   SQL トレース情報のアクセスパスの統計情報が出力されていることを確認してください。アクセスパスの統計情報が出力されているかどうかは，SQL トレース機能に関するサーバ定義の指定で確認できます。詳細については，「10.11.5　SQL トレース情報を出力するための準備」の「(2)　サーバ定義を指定する」を参照してください。

2. 次のSELECT 文を実行する

   ```
   SELECT COUNT(*) FROM ”表名” /*>> WITHOUT INDEX <<*/
   ```

   上記のSELECT 文の実行結果が，表の行数（変数*A* の値）になります。

3. アクセスパスの統計情報を確認する

手順 2.で実行したSELECT 文に対するアクセスパスの統計情報を確認してください。アクセスパスの統計情報の「Data_deleted_rows_cnt」の値を確認します。この値が，無効行数（変数*B* の値）になります。

なお，無効行数（変数*B* の値）がわからない場合は，次の手順で，データの格納効率を求めてください。

■手順（変数*B* の値がわからない場合）

1. 表の使用中セグメント数を確認する

   adbdbstatus コマンドで，表のサマリ情報を出力してください。そして，Used_segments（表の使用中セグメント数）の出力内容を確認してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

2. 表の行数を基に，データ用 DB エリアのセグメント数を求める

   表の行数を求めたあとで，データ用 DB エリアのセグメント数を求めてください。表の行数を求める場合は，上記にある変数*A* の説明を参照してください。

   データ用 DB エリアのセグメント数を求める場合は，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(a)　変数 SGROWTBL の求め方（シングルチャンク表の場合)」を参照してください。

   なお，変数*SGROWTBL* は，データ用 DB エリアに格納されているすべての表の合計を求める計算式となっています。ここでは，対象表だけを求めてください。また，表の行数は，変数*SGROWTBL* にある変数*BP*（*i*）および変数 *VP*（*i*）で使用します。

3. 求めた結果を基に，データの格納効率を求める

   手順 1.および手順 2.で求めた結果を基に，データの格納効率を求めてください。計算式を次に示します。

   計算式

   ```
   データの格納効率＝ 手順2. の結果÷手順1. の結果
   ```

   上記の計算結果が 0 に近い場合，「11.1.11　シングルチャンク表を再編成する方法」を参照して，表の再編成を実行してください。

(3)　シングルチャンク表がカラムストア表の場合

adbdbstatus コマンドを実行して**再編成要否の情報**を確認し，シングルチャンク表の再編成を実行するかどうかを決めてください。確認手順を次に示します。

手順

1. -d reorginfo オプションを指定したadbdbstatus コマンドを実行して，再編成要否の情報を出力する
2. 再編成要否の情報のReorganization_necessity（再編成要否）の出力内容を確認する
   - Recommended が出力されている場合

     シングルチャンク表の再編成を実行してください。再編成手順については，「11.1.11　シングルチャンク表を再編成する方法」を参照してください。

- Not_recommended が出力されている場合
  シングルチャンク表の再編成を実行する必要はありません。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

11.1.11　シングルチャンク表を再編成する方法

シングルチャンク表を再編成する方法を説明します。

重要

再編成中の表に対して，更新系 SQL または表を更新するコマンド（adbimport コマンドや adbmergechunk コマンドなど）を実行しないでください。実行した場合，SQL 文やコマンドで更新した内容が，再編成後に失われるおそれがあります。

(1)　再編成の手順

シングルチャンク表の再編成手順を次に示します。

手順

1. AP やコマンドが再編成対象の表にアクセスできない状態にする
   次の操作を実行して，AP やコマンドが再編成対象の表にアクセスできない状態にしてください。
   - adbchgsrvmode コマンドに--offline オプションを指定して，HADB サーバの稼働モードをオフラインモードに変更してください。さらに，HADB サーバマシン上で動作している，表を更新するコマンドやジョブ（定期的に実行しているadbimport コマンドやadbmergechunk コマンドなど）を停止してください。
   - HADB サーバにアクセスする AP やコマンドをすべて停止してください。

   上記の操作を両方とも実行することを推奨します。
2. 表のデータをエクスポートする
   adbexport コマンドの-n オプションに再編成対象の表を指定して，表のデータをエクスポートしてください。
   adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。
3. エクスポートしたデータを作成モードでインポートする
   adbimport コマンドに-d オプションを指定して，手順 2.でエクスポートしたデータをインポートしてください。-d オプションを指定すると，作成モードでインポートされます（表の既存データをすべて削除してから，手順 2.でエクスポートしたデータがインポートされます)。

adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

> **重要**
>
> adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，「(2) インポートオプションの見積もり」に示す計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

4. AP やコマンドが再編成対象の表にアクセスできる状態にする

次に示す方法で，AP やコマンドが再編成対象の表にアクセスできる状態にしてください。

- 手順 1.で，HADB サーバの稼働モードをオフラインモードに変更した場合

  adbchgsrvmode コマンドに--normal オプションを指定して，HADB サーバの稼働モードを通常モードに変更してください。また，HADB サーバマシン上で動作しているコマンドやジョブを停止した場合は，停止していたコマンドやジョブを開始してください。

- 手順 1.で，HADB サーバにアクセスする AP やコマンドを停止した場合

  停止していた AP やコマンドを開始してください。

> **メモ**
>
> マルチチャンク表を再編成する方法については，次の個所を参照してください。
>
> - 「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合)」
> - 「11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合)」
> - 「11.4.16　マルチチャンク表を再編成する方法（サンプルシェルを使用して再編成する場合)」

(2)　インポートオプションの見積もり

■シングルチャンク表がローストア表の場合

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，次の計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

計算式

インポートオプションadb_import_rthd_numの指定値 ≦
↑ *再編成対象のセグメント数* × *データの格納効率* ↑＋1

再編成対象のセグメント数

表の使用中セグメント数を代入してください。

表の使用中セグメント数を代入するためには，adbdbstatus コマンドで**表のサマリ情報**を出力してください。そして，Used_segments（表の使用中セグメント数）の出力内容を確認してください。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus (データベースの状態解析)』を参照してください。

データの格納効率

「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」の「(2)　シングルチャンク表がロースト表の場合」で求めた値を代入してください。

■シングルチャンク表がカラムストア表の場合

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，次の計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

計算式

```
インポートオプションadb_import_rthd_numの指定値 ≦
    ↑ 再編成対象のセグメント数 × 再編成対象の行数
      ÷ カラムデータ用セグメントに格納されている行数 ↑＋1
```

再編成対象のセグメント数

表の使用中のカラムデータ用セグメント数を代入してください。

表の使用中のカラムデータ用セグメント数を確認するには，-d used オプションを指定した adbdbstatus コマンドで，**DB エリア，表とインデクスの使用量情報**を出力してください。そして，Segment_type（セグメントの種類）がColumn_data（カラムデータ用セグメント）の行の Used_segments（使用中セグメント数）の出力内容を確認してください。

再編成対象の行数

再編成対象の表の行数を代入してください。次のSELECT 文を実行すると，表の行数がわかります。

```
SELECT COUNT(*) FROM ”表名”
```

カラムデータ用セグメントに格納されている行数

再編成対象の表にadbimport コマンドでインポートした行数，または更新行のカラム化機能によってロースト形式からカラムストア形式に変換された行数を代入してください。

adbdbstatus コマンドを実行して，**再編成要否の情報**を取得してください。Column_data_num の値が，カラムデータ用セグメントに格納されている行数です。

11.1.12　ロースト表をカラムストア表に変更する方法

ロースト表をカラムストア表に変更したい場合は，対象のロースト表をいったん削除したあとで，カラムストア表として再定義する必要があります。

重要

ロースト表と比べると，カラムストア表には幾つかの制限事項があります。そのため，カラムストア表の制限事項を確認してから，ロースト表をカラムストア表に変更するようにして

ください。カラムストア表の制限事項については，「5.2.2　ローストア表とカラムストア表の選択基準」の「(1)　カラムストア表の制限事項」を参照してください。

メモ

カラムストア表をローストア表に変更したい場合は，「11.1.13　カラムストア表をローストア表に変更する方法」を参照してください。

次の手順に従って，ローストア表をカラムストア表に変更してください。

手順

1. ローストア表に格納されているすべてのデータを出力する

   adbexport コマンドで，ローストア表に格納されているすべてのデータをファイルに出力してください。

   ローストア表がマルチチャンク表の場合に，adbexport コマンドを実行する際の留意事項を次に示します。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

     留意事項は特にありません。ローストア表に格納されているすべてのデータをファイルに出力してください。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

     ローストア表に格納されているすべてのデータを，チャンク単位でファイルに出力する必要があります。チャンク単位でデータを出力しないと，チャンクの状態・構成を維持したまま，カラムストア表にデータを格納できません。そのため，チャンク数分，エクスポートを行ってください。

     チャンク単位でデータを出力する場合は，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

2. ローストア表の定義情報を確認する

   ローストア表を定義したときのCREATE TABLE 文の指定内容を確認してください。カラムストア表を定義する際に使用します。

   ローストア表を定義したときのCREATE TABLE 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(28)　実表の定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，ローストア表を定義したときのCREATE TABLE 文の指定内容が確認できます。

3. ローストア表に定義されているインデクスの定義情報を確認する

   ローストア表にインデクスを定義している場合，インデクスを定義したときのCREATE INDEX 文の指定内容を確認してください。カラムストア表にインデクスを定義する際に使用します。

   インデクスを定義したときのCREATE INDEX 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，インデクスを定義したときのCREATE INDEX 文の指定内容が確認できます。

4. ローストア表を削除する

   *削除動作*の指定を省略したDROP TABLE 文で，ローストア表を削除してください。

5. カラムストア表を定義する

CREATE TABLE 文で，カラムストア表を定義してください。CREATE TABLE 文を作成する際の留意事項を次に示します。

- 手順 2.で確認したローストア表を定義したときのCREATE TABLE 文の指定内容を流用して，CREATE TABLE 文を作成してください。
- CREATE TABLE 文のSTORAGE FORMAT に，COLUMN を指定してください。

6. カラムストア表にインデクスを定義する

削除したローストア表にインデクスを定義していた場合，カラムストア表にCREATE INDEX 文でインデクスを定義してください。CREATE INDEX 文を作成する際の留意事項を次に示します。

- 手順 3.で確認したインデクスを定義したときのCREATE INDEX 文の指定内容を流用して，CREATE INDEX 文を作成してください。

  なお，ローストア表に B-tree インデクスを定義していた場合でも，カラムストア表には B-tree インデクスを定義しないことを推奨します。詳細については，「5.2.2　ローストア表とカラムストア表の選択基準」の「(3)　カラムストア表の検索時に使用される B-tree インデクス」を参照してください。
- カラムストア表には，テキストインデクスは定義できません。

7. カラムストア表にデータを格納する

adbimport コマンドで，手順 5.で定義したカラムストア表に，手順 1.で出力したすべてのデータを格納してください。

カラムストア表がマルチチャンク表の場合に，adbimport コマンドを実行する際の留意事項を次に示します。

- **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

  adbimport コマンドには，-d オプションおよび-b オプションは指定しないでください。
- **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

  手順 1.でチャンク単位に出力したデータを，チャンク単位にインポートする必要があります。
  - 手順 1.でカレントチャンクだったチャンクのデータは，必ず最後に格納してください。
  - 最初に格納するデータの場合は，-d オプションおよび-b オプションを指定しないで，adbimport コマンドを実行してください。
  - 2 回目以降に格納するデータの場合は，バックグラウンドインポートで格納してください。-b オプションを指定してadbimport コマンドを実行してください。手順 1.でカレントチャンクだったチャンクのデータも，バックグラウンドインポートで格納してください。

8. ビュー表の無効化を解除する

手順 4.で削除したローストア表を基表とするビュー表が無効化されます。そのため，カラムストア表の定義後に，ビュー表の無効化を解除する必要があります。「11.2.8　ビュー表の無効化を解除する方法」の「(6)　表を誤って削除してしまったことによってビュー表が無効化された場合」を参照して，ビュー表の無効化を解除してください。

これで，ローストア表をカラムストア表に変更できます。

メモ

- adbexport コマンドおよびadbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。
- DROP TABLE 文，CREATE TABLE 文およびCREATE INDEX 文については，マニュアル『HADB SQL リファレンス』を参照してください。

11.1.13　カラムストア表をローストア表に変更する方法

カラムストア表をローストア表に変更したい場合は，対象のカラムストア表をいったん削除したあとで，ローストア表として再定義する必要があります。

重要

カラムストア表と比べると，ローストア表の場合，表定義時に指定できる項目が増えます。そのため，ローストア表に変更する前に，「5.2.1　表を設計する際の流れ」を参照してください。

メモ

ローストア表をカラムストア表に変更したい場合は，「11.1.12　ローストア表をカラムストア表に変更する方法」を参照してください。

次の手順に従って，カラムストア表をローストア表に変更してください。

手順

1. カラムストア表に格納されているすべてのデータを出力する

   adbexport コマンドで，カラムストア表に格納されているすべてのデータをファイルに出力してください。

   カラムストア表がマルチチャンク表の場合に，adbexport コマンドを実行する際の留意事項を次に示します。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

     留意事項は特にありません。カラムストア表に格納されているすべてのデータをファイルに出力してください。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

     カラムストア表に格納されているすべてのデータを，チャンク単位でファイルに出力する必要があります。チャンク単位でデータを出力しないと，チャンクの状態・構成を維持したまま，ローストア表にデータを格納できません。そのため，チャンク数分，エクスポートを行ってください。

チャンク単位でデータを出力する場合は，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

2. カラムストア表の定義情報を確認する

カラムストア表を定義したときのCREATE TABLE 文の指定内容を確認してください。ローストア表を定義する際に使用します。

カラムストア表を定義したときのCREATE TABLE 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(28)　実表の定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，カラムストア表を定義したときのCREATE TABLE 文の指定内容が確認できます。

3. カラムストア表に定義されているインデクスの定義情報を確認する

カラムストア表にインデクスを定義している場合，インデクスを定義したときのCREATE INDEX 文の指定内容を確認してください。ローストア表にインデクスを定義する際に使用します。

インデクスを定義したときのCREATE INDEX 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，インデクスを定義したときのCREATE INDEX 文の指定内容が確認できます。

4. カラムストア表を削除する

*削除動作*の指定を省略したDROP TABLE 文で，カラムストア表を削除してください。

5. ローストア表を定義する

CREATE TABLE 文で，ローストア表を定義してください。CREATE TABLE 文を作成する際の留意事項を次に示します。

- 手順 2.で確認したカラムストア表を定義したときのCREATE TABLE 文の指定内容を流用して，CREATE TABLE 文を作成してください。
- CREATE TABLE 文のSTORAGE FORMAT に，ROW を指定してください。

6. ローストア表にインデクスを定義する

削除したカラムストア表にインデクスを定義していた場合，ローストア表にCREATE INDEX 文でインデクスを定義してください。CREATE INDEX 文を作成する際の留意事項を次に示します。

- 手順 3.で確認したインデクスを定義したときのCREATE INDEX 文の指定内容を流用して，CREATE INDEX 文を作成してください。
- CREATE INDEX 文を作成する際，必要に応じて，B-tree インデクスやテキストインデクスを定義することを検討してください。

  カラムストア表には，B-tree インデクスを定義することは非推奨です。また，テキストインデクスを定義できません。通常は，レンジインデクスだけが定義されているため，CREATE INDEX 文を作成する際に留意してください。

7. ローストア表にデータを格納する

adbimport コマンドで，手順 5.で定義したローストア表に，手順 1.で出力したすべてのデータを格納してください。

ローストア表がマルチチャンク表の場合に，adbimport コマンドを実行する際の留意事項を次に示します。

- **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

  adbimport コマンドには，-d オプションおよび-b オプションは指定しないでください。

- **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

  手順 1.でチャンク単位に出力したデータを，チャンク単位にインポートする必要があります。

  - 手順 1.でカレントチャンクだったチャンクのデータは，必ず最後に格納してください。
  - 最初に格納するデータの場合は，-d オプションおよび-b オプションを指定しないで，adbimport コマンドを実行してください。
  - 2 回目以降に格納するデータの場合は，バックグラウンドインポートで格納してください。-b オプションを指定してadbimport コマンドを実行してください。手順 1.でカレントチャンクだったチャンクのデータも，バックグラウンドインポートで格納してください。

8. ビュー表の無効化を解除する

   手順 4.で削除したカラムストア表を基表とするビュー表は無効化されます。そのため，ローストア表の定義後に，ビュー表の無効化を解除する必要があります。「11.2.8　ビュー表の無効化を解除する方法」の「(6)　表を誤って削除してしまったことによってビュー表が無効化された場合」を参照して，ビュー表の無効化を解除してください。

これで，カラムストア表をローストア表に変更できます。

メモ

- adbexport コマンドおよびadbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。
- DROP TABLE 文，CREATE TABLE 文およびCREATE INDEX 文については，マニュアル『HADB SQL リファレンス』を参照してください。

11.1.14　実表の定義情報を調べる方法（ディクショナリ表の検索）

ディクショナリ表を検索すると，実表の定義情報を確認できます。

メモ

各ディクショナリ表の内容については，「付録 B　ディクショナリ表」を参照してください。また，ディクショナリ表の検索例については，「付録 B.22　ディクショナリ表の検索」を参照してください。

ディクショナリ表を検索すると確認できる実表の定義情報を次の表に示します。

表 11-2　ディクショナリ表を検索すると確認できる実表の定義情報

項番	ディクショナリ表	確認できる実表の定義情報
1	SQL_TABLES 表	SQL_TABLES 表を検索すると，次の定義情報を確認できます。 • 表名 TABLE_SCHEMA 列（スキーマ名），およびTABLE_NAME 列（表識別子）を確認してください。 • 表オプションPCTFREE の指定内容 FREE_AREA 列（表のページ内の未使用領域の比率）を確認してください。 • 表オプションBRANCH ALL の指定有無 IS_BRANCH_ALL 列（表オプションBRANCH ALL の指定）を確認してください。 • 表の種類（シングルチャンク表またはマルチチャンク表） IS_CHUNK 列（チャンク指定のCHUNK の指定）を確認してください。 • マルチチャンク表に作成されるチャンク数の最大値 N_CHUNK_RESERVED 列（作成されるチャンク数の最大値）を確認してください。 • アーカイブマルチチャンク表かどうか IS_ARCHIVABLE 列（チャンクアーカイブ指定の有無）を確認してください。 • アーカイブディレクトリのパス名 ARCHIVE_DIRECTORY_PATH 列（アーカイブディレクトリの絶対パス）を確認してください。 • 表の種類（ローストア表またはカラムストア表） STORAGE_FORMAT 列（表データの格納形式）を確認してください。 SQL_TABLES 表の詳細については，「付録 B.2　SQL_TABLES の内容」を参照してください。
2	SQL_COLUMNS 表	SQL_COLUMNS 表を検索すると，次の定義情報を確認できます。 • 列名 COLUMN_NAME 列（列名）を確認してください。 • 列のデータ型と列の定義長 DATA_TYPE_CODE 列（列のデータ型），およびDATA_LENGTH 列（列の定義長）を確認してください。 • DEFAULT 句の指定内容 IS_DEFAULT_COLUMN 列（DEFAULT 句の指定有無），およびDEFAULT_VALUE 列（DEFAULT 句に指定された既定値）を確認してください。 • BRANCH の指定有無 BRANCH 列（列定義のBRANCH の指定値）を確認してください。 • 主キーの指定内容 IS_PRIMARY_KEY_COLUMN 列（主キーを構成する列かどうか），およびPRIMARY_KEY_COLUMN_SEQUENCE_NUMBER 列（主キーの構成順序）を確認してください。 • アーカイブレンジ列かどうか IS_ARCHIVE_RANGE_COLUMN 列（アーカイブレンジ列かどうか）を確認してください。 • カラムストア表の定義時に指定した列データの圧縮方式の指定値 COMPRESSION_TYPE 列（列定義のCOMPRESSION TYPE の指定値）を確認してください。 SQL_COLUMNS 表の詳細については，「付録 B.3　SQL_COLUMNS の内容」を参照してください。

項番	ディクショナリ表	確認できる実表の定義情報
3	• SQL_DIV_TABLE 表 • SQL_DBAREAS 表	左記の表を検索すると，次の定義情報を確認できます。 • 表が格納されている DB エリアの DB エリア ID SQL_DIV_TABLE 表のDBAREA_ID 列（表が格納されている DB エリアの DB エリア ID）を確認してください。 • DB エリアの DB エリア名 SQL_DIV_TABLE 表で確認した DB エリア ID を基に，SQL_DBAREAS 表のDBAREA_NAME 列（DB エリア名）を確認してください。 SQL_DIV_TABLE 表の詳細については，「付録 B.4　SQL_DIV_TABLE の内容」を参照してください。 SQL_DBAREAS 表の詳細については，「付録 B.7　SQL_DBAREAS の内容」を参照してください。
4	• SQL_TABLE_CONSTRAINTS 表 • SQL_KEY_COLUMN_USAGE 表 • SQL_REFERENTIAL_CONSTRAINTS 表	左記の表を検索すると，次の定義情報を確認できます。 • 実表の制約に関する情報（主キーや外部キーの有無） SQL_TABLE_CONSTRAINTS 表を参照してください。 SQL_TABLE_CONSTRAINTS 表の詳細については，「付録 B.14 SQL_TABLE_CONSTRAINTS の内容」を参照してください。 • 主キーと外部キーの構成列に関する情報 SQL_KEY_COLUMN_USAGE 表を参照してください。 SQL_KEY_COLUMN_USAGE 表の詳細については，「付録 B.16 SQL_KEY_COLUMN_USAGE の内容」を参照してください。 • 参照制約に関する情報 SQL_REFERENTIAL_CONSTRAINTS 表を確認してください。 SQL_REFERENTIAL_CONSTRAINTS 表の詳細については，「付録 B.17 SQL_REFERENTIAL_CONSTRAINTS の内容」を参照してください。

11.1.15　実表の表定義を変更する方法（別表を定義して動作検証したあとに運用中の実表の表定義を変更する場合）

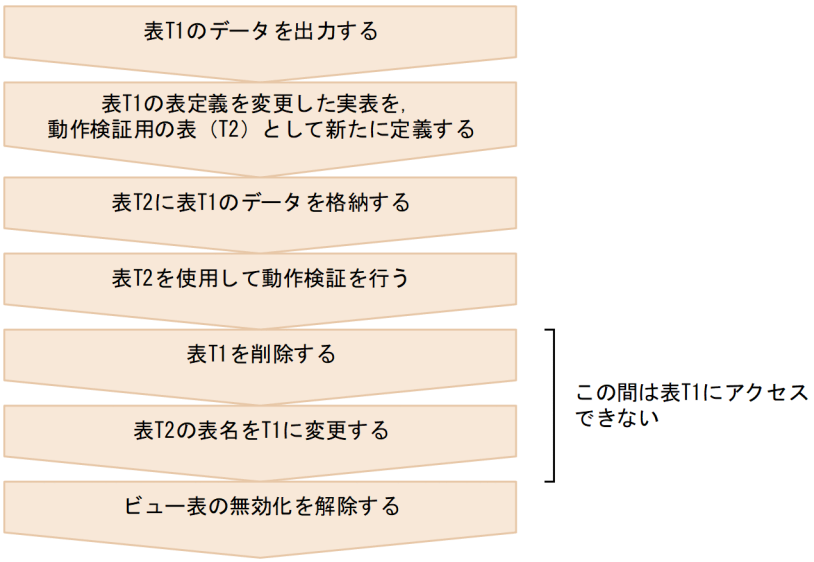
ここでは，次の内容で運用中の実表の表定義を変更する方法を説明します。

- 変更後の表定義を使用して，動作検証用の表を新たに定義する。
- 動作検証用の表を使用して動作検証を行う。
- 表定義を変更する表を削除し，動作検証用の表の名称を削除した表の名称に変更する。

上記の方法で，表定義を変更すると，表定義を変更する表にアクセスできなくなる時間を極力短くできます。

表定義を変更する際の流れと手順を以降で説明しますが，表定義を変更する表をT1，動作検証用の表をT2としています。

図 11-1　表定義を変更する際の流れ



運用中の表T1 の表定義を変更する手順を次に示します。

手順

1. 表T1 に格納されているすべてのデータを出力する

   adbexport コマンドで，表T1 に格納されているすべてのデータをファイルに出力してください。

   表T1 がマルチチャンク表の場合，adbexport コマンドを実行する際の留意事項を次に示します。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

     留意事項は特にありません。表T1 に格納されているすべてのデータをファイルに出力してください。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

     表T1 に格納されているすべてのデータを，チャンク単位でファイルに出力する必要があります。チャンク単位でデータを出力しないと，チャンクの状態・構成を維持したまま，表にデータを格納できません。そのため，チャンク数分，エクスポートを行ってください。

     チャンク単位でデータを出力する場合は，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

2. 表T1 の定義情報を確認する

   表T1 を定義したときのCREATE TABLE 文の指定内容を確認してください。動作検証用の表T2 を定義する際に使用します。

   表T1 を定義したときのCREATE TABLE 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(28)　実表の定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，表T1 を定義したときのCREATE TABLE 文の指定内容が確認できます。

3. 表T1 に定義されているインデクスの定義情報を確認する

   表T1 にインデクスを定義している場合，インデクスを定義したときのCREATE INDEX 文の指定内容を確認してください。表T2 にインデクスを定義する際に使用します。

インデクスを定義したときのCREATE INDEX 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，インデクスを定義したときのCREATE INDEX 文の指定内容が確認できます。

4. 表T1 の表定義を変更して，動作検証用の表（表T2）を定義する

   CREATE TABLE 文で表T2 を定義してください。

   手順 2.で確認した表T1 を定義したときのCREATE TABLE 文の指定内容を流用して，CREATE TABLE 文を作成してください。

5. 表T2 にインデクスを定義する

   表T1 にインデクスが定義されている場合，表T2 にCREATE INDEX 文でインデクスを定義してください。

   手順 3.で確認したインデクスを定義したときのCREATE INDEX 文の指定内容を流用して，CREATE INDEX 文を作成してください。

6. 表T2 にデータを格納する

   adbimport コマンドで，表T2 に手順 1.で出力したすべてのデータを格納してください。

   表T2 がマルチチャンク表の場合，adbimport コマンドを実行する際の留意事項を次に示します。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

     adbimport コマンドには，-d オプションおよび-b オプションは指定しないでください。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

     手順 1.でチャンク単位に出力したデータを，チャンク単位にインポートする必要があります。

     - 手順 1.でカレントチャンクだったチャンクのデータは，必ず最後に格納してください。
     - 最初に格納するデータの場合は，-d オプションおよび-b オプションを指定しないで，adbimport コマンドを実行してください。
     - 2 回目以降に格納するデータの場合は，バックグラウンドインポートで格納してください。-b オプションを指定してadbimport コマンドを実行してください。手順 1.でカレントチャンクだったチャンクのデータも，バックグラウンドインポートで格納してください。

   メモ

   表T2 を定義する際，列を新たに追加したり，列定義を変更したりした場合，列の追加，または列定義の変更に合わせて入力データファイルを修正してください。そのあとに，adbimport コマンドで，表T2 にデータを格納してください。

   例えば，次のケースなどが該当します。

   - 列を新規に追加し，追加した列にデータを格納する場合
   - 列定義を変更し，変更した列定義に合わせたデータを格納する場合

7. 表T2 を使用して動作検証を行う

   動作検証の結果，問題がない場合は，次の手順に進んでください。

表T2 の表定義に問題がある場合は，削除動作にCASCADE を指定してDROP TABLE 文を実行し，表T2 を削除してください。そのあと，手順 4.に戻り，表T2 の表定義を変更して，表T2 を定義し直してください。

8. 表T1 を削除する

   削除動作の指定を省略してDROP TABLE 文を実行し，表T1 を削除してください。

9. 表T2 の表名をT1 に変更する

   ALTER TABLE 文で，表T2 の表名をT1 に変更してください。

10. ビュー表の無効化を解除する

    手順 8.で表T1 を削除したため，表T1 を基表とするビュー表が無効化されています。次のどちらかの方法でビュー表の無効化を解除してください。

    - 表T1 の列定義を変更した場合

      DROP VIEW 文で無効化されたビュー表を削除してください。そのあと，列定義の変更内容を基にCREATE VIEW 文を作成して実行し，ビュー表を再作成してください。

    - 表T1 の列定義以外の表定義を変更した場合

      ALTER VIEW 文でビュー表を再作成し，ビュー表の無効化を解除してください。

ヒント

- 無効化されたビュー表の特定方法，および無効化されたビュー表の削除方法については，「11.2.9　無効化されたビュー表を削除する方法」を参照してください。
- 無効化を解除するビュー表が複数ある場合，ビューレベルの値が小さい順にCREATE VIEW 文またはALTER VIEW 文を実行してください。

メモ

- adbexport コマンドおよびadbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。
- DROP TABLE 文，CREATE TABLE 文，およびCREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』を参照してください。

11.1.16　実表が格納されているデータ用 DB エリアを変更する方法

実表が格納されているデータ用 DB エリアを別のデータ用 DB エリアに変更したい場合は，対象の実表をいったん削除したあとで再定義する必要があります。

> **メモ**
>
> インデクスが格納されているデータ用 DB エリアを別のデータ用 DB エリアに変更したい場合は，「11.3.8　インデクスが格納されているデータ用 DB エリアを変更する方法」を参照してください。

次の手順に従って，実表が格納されているデータ用 DB エリアを，別のデータ用 DB エリアに変更してください。

なお，新たなデータ用 DB エリアを追加する必要がある場合は，「11.10.1　データ用 DB エリアを追加する方法」を参照してください。そのあとで，実表が格納されているデータ用 DB エリアを変更してください。

手順

1. 実表に格納されているすべてのデータを出力する

   adbexport コマンドで，実表に格納されているすべてのデータをファイルに出力してください。

   実表がマルチチャンク表の場合に，adbexport コマンドを実行する際の留意事項を次に示します。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

     留意事項は特にありません。実表に格納されているすべてのデータをファイルに出力してください。

   - **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

     実表に格納されているすべてのデータを，チャンク単位でファイルに出力する必要があります。チャンク単位でデータを出力しないと，チャンクの状態・構成を維持したまま，実表にデータを格納できません。そのため，チャンク数分，エクスポートを行ってください。

     チャンク単位でデータを出力する場合は，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

2. 実表の定義情報を確認する

   実表を定義したときのCREATE TABLE 文の指定内容を確認してください。別のデータ用 DB エリアに実表を定義する際に使用します。

   実表を定義したときのCREATE TABLE 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(28)　実表の定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，実表を定義したときのCREATE TABLE 文の指定内容が確認できます。

3. 実表に定義されているインデクスの定義情報を確認する

   実表にインデクスを定義している場合，インデクスを定義したときのCREATE INDEX 文の指定内容を確認してください。別のデータ用 DB エリアに定義した実表に，インデクスを定義する際に使用します。

   インデクスを定義したときのCREATE INDEX 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，インデクスを定義したときのCREATE INDEX 文の指定内容が確認できます。

4. 実表を削除する

   *削除動作*の指定を省略したDROP TABLE 文で，実表を削除してください。

5. 実表を再定義する

CREATE TABLE 文で，別のデータ用 DB エリアに実表を定義してください。CREATE TABLE 文を作成する際の留意事項を次に示します。

- 手順 2.で確認した実表を定義したときのCREATE TABLE 文の指定内容を流用して，CREATE TABLE 文を作成してください。
- CREATE TABLE 文のIN *DB エリア名*に，実表を格納する別のデータ用 DB エリアの DB エリア名を指定してください。

6. 実表にインデクスを定義する

削除した実表にインデクスを定義していた場合，再定義した実表にCREATE INDEX 文でインデクスを定義してください。CREATE INDEX 文を作成する際の留意事項を次に示します。

- 手順 3.で確認したインデクスを定義したときのCREATE INDEX 文の指定内容を流用して，CREATE INDEX 文を作成してください。

7. 実表にデータを格納する

adbimport コマンドで，手順 5.で定義した実表に，手順 1.で出力したすべてのデータを格納してください。

実表がマルチチャンク表の場合に，adbimport コマンドを実行する際の留意事項を次に示します。

- **マルチチャンク表のチャンクの状態・構成を維持する必要がない場合**

  adbimport コマンドには，-d オプションおよび-b オプションは指定しないでください。
- **マルチチャンク表のチャンクの状態・構成を維持する必要がある場合**

  手順 1.でチャンク単位に出力したデータを，チャンク単位にインポートする必要があります。
  - 手順 1.でカレントチャンクだったチャンクのデータは，必ず最後に格納してください。
  - 最初に格納するデータの場合は，-d オプションおよび-b オプションを指定しないで，adbimport コマンドを実行してください。
  - 2 回目以降に格納するデータの場合は，バックグラウンドインポートで格納してください。-b オプションを指定してadbimport コマンドを実行してください。手順 1.でカレントチャンクだったチャンクのデータも，バックグラウンドインポートで格納してください。

8. ビュー表の無効化を解除する

手順 4.で削除した実表を基表とするビュー表は無効化されます。そのため，実表の定義後に，ビュー表の無効化を解除する必要があります。「11.2.8　ビュー表の無効化を解除する方法」の「(6)　表を誤って削除してしまったことによってビュー表が無効化された場合」を参照して，ビュー表の無効化を解除してください。

これで，実表が格納されているデータ用 DB エリアを，別のデータ用 DB エリアに変更できます。

メモ

- adbexport コマンドおよびadbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

- DROP TABLE 文，CREATE TABLE 文およびCREATE INDEX 文については，マニュアル『HADB SQL リファレンス』を参照してください。

11.1.17　実表を削除する方法

実表を削除するには，定義系 SQL のDROP TABLE 文を実行します。

重要

- 実表を削除した場合，削除対象の実表に依存するビュー表が削除されます（または無効化されます）。そのため，実表を削除する前に，削除対象の実表に依存するビュー表があるかどうかを確認し，削除または無効化されても問題ないことを確認してください。
- 主キーを定義している実表を削除した場合，削除した実表を被参照表とする外部キーも削除されます（ほかのスキーマの外部キーも削除されます）。そのため，実表を削除する前に，外部キーが削除されても問題ないことを確認してください。ただし，削除動作にRESTRICT を指定した場合は SQL 文がエラーになります。

実表の削除例を次に示します。

例題

店舗表（SHOPSLIST）を削除します。

手順

1. 削除対象の実表に依存するビュー表を確認する

   ディクショナリ表を検索すると，削除対象の実表に依存するビュー表を確認できます。ディクショナリ表の検索例については，「11.2.11　依存するビュー表を確認する方法」を参照してください。

2. 削除対象の実表を被参照表とする外部キーを確認する

   主キーを定義している実表を削除する場合に必要な作業です。

   ディクショナリ表を検索すると，削除対象の実表を被参照表とする外部キーを確認できます。ディクショナリ表の検索例については，「付録 B.22　ディクショナリ表の検索」の「(21)　主キーを参照する外部キーを調べる場合」を参照してください。

3. 削除対象の実表が更新不可状態かどうかを確認する

   実表を削除する前に，削除する実表に対して，コマンドの再実行が必要かどうか（更新不可状態かどうか）を確認してください。確認する方法については，「10.9.2　実表の状態や使用量を確認したい場合」の「(1)　実表が更新不可状態かどうかを確認する場合」を参照してください。

重要

コマンドの再実行が必要な場合

コマンドを再実行してください。そのあとで，実表を削除してください。

コマンドの再実行が必要な状態のままで（更新不可状態のままで）実表を削除した場合

中断したコマンドによって作成された作業用一時ファイルが残ることがあります。作業用一時ファイルを削除するには，すべての実表の更新不可状態を解除する必要があります。作業用一時ファイルの削除については，「15.2.6　コマンド実行時の作業用一時ファイルを格納するディスクの空き容量が不足した場合の対処方法」の「(2)　ディスクに不要な作業用一時ファイルが残っている場合」を参照してください。

4. DROP TABLE 文を実行する

```
DROP TABLE ”SHOPSLIST” CASCADE
```

メモ

上記の例では，削除動作にCASCADE を指定しているため，削除対象の実表に依存するすべてのビュー表が削除されます。削除動作の指定を省略した場合は，削除対象の実表に依存するすべてのビュー表が無効化されます。

11.1.18　配列型の列を定義した実表の運用

配列型の列を定義した実表の運用方法を説明します。

なお，ここでは，実表を表と表記しています。

(1)　表の定義

CREATE TABLE 文で表を定義します。定義する表に配列型の列を定義する場合は，CREATE TABLE 文の列定義で，列のデータ型に配列型（ARRAY 型）を指定します。配列型の列を定義した商品表（PRODUCT）の定義例を次に示します。

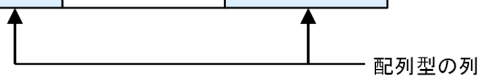
(例)

商品表（PRODUCT）の定義条件を次に示します。

- SIZE 列（サイズ）とCOLOR 列（色）を配列型の列として定義します。
- SIZE 列は，要素データ型をCHAR(1)とし，最大要素数を 3 とします。
- COLOR 列は，要素データ型をVARCHAR(8)とし，最大要素数を 6 とします。

商品表（PRODUCT）

PCODE	PNAME	SIZE	PRICE	COLOR
101	BLOUSE	S	3500	GRAY
		M		RED
		L		BLUE
202	SHIRT	M	3640	GRAY
353	SKIRT	S	4760	PINK
		M		YELLOW
412	SWEATER	M	8400	GREEN
591	SOCKS	S	300	GRAY
		M		BROWN
		L		BLUE
671	SWEAT SHIRT	S	4500	RED
		M		BROWN



CREATE TABLE 文の指定例

```
CREATE TABLE ”PRODUCT” (”PCODE” CHAR(4),
                        ”PNAME” VARCHAR(20),
                        ”SIZE”  CHAR(1) ARRAY[3],        ...1
                        ”PRICE” INTEGER,
                        ”COLOR” VARCHAR(8) ARRAY[6])     ...2
               IN ”DBAREA01”
               CHUNK=120
               STORAGE FORMAT COLUMN                     ...3
```

[説明]

1. 配列型の列であるSIZE 列を定義します。
2. 配列型の列であるCOLOR 列を定義します。
3. 配列型の列を定義した表は，カラムストア表として定義する必要があります。

重要

配列型の列を定義した表を定義する前に，「5.2.13　配列型の列の定義【カラムストア表】」を参照して制限事項などを確認してください。

メモ

CREATE TABLE 文の指定形式および各オプションの詳細については，マニュアル『HADB　SQL リファレンス』の『CREATE TABLE（表の定義）』を参照してください。

(2)　表の検索

配列型の列を定義した表の検索例を次に示します。

（例 1）

商品表（PRODUCT）から，商品名（PNAME）とサイズ（SIZE）を検索します。SIZE 列は配列型の列です。

SELECT 文の指定例

```
SELECT "PNAME","SIZE" FROM "PRODUCT"
```

上記の下線部分のように配列型の列名だけを指定すると，SIZE 列中の配列要素の値がすべて出力されます。

検索結果

PNAME	SIZE
BLOUSE	{S,M,L}
SHIRT	{M}
SKIRT	{S,M}
SWEATER	{M}
SOCKS	{S,M,L}
SWEAT SHIRT	{S,M}

（例 2）

商品表（PRODUCT）から，商品名（PNAME）とサイズ（SIZE）の先頭の配列要素を検索します。

SELECT 文の指定例

```
SELECT "PNAME","SIZE"[1] FROM "PRODUCT"
```

上記の下線部分のように，配列型の列名と，検索対象とする配列要素の要素番号を指定します。この例の場合，先頭の配列要素を検索するため，要素番号に 1 を指定しています。下線部分の指定を配列要素参照といいます。

検索結果

PNAME	SIZE
BLOUSE	S
SHIRT	M
SKIRT	S
SWEATER	M
SOCKS	S
SWEAT SHIRT	S

（例 3）

商品表（PRODUCT）から，色（COLOR）の配列要素に'BLUE'がある商品名（PNAME）と単価（PRICE）を検索します。

SELECT 文の指定例

```
SELECT "PNAME","PRICE" FROM "PRODUCT" WHERE "COLOR"[ANY] = 'BLUE'
```

COLOR 列の全配列要素を検索対象とする場合は，上記の下線部分のように，配列型の列名と[ANY]を指定します。下線部分の指定を配列要素参照といいます。

検索結果

PNAME	PRICE
BLOUSE	3500
SOCKS	300

メモ

配列要素参照の指定形式の詳細については，マニュアル『HADB　SQL リファレンス』の『配列要素参照』を参照してください。

(3)　表の全行削除

表の全行を削除する場合は，TRUNCATE TABLE 文を実行します。

(例)

商品表（PRODUCT）の全行を削除します。

```
TRUNCATE TABLE "PRODUCT"
```

重要

配列型の列を定義した表には，DELETE 文を実行できません。また，INSERT 文およびUPDATE 文も実行できません。

(4)　最大要素数の変更

配列型の列の最大要素数を変更する場合は，ALTER TABLE 文を実行します。

(例)

商品表（PRODUCT）のSIZE 列の最大要素数を 5 に変更します。

```
ALTER TABLE "PRODUCT"
    CHANGE COLUMN "SIZE" ARRAY[5]
```

重要

- 最大要素数を大きくすることだけができます。最大要素数を小さくすることはできません。
- 表を定義した HADB ユーザだけが，ALTER TABLE 文で最大要素数を変更できます。

メモ

ALTER TABLE 文の指定形式および規則については，マニュアル『HADB　SQL リファレンス』の『ALTER TABLE（表定義の変更)』を参照してください。

なお，最大要素数を変更する場合，次に示す影響があります。影響がないかどうかを確認してください。

- 最大要素数を変更した表に依存するビュー表がある場合

  最大要素数を変更した表に依存するビュー表は無効化されます。ビュー表の定義時に，最大要素数を変更した列名を明示的に指定しているかどうかに関係なく，ビュー表は無効化されます。

  無効化されるビュー表（依存するビュー表）を調べる場合は，「11.2.11　依存するビュー表を確認する方法」を参照してください。無効化されるビュー表がある場合は，ALTER TABLE 文で最大要素数を変更したあとで，ビュー表の無効化を解除する必要があります。

  ビュー表の無効化を解除する方法については，「11.2.8　ビュー表の無効化を解除する方法」の「(4) ALTER TABLE 文で配列型の列の最大要素数を変更したことによってビュー表が無効化された場合」を参照してください。

また，次に示す場合は，ALTER TABLE 文で最大要素数を変更できません。

- 操作対象の実表が更新不可状態になっている場合

  実表の更新不可状態を解除したあとで，ALTER TABLE 文で最大要素数を変更してください。実表の更新不可状態を解除する方法については，「15.8.1　実表が更新不可状態になった場合」の「(1)　実表の更新不可状態を解除する方法」を参照してください。

(5)　配列型の列の追加

表に配列型の列を追加する場合は，ALTER TABLE 文を実行します。

(例)

商品表（PRODUCT）に，配列型の列であるITEM_CODE 列を追加します。ITEM_CODE 列の要素データ型はCHAR(8)，最大要素数は 10 とします。

```
ALTER TABLE ”PRODUCT”
    ADD COLUMN ”ITEM_CODE” CHAR(8) ARRAY[10]
```

重要

- カラムストア表に配列型の列を追加できます。ローストア表には配列型の列を追加できません。
- B-tree インデクスまたは一意性制約を定義している表には，配列型の列を追加できません。

メモ

- 表に列を追加する際の留意事項については，「11.1.2　実表に列を追加する方法」を参照してください。
- ALTER TABLE 文の指定形式および規則については，マニュアル『HADB　SQL リファレンス』の『ALTER TABLE（表定義の変更)』を参照してください。

11.2　ビュー表の運用

ここでは，ビュー表の運用について説明します。

ビュー表については，「2.1.2　ビュー表」を参照してください。

11.2.1　ビュー表を定義する方法

ビュー表を定義するには，定義系 SQL のCREATE VIEW 文を実行します。

ビュー表の定義例を次に示します。

例題

実表のCUSTOMER 表およびSTOCK 表から，30 歳代の顧客が購入した商品一覧を示すビュー表として，CUSTOMER30s 表を定義します。

図 11-2　実表の CUSTOMER 表および STOCK 表の例

■CUSTOMER（実表）

AGE（年代）	SEX（性別）	GNO（品番）	PNUM（購入数）
30	F	20180	1
20	F	20130	1
30	M	20240	1
40	F	20200	1
50	F	20240	1
50	M	20130	1
60	M	20200	1
20	F	20220	1
30	F	20150	1
50	M	20290	1
60	F	20250	1
50	F	20310	1
40	M	20210	1
20	F	20280	1
30	M	20180	1

■STOCK（実表）

GNO（品番）	PCODE（商品コード）	PNAME（商品名）	PRICE（単価）	QUANTITY（在庫量）
20180	C20	Blouse	2000	26
20190	C77	Blouse	2800	105
20130	P10	Shirt	3000	70
20230	K18	Sweater	3500	12
20200	C89	Blouse	3500	30
20140	P23	Shirt	3500	60
20280	L10	Socks	380	200
20150	P32	Shirt	4800	50
20290	L50	Socks	490	260
20240	K20	Sweater	5000	15
20160	P35	Shirt	5580	120
20250	K22	Sweater	7500	36
20300	L80	Socks	750	170
20210	C91	Blouse	3000	35
20260	K24	Sweater	6000	20
20310	L90	Socks	800	230
20220	C95	Blouse	2500	110

■ビュー表（CUSTOMER30s 表）を定義する SQL 文

```
CREATE VIEW ”CUSTOMER30s”
  (”AGE”,”SEX”,”GNO”,”PNAME”,”PRICE”)
  AS SELECT
   ”CUSTOMER”.”AGE”, ”CUSTOMER”.”SEX”, ”CUSTOMER”.”GNO”,
   ”STOCK”.”PNAME”, ”STOCK”.”PRICE”
  FROM ”CUSTOMER”,”STOCK”
    WHERE ”CUSTOMER”.”AGE”=30 AND ”CUSTOMER”.”GNO”=”STOCK”.”GNO”
```

図 11-3　ビュー表（CUSTOMER30s 表）の定義例

■CUSTOMER30s（ビュー表）

AGE（年代）	SEX（性別）	GNO（品番）	PNAME（商品名）	PRICE（単価）
30	F	20180	Blouse	2000
30	M	20240	Sweater	5000
30	F	20150	Shirt	4800
30	M	20180	Blouse	2000

なお，定義するビュー表が，更新可能ビュー表または読み取り専用ビュー表のどちらになるかは，CREATE VIEW 文中に指定する「AS *問合せ指定*」の指定内容によって決まります。詳細については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE VIEW（ビュー表の定義）』を参照してください。

11.2.2　ビュー表のデータを検索，更新，追加，削除する方法

ビュー表を検索するには，操作系 SQL のSELECT 文を実行します。実表を検索する場合と同様です。

ビュー表の検索例を次に示します。

例題

「11.2.1　ビュー表を定義する方法」で定義したCUSTOMER30s 表から，性別（SEX）が女性（F）の行を検索します。

```
SELECT * FROM "CUSTOMER30s" WHERE "SEX"= 'F'
```

更新可能ビュー表で，かつビュー表に対する必要なアクセス権限がある場合，ビュー表のデータを更新，追加，または削除できます。ビュー表のデータを更新，追加，または削除するには，操作系 SQL のUPDATE 文，INSERT 文，DELETE 文を実行します。実表を更新，追加，または削除する場合と同様です。

11.2.3　ビュー表のデータをファイルに出力する方法（データをエクスポートする方法）

ビュー表のデータをファイルに出力したい場合は，「11.1.9　実表のデータをファイルに出力する方法（データをエクスポートする方法）」を参照してください。実表のデータをファイルに出力する場合と同様です。

11.2.4　ビュー表を再作成する方法

ビュー表を再作成するには，定義系 SQL のALTER VIEW 文を実行します。ALTER VIEW 文については，マニュアル『HADB　SQL リファレンス』の『ALTER VIEW（ビュー表の再作成）』を参照してください。

自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有しているビュー表だけを再作成できます。ほかの HADB ユーザが所有しているビュー表は再作成できません。

ビュー表の無効化要因を対策したあとに，ALTER VIEW 文でビュー表を再作成すると，ビュー表の無効化を解除できます。詳細については，「11.2.8　ビュー表の無効化を解除する方法」を参照してください。

なお，ALTER VIEW 文では，既存のビュー定義を変更できません。

> **重要**
>
> ビュー表を再作成した場合，再作成対象のビュー表に依存するすべてのビュー表が無効化されます。そのため，ビュー表を再作成する前に，再作成対象のビュー表に依存するビュー表があるかどうかを確認し，無効化されても問題ないことを確認してください。

11.2.5　ビュー表を削除する方法

ビュー表を削除するには，定義系 SQL のDROP VIEW 文を実行します。

自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有しているビュー表だけを削除できます。ほかの HADB ユーザが所有しているビュー表は削除できません。

> **重要**
>
> ビュー表を削除した場合，削除対象のビュー表に依存するすべてのビュー表が削除されます（または無効化されます）。そのため，ビュー表を削除する前に，削除対象のビュー表に依存するビュー表があるかどうかを確認し，削除または無効化されても問題ないことを確認してください。

ビュー表の削除例を次に示します。

例題

「11.2.1　ビュー表を定義する方法」で定義したCUSTOMER30s 表を削除します。

手順

1. CUSTOMER30s に依存するビュー表を確認する

   ディクショナリ表を検索すると，削除対象のビュー表に依存するビュー表を確認できます。ディクショナリ表の検索例については，「11.2.11　依存するビュー表を確認する方法」を参照してください。

2. CUSTOMER30s をDROP VIEW 文で削除する

```
DROP VIEW "CUSTOMER30s" CASCADE
```

> **メモ**
>
> 上記の例では，削除動作にCASCADE を指定しているため，削除対象のビュー表（CUSTOMER30s）に依存するすべてのビュー表が削除されます。削除動作の指定を省略した場合は，削除対象のビュー表に依存するすべてのビュー表が無効化されます。

11.2.6　ビュー表の更新可否を調べる方法

INSERT 文，UPDATE 文およびDELETE 文で，ビュー表を更新できるかどうかを調べるには，SELECT 文でディクショナリ表SQL_VIEWS を検索します。

ビュー表の更新可否を調べる例を次に示します。

例題

ADBUSER01 が定義したビュー表CUSTOMER30s が，更新可能ビュー表かどうかを調べます。

```
SELECT ”IS_UPDATABLE” FROM ”MASTER”.”SQL_VIEWS”
  WHERE ”TABLE_SCHEMA”=’ADBUSER01’ AND ”TABLE_NAME”=’CUSTOMER30s’
```

検索結果が'Y'の場合は，更新可能ビュー表であるため更新できます。'N'の場合は，読み取り専用ビュー表であるため更新できません。

ディクショナリ表SQL_VIEWS については，「付録 B.9　SQL_VIEWS の内容」を参照してください。

11.2.7　ビュー表が無効化されているかどうかを確認する方法

ディクショナリ表を検索すると，ビュー表が無効化されているかどうかを確認できます。ディクショナリ表の検索例を次に示します。

例題 1

HADB ユーザADBUSER01 が定義したビュー表CUSTOMER30s が，無効化されているかどうかを調べます。

次のSELECT 文を実行して，ディクショナリ表を検索してください。

```
SELECT ”IS_INVALID” FROM ”MASTER”.”SQL_VIEWS”
    WHERE ”TABLE_SCHEMA”=’ADBUSER01’ AND ”TABLE_NAME”=’CUSTOMER30s’
```

検索結果

```
IS_INVALID
----------
Y
```

検索結果がY の場合は，CUSTOMER30s が無効化されています。

検索結果がナル値の場合は，CUSTOMER30s は無効化されていません。

例題 2

HADB ユーザADBUSER01 が定義したビュー表のうち，無効化されているビュー表の一覧を表示します。

```
SELECT ”TABLE_NAME” FROM ”MASTER”.”SQL_VIEWS”
    WHERE ”TABLE_SCHEMA”=’ADBUSER01’ AND ”IS_INVALID”=’Y’
```

検索結果

```
TABLE_NAME
-----------
CUSTOMER30s
CUSTOMER40s
CUSTOMER50s
```

検索結果には，無効化されているビュー表の一覧が表示されます。

11.2.8　ビュー表の無効化を解除する方法

ビュー表の無効化を解除する方法は，ビュー表が無効化された原因によって異なります。それぞれの方法について説明します。

メモ

ビュー表の無効化要因がわからなくなってしまった場合

ビュー表にALTER VIEW 文を実行すると，ALTER VIEW 文がエラーになります。そのとき出力されるエラーメッセージを参照すると，ビュー表が無効化された要因を特定できます。

(1)　ALTER TABLE 文で表の種別を変更したことによってビュー表が無効化された場合

次に示す場合は，種別を変更した表に依存するすべてのビュー表が無効化されます。

- ALTER TABLE 文を実行して，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更した場合
- ALTER TABLE 文を実行して，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更した場合

ビュー表の無効化を解除する手順を次に示します。

例題

CUSTOMER 表をレギュラーマルチチャンク表からアーカイブマルチチャンク表に変更したため，CUSTOMER 表に依存するすべてのビュー表が無効化されました。CUSTOMER 表に依存するすべてのビュー表の無効化を解除します。

手順

1. 無効化されたビュー表の一覧を確認する

ディクショナリ表を検索すると，無効化されたビュー表の一覧を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

この例では，ビュー表CUSTOMER30s，CUSTOMER40s，およびCUSTOMER50s が無効化されたとします。

2. 無効化されたビュー表をALTER VIEW 文で再作成する

```
ALTER VIEW ”CUSTOMER30s” RECREATE
```

同様に，ビュー表CUSTOMER40s およびCUSTOMER50s も，ALTER VIEW 文で再作成してください。ビュー表の無効化を解除できます。

ヒント

再作成するビュー表が複数ある場合，ビューレベルの値が小さい順にALTER VIEW 文を実行してください。

(2)　ALTER TABLE 文で表の列名を変更したことによってビュー表が無効化された場合

ALTER TABLE 文で表の列名を変更した場合，列名を変更した表に依存するすべてのビュー表が無効化されます。ビュー表の無効化を解除する手順を次に示します。

なお，変更対象の列名をCREATE VIEW 文中に指定しているかどうかによって，ビュー表の無効化を解除する手順が異なります。

例題

CUSTOMER 表の列名を変更したため，CUSTOMER 表に依存するすべてのビュー表が無効化されました。CUSTOMER 表に依存するすべてのビュー表の無効化を解除します。

手順（変更対象の列名をCREATE VIEW 文中に指定していない場合）

1. 無効化されたビュー表の一覧を確認する

ディクショナリ表を検索すると，無効化されたビュー表の一覧を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

この例では，ビュー表CUSTOMER30s，CUSTOMER40s，およびCUSTOMER50s が無効化されたとします。

2. 無効化されたビュー表をALTER VIEW 文で再作成する

```
ALTER VIEW ”CUSTOMER30s” RECREATE
```

同様に，ビュー表CUSTOMER40s およびCUSTOMER50s も，ALTER VIEW 文で再作成してください。ビュー表の無効化を解除できます。

ヒント

再作成するビュー表が複数ある場合，ビューレベルの値が小さい順にALTER VIEW 文を実行してください。

手順（変更対象の列名をCREATE VIEW 文中に指定している場合）

1. 無効化されたビュー表の一覧と，それらのビュー表の定義情報を確認する

   ディクショナリ表を検索すると，無効化されたビュー表の一覧と，それらのビュー表の定義情報を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(32)　依存するビュー表の一覧，およびそれらのビュー表の定義情報を調べる場合」を参照してください。

   この例では，ビュー表CUSTOMER30s，CUSTOMER40s，およびCUSTOMER50s が無効化されたとします。

2. 無効化されたビュー表をDROP VIEW 文で削除する

```
DROP VIEW "CUSTOMER30s"
DROP VIEW "CUSTOMER40s"
DROP VIEW "CUSTOMER50s"
```

   手順 1.で確認した無効化されたビュー表をすべて削除してください。

3. CREATE VIEW 文の指定内容を修正する

   手順 1.で調べたCREATE VIEW 文の指定内容のうち，列名変更に該当する個所を修正します。

   （例）表T1 の列名をC3 からC33 に変更した場合

   ＜修正前＞

```
CREATE VIEW "VT1" ("VC1","VC2","VC3")
    AS SELECT "C1","C2","C3" FROM "T1"
```

   ＜修正後＞

```
CREATE VIEW "VT1" ("VC1","VC2","VC3")
    AS SELECT "C1","C2","C33" FROM "T1"
```

4. 無効化されたビュー表をCREATE VIEW 文で再定義する

```
CREATE VIEW "CUSTOMER30s" ("AGE","SEX","GNO","PNAME","PRICE")
  AS SELECT "CUSTOMER"."AGE",
            "CUSTOMER"."SEX",
            "CUSTOMER"."GNO",
            "STOCK"."PNAME",
            "STOCK"."PRICE"
       FROM "CUSTOMER","STOCK"
         WHERE "CUSTOMER"."AGE"=30 AND "CUSTOMER"."GNO"="STOCK"."GNO"
```

   同様に，CUSTOMER40s およびCUSTOMER50s もCREATE VIEW 文でビュー表を再定義してください。

ヒント

再定義するビュー表が複数ある場合，ビューレベルの値が小さい順にCREATE VIEW 文を実行してください。

(3) ALTER TABLE 文で表の表名を変更したことによってビュー表が無効化された場合

ALTER TABLE 文で表の表名を変更した場合，表名を変更した表に依存するすべてのビュー表が無効化されます。ビュー表の無効化を解除する手順を次に示します。

例題

店舗表（SHOPSLIST）の表名を，STORELIST に変更したため，SHOPSLIST 表に依存するすべてのビュー表が無効化されました。SHOPSLIST 表に依存するすべてのビュー表を削除し，STORELIST 表のビュー表を作成します。

手順

1. 無効化されたビュー表の一覧と，それらのビュー表の定義情報を確認する

   ディクショナリ表を検索すると，無効化されたビュー表の一覧と，それらのビュー表の定義情報を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(32)　依存するビュー表の一覧，およびそれらのビュー表の定義情報を調べる場合」を参照してください。

   この例では，ビュー表VLIST01，VLIST02，およびVLIST03 が無効化されたとします。

2. 無効化されたビュー表をDROP VIEW 文で削除する

```
DROP VIEW "VLIST01"
DROP VIEW "VLIST02"
DROP VIEW "VLIST03"
```

   手順 1.で確認した無効化されたビュー表をすべて削除してください。

3. CREATE VIEW 文の指定内容を修正する

   手順 1.で調べたCREATE VIEW 文の指定内容のうち，表名変更に該当する個所を修正します。

   （例）

   ＜修正前＞

```
CREATE VIEW "VLIST01" ("VC1","VC2","VC3")
    AS SELECT "C1","C2","C3" FROM "SHOPSLIST"
```

   ＜修正後＞

```
CREATE VIEW "VLIST01" ("VC1","VC2","VC3")
    AS SELECT "C1","C2","C3" FROM "STORELIST"
```

4. 無効化されたビュー表をCREATE VIEW 文で再定義する

   手順 3.の修正後のCREATE VIEW 文を実行します。

ヒント

再定義するビュー表が複数ある場合，ビューレベルの値が小さい順にCREATE VIEW 文を実行してください。

(4) ALTER TABLE 文で配列型の列の最大要素数を変更したことによってビュー表が無効化された場合

ALTER TABLE 文で配列型の列の最大要素数を変更した場合，最大要素数を変更した表に依存するすべてのビュー表が無効化されます。ビュー表の無効化を解除する手順を次に示します。

例題

CUSTOMER 表に定義されている配列型の列の最大要素数を変更したため，CUSTOMER 表に依存するすべてのビュー表が無効化されました。CUSTOMER 表に依存するすべてのビュー表の無効化を解除します。

手順

1. 無効化されたビュー表の一覧を確認する

   ディクショナリ表を検索すると，無効化されたビュー表の一覧を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

   この例では，ビュー表CUSTOMER30s，CUSTOMER40s，およびCUSTOMER50s が無効化されたとします。

2. 無効化されたビュー表をALTER VIEW 文で再作成する

```
ALTER VIEW ”CUSTOMER30s” RECREATE
```

   同様に，ビュー表CUSTOMER40s およびCUSTOMER50s も，ALTER VIEW 文で再作成してください。ビュー表の無効化を解除できます。

ヒント

再作成するビュー表が複数ある場合，ビューレベルの値が小さい順にALTER VIEW 文を実行してください。

(5) 基表に対する SELECT 権限を誤って取り消したことによってビュー表が無効化された場合

基表に対するSELECT 権限を誤って取り消してしまった場合，その表に依存するすべてのビュー表が無効化されます。ビュー表の無効化を解除する手順を次に示します。

例題

基表に対するSELECT 権限を誤って取り消してしまったため，基表に依存するすべてのビュー表が無効化されました。ビュー表と基表を特定し，基表に依存するすべてのビュー表の無効化を解除します。

手順

1. 無効化されたビュー表の一覧を確認する

   ディクショナリ表を検索すると，無効化されたビュー表の一覧を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

   この例では，ビュー表CUSTOMER30s が無効化されたとします。

2. 無効化されたビュー表の基表を確認する

   手順 1.で確認した無効化されたビュー表を基に，ディクショナリ表を検索すると，無効化されたビュー表の基表を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(35)　ビュー表の基表である実表とビュー表を調べる場合」を参照してください。

   この例では，無効化されたビュー表CUSTOMER30s の基表が，CUSTOMER 表であるとします。

3. 基表に対するSELECT 権限をGRANT 文で付与する

   手順 2.で確認した基表のCUSTOMER 表に対して，GRANT 文でSELECT 権限を付与します。

   ```
   GRANT SELECT ON ”CUSTOMER” TO ”ADBUSER01”
   ```

   下線部分は，ビュー表CUSTOMER30s を所有している HADB ユーザの認可識別子を指定します。

4. 無効化されたビュー表をALTER VIEW 文で再作成する

   ```
   ALTER VIEW ”CUSTOMER30s” RECREATE
   ```

   ビュー表CUSTOMER30s をALTER VIEW 文で再作成してください。ビュー表の無効化を解除できます。

   > **ヒント**
   >
   > 再作成するビュー表が複数ある場合，ビューレベルの値が小さい順にALTER VIEW 文を実行してください。

(6)　表を誤って削除してしまったことによってビュー表が無効化された場合

表を誤って削除してしまった場合※，削除してしまった表に依存するすべてのビュー表が無効化されます。ビュー表の無効化を解除する手順を次に示します。

注※

削除動作の指定を省略したDROP TABLE 文，またはDROP VIEW 文を実行した場合が該当します。

例題

実表のCUSTOMER 表を誤って削除してしまったため，削除してしまった表に依存するすべてのビュー表が無効化されました。実表を再定義して，依存するすべてのビュー表の無効化を解除します。

手順

1. 無効化されたビュー表の一覧を確認する

ディクショナリ表を検索すると，無効化されたビュー表の一覧を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

この例では，ビュー表CUSTOMER30s が無効化されたとします。

2. 誤って削除してしまった表を再定義する

   - 削除してしまった表が実表の場合は，実表をCREATE TABLE 文で再定義してください。
   - 削除してしまった表がビュー表の場合は，ビュー表をCREATE VIEW 文で再定義してください。

   この例では，実表のCUSTOMER 表をCREATE TABLE 文で再定義します。

3. ほかの HADB ユーザが再定義した表に対するSELECT 権限をGRANT 文で付与してもらう

   手順 2.で再定義した表がほかの HADB ユーザの表である場合，その表に対するSELECT 権限をGRANT 文で付与してもらってください。

   なお，自分で表を再定義した場合は，この手順は実施不要です。

```
GRANT SELECT ON ”CUSTOMER” TO ”ADBUSER01”
```

   下線部分は，ビュー表CUSTOMER30s を所有している HADB ユーザの認可識別子を指定します。

4. 無効化されたビュー表をALTER VIEW 文で再作成する

```
ALTER VIEW ”CUSTOMER30s” RECREATE
```

   ビュー表CUSTOMER30s をALTER VIEW 文で再作成してください。ビュー表の無効化を解除できます。

> **ヒント**
>
> 再作成するビュー表が複数ある場合，ビューレベルの値が小さい順にALTER VIEW 文を実行してください。

(7)　ALTER VIEW 文で基表であるビュー表を再作成したことによってビュー表が無効化された場合

ALTER VIEW 文で基表であるビュー表を再作成した場合，再作成したビュー表に依存するすべてのビュー表が無効化されます。ビュー表の無効化を解除する手順を次に示します。

なお，基表であるビュー表の列名が変更されているかどうか，および変更対象の列名をCREATE VIEW 文中に指定しているかどうかによって，ビュー表の無効化を解除する手順が異なります。

手順（基表であるビュー表の列名が変更されていない場合，または変更対象の列名をCREATE VIEW 文中に指定していない場合）

基表であるビュー表の列名が変更されていない場合は，次の手順でビュー表の無効化を解除してください。また，基表であるビュー表の列名が変更されているが，変更対象の列名をCREATE VIEW 文中に指定していない場合も，次の手順でビュー表の無効化を解除してください。

1. 無効化されたビュー表の一覧を確認する

ディクショナリ表を検索すると，無効化されたビュー表の一覧を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(33)　依存するビュー表の一覧を調べる場合」を参照してください。

この例では，ビュー表CUSTOMER30s，CUSTOMER40s，およびCUSTOMER50s が無効化されたとします。

2. 無効化されたビュー表をALTER VIEW 文で再作成する

```
ALTER VIEW ”CUSTOMER30s” RECREATE
```

同様に，ビュー表CUSTOMER40s およびCUSTOMER50s も，ALTER VIEW 文で再作成してください。ビュー表の無効化を解除できます。

ヒント

再作成するビュー表が複数ある場合，ビューレベルの値が小さい順にALTER VIEW 文を実行してください。

手順（変更対象の列名をCREATE VIEW 文中に指定している場合）

基表であるビュー表の列名が変更されて，かつ変更対象の列名をCREATE VIEW 文中に指定している場合は，次の手順でビュー表の無効化を解除してください。

1. 無効化されたビュー表の一覧と，それらのビュー表の定義情報を確認する

ディクショナリ表を検索すると，無効化されたビュー表の一覧と，それらのビュー表の定義情報を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(32)　依存するビュー表の一覧，およびそれらのビュー表の定義情報を調べる場合」を参照してください。

この例では，ビュー表CUSTOMER30s，CUSTOMER40s，およびCUSTOMER50s が無効化されたとします。

2. 無効化されたビュー表をDROP VIEW 文で削除する

```
DROP VIEW ”CUSTOMER30s”
DROP VIEW ”CUSTOMER40s”
DROP VIEW ”CUSTOMER50s”
```

手順 1.で確認した無効化されたビュー表をすべて削除してください。

3. CREATE VIEW 文の指定内容を修正する

手順 1.で調べたCREATE VIEW 文の指定内容のうち，列名変更に該当する個所を修正します。

（例）表T1 の列名をC3 からC33 に変更した場合

＜修正前＞

```
CREATE VIEW ”VT1” (”VC1”,”VC2”,”VC3”)
    AS SELECT ”C1”,”C2”,”C3” FROM ”T1”
```

＜修正後＞

```
CREATE VIEW ”VT1” (”VC1”,”VC2”,”VC3”)
    AS SELECT ”C1”,”C2”,”C33” FROM ”T1”
```

4. 無効化されたビュー表をCREATE VIEW 文で再定義する

```
CREATE VIEW "CUSTOMER30s" ("AGE","SEX","GNO","PNAME","PRICE")
  AS SELECT "CUSTOMER"."AGE",
            "CUSTOMER"."SEX",
            "CUSTOMER"."GNO",
            "STOCK"."PNAME",
            "STOCK"."PRICE"
       FROM "CUSTOMER","STOCK"
         WHERE "CUSTOMER"."AGE"=30 AND "CUSTOMER"."GNO"="STOCK"."GNO"
```

同様に，CUSTOMER40s およびCUSTOMER50s もCREATE VIEW 文でビュー表を再定義してください。

ヒント

再定義するビュー表が複数ある場合，ビューレベルの値が小さい順にCREATE VIEW 文を実行してください。

(8)　バージョンアップ時に行われるビュー表の再作成が失敗した場合

HADB サーバのバージョンアップ時にビュー表が再作成されることがあります。その際，ビュー表の再作成に失敗すると，対象のビュー表は無効化されます。詳細については，「8.6.5　バージョンアップ時の注意事項」の「(5)　バージョンアップ時に行われるビュー表の再作成」を参照してください。

再作成に失敗したビュー表の無効化を解除する手順を次に示します。

手順

1. 無効化されたビュー表の一覧と，それらのビュー表の定義情報を確認する

   ディクショナリ表を検索すると，無効化されたビュー表の一覧と，それらのビュー表の定義情報を取得できます。ディクショナリ表の検索については，「付録 B.22　ディクショナリ表の検索」の「(32)　依存するビュー表の一覧，およびそれらのビュー表の定義情報を調べる場合」を参照してください。

2. 無効化されたビュー表をDROP VIEW 文で削除する

   手順 1.で確認した，無効化されたビュー表をすべて削除してください。

3. CREATE VIEW 文を作成する

   手順 1.で確認した，無効化されたビュー表のCREATE VIEW 文を作成してください。

4. 出力されたメッセージを基に，CREATE VIEW 文の指定内容を修正する

   手順 3.で作成したCREATE VIEW 文を修正し，ビュー表の無効化要因を取り除いてください。

   バージョンアップ時に行われるビュー表の再作成に失敗した場合，ビュー表の再作成に失敗したことを知らせるKFAA51312-W メッセージが出力されます。KFAA51312-W メッセージの直前に出力されているメッセージに，ビュー表の無効化要因と対処方法が説明されています。それに従って，CREATE VIEW 文を修正してください。

5. 無効化されたビュー表をCREATE VIEW 文で再定義する

   手順 4.で修正したCREATE VIEW 文で，無効化されたビュー表を再定義してください。

ヒント

再定義するビュー表が複数ある場合，ビューレベルの値が小さい順にCREATE VIEW 文を実行してください。

11.2.9　無効化されたビュー表を削除する方法

無効化されたビュー表が不要な場合は，次に示す手順でビュー表を削除してください。

例題

HADB ユーザADBUSER01 が定義したビュー表のうち，無効化されているビュー表を削除します。

手順

1. 無効化されているビュー表の一覧を取得する

   ディクショナリ表を検索すると，無効化されているビュー表の一覧を取得できます。ディクショナリ表の検索例については，「11.2.7　ビュー表が無効化されているかどうかを確認する方法」の「例題 2」を参照してください。

2. 無効化されているビュー表をDROP VIEW 文で削除する

```
DROP VIEW ”CUSTOMER30s”
DROP VIEW ”CUSTOMER40s”
DROP VIEW ”CUSTOMER50s”
```

   無効化されているすべてのビュー表に対してDROP VIEW 文を実行します。

■削除するビュー表が多数ある場合

ビューレベルが最も小さいビュー表を削除すると，削除するビュー表に依存し，無効化されているすべてのビュー表を一括削除できます。

（例）

```
DROP VIEW ”CUSTOMER30s” CASCADE
```

上記のように削除動作にCASCADE を指定すると，CUSTOMER30s に依存するすべてのビュー表（無効化されているすべてのビュー表）を削除できます。

無効化されているビュー表のうち，ビューレベルが最も小さいビュー表を調べるには，次に示すSELECT 文を実行します。下線部分には，スキーマ名を指定します。

```
SELECT ”TABLE_NAME”,”VIEW_LEVEL” FROM ”MASTER”.”SQL_VIEWS”
    WHERE ”TABLE_SCHEMA”=’ADBUSER01’ AND IS_INVALID = ’Y’
    ORDER BY ”VIEW_LEVEL”
```

検索結果

```
TABLE_NAME  VIEW_LEVEL
----------- ----------
```

```
CUSTOMER30s 3
CUSTOMER40s 4
    :       :
```

検索結果の 1 行目に表示されているビュー表（CUSTOMER30s）が，ビューレベルが最も小さいビュー表です。

11.2.10　ビュー表の基表を確認する方法

ディクショナリ表を検索すると，ビュー表の基表を調べることができます。ディクショナリ表の検索例を次に示します。

例題

HADB ユーザADBUSER01 が定義したビュー表CUSTOMER30s の基表を調べます。

次のSELECT 文を実行して，ディクショナリ表を検索してください。

```
SELECT "MVU"."TABLE_SCHEMA","MVU"."TABLE_NAME"
  FROM  "MASTER"."SQL_VIEW_TABLE_USAGE" "MVU"
    WHERE "MVU"."VIEW_SCHEMA"='ADBUSER01'
      AND "MVU"."VIEW_NAME" = 'CUSTOMER30s'
      AND "MVU"."IS_DIRECT" ='Y'
```

検索結果

```
TABLE_SCHEMA TABLE_NAME
------------ ----------
ADBUSER01    CUSTOMER
ADBUSER01    STOCK
```

CUSTOMER30s の基表は，ADBUSER01 が所有しているCUSTOMER 表とSTOCK 表であることがわかります。

11.2.11　依存するビュー表を確認する方法

ディクショナリ表を検索すると，依存するビュー表を調べることができます。ディクショナリ表の検索例を次に示します。

例題

HADB ユーザADBUSER01 が定義したCUSTOMER 表に依存するビュー表を調べます。

次のSELECT 文を実行して，ディクショナリ表を検索してください。

```
SELECT "MV"."TABLE_SCHEMA","MV"."TABLE_NAME"
    FROM  "MASTER"."SQL_VIEW_TABLE_USAGE" "MVU"
         ,"MASTER"."SQL_VIEWS" "MV"
      WHERE "MVU"."TABLE_SCHEMA"='ADBUSER01'
        AND "MVU"."TABLE_NAME" = 'CUSTOMER'
        AND "MV"."TABLE_SCHEMA" = "MVU"."VIEW_SCHEMA"
        AND "MV"."TABLE_NAME" = "MVU"."VIEW_NAME"
```

検索結果

```
TABLE_SCHEMA TABLE_NAME
------------ ----------
ADBUSER01    CUSTOMER30s
ADBUSER01    CUSTOMER40s
ADBUSER02    CUSTOMER90s
```

検索結果には，依存するビュー表の一覧が表示されます。

11.2.12　ビュー表に対するアクセス権限を確認する方法

ディクショナリ表を検索すると，ビュー表に対するアクセス権限を確認できます。ディクショナリ表の検索例については，「付録 B.22　ディクショナリ表の検索」の「(23)　表に対するアクセス権限を調べる場合」を参照してください。

11.2.13　ビュー表を使用した運用例

次に示す運用条件を満たす場合の，ビュー表の定義方法とアクセス権限の付与方法について説明します。

■運用条件

- 顧客データが格納されている顧客表（CUSTOMERS）を定義し，顧客データを HADB ユーザが参照できるようにする
- 顧客データを参照する HADB ユーザが，顧客表の顧客名列（CUSTOMER_NAME）に格納されているデータを参照できないようにする
- 顧客データを参照する HADB ユーザには，顧客表の表名（CUSTOMERS）を公開しないようにする

■この運用例で定義する表

- 顧客表（CUSTOMERS）

  実際の顧客データが格納されている実表です。
- 顧客表のビュー表（V1_CUSTOMERS）

  V1_CUSTOMERS は，CUSTOMERS を基表としたビュー表です。顧客表から顧客名列（CUSTOMER_NAME）を除いた表をビュー表として定義します。
- 顧客表のビュー表を基表としたビュー表（V2_CUSTOMERS）

  V2_CUSTOMERS は，V1_CUSTOMERS を基表としたビュー表です。

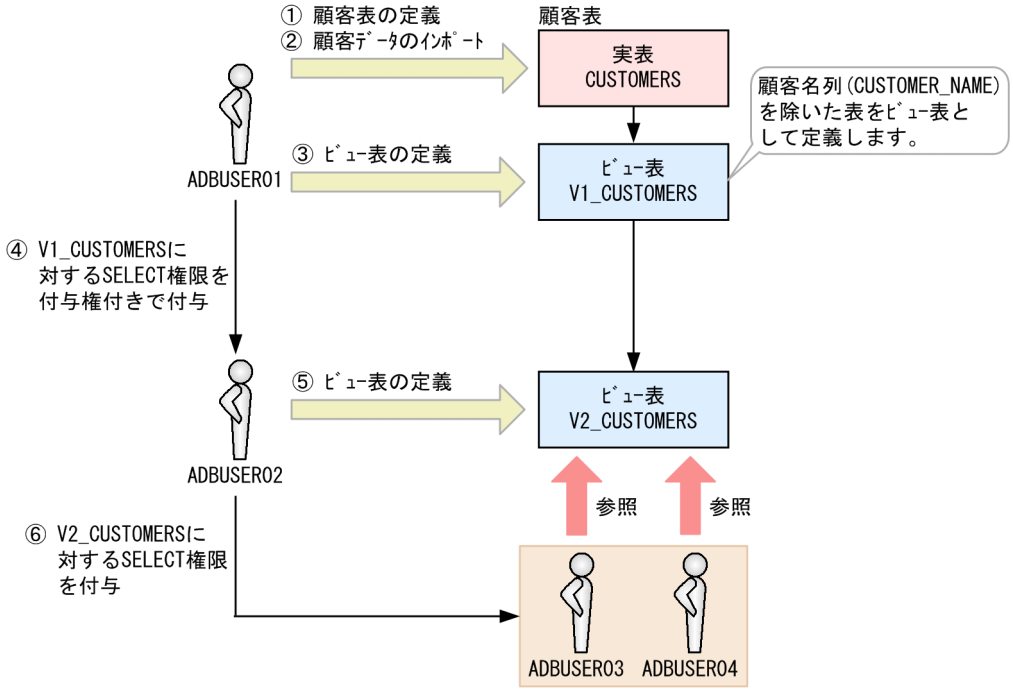
  顧客データを参照する HADB ユーザには，V2_CUSTOMERS にアクセスさせます。

■この運用例で登場する HADB ユーザ

- HADB ユーザADBUSER01：顧客表CUSTOMERS，ビュー表V1_CUSTOMERS の所有者
- HADB ユーザADBUSER02：ビュー表V2_CUSTOMERS の所有者

- HADB ユーザADBUSER03：顧客データを参照する HADB ユーザ
- HADB ユーザADBUSER04：顧客データを参照する HADB ユーザ

■運用の関連図



注　図中の丸付き数字は，次の手順の説明の番号と対応しています。

上記の運用条件を実現するには，次の手順で顧客表とビュー表を定義し，各表に対するアクセス権限を各 HADB ユーザに付与します。

手順

1. ADBUSER01 が顧客表を定義する

```
CREATE TABLE ”ADBUSER01”.”CUSTOMERS”(”CUSTOMER_ID” INTEGER,
                                     ”CUSTOMER_NAME” CHAR(100),
                                     ”REGISTERED_DATE” TIMESTAMP)
             IN ”DBAREA01”
```

2. ADBUSER01 が，顧客表に顧客データをインポートする

   adbimport コマンドを実行して，顧客表に顧客データをインポートしてください。

3. ADBUSER01 が，顧客表のビュー表V1_CUSTOMERS を定義する

```
CREATE VIEW ”ADBUSER01”.”V1_CUSTOMERS”
    AS SELECT ”CUSTOMER_ID”,”REGISTERED_DATE” FROM ”ADBUSER01”.”CUSTOMERS”
```

   顧客表からCUSTOMER_NAME 列を除いた表が，ビュー表V1_CUSTOMERS として定義されます。

4. ADBUSER01 が，ビュー表V1_CUSTOMERS に対するSELECT 権限を付与権付きでADBUSER02 に付与する

```
GRANT SELECT ON ”ADBUSER01”.”V1_CUSTOMERS” TO ”ADBUSER02” WITH GRANT OPTION
```

ビュー表V1_CUSTOMERS に対するSELECT 権限が付与権付きでADBUSER02 に付与されます。

メモ

- ビュー表V1_CUSTOMERS に対するSELECT 権限の付与権をADBUSER02 に付与するために，WITH GRANT OPTION を指定しています。
- 手順の 5.で，ADBUSER02 が，ビュー表V1_CUSTOMERS を基表としたビュー表V2_CUSTOMERS を定義するために，ビュー表V1_CUSTOMERS に対するSELECT 権限をADBUSER02 に付与しています。
- この例では，SELECT 権限だけを付与していますが，必要に応じてほかの種類のアクセス権限も一緒に付与できます。

5. ADBUSER02 が，ビュー表V2_CUSTOMERS を定義する

```
CREATE VIEW ”ADBUSER02”.”V2_CUSTOMERS”
    AS SELECT * FROM ”ADBUSER01”.”V1_CUSTOMERS”
```

ビュー表V1_CUSTOMERS を基表としたビュー表V2_CUSTOMERS が定義されます。

メモ

- CREATE VIEW 文中の問合せ式中のSELECT に*が指定されていますが，CUSTOMER_NAME 列を除いたV1_CUSTOMERS が基表のため，定義されたV2_CUSTOMERS もCUSTOMER_NAME 列を除いたビュー表として定義されます。
- ビュー表V2_CUSTOMERS の基表にビュー表V1_CUSTOMERS を指定することによって，実際に顧客データが格納されている顧客表の表名（CUSTOMERS）を隠すことができます。

6. ADBUSER02 が，ビュー表V2_CUSTOMERS に対するSELECT 権限をADBUSER03 とADBUSER04 に付与する

```
GRANT SELECT ON ”ADBUSER02”.”V2_CUSTOMERS” TO ”ADBUSER03”,”ADBUSER04”
```

ADBUSER03 とADBUSER04 は，ビュー表V2_CUSTOMERS に対するSELECT 権限を付与されたため，顧客データを参照できるようになります（ビュー表V2_CUSTOMERS を参照できるようになります）。

メモ

ADBUSER03 とADBUSER04 は，顧客データを参照するだけの HADB ユーザのため，ビュー表V2_CUSTOMERS に対するSELECT 権限の付与権は付与しません。そのため，上記のGRANT 文では，WITH GRANT OPTION を指定していません。

上記の運用を継続中に，顧客データを参照する HADB ユーザADBUSER05 が追加された場合，ADBUSER02 が，ビュー表V2_CUSTOMERS に対するSELECT 権限をADBUSER05 に付与してください。そうすると，ADBUSER05 は，顧客データを参照できるようになります。

メモ

- この例の場合，顧客表中の顧客データを更新，追加，削除できるのは，ADBUSER01 だけになります。
- 顧客データを参照する HADB ユーザの管理（ビュー表V2_CUSTOMERS に対するアクセス権限の管理）は，ADBUSER02 が実施します。
- ADBUSER02～ADBUSER05 は，顧客データを参照だけできます。また，CUSTOMER_NAME 列を参照することはできません。

11.3　インデクスの運用

ここでは，インデクスの運用について説明します。

11.3.1　インデクスの定義

実表にインデクス（B-tree インデクス，テキストインデクスまたはレンジインデクス）を定義するには，CREATE INDEX 文を実行します。CREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義)』を参照してください。

自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有している実表に対してだけ，インデクスを定義できます。ほかの HADB ユーザが所有している実表に対しては，インデクスを定義できません。

また，カラムストア表にはテキストインデクスを定義できません。

なお，インデクスを定義する際は，次の個所を参照して，適切な列にインデクスを定義するようにしてください。

- 「5.3　B-tree インデクスの設計」
- 「5.4　テキストインデクスの設計」
- 「5.5　レンジインデクスの設計」

重要

行を格納するセグメントが割り当てられている状態の表に対してインデクスを定義すると，インデクスが未完状態（インデクスデータが作成されない状態）になります。インデクスが未完状態の場合，未完状態のインデクスを使った検索や，データの追加，更新，削除などができなくなります。

インデクスの未完状態の解除方法については，次の個所を参照してください。

- 「15.9.1　B-tree インデクスが未完状態になったときの対処方法」
- 「15.10.1　テキストインデクスが未完状態になったときの対処方法」
- 「15.11.1　レンジインデクスが未完状態になったときの対処方法」

行を格納するセグメントが割り当てられている状態については，「5.3.1　B-tree インデクスを定義する場合の注意点（B-tree インデクスの未完状態)」を参照してください。

11.3.2　B-tree インデクスの再作成

B-tree インデクスを定義した表に対して，行の挿入，行の更新，および行の削除を頻繁に実施した場合は，B-tree インデクスを再作成することを検討してください。

B-tree インデクスを再作成すると，次に示す効果があります。

- B-tree インデクスを使用した検索性能の低下を防ぐことができる
- データ用 DB エリアの使用量を削減できる（行の更新や行の削除で発生した無効な状態の B-tree インデクスのデータを削除できる）
- インデクスページスプリットの発生を抑えることができる（B-tree インデクスのインデクスページ内の未使用領域を再度確保できる）

B-tree インデクスを再作成する場合は，adbidxrebuild コマンドを実行してください。adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

> **メモ**
>
> - 未完状態になった B-tree インデクスを再作成したい場合は，「15.9.1　B-tree インデクスが未完状態になったときの対処方法」を参照してください。
> - 一意性制約に違反したユニークインデクスを再作成した場合は，「15.9.2　一意性制約に違反した場合の対処方法（KFAA61205-W メッセージが出力された場合)」を参照してください。

B-tree インデクスを再作成するかどうかを判断する方法について，次に示します。

(1)　B-tree インデクスを使用した検索性能の低下を防ぐ場合

B-tree インデクスを定義した表に対して，行の挿入，行の更新，または行の削除を頻繁に実施した場合は，B-tree インデクスを使用した検索性能が低下します。

その場合は，次に示す出力項目のうち，どちらか片方を確認してください。確認した結果，出力項目の値が大きい場合は，B-tree インデクスを再作成してください。B-tree インデクスを再作成すると，B-tree インデクスを使用した検索性能の低下を防ぐことができます。

- SQL 文の統計情報に出力されるBidx_validation_check_cnt の値（B-tree インデクスを使用した検索で，行データが有効かどうかを判定した回数）

  SQL 文の統計情報は，統計ログファイルおよび SQL トレースファイルに出力されます。

  統計ログファイルに出力された SQL 文の統計情報を確認する場合は，adbstat コマンドを実行してください。adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat (HADB サーバの統計解析)』を参照してください。

SQL トレースファイルに出力された SQL 文の統計情報を確認する場合は，SQL トレースファイル中の SQL トレース情報を確認してください。SQL トレース情報については，「10.11.2　SQL トレース情報に出力される情報」の「(10)　SQL 文の統計情報」を参照してください。

- アクセスパスの統計情報のデータアクセス情報に出力されるData_bidx_validation_check_cnt の値（B-tree インデクスを使用した検索で，行が有効かどうかを判定した回数）

  アクセスパスの統計情報は，SQL トレースファイルに出力されます。SQL トレースファイル中の SQL トレース情報を確認してください。アクセスパスの統計情報については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」を参照してください。

SQL 文の統計情報とアクセスパスの統計情報の両方を確認できる場合は，アクセスパスの統計情報を確認することを推奨します。アクセスパスの統計情報を確認すると，複数の表を検索している場合に，どの表に定義した B-tree インデクスを再作成すればよいかを確認できます。

(2)　データ用 DB エリアの使用量を削減する場合

B-tree インデクスを定義した表に対して，行の更新，または行の削除を頻繁に実施した場合は，次に示す手順に従って，B-tree インデクスを再作成するかどうかを判断してください。

手順

1. adbdbstatus コマンドで B-tree インデクスの使用量を確認する

   adbdbstatus コマンドで，インデクスのサマリ情報を出力してください。そして，Used_segments（インデクスの使用中セグメント数）の出力内容を確認してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

2. 実表内の有効なデータを基に見積もった B-tree インデクスの容量を確認する

   B-tree インデクスの容量を求める場合は，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある次の個所を参照してください。なお，表の種類によって参照先が異なります。

   - シングルチャンク表の場合

     「(c)　変数 SGIDX の求め方（シングルチャンク表の場合）」

   - マルチチャンク表の場合

     「(h)　変数 SGIDX の求め方（マルチチャンク表の場合）」

   なお，変数 *SGIDX* は，データ用 DB エリアに格納されているすべての B-tree インデクスの合計を求める計算式となっています。ここでは，対象の B-tree インデクスだけを求めてください。

手順 1.および手順 2.で確認した結果を比較して，その差が大きい場合は，B-tree インデクスを再作成してください。

(3)　B-tree インデクスのインデクスページスプリットの発生を抑える場合

B-tree インデクスを定義した表に対して，行の挿入，または行の更新を頻繁に実施した場合は，インデクスページスプリットが発生するおそれがあります。

インデクスページスプリットの発生有無については，SQL 文の統計情報のBidx_page_split_cnt（B-tree インデクスのインデクスページスプリットが発生した回数）を確認してください。確認した結果，インデクスページスプリットが発生している場合は，B-tree インデクスを再作成してください。B-tree インデクスを再作成すると，インデクスページ内の未使用領域を再度確保できるため，インデクスページスプリットの発生を抑えることができます。

SQL 文の統計情報は，統計ログファイルおよび SQL トレースファイルに出力されます。

統計ログファイルに出力された SQL 文の統計情報を確認する場合は，adbstat コマンドを実行してください。adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』を参照してください。

SQL トレースファイルに出力された SQL 文の統計情報を確認する場合は，SQL トレースファイル中の SQL トレース情報を確認してください。SQL トレース情報については，「10.11.2　SQL トレース情報に出力される情報」の「(10)　SQL 文の統計情報」を参照してください。

11.3.3　テキストインデクスの再作成

テキストインデクスを定義した表に対して，行の挿入，行の更新，および行の削除を頻繁に実施した場合は，テキストインデクスを再作成することを検討してください。

テキストインデクスを再作成すると，次に示す効果があります。

- テキストインデクスを使用した検索性能の低下を防ぐことができる
- データ用 DB エリアの使用量を削減できる（行の更新や行の削除で発生した無効な状態のテキストインデクスのデータを削除できる）
- インデクスページスプリットの発生を抑えることができる（テキストインデクスのインデクスページ内の未使用領域を再度確保できる）

テキストインデクスを再作成する場合は，adbidxrebuild コマンドを実行してください。adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

> **メモ**
>
> 未完状態になったテキストインデクスを再作成したい場合は，「15.10.1　テキストインデクスが未完状態になったときの対処方法」を参照してください。

テキストインデクスを再作成するかどうかを判断する方法について，次に示します。

(1)　テキストインデクスを使用した検索性能の低下を防ぐ場合

テキストインデクスを定義した表に対して，行の挿入，行の更新，または行の削除を頻繁に実施した場合は，テキストインデクスを使用した検索性能が低下します。

テキストインデクスを定義した表に格納されているデータのうち，0.01%以上のデータを更新（行の挿入および行の更新）した場合は，テキストインデクスを再作成することを検討してください。

また，テキストインデクスを定義した表に格納されているデータのうち，30%以上のデータを削除（行の削除）した場合も，テキストインデクスを再作成することを検討してください。

テキストインデクスを再作成すると，テキストインデクスを使用した検索性能の低下を防ぐことができます。

(2)　データ用 DB エリアの使用量を削減する場合

テキストインデクスを定義した表に対して，行の更新，または行の削除を頻繁に実施した場合は，次に示す手順に従って，テキストインデクスを再作成するかどうかを判断してください。

手順

1. adbdbstatus コマンドでテキストインデクスの使用量を確認する

   adbdbstatus コマンドで，インデクスのサマリ情報を出力してください。そして，Used_segments（インデクスの使用中セグメント数）の出力内容を確認してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

2. 実表内の有効なデータを基に見積もったテキストインデクスの容量を確認する

   テキストインデクスの容量を求める場合は，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」にある次の個所を参照してください。なお，表の種類によって参照先が異なります。

   - シングルチャンク表の場合

     「(e)　変数 SGTIX の求め方（シングルチャンク表の場合）」

   - マルチチャンク表の場合

     「(j)　変数 SGTIX の求め方（マルチチャンク表の場合）」

   なお，変数 *SGTIX* は，データ用 DB エリアに格納されているすべてのテキストインデクスの合計を求める計算式となっています。ここでは，対象のテキストインデクスだけを求めてください。

手順 1.および手順 2.で確認した結果を比較して，その差が大きい場合は，テキストインデクスを再作成してください。

(3)　テキストインデクスのインデクスページスプリットの発生を抑える場合

テキストインデクスを定義した表に対して，行の挿入，または行の更新を頻繁に実施した場合は，インデクスページスプリットが発生するおそれがあります。

インデクスページスプリットの発生有無については，SQL 文の統計情報のTidx_page_split_cnt（テキストインデクスのインデクスページスプリットが発生した回数）を確認してください。確認した結果，インデクスページスプリットが発生している場合は，テキストインデクスを再作成してください。テキストインデクスを再作成すると，インデクスページ内の未使用領域を再度確保できるため，インデクスページスプリットの発生を抑えることができます。

SQL 文の統計情報は，統計ログファイルおよび SQL トレースファイルに出力されます。

統計ログファイルに出力された SQL 文の統計情報を確認する場合は，adbstat コマンドを実行してください。adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』を参照してください。

SQL トレースファイルに出力された SQL 文の統計情報を確認する場合は，SQL トレースファイル中の SQL トレース情報を確認してください。SQL トレース情報については，「10.11.2　SQL トレース情報に出力される情報」の「(10)　SQL 文の統計情報」を参照してください。

11.3.4　テキストインデクスの確認（指定したインデクスオプションの確認）

テキストインデクスの定義時に指定したインデクスオプションを確認する場合は，「付録 B.22　ディクショナリ表の検索」の「(34)　テキストインデクスに指定したインデクスオプションを調べる場合」を参照してください。

指定したインデクスオプションによって，表記ゆれ補正検索やワード検索を行う場合に，テキストインデクスが使用されるかどうかが決まります。

11.3.5　レンジインデクスの再作成

レンジインデクスを定義した表に対して，行の挿入，行の更新，および行の削除を頻繁に実施した場合は，レンジインデクスを再作成することを検討してください。

レンジインデクスを再作成すると，次に示す効果があります。

- レンジインデクスを使用した検索性能の低下を防ぐことができる
- データ用 DB エリアの使用量を削減できる（行の更新や行の削除で発生した無効な状態のレンジインデクスのデータを削除できる）

レンジインデクスを再作成する場合は，adbidxrebuild コマンドを実行してください。adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

メモ

未完状態になったレンジインデクスを再作成したい場合は，「15.11.1　レンジインデクスが未完状態になったときの対処方法」を参照してください。

レンジインデクスを再作成するかどうかを判断する方法について，次に示します。

(1)　レンジインデクスを使用した検索性能の低下を防ぐ場合

レンジインデクスを定義した表に対して，行の挿入，行の更新，または行の削除を頻繁に実施した場合は，レンジインデクスを使用した検索性能が低下します。

レンジインデクスを使用した検索処理の状況を確認する場合は，「13.1.2　定義したレンジインデクスの見直し」を参照してください。SQL 文の統計情報またはアクセスパスの統計情報のどちらかを確認して，レンジインデクスの使用状況を確認してください。確認した結果，レンジインデクスの効果がない場合（計算結果が 0 である，または 0 に近い）は，レンジインデクスを再作成してください。

(2)　データ用 DB エリアの使用量を削減する場合

レンジインデクスを定義した表に対して，行の更新，または行の削除を頻繁に実施した場合は，次に示す手順に従って，レンジインデクスを再作成するかどうかを判断してください。

手順

1. adbdbstatus コマンドでレンジインデクスの使用量を確認する

   adbdbstatus コマンドで，インデクスのサマリ情報を出力してください。そして，Used_segments（インデクスの使用中セグメント数）の出力内容を確認してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

2. 実表内の有効なデータを基に見積もったレンジインデクスの容量を確認する

   レンジインデクスの容量を求める場合は，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2) 変数の説明」にある次の個所を参照してください。なお，表の種類によって参照先が異なります。

   - シングルチャンク表の場合

     「(d)　変数 SGRIX の求め方（シングルチャンク表の場合）」

   - マルチチャンク表の場合

     「(i)　変数 SGRIX の求め方（マルチチャンク表の場合）」

   なお，変数 *SGRIX* は，データ用 DB エリアに格納されているすべてのレンジインデクスの合計を求める計算式となっています。ここでは，対象のレンジインデクスだけを求めてください。

手順 1.および手順 2.で確認した結果を比較して，その差が大きい場合は，レンジインデクスを再作成してください。

11.3.6　レンジインデクスの確認（チャンクのスキップの可否）

ここでは，定義したレンジインデクスが，チャンクのスキップができるレンジインデクスかどうかを確認する方法について説明します。

> **メモ**
>
> バージョン 02-02 より前のバージョンで定義されたレンジインデクスは，チャンクのスキップができません。

ディクショナリ表SQL_INDEXES のIS_CHUNK_SKIP 列を検索することで，チャンクのスキップができるレンジインデクスかどうかを確認できます。SELECT 文の指定例を次に示します。

■SELECT 文の指定例

```
SELECT "TABLE_SCHEMA","TABLE_NAME","INDEX_NAME","IS_CHUNK_SKIP"
  FROM "MASTER"."SQL_INDEXES" WHERE "INDEX_TYPE" = 'R'
```

検索結果のIS_CHUNK_SKIP 列がY の場合は，チャンクのスキップができるレンジインデクスです。
IS_CHUNK_SKIP 列がナル値の場合は，チャンクのスキップができないレンジインデクスです。

チャンクのスキップができないレンジインデクスを，チャンクのスキップができるレンジインデクスに変更したい場合は，「11.3.7　レンジインデクスの再定義（チャンクのスキップの設定）」を参照してください。

ディクショナリ表SQL_INDEXES については，「付録 B.5　SQL_INDEXES の内容」を参照してください。

> **メモ**
>
> 対象のレンジインデクスのスキーマ名，およびインデクス識別子がわかっている場合は，「付録 B.22　ディクショナリ表の検索」の「(19)　チャンクのスキップができるレンジインデクスかどうかを調べる場合」を参照して検索することもできます。

11.3.7　レンジインデクスの再定義（チャンクのスキップの設定）

定義したレンジインデクスが，チャンクのスキップができないレンジインデクスの場合で，チャンクのスキップができるレンジインデクスに変更したいときは，レンジインデクスを再定義したあとで，adbidxrebuild コマンドを実行してください。手順を次に示します。

1. レンジインデクスを削除する
   チャンクのスキップができないレンジインデクスを，DROP INDEX 文で削除します。「11.3.9　インデクスの削除」を参照してください。
2. レンジインデクスを再定義する

手順 1.で削除したレンジインデクスが定義されていた表に対して，CREATE INDEX 文でレンジインデクスを再定義してください。

CREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義)』を参照してください。

3. レンジインデクスを再作成する

手順 2.でレンジインデクスを再定義した表に対して，adbidxrebuild コマンドでレンジインデクスを再作成してください。

adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

レンジインデクスを再定義したあとで，adbidxrebuild コマンドを実行すると，チャンクのスキップができるレンジインデクスになります。

11.3.8　インデクスが格納されているデータ用 DB エリアを変更する方法

インデクスが格納されているデータ用 DB エリアを別のデータ用 DB エリアに変更したい場合は，対象のインデクスをいったん削除したあとで再定義する必要があります。そのあとで，インデクスを再作成する必要があります。

> **メモ**
>
> 実表が格納されているデータ用 DB エリアを別のデータ用 DB エリアに変更したい場合は，「11.1.16　実表が格納されているデータ用 DB エリアを変更する方法」を参照してください。

次の手順に従って，インデクスが格納されているデータ用 DB エリアを，別のデータ用 DB エリアに変更してください。

なお，新たなデータ用 DB エリアを追加する必要がある場合は，「11.10.1　データ用 DB エリアを追加する方法」を参照してください。そのあとで，インデクスが格納されているデータ用 DB エリアを変更してください。

手順

1. インデクスの定義情報を確認する

実表にインデクスを定義したときのCREATE INDEX 文の指定内容を確認してください。実表にインデクスを再定義する際に使用します。

インデクスを定義したときのCREATE INDEX 文の指定内容がわからない場合は，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。ディクショナリ表を検索すると，インデクスを定義したときのCREATE INDEX 文の指定内容が確認できます。

2. インデクスを削除する

DROP INDEX 文で，インデクスを削除してください。

3. インデクスを再定義する

実表にCREATE INDEX 文でインデクスを再定義してください。CREATE INDEX 文を作成する際の留意事項を次に示します。

- 手順 1.で確認したインデクスを定義したときのCREATE INDEX 文の指定内容を使用して，CREATE INDEX 文を作成してください。
- CREATE INDEX 文のIN *DB エリア名*に，インデクスを格納する別のデータ用 DB エリアの DB エリア名を指定してください。

4. インデクスを再作成する

adbidxrebuild コマンドで，手順 3.で定義したインデクスを再作成してください。

これで，インデクスが格納されているデータ用 DB エリアを，別のデータ用 DB エリアに変更できます。

メモ

- adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。
- DROP INDEX 文およびCREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』を参照してください。

11.3.9　インデクスの削除

表に定義しているインデクス（B-tree インデクス，テキストインデクスまたはレンジインデクス）を削除するには，DROP INDEX 文を実行します。

DROP INDEX 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『DROP INDEX（インデクスの削除)』を参照してください。

自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が所有しているインデクスだけを削除できます。ほかの HADB ユーザが所有しているインデクスは削除できません。

また，次に示すインデクスは，DROP INDEX 文で削除できません。削除したい場合は，DROP TABLE 文で実表ごと削除してください。

- 主キーを構成する列に自動的に定義された B-tree インデクス
- アーカイブマルチチャンク表のアーカイブレンジ列に自動的に定義されたレンジインデクス

重要

インデクスを削除する前に，削除するインデクス（B-tree インデクス，またはテキストインデクス）が定義されている実表に対して，コマンドの再実行が必要かどうか（実表が更新不可状

態かどうか）を確認してください。確認する方法については，「10.9.2　実表の状態や使用量を確認したい場合」の「(1)　実表が更新不可状態かどうかを確認する場合」を参照してください。

コマンドの再実行が必要な場合

コマンドを再実行してください。そのあとで，インデクスを削除してください。

コマンドの再実行が必要な状態のままで（実表が更新不可状態のままで）インデクスを削除した場合

中断したコマンドによって作成された作業用一時ファイルが残ることがあります。作業用一時ファイルを削除するには，すべての実表の更新不可状態を解除する必要があります。作業用一時ファイルの削除については，「15.2.6　コマンド実行時の作業用一時ファイルを格納するディスクの空き容量が不足した場合の対処方法」の「(2)　ディスクに不要な作業用一時ファイルが残っている場合」を参照してください。

11.4　マルチチャンク表の運用

ここでは，マルチチャンク表特有の運用について説明します。マルチチャンク表とは，次に示す表の総称です。

- レギュラーマルチチャンク表
- アーカイブマルチチャンク表

なお，実表の運用については，「11.1　実表の運用」を参照してください。

11.4.1　マルチチャンク表を定義する方法

マルチチャンク表を定義するには，チャンク指定を指定したCREATE TABLE 文を実行します。

なお，レギュラーマルチチャンク表とアーカイブマルチチャンク表とで，チャンク指定の指定内容が異なります。

レギュラーマルチチャンク表を定義する場合

「*チャンク指定*」のうち，「CHUNK=*チャンク数の最大値*」だけを指定してください。

アーカイブマルチチャンク表を定義する場合

次の指定をしてください。

- 「CHUNK=*チャンク数の最大値*」
- 「*チャンクアーカイブ指定*」

マルチチャンク表の定義例については，マニュアル『HADB　SQL リファレンス』の『CREATE TABLE 文の指定形式および規則』の『例題』を参照してください。

アーカイブマルチチャンク表を定義する場合，アーカイブディレクトリを作成しておく必要があります。アーカイブディレクトリに関する注意事項を次に示します。

■アーカイブディレクトリに関する注意事項

- 使用中のアーカイブディレクトリは削除しないでください。使用中のアーカイブディレクトリを削除した場合，データベースの整合性が失われるため，HADB サーバは動作を保証しません。
- アーカイブディレクトリ下に格納されているファイルを，更新または削除しないでください。ファイルを更新または削除した場合，HADB サーバは動作を保証しません。
- アーカイブディレクトリ下にディレクトリを作成したり，またはファイルを格納したりしないでください。アーカイブマルチチャンク表に対する操作を実行した際に，ディレクトリやファイルが削除されてしまうおそれがあります。

11.4.2　マルチチャンク表にデータを格納する方法（バックグラウンドインポート）

バックグラウンドインポートでマルチチャンク表にデータを格納する方法について説明します。

> **メモ**
>
> マルチチャンク表にデータを格納する方法として，バックグラウンドインポートではなく，追加モードによるデータインポートもあります。追加モードによるデータインポートについては，「11.1.8　実表にデータを格納する方法（データインポート）」を参照してください。

バックグラウンドインポートを実行する場合，adbimport コマンドに-b オプションを指定してください。adbimport コマンドに-b オプションを指定して，バックグラウンドインポートを実行する例を次に示します。

adbimport コマンドの指定形式については，マニュアル『HADB　コマンドリファレンス』の『adbimport(データのインポート)』を参照してください。

■バックグラウンドインポートの実行例

HADB ユーザ（ADBUSER01）が，店舗表（SHOPSLIST）に対して，バックグラウンドインポートを実行します。

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file.txt
          -b -m '2014/01/01-2014/01/31'
          SHOPSLIST
          /home/adbmanager/imp_file/imp_data_path.txt
```

> **メモ**
>
> **■チャンクにコメントを設定する方法**
>
> マルチチャンク表に対するデータインポート時に，データが格納されるチャンクに対して，コメントを設定できます。コメントは，-m オプションで設定できます。
>
> 特定のチャンクを対象にした運用をするような場合，コメントを設定しておくと，対象とするチャンクを特定しやすくなります。

> **重要**
>
> バックグラウンドインポートを実行する場合，インポートオプションadb_import_rthd_num には，次の計算式を満たす値を指定してください。次の計算式を満たす値をインポートオプションadb_import_rthd_num に指定すると，バックグラウンドインポートで追加するデータの格納効率が良くなります。
>
> **計算式**

```
インポートオプションadb_import_rthd_numの指定値 ≦
      ↑ MIN ( C × db_org  , db_org × index_num × 0.25 ) ↑ +1
```

変数の説明

C

バックグラウンドインポートの対象表が，ローストア表の場合は 2 を代入してください。カラムストア表の場合は 0.5 を代入してください。

db_org

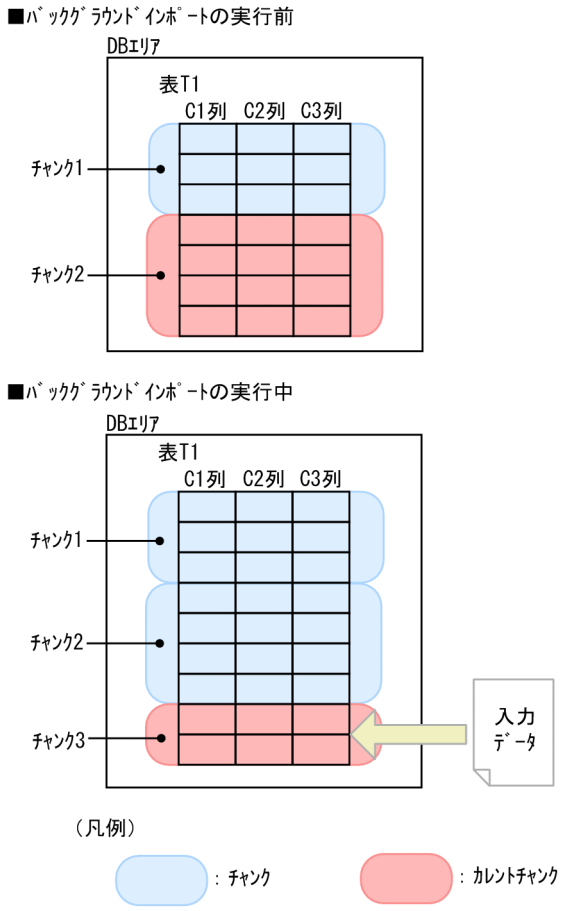
バックグラウンドインポート時に使用する入力データファイルのファイルサイズ（単位：メガバイト）

index_num

バックグラウンドインポートの対象表に定義されている B-tree インデクスとテキストインデクスの数

バックグラウンドインポートを実行すると，HADB はチャンクを新規に作成し，そのチャンクにデータを格納します。バックグラウンドインポートによるデータの格納先を次の図に示します。

図 11-4　バックグラウンドインポートによるデータの格納先



[説明]

バックグラウンドインポート機能を適用したadbimport コマンドを実行すると，新規に作成されたチャンク 3 にデータが格納されます。その際，カレントチャンクがチャンク 2 からチャンク 3 に変更されます。

なお，チャンク 3 にデータを格納しているときに，チャンク 1 とチャンク 2 に格納されているデータを検索することができます。

重要

バックグラウンドインポートを実行するたびに 1 つのチャンクが新規に作成されるため，バックグラウンドインポートを繰り返すと，その分だけチャンク数が増加します。チャンク数が増加すると，B-tree インデクスやテキストインデクスを使用した検索時に性能が低下するおそれがあります。そのため，チャンクをマージして，チャンク数を減らすことを検討してください。チャンクをマージする場合は，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」を参照してください。

11.4.3　アーカイブマルチチャンク表のデータを検索・更新するときの注意事項

アーカイブマルチチャンク表のデータを検索・更新するときの注意事項を説明します。

(1)　SELECT 文による行の検索

アーカイブマルチチャンク表の検索処理を実行する際，HADB サーバは最初に各チャンクのアーカイブレンジ列の範囲とSELECT 文の探索条件に従って，アクセス対象となるチャンクを絞り込みます。また，アーカイブ状態のチャンク内のデータを検索する場合，インデクスを使用した検索ができません。

そのため，アーカイブマルチチャンク表を検索する場合は，SELECT 文の探索条件にアーカイブレンジ列を指定した条件を指定して，アクセス対象となるチャンクを絞り込むようにしてください。詳細については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『アーカイブマルチチャンク表を検索する際の考慮点』を参照してください。

なお，アーカイブ状態のチャンク内のデータに対して，検索する頻度が一時的に増える場合は，adbunarchivechunk コマンドでチャンクのアーカイブ状態を解除することを検討してください。

(2)　INSERT 文による行の挿入

INSERT 文を実行してアーカイブマルチチャンク表に行を挿入する際は特別な留意事項はありません。挿入した行データは，カレントチャンクに格納されます。

(3) UPDATE 文による行の更新

アーカイブ状態ではないチャンク内のデータはUPDATE 文で更新できますが，UPDATE 文の探索条件の指定に特別な規則があります。詳細については，マニュアル『HADB　SQL リファレンス』の『UPDATE 文の指定形式および規則』の『規則』を参照してください。

また，アーカイブ状態のチャンク内のデータはUPDATE 文で更新できません。アーカイブ状態のチャンク内のデータをUPDATE 文で更新したい場合は，adbunarchivechunk コマンドでチャンクのアーカイブ状態を解除してください。

(4) DELETE 文による行の削除

アーカイブ状態ではないチャンク内のデータはDELETE 文で削除できますが，DELETE 文の探索条件の指定に特別な規則があります。詳細については，マニュアル『HADB　SQL リファレンス』の『DELETE 文の指定形式および規則』の『規則』を参照してください。

また，アーカイブ状態のチャンク内のデータはDELETE 文で削除できません。アーカイブ状態のチャンク内のデータをDELETE 文で削除したい場合は，adbunarchivechunk コマンドでチャンクのアーカイブ状態を解除してください。

11.4.4 マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成）

バックグラウンドインポートで格納するデータを検索対象外としておき，任意のタイミングで検索対象に変更したい場合は，adbimport コマンドに-b オプション，および--status wait オプションを指定してください。

待機状態のチャンクが作成されて，データが格納されます。その際，カレントチャンクは変更されません。また，システム表STATUS_CHUNKS にあるチャンクの作成日時，およびカレントチャンクが切り替わった日時は設定されません。

重要

- 待機状態のチャンクは，操作系 SQL で操作できません。待機状態のチャンク内にあるデータに対して，SELECT 文やUPDATE 文などを実行できません。ただし，PURGE CHUNK 文，およびTRUNCATE TABLE 文で待機状態のチャンクを削除することはできます。待機状態のチャンクを，操作系 SQL で操作したい場合は，通常状態のチャンクに変更してください。
- 待機状態のチャンクを通常状態のチャンク（操作系 SQL の対象となるチャンク）に変更したい場合は，adbchgchunkstatus コマンドを実行してください。詳細については，「11.4.12 チャンクの状態を変更する方法」を参照してください。
- adbimport コマンドで作成した待機状態のチャンクとadbchgchunkstatus コマンドを組み合わせて利用すると，任意のタイミングでデータを検索対象にしたり，検索対象外にしたりす

る運用を行うこともできます。詳細については，「11.4.24　バックグラウンドインポートとチャンクを意識した運用例 2（待機状態のチャンクを利用した運用）」を参照してください。

メモ

システム表STATUS_CHUNKS の詳細については，「付録 C.5　STATUS_CHUNKS の内容」を参照してください。

adbimport コマンドに-b オプション，および--status wait オプションを指定して，バックグラウンドインポートを実行する例を次に示します。

adbimport コマンドの指定形式については，マニュアル『HADB　コマンドリファレンス』の『adbimport (データのインポート)』を参照してください。

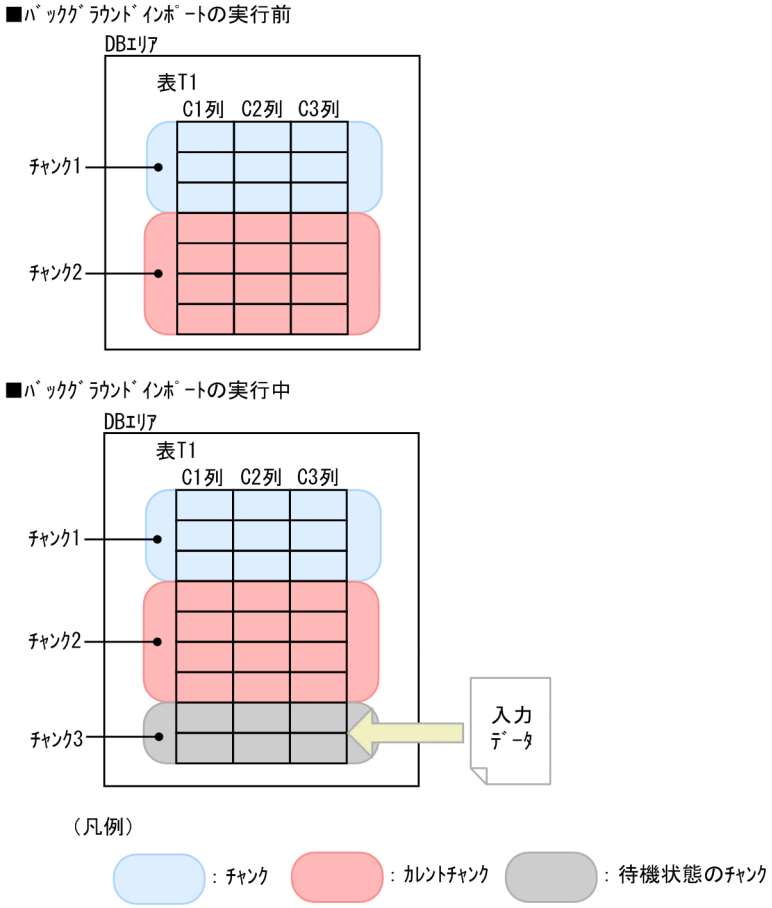
■待機状態のチャンクを作成するバックグラウンドインポートの実行例

HADB ユーザ（ADBUSER01）が，店舗表（SHOPSLIST）に対して，検索対象外のデータをバックグラウンドインポートで格納します。

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file.txt
          -b --status wait
          -m '2014/06/01-2014/06/30'
          SHOPSLIST
          /home/adbmanager/imp_file/imp_data_path.txt
```

待機状態のチャンクを作成するバックグラウンドインポートを実行すると，HADB は待機状態のチャンクを新規に作成し，そのチャンクにデータを格納します。待機状態のチャンクを作成するバックグラウンドインポートを次の図に示します。

図 11-5　待機状態のチャンクを作成するバックグラウンドインポート



[説明]

バックグラウンドインポート機能を適用したadbimport コマンドを実行して待機状態のチャンクを作成すると，新規に作成された待機状態のチャンク（チャンク 3）にデータが格納されます。その際，カレントチャンクはチャンク 2 のままです。

なお，チャンク 3 にデータを格納しているときに，チャンク 1 とチャンク 2 に格納されているデータを検索できます。また，チャンク 3 は待機状態のチャンクであるため，操作系 SQL で操作できません。

重要

-b オプションおよび--status wait オプションを指定して，バックグラウンドインポートを実行するたびに，待機状態のチャンクが 1 つ新規に作成されます。そのため，バックグラウンドインポートを繰り返すと，その分だけ待機状態のチャンク数が増加します。待機状態のチャンク数が増加すると，adbidxrebuild コマンドの実行に時間が掛かるおそれがあります。インデクスの再作成処理では，通常状態のチャンクだけでなく，待機状態のチャンクも対象となるためです。待機状態のチャンクが大量にある場合は，チャンクをマージして，待機状態のチャンク数を減らすことを検討してください。待機状態のチャンクをマージする場合は，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」を参照してください。

11.4.5　チャンク単位にデータをエクスポートする方法

マルチチャンク表に格納されているデータをチャンク単位でファイルに出力する場合，次に示す 2 つの方法があります。どちらの方法も，出力したいチャンクのチャンク ID を指定する必要があります。

- **-c オプションにチャンク ID を指定してadbexport コマンドを実行する**

  マルチチャンク表に格納されているデータをチャンク単位でファイルに出力する場合は，-c オプションにチャンク ID を指定してadbexport コマンドを実行してください。なお，adbexport コマンドは，HADB サーバで実行できますが，HADB クライアントでは実行できません。

  adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。

- **adbsql コマンドのサブコマンド#GETDATA を実行する**

  マルチチャンク表に格納されているデータをチャンク単位でファイルに出力する場合は，adbsql コマンドのサブコマンド#GETDATA を実行してください。出力対象のチャンクのチャンク ID を指定して #GETDATA を実行し，実行結果をリダイレクションすると，実行結果をファイルに出力できます。なお，adbsql コマンドは，HADB サーバおよび HADB クライアント（Linux 版）で実行できます。

  adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』を参照してください。

> **メモ**
>
> 上記 2 つの方法のうち，adbexport コマンドを使用することを推奨します。adbexport コマンドを使用した方が，adbsql コマンドよりも，短時間でデータを出力できます。HADB クライアント（Linux 版）で，マルチチャンク表のデータをファイルに出力する必要がある場合に限り，adbsql コマンドを使用してください。

上記 2 つの方法で，データを出力できるチャンクの種類は次のとおりです。

データを出力できるチャンクの種類

- 通常状態のチャンク
- 待機状態のチャンク

上記 2 つの方法の場合，通常状態のチャンクだけでなく，待機状態のチャンクについてもデータを出力できます。なお，「11.1.9　実表のデータをファイルに出力する方法（データをエクスポートする方法)」の場合，待機状態のチャンクについては，データを出力できません。

adbexport コマンドまたはadbsql コマンドで，チャンク単位にデータを出力する手順を，それぞれ次に示します。

> **重要**
>
> adbmergechunk コマンドでマージ処理中のチャンクのデータは，ファイルに出力しないことを推奨します。adbmergechunk コマンドによるチャンクのマージ処理が完了したあとに，チャン

ク内のデータをファイルに出力するようにしてください。adbmergechunk コマンドでマージ処理中のチャンクのデータを出力した場合，マージ処理のタイミングによって，ファイルに出力されるデータが異なります。詳細については，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」の「(4)　adbmergechunk コマンド実行中に検索対象となるチャンク」を参照してください。

■出力手順（-c オプションを指定したadbexport コマンドを実行する場合）

1. 出力したいチャンクのチャンク ID を確認する

   adbexport コマンドの-c オプションに指定するチャンク ID を確認してください。チャンク ID の情報は，システム表のSTATUS_CHUNKS に格納されています。

   チャンク ID を確認する方法については，「付録 C.9　システム表の検索」を参照してください。

2. adbexport コマンドを実行する

   確認したチャンク ID を基に-c オプションを指定したadbexport コマンドを実行して，出力したいチャンクに格納されたデータを出力してください。

   また，GZIP 形式で圧縮されたファイル（CSV 形式）として出力したい場合は，--compress GZIP オプションも指定してください。

   adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。

■出力手順（adbsql コマンドのサブコマンド#GETDATA を実行する場合）

1. 出力したいチャンクのチャンク ID を確認する

   adbsql コマンドのサブコマンド#GETDATA のCHUNKID に指定するチャンク ID を確認してください。チャンク ID の情報は，システム表のSTATUS_CHUNKS に格納されています。

   チャンク ID を確認する方法については，「付録 C.9　システム表の検索」を参照してください。

2. adbsql コマンドを実行する

   確認したチャンク ID を基にadbsql コマンドのサブコマンド#GETDATA を実行して，出力したいチャンクに格納されたデータを出力してください。

   adbsql コマンドのサブコマンド#GETDATA については，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』の『adbsql サブコマンド』を参照してください。

11.4.6　チャンク単位にデータを削除する方法

チャンク単位にデータを削除する場合は，PURGE CHUNK 文を実行してください。PURGE CHUNK 文を使用したデータの削除方法を次に示します。

削除手順

1. 削除したいチャンクのチャンク ID を確認する

PURGE CHUNK 文を実行してチャンク単位でデータを削除するため，削除したいチャンクのチャンク ID を確認してください。チャンク ID の情報は，システム表のSTATUS_CHUNKS に格納されています。

チャンク ID を確認する方法については，「付録 C.9　システム表の検索」を参照してください。

2. PURGE CHUNK 文を実行する

確認したチャンク ID を基にPURGE CHUNK 文を実行して，削除したいチャンクを削除してください。

なお，カレントチャンクは削除できません。

PURGE CHUNK 文については，マニュアル『HADB　SQL リファレンス』の『PURGE CHUNK（チャンク内の全行削除)』を参照してください。

PURGE CHUNK 文のWHERE 句に条件（IN 述語，限定述語）を指定して，条件に合致するチャンクを一括して削除することができます。PURGE CHUNK 文の指定方法の例については，マニュアル『HADB　SQL リファレンス』の『PURGE CHUNK（チャンク内の全行削除)』の『PURGE CHUNK 文の指定形式および規則』の『例題』を参照してください。

メモ

- PURGE CHUNK 文でチャンクを削除すると，チャンクが利用していた格納領域を再利用できます。
- アーカイブ状態のチャンクを削除した場合，対象のアーカイブファイルも削除されます。

11.4.7　マルチチャンク表に格納されているデータを一括削除する方法

マルチチャンク表に格納されているデータを一括削除する場合は，TRUNCATE TABLE 文を実行してください。

TRUNCATE TABLE 文を使用した実表の全行削除については，「11.1.6　実表の全行を削除する方法」を参照してください。

メモ

- TRUNCATE TABLE 文を実行すると，マルチチャンク表内のすべてのチャンクが削除されます。そのあとで，空のチャンクが 1 つ作成されます。
- マルチチャンク表に定義したインデクスのチャンクについてもすべて削除されます。
- アーカイブ状態のチャンクがある場合，対象のアーカイブファイルも削除されます。

11.4.8　チャンクの状態や作成されているチャンク数を確認する方法

ここでは，マルチチャンク表に作成されたチャンクの状態や，作成されているチャンク数を確認する方法について説明します。

チャンクの情報（チャンクの状態やチャンク数）は，次に示す方法で確認できます。

- adbdbstatus コマンドを実行する
- ディクショナリ表のSQL_TABLES を検索する
- システム表のSTATUS_CHUNKS を検索する
- ディクショナリ表のSQL_COLUMNS を検索する
- ディクショナリ表のSQL_INDEXES を検索する

チャンクの情報を確認する方法を次に示します。

(1) adbdbstatus コマンドを実行する場合

adbdbstatus コマンドを実行すると，次に示すチャンクの情報を確認できます。

「DB エリアのサマリ情報」で確認できるチャンクの情報

-c dbarea オプションを指定してadbdbstatus コマンドを実行してください。コマンドの実行結果から，マルチチャンク表に関する次の情報を確認できます。

- Creatable_chunks（DB エリア内に作成できるチャンク数）
- Created_chunks（DB エリア内に作成されたチャンク数）

「表のサマリ情報」で確認できるチャンクの情報

-c table オプションを指定してadbdbstatus コマンドを実行してください。コマンドの実行結果から，マルチチャンク表に関する次の情報を確認できます。

- Creatable_chunks（実表に作成できるチャンク数）
- Created_chunks（実表に作成されたチャンク数）
- Current_chunk_ID（実表のカレントチャンク ID）
- Storage_format（表データの格納形式）

また，アーカイブマルチチャンク表の場合は，コマンドの実行結果から次の情報も確認できます。

- Archive_chunks（アーカイブ状態のチャンク数）

「アーカイブ状態のチャンクのサマリ情報」で確認できるチャンクの情報

-c archivechunk オプションを指定してadbdbstatus コマンドを実行してください。コマンドの実行結果から，アーカイブマルチチャンク表に関する次の情報を確認できます。

- Chunk_ID（チャンク ID）
- Chunk_status（チャンクの状態）
- Archive_status（チャンクのアーカイブ状態）
- Range_min（アーカイブレンジ列の最小値）
- Range_max（アーカイブレンジ列の最大値）
- Rows（アーカイブ状態のチャンクに格納されている行数）
- Archive_file_num（アーカイブ状態のチャンクに対応しているアーカイブファイル数の合計値）

- Archive_file_size（アーカイブ状態のチャンクに対応している全アーカイブファイルのファイルサイズの合計値）
- Unarchive_table_size（アーカイブ状態になる前のチャンクで，表を格納するために使用していたセグメントの容量）
- Unarchive_index_size（アーカイブ状態になる前のチャンクで，インデクスを格納するために使用していたセグメントの容量）
- Compression_ratio（アーカイブ状態のチャンクの圧縮率）
- Chunk_comment（チャンクに設定されたコメント）

「DB エリア，表とインデクスの使用量情報」で確認できるチャンクの情報

-d used オプションを指定してadbdbstatus コマンドを実行してください。コマンドの実行結果から，マルチチャンク表に関する次の情報を確認できます。

- Chunk_ID（チャンク ID）
- Chunk_create_time（チャンクの作成日時）
- Chunk_swap_time（カレントチャンクが切り替わった日時）
- Chunk_status（チャンクの状態）

また，Chunk_ID をキーにすると，次の情報も確認できます。

- Used_segments（チャンクごとの使用中セグメント数）
- Used_pages（チャンクごとの使用中ページ数）
- Storage_format（チャンク内のデータ格納形式）

「アーカイブ状態のチャンクの使用量情報」で確認できるチャンクの情報

-d used オプションと-c archivechunk オプションを指定して，adbdbstatus コマンドを実行してください。コマンドの実行結果から，アーカイブマルチチャンク表に関する次の情報を確認できます。

- Chunk_status（チャンクの状態）
- Range_min（アーカイブレンジ列の最小値）
- Range_max（アーカイブレンジ列の最大値）
- Archive_file_name（アーカイブファイルのファイル名）
- Archive_file_size（アーカイブファイルのファイルサイズ）

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

(2)　ディクショナリ表 SQL_TABLES を検索する場合

SQL_TABLES 表をSELECT 文で検索すると，次に示すチャンクの情報を確認できます。

- IS_CHUNK（マルチチャンク表かどうか）

- N_CHUNK_RESERVED（チャンク数の最大値）

また，アーカイブマルチチャンク表の場合は，次に示す情報も確認できます。

- IS_ARCHIVABLE（アーカイブマルチチャンク表かどうか）
- ARCHIVE_DIRECTORY_PATH（アーカイブディレクトリの絶対パス）

SQL_TABLES 表については，「付録 B.2　SQL_TABLES の内容」を参照してください。

SQL_TABLES 表を検索する方法については，「付録 B.22　ディクショナリ表の検索」を参照してください。

(3)　システム表 STATUS_CHUNKS を検索する場合

STATUS_CHUNKS 表をSELECT 文で検索すると，次に示すチャンクの情報を確認できます。

- CHUNK_ID（チャンク ID）
- CREATE_TIME（チャンクの作成日時）
- SWAP_TIME（カレントチャンクが切り替わった日時）
- CHUNK_COMMENT（チャンクに設定されたコメント）
- CHUNK_STATUS（チャンクの状態）
- STORAGE_FORMAT（チャンク内のデータ格納形式）

また，アーカイブマルチチャンク表の場合は，次に示す情報も確認できます。

- ARCHIVE_STATUS（チャンクのアーカイブ状態）

STATUS_CHUNKS 表については，「付録 C.5　STATUS_CHUNKS の内容」を参照してください。

STATUS_CHUNKS 表を検索する方法については，「付録 C.9　システム表の検索」を参照してください。

(4)　ディクショナリ表 SQL_COLUMNS を検索する場合

アーカイブマルチチャンク表の場合は，SQL_COLUMNS 表をSELECT 文で検索すると，次に示すチャンクの情報を確認できます。

- IS_ARCHIVE_RANGE_COLUMN（アーカイブレンジ列かどうか）

SQL_COLUMNS 表については，「付録 B.3　SQL_COLUMNS の内容」を参照してください。

SQL_COLUMNS 表を検索する方法については，「付録 B.22　ディクショナリ表の検索」を参照してください。

(5)　ディクショナリ表 SQL_INDEXES を検索する場合

アーカイブマルチチャンク表の場合は，SQL_INDEXES 表をSELECT 文で検索すると，次に示すチャンクの情報を確認できます。

- IS_ARCHIVE_RANGE（アーカイブレンジ列に対して自動的に定義されたレンジインデクスかどうか）

SQL_INDEXES 表については，「付録 B.5　SQL_INDEXES の内容」を参照してください。

SQL_INDEXES 表を検索する方法については，「付録 B.22　ディクショナリ表の検索」を参照してください。

11.4.9　チャンクをマージする方法（チャンク数を減らす方法）

バックグラウンドインポートを繰り返してチャンク数が増加すると，B-tree インデクスやテキストインデクスを使用した検索時に性能が低下するおそれがあります。インデクスがチャンクごとに分割されると，インデクスのデータの総容量が増え，I/O 回数がチャンク数に比例して増加するためです。

そのため，チャンク数が増えた場合は，adbmergechunk コマンドでチャンクをマージして，チャンク数を減らすことを検討してください。

(1)　チャンクの状態に関する留意事項

次に示すチャンクをマージできます（状態が同じである複数のチャンクをマージできます）。

レギュラーマルチチャンク表の場合

- 複数の通常状態のチャンクを，1 つの通常状態のチャンクにマージできます。
- 複数の待機状態のチャンクを，1 つの待機状態のチャンクにマージできます。

アーカイブマルチチャンク表の場合

- 通常状態でアーカイブ状態ではない複数のチャンクを，通常状態でアーカイブ状態ではない 1 つのチャンクにマージできます。
- 通常状態でアーカイブ状態の複数のチャンクを，通常状態でアーカイブ状態の 1 つのチャンクにマージできます。
- 待機状態でアーカイブ状態ではない複数のチャンクを，待機状態でアーカイブ状態ではない 1 つのチャンクにマージできます。
- 待機状態でアーカイブ状態の複数のチャンクを，待機状態でアーカイブ状態の 1 つのチャンクにマージできます。

重要

- 通常状態のチャンクと待機状態のチャンクを，マージすることはできません。
- アーカイブ状態ではないチャンクとアーカイブ状態のチャンクを，マージすることはできません。

(2) カラムストア表のチャンクをマージする場合の留意事項

次に示す条件をすべて満たす場合は，マージ元チャンクの再編成を実行したあとにチャンクをマージしてください。

- adbmergechunk コマンドの対象表がカラムストア表である
- マージ元チャンクのデータに対して更新系 SQL を実行したことがある

チャンクをマージしたあとにマージ先チャンクの再編成を実行した場合，次に示す影響があります。

- 再編成時に必要なファイルの容量やデータ用 DB エリアの空き領域がより必要になる
- 再編成に掛かる時間が長くなる

マージ元チャンクのデータに対して更新系 SQL を実行したことがあるかどうかは，adbdbstatus コマンドを実行して，**再編成要否の情報**を取得して確認してください。取得した情報のうち，次のどちらかの値が 1 以上の場合は，対象のチャンクに対して更新系 SQL が実行されたことを示しています。

- Base_row_pages（ローデータ用セグメントに含まれる基本行用ページのページ数）
- Invalid_row_information_pages（ローデータ用セグメントに含まれる無効情報管理用ページのページ数）

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

(3) adbmergechunk コマンドによるチャンクのマージ

adbmergechunk コマンドを使用してチャンクをマージする例を次に示します。

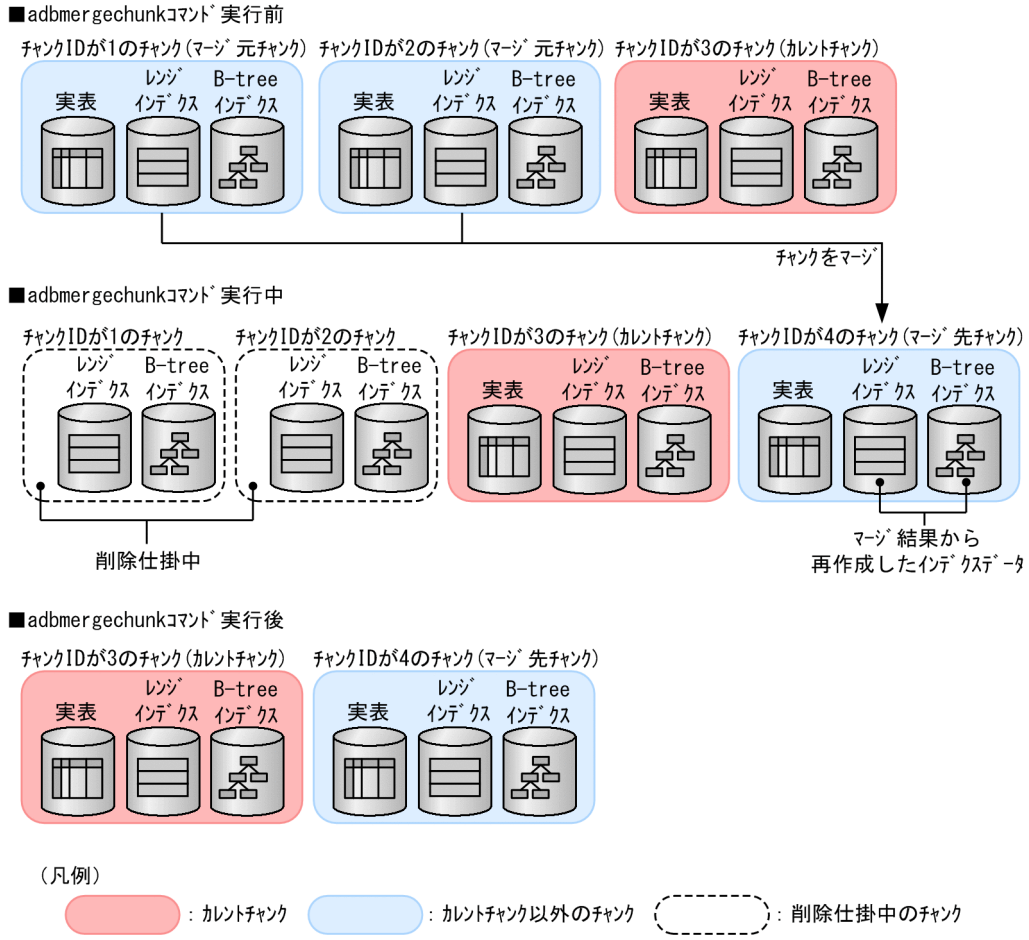
adbmergechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmergechunk (チャンクのマージ)』を参照してください。

■チャンクのマージの実行例

HADB ユーザ（ADBUSER01）が，店舗表（SHOPSLIST）の 2 つのチャンク（チャンク ID が1 および2）をマージします。

```
adbmergechunk -u ADBUSER01 -p '#HelloHADB_01' -g 2
              -w /home/adbmanager/tmp
              -z /home/adbmanager/merge_file/merge_opt_file.txt
              -m 'January 2014'
              -c 1,2
              SHOPSLIST
```

図 11-6　マージチャンク処理の例



[説明]

adbmergechunk コマンドを実行すると，マージ元チャンクである 2 つのチャンク（チャンク ID が1 および2）が，1 つのチャンク（チャンク ID が4）にマージされます。マージが完了すると，マージ元チャンクは削除されます。

メモ

■削除仕掛中のチャンク

実行中のadbmergechunk コマンドが，マージ先チャンクの完成からマージ元チャンクの削除完了までの間に中断した場合，マージ元チャンクの状態は「削除仕掛中」となります。このとき，マージ元チャンクは削除されないで残ります。adbcancel コマンドを実行したとき，HADB サーバでメモリ不足が発生したときなどが該当します。

また，adbmergechunk コマンドの--purge-chunk オプションにNOWAIT を指定して実行した場合，マージ先チャンクの完成からマージ元チャンクの削除開始までの間に，マージチャンク対象の表を参照している処理が存在したときも，マージ元チャンクの状態は「削除仕掛中」となります。そのため，マージ元チャンクは削除されないで残ります。

「削除仕掛中」となったマージ元チャンクは手動で削除する必要があります。

手動でマージ元チャンクを削除する場合は，「付録 C.9　システム表の検索」の「(12)　削除仕掛中のチャンクを調べる場合」を参照してマージ元チャンクのチャンク ID を調べ，PURGE CHUNK 文で削除してください。

なお，PURGE CHUNK 文のWHERE 句に条件（IN 述語，限定述語）を指定して，「削除仕掛中」のチャンクを一括して削除することができます。PURGE CHUNK 文の指定方法の例については，マニュアル『HADB　SQL リファレンス』の『PURGE CHUNK（チャンク内の全行削除)』の『PURGE CHUNK 文の指定形式および規則』の『例題』を参照してください。

■チャンクのマージに伴うインデクスのデータ量の一時的な増加

adbmergechunk コマンドの実行中は，マージ元チャンクのインデクスのデータ，およびマージ先チャンクのインデクスのデータの両方が一時的に存在するときがあります。そのため，インデクスのデータ量が一時的に増加します。

このとき，実行中のadbmergechunk コマンドを中断し，削除仕掛中のチャンクが残ると，一時的に作成されたインデクスのデータも残るため，削除仕掛中のチャンクを手動で削除する必要があります。

■マージ元チャンクの削除と検索処理・更新処理の関係

adbmergechunk コマンドは，マージ先チャンクが完成したタイミングで，マージチャンク対象の表に対する検索処理（SQL 文，adbexport コマンド，--shared-lock オプションを指定したadbdbstatus コマンドなど）の存在有無を確認します。検索処理が存在した場合は，確認したすべての検索処理の完了を待ってから，マージ元チャンクの削除を行います。そのため，検索処理によっては，マージ元チャンクの削除を行うまでに時間が掛かることがあります。なお，マージ先チャンクが完成したタイミング以降の検索処理については，マージ元チャンクの削除に影響しません。

また，マージ先チャンクの完成後，マージ元チャンクの削除完了を待たないで，マージチャンク対象の表に対して更新処理を行いたい場合は，adbmergechunk コマンドをadbcancel コマンドでキャンセルしてください。キャンセルすると，マージチャンク対象の表を更新できるようになります。

ただし，マージ元チャンクの状態は「削除仕掛中」となり，マージ元チャンクは削除されないで残ります。「削除仕掛中」となったマージ元チャンクは手動で削除してください。

■システム用のチャンク

データ用 DB エリアには，通常のチャンクとは別にシステム用のチャンクが存在します。システム用のチャンクは，実表やデータ用 DB エリアに作成できるチャンクがない状態でもadbmergechunk コマンドを実行できるようにするためのチャンクです。

実表やデータ用 DB エリアに作成できるチャンクがない状態でadbmergechunk コマンドを実行した場合，マージ先チャンクとしてシステム用のチャンクが使用されます。そして，マージ元チャンクが削除されたときに，削除されたチャンクの 1 つがシステム用のチャンクとなります。

(4) adbmergechunk コマンド実行中に検索対象となるチャンク

adbmergechunk コマンドの実行中に，チャンク ID を指定した検索（-c オプションを指定したadbexport コマンドやadbsql コマンドのサブコマンド#GETDATA など）を実行した場合，検索対象となるチャンクを次の表に示します。

表 11-3 adbmergechunk コマンド実行中に検索対象となるチャンク

項番	adbmergechunk コマンドの処理状況	検索対象のチャンク	検索の可否
1	コマンド開始からマージ先チャンクの完成前まで KFAA90000-I メッセージ（adbmergechunk コマンドの処理の開始）の出力から，KFAA51242-I メッセージ（マージ先チャンクの作成）の出力まで	マージ元チャンク	○
		マージ先チャンク	×
		マージ対象外のチャンク	○
2	マージ先チャンクの完成からマージ元チャンクの削除前まで KFAA51242-I メッセージ（マージ先チャンクの作成）の出力から，KFAA80243-I メッセージ（マージ元チャンクの削除）の出力まで	マージ元チャンク（削除仕掛中）	×
		マージ先チャンク	○
		マージ対象外のチャンク	○
3	マージ元チャンクの削除完了からコマンド終了まで KFAA80243-I メッセージ（マージ元チャンクの削除）の出力から，KFAA90001-I メッセージ（adbmergechunk コマンドの処理の終了）の出力まで	マージ元チャンク	×
		マージ先チャンク	○
		マージ対象外のチャンク	○

（凡例）

○：検索できます。

×：検索対象外のため，検索結果は 0 件となります。

(5) 待機状態のチャンクをマージする場合の注意点

複数の待機状態のチャンク（マージ元チャンク）をマージした場合，マージ先チャンクは待機状態のチャンクとなります。その際，カレントチャンクは変更されません。

また，待機状態のチャンクをマージした場合，マージするチャンクの状態の組み合わせによって，システム表STATUS_CHUNKS にあるチャンクの作成日時，およびカレントチャンクが切り替わった日時に設定される情報が異なります。詳細を次に示します。

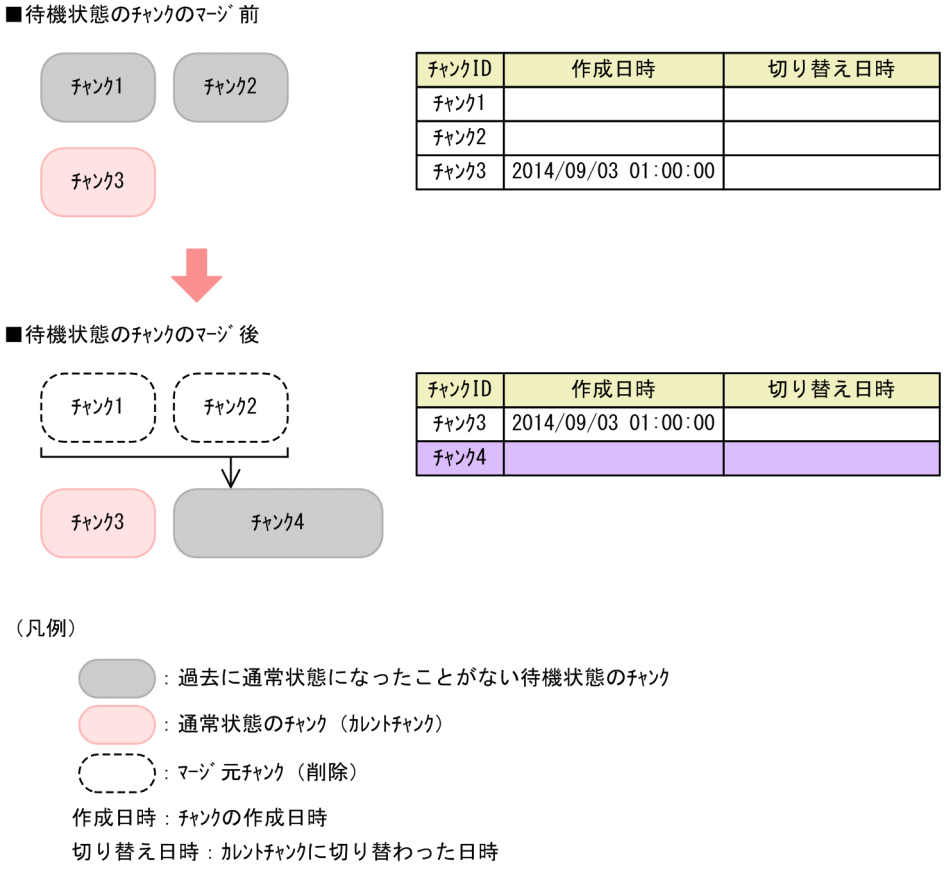
メモ

システム表STATUS_CHUNKS の詳細については，「付録 C.5 STATUS_CHUNKS の内容」を参照してください。

(a) 過去に通常状態になったことがない待機状態のチャンクだけをマージしたとき

過去に通常状態になったことがない待機状態のチャンクだけをマージしたときは，マージ先チャンクに，チャンクの作成日時およびカレントチャンクが切り替わった日時は設定されません。

図 11-7　過去に通常状態になったことがない待機状態のチャンクだけをマージした例



[説明]

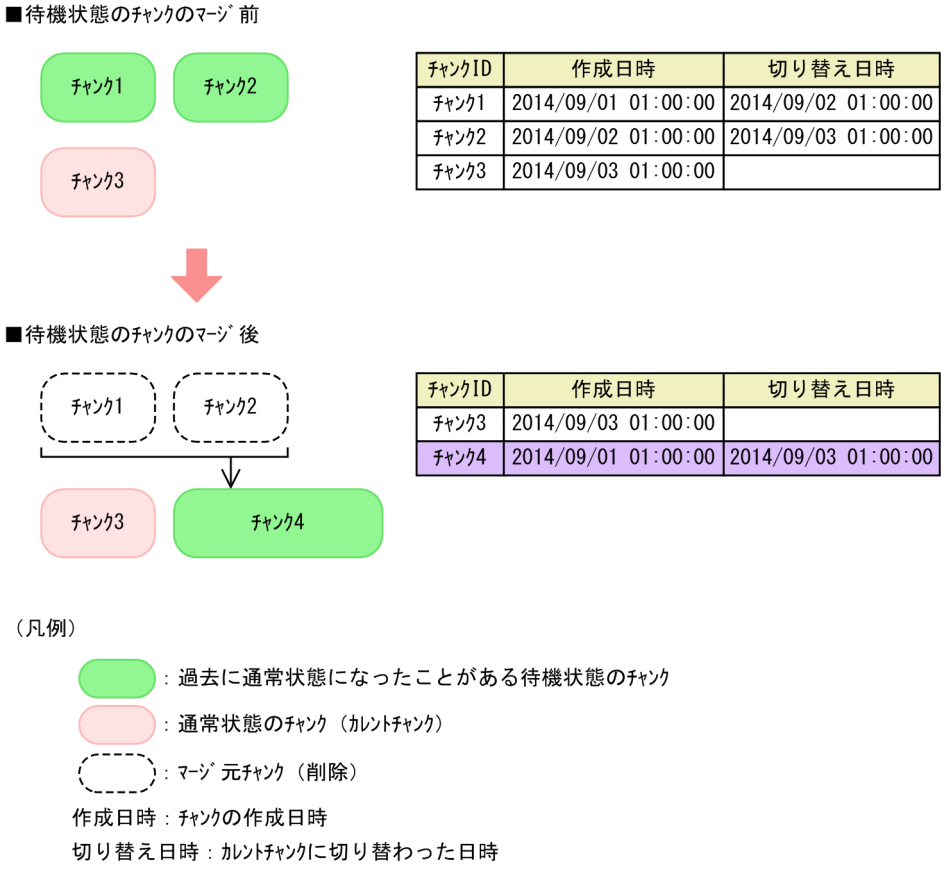
過去に通常状態になったことがない待機状態のチャンク（チャンク 1 と 2）をマージすると，マージ先チャンクとして待機状態のチャンク（チャンク 4）が作成されます。

その際，カレントチャンクは変更されません。また，チャンク 4 に，チャンクの作成日時およびカレントチャンクが切り替わった日時は設定されません。

(b)　過去に通常状態になったことがある待機状態のチャンクだけをマージしたとき

過去に通常状態になったことがある待機状態のチャンクだけをマージしたときは，マージ先チャンクに，チャンクの作成日時およびカレントチャンクが切り替わった日時が設定されます。

図 11-8　過去に通常状態になったことがある待機状態のチャンクだけをマージした例



ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ1	2014/09/01 01:00:00	2014/09/02 01:00:00
ﾁｬﾝｸ2	2014/09/02 01:00:00	2014/09/03 01:00:00
ﾁｬﾝｸ3	2014/09/03 01:00:00	

ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ3	2014/09/03 01:00:00	
ﾁｬﾝｸ4	2014/09/01 01:00:00	2014/09/03 01:00:00

[説明]

過去に通常状態になったことがある待機状態のチャンク（チャンク 1 と 2）をマージすると，マージ先チャンクとして待機状態のチャンク（チャンク 4）が作成されます。

その際，カレントチャンクは変更されません。また，チャンク 4 に，チャンクの作成日時およびカレントチャンクが切り替わった日時が設定されます。

- **チャンクの作成日時に設定される情報**

  マージ元チャンクの「チャンクの作成日時」のうち，最も古い日時が設定されます。

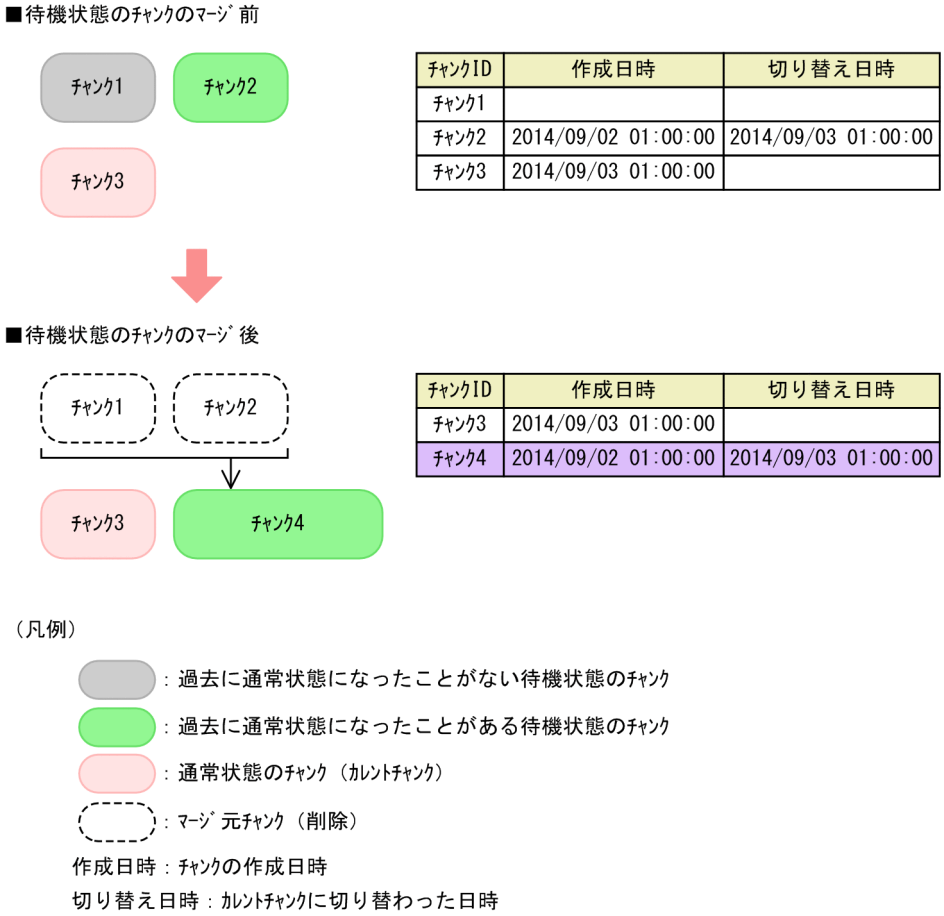
- **カレントチャンクが切り替わった日時に設定される情報**

  マージ元チャンクの「カレントチャンクが切り替わった日時」のうち，最も新しい日時が設定されます。

(c)　過去に通常状態になったことがない待機状態のチャンクと，過去に通常状態になったことがある待機状態のチャンクをマージしたとき

過去に通常状態になったことがない待機状態のチャンクと，過去に通常状態になったことがある待機状態のチャンクをマージしたときは，マージ先チャンクに，チャンクの作成日時およびカレントチャンクが切り替わった日時が設定されます。

図 11-9　過去に通常状態になったことがない待機状態のチャンクと，過去に通常状態になったことがある待機状態のチャンクをマージした例



ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ1		
ﾁｬﾝｸ2	2014/09/02 01:00:00	2014/09/03 01:00:00
ﾁｬﾝｸ3	2014/09/03 01:00:00	

ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ3	2014/09/03 01:00:00	
ﾁｬﾝｸ4	2014/09/02 01:00:00	2014/09/03 01:00:00

[説明]

過去に通常状態になったことがない待機状態のチャンク（チャンク 1）と，過去に通常状態になったことがある待機状態のチャンク（チャンク 2）をマージすると，マージ先チャンクとして待機状態のチャンク（チャンク 4）が作成されます。

その際，カレントチャンクは変更されません。また，チャンク 4 に，チャンクの作成日時およびカレントチャンクが切り替わった日時が設定されます。

- **チャンクの作成日時に設定される情報**

  マージ元チャンクの「チャンクの作成日時」のうち，最も古い日時が設定されます。

- **カレントチャンクが切り替わった日時に設定される情報**

  マージ元チャンクの「カレントチャンクが切り替わった日時」のうち，最も新しい日時が設定されます。

重要

過去に通常状態になったことがない待機状態のチャンクと，過去に通常状態になったことがある待機状態のチャンクをマージした場合，マージ先チャンク（待機状態のチャンク）を通常状態のチャンクに変更しても，カレントチャンクにはなりません。

そのため，過去に通常状態になったことがない待機状態のチャンクと，過去に通常状態になったことがある待機状態のチャンクをマージすることは推奨しません。

(6) adbmergechunk コマンドの--purge-chunk オプション指定時の考慮点

ここでは，adbmergechunk コマンドに--purge-chunk オプションを指定する場合の考慮点について説明します。

--purge-chunk オプションには，WAIT またはNOWAIT のどちらかを指定できます。通常は，WAIT を指定してください（または，--purge-chunk オプションの指定を省略してください)。

--purge-chunk オプションにWAIT を指定してadbmergechunk コマンドを実行すると，マージチャンク処理は次の順番で行われます。--purge-chunk オプションの指定を省略した場合も同様です。

■--purge-chunk オプションにWAIT を指定した場合のマージチャンク処理

1. 複数のチャンク（マージ元チャンク）が，1 つのチャンク（マージ先チャンク）にマージされます。
2. マージ完了後に，ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在しない場合は，マージ元チャンクが削除されます。

   なお，ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在する場合は，マージ元チャンクの削除を待ちます。マージチャンク対象表を参照する処理が 1 つも存在しない状況になったら，マージ元チャンクが削除されます。
3. マージ元チャンクの削除が完了すると，adbmergechunk コマンドが終了します。

--purge-chunk オプションにWAIT を指定すると，上記のとおり，マージ元チャンクの削除を待つことがあります。マージ元チャンクの削除を待つ間は，adbmergechunk コマンドが終了しません。そのため，マージチャンク対象表を更新する処理は実行できません。例えば，adbimport コマンドによるバックグラウンドインポートなどは実行できません。

ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在することで，adbmergechunk コマンドがマージ元チャンクの削除を待つと困る場合は，--purge-chunk オプションにNOWAIT を指定することを検討してください。--purge-chunk オプションにNOWAIT を指定してadbmergechunk コマンドを実行すると，マージチャンク処理は次の順番で行われます。

■--purge-chunk オプションにNOWAIT を指定した場合のマージチャンク処理

1. 複数のチャンク（マージ元チャンク）が，1 つのチャンク（マージ先チャンク）にマージされます。

   マージ完了後に，ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在しない場合は，2.に進みます。

   ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在する場合は，3.に進みます。
2. ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在しない場合は，マージ元チャンクが削除されます。

マージ元チャンクの削除が完了すると，adbmergechunk コマンドが終了します。

3. ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在する場合は，マージ元チャンクの削除を行わないで，adbmergechunk コマンドが終了します。

   マージ元チャンクは削除されないため，削除仕掛中のチャンクとして残ります。

--purge-chunk オプションにNOWAIT を指定すると，上記のとおり，マージ元チャンクの削除を行わないで adbmergechunk コマンドが終了することがあります。adbmergechunk コマンドは終了するため，マージチャンク対象表を更新する処理を実行できます。例えば，adbimport コマンドによるバックグラウンドインポートなどが実行できます。このように，--purge-chunk オプションにNOWAIT を指定すると，ほかの SQL 文やコマンドによるマージチャンク対象表を参照する処理が存在することで，adbmergechunk コマンドがマージ元チャンクの削除を待つというケースを防げます。

ただし，マージ元チャンクは削除されないで削除仕掛中のチャンクとして残るため，PURGE CHUNK 文で削除する必要があります。マージ元チャンクを削除しないと，adbmergechunk コマンドを実行できません。また，残っているマージ元チャンクの数だけ，作成できるチャンク数が減ります。adbmergechunk コマンドが終了したあとで，任意のタイミングで，マージ元チャンクをPURGE CHUNK 文で削除してください。マージ元チャンクを削除する場合は，「付録 C.9　システム表の検索」の「(12)　削除仕掛中のチャンクを調べる場合」を参照して，マージ元チャンクのチャンク ID を調べてください。そして，調べたチャンク ID を基にPURGE CHUNK 文で，マージ元チャンクを削除してください。なお，マージ元チャンクのチャンク ID は，KFAA96785-E メッセージで調べることもできます。

> **重要**
>
> adbmergechunk コマンドの終了後から次のadbmergechunk コマンドを実行するまでの間に，残っているマージ元チャンクをPURGE CHUNK 文で削除するタイミングが存在しない場合は，--purge-chunk オプションにNOWAIT を指定しないでください。マージ元チャンクが残っているマージチャンク対象表を参照・更新する処理が存在する場合には，PURGE CHUNK 文は実行できません。

11.4.10　チャンク数の最大値を変更する方法

マルチチャンク表の定義時に指定したチャンク数の最大値（CHUNK の指定値）を変更する場合は，ALTER TABLE 文を実行してください。変更手順を次に示します。

手順

1. マルチチャンク表の定義時に指定したチャンク数の最大値を確認する

   「付録 B.22　ディクショナリ表の検索」の「(13)　全マルチチャンク表に作成されるチャンク数の最大値を調べる場合」を参照して，マルチチャンク表の定義時に指定したチャンク数の最大値を確認してください。

2. ALTER TABLE 文でチャンク数の最大値を変更する

   ALTER TABLE 文の指定例を次に示します。

指定例

店舗表（SHOPSLIST）のチャンク数の最大値を500 に変更します。

```
ALTER TABLE "SHOPSLIST"
      CHANGE OPTION CHUNK=500
```

重要

- ALTER TABLE 文のCHANGE OPTION CHUNK に指定する値については，「5.2.4　マルチチャンク表を定義する場合の考慮点」の「(4)　チャンク数の最大値の求め方」を参照して求めてください。
- ALTER TABLE 文のCHANGE OPTION CHUNK に指定する値が「5.6.2　データ用 DB エリアにマルチチャンク表を格納する場合の考慮点」で説明している『1 つのデータ用 DB エリアが管理できるチャンク数の上限』を超えないかどうかを確認してください。

11.4.11　チャンクのコメントを変更する方法

チャンクのコメントを変更（新規設定，削除も含む）する場合は，adbchgchunkcomment コマンドを実行してください。変更手順を次に示します。

手順

1. コメントを変更したいチャンクのチャンク ID を調べる

   「付録 C.9　システム表の検索」の「(11)　指定したコメントが含まれているチャンクを調べる場合」を参照して，コメントを変更したいチャンクのチャンク ID を確認してください。

2. adbchgchunkcomment コマンドでコメントを変更する

   手順 1.で確認したチャンク ID を基に，adbchgchunkcomment コマンドを使用して，チャンクに設定したコメントを変更してください。adbchgchunkcomment コマンドの指定例を次に示します。

   指定例

   HADB ユーザ（ADBUSER01）が，店舗表（SHOPSLIST）のチャンク（チャンク ID が5）に対するコメントを，「'2014/09/01-2014/09/30'」に変更します。

   ```
   adbchgchunkcomment -u ADBUSER01 -p '#HelloHADB_01'
                      -m '2014/09/01-2014/09/30'
                      -c 5 SHOPSLIST
   ```

これで，チャンクに設定したコメントを変更できます。

メモ

adbchgchunkcomment コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbchgchunkcomment（チャンクのコメントの設定・変更・削除)』を参照してください。

11.4.12　チャンクの状態を変更する方法

チャンクの状態を変更する場合，adbchgchunkstatus コマンドを実行してください。adbchgchunkstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbchgchunkstatus（チャンクの状態変更)』を参照してください。また，チャンクの状態の詳細については，「2.14.2　データインポート単位のデータ管理（チャンク)」の「(4)　チャンクの状態変更」を参照してください。

adbchgchunkstatus コマンドを実行すると，次に示すチャンクの状態変更ができます。

- 待機状態のチャンクを通常状態のチャンクに変更する
  adbchgchunkstatus コマンドに，-n オプションを指定します。
- 通常状態のチャンクを待機状態のチャンクに変更する
  adbchgchunkstatus コマンドに，-w オプションを指定します。

重要

- adbchgchunkstatus コマンドの-n オプションおよび-w オプションは，同時に指定できます。
- チャンクをアーカイブ状態にする場合は，adbarchivechunk コマンドを実行します。また，チャンクのアーカイブ状態を解除する場合は，adbunarchivechunk コマンドを実行します。
  詳細については，「11.4.17　チャンクをアーカイブする方法（アーカイブマルチチャンク表使用時)」および「11.4.18　チャンクをアンアーカイブする方法（アーカイブマルチチャンク表使用時)」を参照してください。

メモ

- adbimport コマンドによるバックグラウンドインポートで待機状態のチャンクを作成する方法については，「11.4.4　マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成)」を参照してください。
- adbimport コマンドで作成した待機状態のチャンクとadbchgchunkstatus コマンドを組み合わせて利用すると，任意のタイミングでデータを検索対象にしたり，検索対象外にしたりする運用を行うこともできます。詳細については，「(4)　運用例（チャンク内のデータを入れ替える)」を参照してください。

(1)　adbchgchunkstatus コマンドによるチャンクの状態変更

adbchgchunkstatus コマンドを使用して，チャンクの状態を変更する手順を次に示します。

■チャンクの状態を変更する手順

1. 状態を変更したいチャンクのチャンク ID を調べる

「付録 C.9　システム表の検索」の「(13)　通常状態のチャンクを調べる場合」または「(14)　待機状態のチャンクを調べる場合」を参照して，状態を変更したいチャンクのチャンク ID を確認してください。

2. adbchgchunkstatus コマンドでチャンクの状態を変更する

手順 1.で確認したチャンク ID を基に，adbchgchunkstatus コマンドを使用して，チャンクの状態を変更してください。adbchgchunkstatus コマンドの指定例を次に示します。

指定例 1（待機状態のチャンクを通常状態のチャンクに変更する）

HADB ユーザ（ADBUSER01）が，店舗表（SHOPSLIST）の待機状態のチャンク（チャンク ID が1）を，通常状態のチャンクに変更します。

```
adbchgchunkstatus -u ADBUSER01 -p '#HelloHADB_01'
                  -n 1 SHOPSLIST
```

重要

待機状態のチャンクが大量にある場合，状態を変更する前に，待機状態のチャンクをマージしてチャンク数を減らしてください。通常状態のチャンクが大量になると，検索性能が低下するおそれがあります。チャンクをマージする場合は，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」を参照してください。

指定例 2（通常状態のチャンクを待機状態のチャンクに変更する）

HADB ユーザ（ADBUSER01）が，店舗表（SHOPSLIST）の通常状態のチャンク（チャンク ID が2）を，待機状態のチャンクに変更します。

```
adbchgchunkstatus -u ADBUSER01  -p '#HelloHADB_01'
                  -w 2 SHOPSLIST
```

これで，チャンクの状態を変更できます。

なお，チャンクの状態変更によって，カレントチャンクが変更されることがあります。詳細については，「(3)　チャンクの状態変更によるカレントチャンクの切り替え」を参照してください。また，カレントチャンクを元に戻す（チャンクの状態変更の前にカレントチャンクだったチャンクを，状態変更のあともカレントチャンクとする）手順については，「(4)　運用例（チャンク内のデータを入れ替える）」を参照してください。

(2)　チャンクの状態と実行できる SQL 文・コマンドの関係

チャンクの状態によって，チャンクに含まれるデータに対して，実行できる SQL 文およびコマンドが異なります。チャンクの状態と，実行できる SQL 文およびコマンドの関係を次の 2 つの表に示します。

- アーカイブ状態ではないチャンクの場合

  「表 11-4　チャンクの状態と，実行できる SQL 文およびコマンドの関係（アーカイブ状態ではないチャンクの場合）」を参照してください。

- アーカイブ状態のチャンクの場合

「表 11-5　チャンクの状態と，実行できる SQL 文およびコマンドの関係（アーカイブ状態のチャンクの場合）」を参照してください。

表 11-4　チャンクの状態と，実行できる SQL 文およびコマンドの関係（アーカイブ状態ではないチャンクの場合）

項番	実行する SQL 文およびコマンド	チャンクの状態		
		通常状態	待機状態	削除仕掛中
1	SELECT 文	○	×	×
2	INSERT 文	○	×	×
3	UPDATE 文	○	×	×
4	DELETE 文	○	×	×
5	PURGE CHUNK 文	○	○	○
6	TRUNCATE TABLE 文	○	○	○
7	adbstart コマンド （レンジインデクスの一括先読みを実行する場合）	○	○	×
8	adbsql コマンド	○	△※1	×
9	adbmergechunk コマンド	△※2	△※3	×
10	adbexport コマンド	○	△※4	×
11	adbidxrebuild コマンド	○	○※5	×
12	adbdbstatus コマンド （チャンクの情報を参照する場合）	○	○	○
13	adbchgchunkcomment コマンド	○	○	○
14	adbchgchunkstatus コマンド	○	○	×
15	adbarchivechunk コマンド	○	△※6	×
16	adbunarchivechunk コマンド	×	×	×

（凡例）

○：実行できます。

△：実行できますが，制限があります。

×：実行できません。

注※1

adbsql コマンドのサブコマンド#GETDATA および#GETCOUNT だけが実行できます。

注※2

通常状態で，かつアーカイブ状態ではない複数のチャンクを，通常状態で，かつアーカイブ状態ではない 1 つのチャンクにマージできます。それ以外のチャンクのマージはできません。

注※3

待機状態で，かつアーカイブ状態ではない複数のチャンクを，待機状態で，かつアーカイブ状態ではない 1 つのチャンクにマージできます。それ以外のチャンクのマージはできません。

注※4

adbexport コマンドの-c オプションを指定した検索だけ実行できます。

注※5

待機状態のチャンクに含まれるインデクスも再作成されます。

注※6

一度も通常状態になったことがない待機状態のチャンクは，アーカイブ状態にできません。

表 11-5　チャンクの状態と，実行できる SQL 文およびコマンドの関係（アーカイブ状態のチャンクの場合）

項番	実行する SQL 文およびコマンド	チャンクの状態		
		通常状態	待機状態	削除仕掛中
1	SELECT 文	○	×	×
2	INSERT 文	×	×	×
3	UPDATE 文	×	×	×
4	DELETE 文	×	×	×
5	PURGE CHUNK 文	○	○	○
6	TRUNCATE TABLE 文	○	○	○
7	adbstart コマンド (レンジインデクスの一括先読みを実行する場合)	×※1	×※1	×※1
8	adbsql コマンド	○	△※2	×
9	adbmergechunk コマンド	△※3	△※4	×
10	adbexport コマンド	○	△※5	×
11	adbidxrebuild コマンド	×※6	×※6	×※6
12	adbdbstatus コマンド (チャンクの情報を参照する場合)	○	○	○
13	adbchgchunkcomment コマンド	○	○	○
14	adbchgchunkstatus コマンド	○	○	×
15	adbarchivechunk コマンド	×	×	×
16	adbunarchivechunk コマンド	○	○	×

(凡例)

○：実行できます。

△：実行できますが，制限があります。

×：実行できません。

注※1

アーカイブ状態のチャンク内のデータにはインデクスが存在しません。そのため，adbstart コマンドによるレンジインデクスの一括先読みは実行できません。

注※2

adbsql コマンドのサブコマンド#GETDATA および#GETCOUNT だけが実行できます。

注※3

通常状態で，かつアーカイブ状態の複数のチャンクを，通常状態で，かつアーカイブ状態の 1 つのチャンクにマージできます。それ以外のチャンクのマージはできません。

注※4

待機状態で，かつアーカイブ状態の複数のチャンクを，待機状態で，かつアーカイブ状態の 1 つのチャンクにマージできます。それ以外のチャンクのマージはできません。

注※5

adbexport コマンドの-c オプションを指定した検索だけが実行できます。

注※6

アーカイブ状態のチャンク内のデータにはインデクスが存在しません。そのため，adbidxrebuild コマンドによるインデクスの再作成は実行できません。

(3)　チャンクの状態変更によるカレントチャンクの切り替え

adbchgchunkstatus コマンドでチャンクの状態を変更する場合，状態を変更するチャンクの種類と状態変更の処理内容によって，カレントチャンクの切り替えが発生することがあります。また，システム表 STATUS_CHUNKS にあるチャンクの作成日時およびカレントチャンクが切り替わった日時に設定される情報が異なります。

> **メモ**
>
> システム表STATUS_CHUNKS の詳細については，「付録 C.5　STATUS_CHUNKS の内容」を参照してください。

詳細を次の表に示します。

表 11-6　チャンクの状態変更の実行条件

状態変更するチャンクの種類と状態変更の処理内容		通常状態のチャンクへの変更			
		なし	過去に通常状態になったことがある待機状態のチャンク※1	過去に通常状態になったことがない待機状態のチャンク※2	過去に通常状態になったことがある待機状態のチャンク※1，および過去に通常状態になったことがない待機状態のチャンク※2
待機状態のチャンクへの変更	なし	－	(a)	(b)	(c)
	カレントチャンクではない通常状態のチャンク	(d)	(e)	(f)	(g)
	カレントチャンクである通常状態のチャンク	(h)	(i)	(j)	(k)
	カレントチャンク，およびカレントチャンクではない通常状態のチャンク	(l)	(m)	(n)	(o)

（凡例）

－：該当しません。

注※1

チャンクの作成日時，およびカレントチャンクが切り替わった日時が設定された待機状態のチャンクのことです。

注※2

チャンクの作成日時，およびカレントチャンクが切り替わった日時が設定されていない待機状態のチャンクのことです。

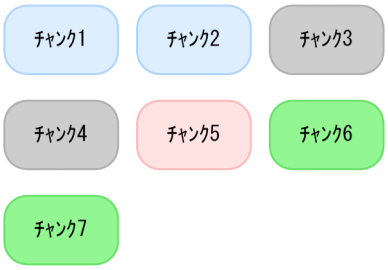
■（*a*）～（*o*）の詳細

上記の表で示した（*a*）～（*o*）の詳細については，「(a)　adbchgchunkstatus コマンドの実行例 1」～「(o)　adbchgchunkstatus コマンドの実行例 15」で示します。

なお，各実行例は，次の図で示すチャンクに対して，adbchgchunkstatus コマンドを実行しています。

図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態

■ﾁｬﾝｸの構成とﾁｬﾝｸの情報



ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ1	2014/09/01 01:00:00	2014/09/02 01:00:00
ﾁｬﾝｸ2	2014/09/02 01:00:00	2014/09/03 01:00:00
ﾁｬﾝｸ3	2014/09/03 01:00:00	2014/09/04 01:00:00
ﾁｬﾝｸ4	2014/09/04 01:00:00	2014/09/05 01:00:00
ﾁｬﾝｸ5	2014/09/05 01:00:00	
ﾁｬﾝｸ6		
ﾁｬﾝｸ7		

（凡例）

：通常状態のﾁｬﾝｸ

：過去に通常状態になったことがある待機状態のﾁｬﾝｸ

：通常状態のﾁｬﾝｸ（ｶﾚﾝﾄﾁｬﾝｸ）

：過去に通常状態になったことがない待機状態のﾁｬﾝｸ

作成日時：ﾁｬﾝｸの作成日時

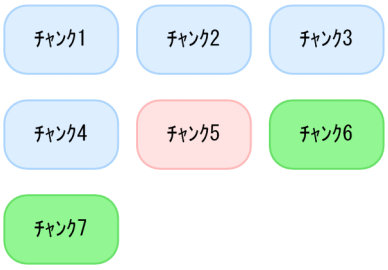
切り替え日時：ｶﾚﾝﾄﾁｬﾝｸに切り替わった日時

(a)　adbchgchunkstatus コマンドの実行例 1

「表 11-6　チャンクの状態変更の実行条件」の（*a*）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-11　(a) の条件で adbchgchunkstatus コマンドを実行した場合

■ﾁｬﾝｸの構成とﾁｬﾝｸの情報



ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ1	2014/09/01 01:00:00	2014/09/02 01:00:00
ﾁｬﾝｸ2	2014/09/02 01:00:00	2014/09/03 01:00:00
ﾁｬﾝｸ3	2014/09/03 01:00:00	2014/09/04 01:00:00
ﾁｬﾝｸ4	2014/09/04 01:00:00	2014/09/05 01:00:00
ﾁｬﾝｸ5	2014/09/05 01:00:00	
ﾁｬﾝｸ6		
ﾁｬﾝｸ7		

（凡例）

：通常状態のﾁｬﾝｸ

：通常状態のﾁｬﾝｸ（ｶﾚﾝﾄﾁｬﾝｸ）

：過去に通常状態になったことがない待機状態のﾁｬﾝｸ

作成日時：ﾁｬﾝｸの作成日時

切り替え日時：ｶﾚﾝﾄﾁｬﾝｸに切り替わった日時

[説明]

過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4）を，通常状態のチャンクに変更します。

■（*a*）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

変更されません。

チャンクの作成日時

設定されません。

カレントチャンクが切り替わった日時

設定されません。

(b)　adbchgchunkstatus コマンドの実行例 2

「表 11-6　チャンクの状態変更の実行条件」の（*b*）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-12　(b) の条件で adbchgchunkstatus コマンドを実行した場合

[説明]

過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■（*b*）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンク

カレントチャンクが切り替わった日時

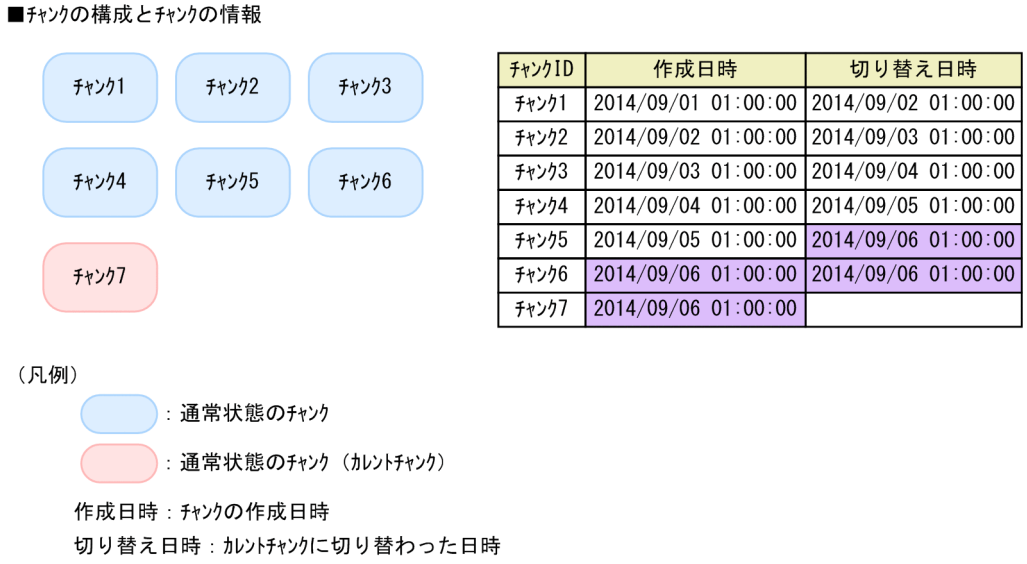
次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならなかったチャンク
- 変更される前にカレントチャンクだったチャンク

(c)　adbchgchunkstatus コマンドの実行例 3

「表 11-6　チャンクの状態変更の実行条件」の（c）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-13　（c）の条件で adbchgchunkstatus コマンドを実行した場合



[説明]

過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4），および過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■（c）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

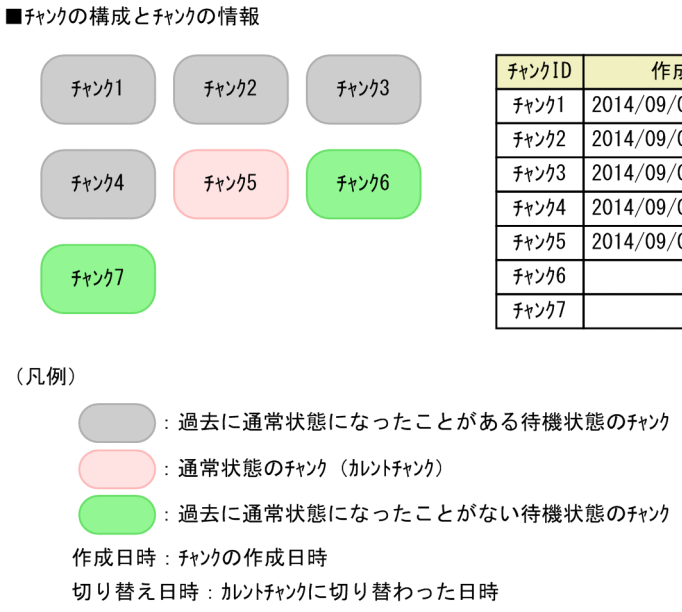
次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク
- 変更される前にカレントチャンクだったチャンク

(d)　adbchgchunkstatus コマンドの実行例 4

「表 11-6　チャンクの状態変更の実行条件」の（d）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-14　（d）の条件で adbchgchunkstatus コマンドを実行した場合



チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	
チャンク6		
チャンク7		

[説明]

通常状態のチャンク（チャンク 1 と 2）を，待機状態のチャンクに変更します。

■（d）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

変更されません。

チャンクの作成日時

設定されません。

カレントチャンクが切り替わった日時

設定されません。

(e) adbchgchunkstatus コマンドの実行例 5

「表 11-6 チャンクの状態変更の実行条件」の（e）の条件で，「図 11-10 adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-15 （e）の条件で adbchgchunkstatus コマンドを実行した場合

ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ1	2014/09/01 01:00:00	2014/09/02 01:00:00
ﾁｬﾝｸ2	2014/09/02 01:00:00	2014/09/03 01:00:00
ﾁｬﾝｸ3	2014/09/03 01:00:00	2014/09/04 01:00:00
ﾁｬﾝｸ4	2014/09/04 01:00:00	2014/09/05 01:00:00
ﾁｬﾝｸ5	2014/09/05 01:00:00	
ﾁｬﾝｸ6		
ﾁｬﾝｸ7		

［説明］

通常状態のチャンク（チャンク 1 と 2）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4）を，通常状態のチャンクに変更します。

■（e）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

変更されません。

チャンクの作成日時

設定されません。

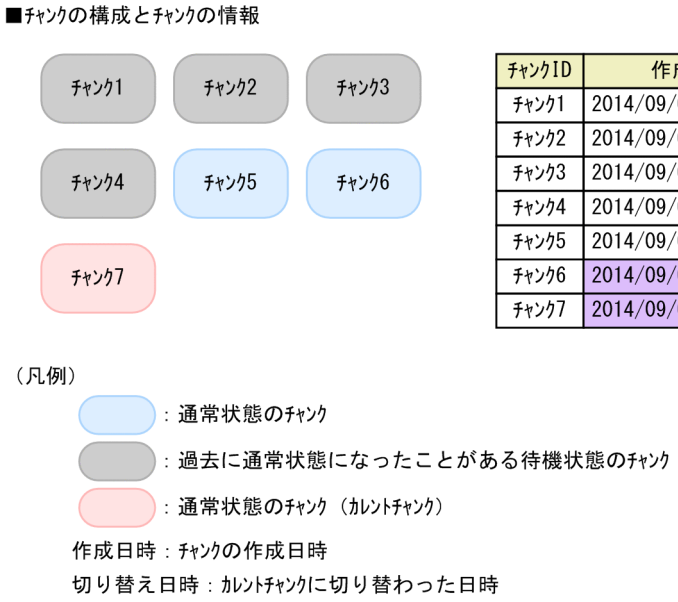
カレントチャンクが切り替わった日時

設定されません。

(f)　adbchgchunkstatus コマンドの実行例 6

「表 11-6　チャンクの状態変更の実行条件」の（f）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-16　(f) の条件で adbchgchunkstatus コマンドを実行した場合



チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	2014/09/06 01:00:00
チャンク6	2014/09/06 01:00:00	2014/09/06 01:00:00
チャンク7	2014/09/06 01:00:00	

［説明］

通常状態のチャンク（チャンク 1 と 2）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■ (f) の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

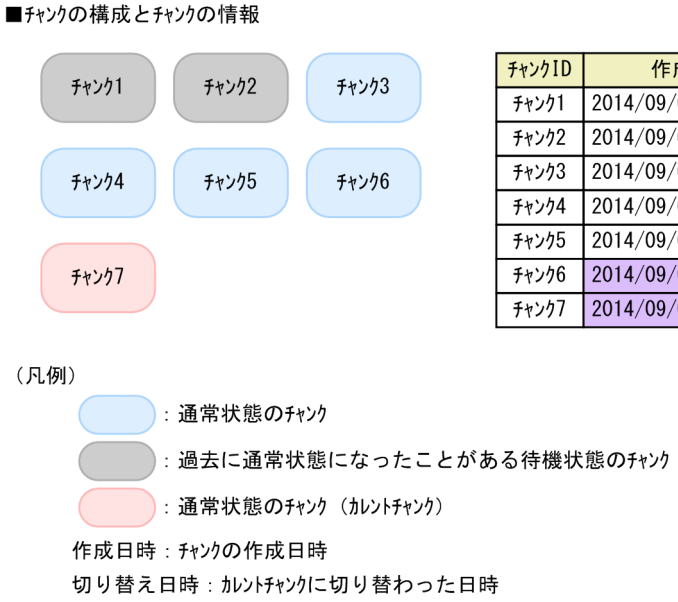
- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク

- 変更される前にカレントチャンクだったチャンク

(g)　adbchgchunkstatus コマンドの実行例 7

「表 11-6　チャンクの状態変更の実行条件」の（g）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-17　（g）の条件で adbchgchunkstatus コマンドを実行した場合



チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	2014/09/06 01:00:00
チャンク6	2014/09/06 01:00:00	2014/09/06 01:00:00
チャンク7	2014/09/06 01:00:00	

［説明］

通常状態のチャンク（チャンク 1 と 2）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4），および過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■ （g）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

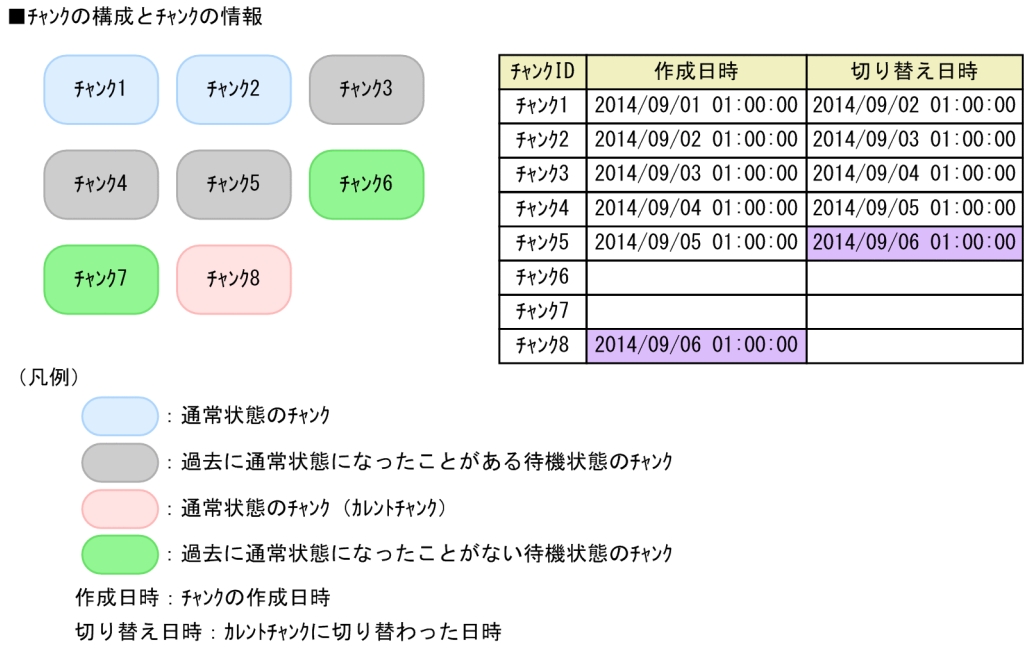
次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク
- 変更される前にカレントチャンクだったチャンク

(h)　adbchgchunkstatus コマンドの実行例 8

「表 11-6　チャンクの状態変更の実行条件」の（h）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-18　（h）の条件で adbchgchunkstatus コマンドを実行した場合



[説明]

カレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。

このとき，新規にチャンクが作成されて，カレントチャンクが変更されます（チャンク 8 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■（h）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

新規にチャンクが作成されて，そのチャンクがカレントチャンクになります。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 新規に作成されたチャンク

カレントチャンクが切り替わった日時

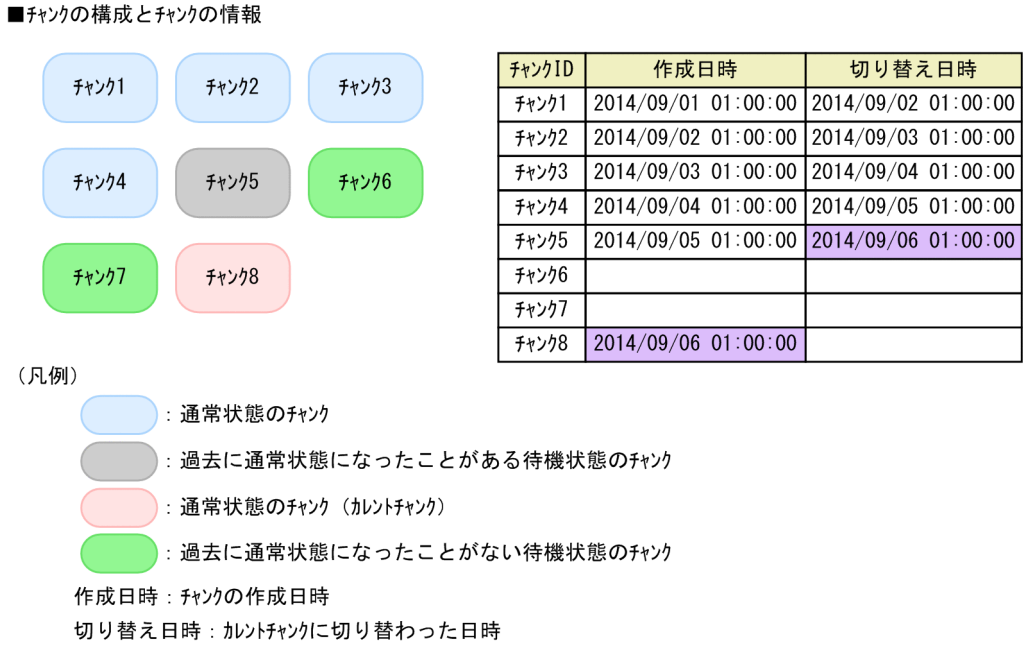
次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 変更される前にカレントチャンクだったチャンク

(i)　adbchgchunkstatus コマンドの実行例 9

「表 11-6　チャンクの状態変更の実行条件」の（*i*）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-19　(i) の条件で adbchgchunkstatus コマンドを実行した場合



チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	2014/09/06 01:00:00
チャンク6		
チャンク7		
チャンク8	2014/09/06 01:00:00	

[説明]

カレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4）を，通常状態のチャンクに変更します。

このとき，新規にチャンクが作成されて，カレントチャンクが変更されます（チャンク 8 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■ (*i*) の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

新規にチャンクが作成されて，そのチャンクがカレントチャンクになります。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 新規に作成されたチャンク

カレントチャンクが切り替わった日時

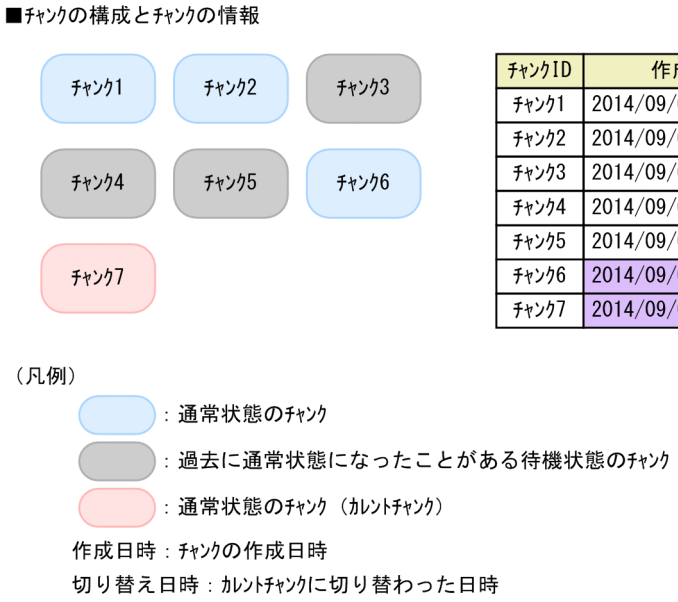
次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 変更される前にカレントチャンクだったチャンク

(j) adbchgchunkstatus コマンドの実行例 10

「表 11-6　チャンクの状態変更の実行条件」の（j）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-20　（j）の条件で adbchgchunkstatus コマンドを実行した場合



ﾁｬﾝｸID	作成日時	切り替え日時
ﾁｬﾝｸ1	2014/09/01 01:00:00	2014/09/02 01:00:00
ﾁｬﾝｸ2	2014/09/02 01:00:00	2014/09/03 01:00:00
ﾁｬﾝｸ3	2014/09/03 01:00:00	2014/09/04 01:00:00
ﾁｬﾝｸ4	2014/09/04 01:00:00	2014/09/05 01:00:00
ﾁｬﾝｸ5	2014/09/05 01:00:00	2014/09/06 01:00:00
ﾁｬﾝｸ6	2014/09/06 01:00:00	2014/09/06 01:00:00
ﾁｬﾝｸ7	2014/09/06 01:00:00	

［説明］

カレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■ (j) の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

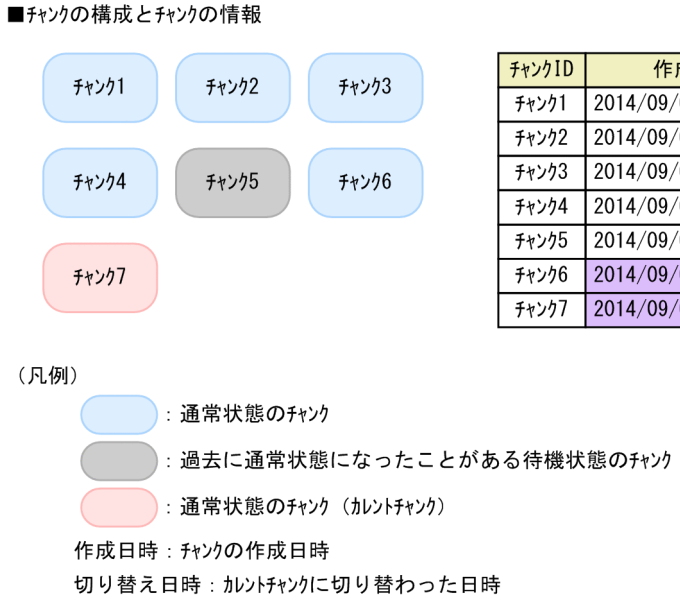
- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク

- 変更される前にカレントチャンクだったチャンク

(k)　adbchgchunkstatus コマンドの実行例 11

「表 11-6　チャンクの状態変更の実行条件」の（k）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-21　（k）の条件で adbchgchunkstatus コマンドを実行した場合



チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	2014/09/06 01:00:00
チャンク6	2014/09/06 01:00:00	2014/09/06 01:00:00
チャンク7	2014/09/06 01:00:00	

[説明]

カレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4），および過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■ （k）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク
- 変更される前にカレントチャンクだったチャンク

(l)　adbchgchunkstatus コマンドの実行例 12

「表 11-6　チャンクの状態変更の実行条件」の（l）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-22　(l) の条件で adbchgchunkstatus コマンドを実行した場合

[説明]

通常状態のチャンク（チャンク 1 と 2），およびカレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。

このとき，新規にチャンクが作成されて，カレントチャンクが変更されます（チャンク 8 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■ (l) の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

新規にチャンクが作成されて，そのチャンクがカレントチャンクになります。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 新規に作成されたチャンク

カレントチャンクが切り替わった日時

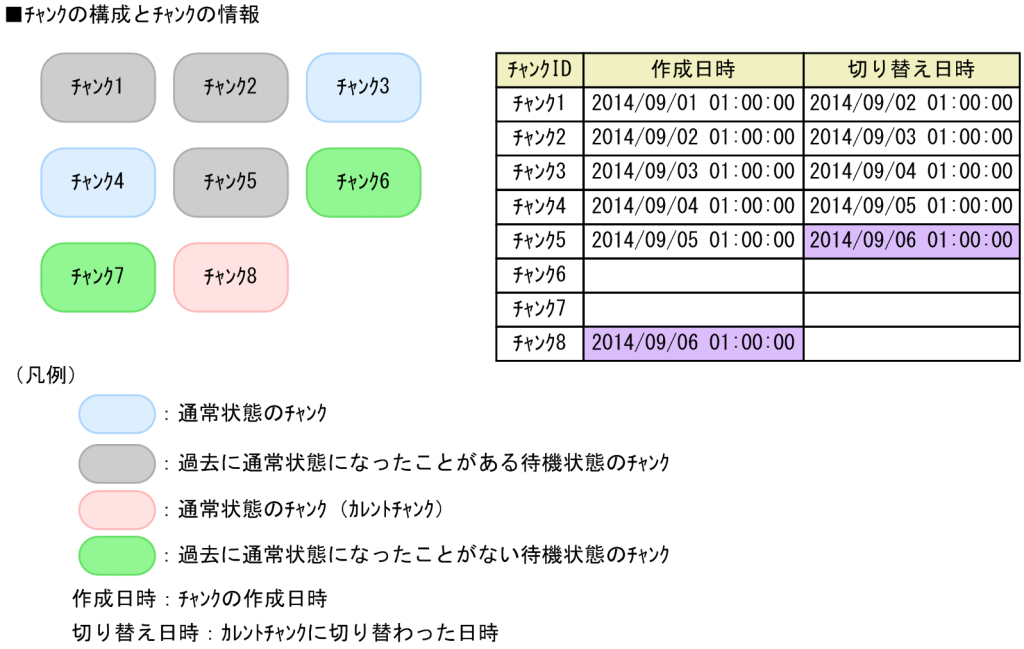
次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 変更される前にカレントチャンクだったチャンク

(m)　adbchgchunkstatus コマンドの実行例 13

「表 11-6　チャンクの状態変更の実行条件」の（*m*）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-23　（m）の条件で adbchgchunkstatus コマンドを実行した場合



チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	2014/09/06 01:00:00
チャンク6		
チャンク7		
チャンク8	2014/09/06 01:00:00	

[説明]

通常状態のチャンク（チャンク 1 と 2），およびカレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4）を，通常状態のチャンクに変更します。

このとき，新規にチャンクが作成されて，カレントチャンクが変更されます（チャンク 8 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■（*m*）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

新規にチャンクが作成されて，そのチャンクがカレントチャンクになります。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 新規に作成されたチャンク

カレントチャンクが切り替わった日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- 変更される前にカレントチャンクだったチャンク

(n)　adbchgchunkstatus コマンドの実行例 14

「表 11-6　チャンクの状態変更の実行条件」の（*n*）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-24　（n）の条件で adbchgchunkstatus コマンドを実行した場合

[説明]

通常状態のチャンク（チャンク 1 と 2），およびカレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■（*n*）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク

- 変更される前にカレントチャンクだったチャンク

(o)　adbchgchunkstatus コマンドの実行例 15

「表 11-6　チャンクの状態変更の実行条件」の（o）の条件で，「図 11-10　adbchgchunkstatus コマンド実行前のチャンクの状態」のチャンクに対して，adbchgchunkstatus コマンドを実行した場合の例を次に示します。

図 11-25　（o）の条件で adbchgchunkstatus コマンドを実行した場合

チャンクID	作成日時	切り替え日時
チャンク1	2014/09/01 01:00:00	2014/09/02 01:00:00
チャンク2	2014/09/02 01:00:00	2014/09/03 01:00:00
チャンク3	2014/09/03 01:00:00	2014/09/04 01:00:00
チャンク4	2014/09/04 01:00:00	2014/09/05 01:00:00
チャンク5	2014/09/05 01:00:00	2014/09/06 01:00:00
チャンク6	2014/09/06 01:00:00	2014/09/06 01:00:00
チャンク7	2014/09/06 01:00:00	

[説明]

通常状態のチャンク（チャンク 1 と 2），およびカレントチャンク（チャンク 5）を，待機状態のチャンクに変更します。また，過去に通常状態になったことがある待機状態のチャンク（チャンク 3 と 4），および過去に通常状態になったことがない待機状態のチャンク（チャンク 6 と 7）を，通常状態のチャンクに変更します。

このとき，カレントチャンクが変更されます（チャンク 7 になります）。また，チャンクの作成日時とカレントチャンクの切り替え日時が設定されます。

■（o）の条件でadbchgchunkstatus コマンドを実行した場合の処理結果

カレントチャンク

過去に通常状態になったことがない待機状態のチャンクのうち，adbchgchunkstatus コマンドの-n オプションに指定されたチャンク ID が最も大きいチャンクに変更されます。

チャンクの作成日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，過去に通常状態になったことがないチャンク

カレントチャンクが切り替わった日時

次のチャンクにadbchgchunkstatus コマンドを実行した日時が設定されます。

- adbchgchunkstatus コマンドの-n オプションに指定された待機状態のチャンクのうち，カレントチャンクにならないで，かつ過去に通常状態になったことがないチャンク
- 変更される前にカレントチャンクだったチャンク

(4) 運用例（チャンク内のデータを入れ替える）

チャンク内のデータを入れ替える場合の運用例を説明します。チャンク内のデータを入れ替える際，adbchgchunkstatus コマンドによるチャンクの状態変更を利用します。

運用例

店舗数の拡大および統廃合によって，C 地区の店舗情報が変更されました。C 地区の店舗情報は，店舗表（SHOPSLIST）のチャンク（チャンク ID が 3）に格納されています。C 地区の新店舗情報は，CSV 形式のファイルに格納されているため，C 地区の旧店舗情報が格納されているチャンク（チャンク ID が 3）内のデータをすべて入れ替えます。

データの入れ替え中は，C 地区の旧店舗情報を参照できるようにします。

チャンク内のデータを入れ替えるときの流れ

次の流れでチャンク内のデータを入れ替えます。

1. C 地区の新店舗情報をバックグラウンドインポートする
2. チャンクの状態を変更して，C 地区の旧店舗情報を参照できなくする（同時に C 地区の新店舗情報を参照できるようにする）
3. C 地区の旧店舗情報を削除する

データの入れ替えを始める前に確認すること

『チャンク内のデータの入れ替え手順』を実施すると，カレントチャンクが変わります。カレントチャンクを変えたくない場合は，カレントチャンクのチャンク ID とチャンクのコメントを確認してください。「付録 C.9　システム表の検索」の「(16)　カレントチャンクの情報を調べる場合」を参照し，カレントチャンクのCHUNK_ID（チャンク ID）とCHUNK_COMMENT（チャンクに設定されたコメント）を確認してください。

なお，入れ替え前のデータがカレントチャンク内にある場合は，カレントチャンクのチャンク ID とチャンクのコメントを確認する必要はありません。

チャンク内のデータの入れ替え手順

1. C 地区の新店舗情報をバックグラウンドインポートする

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file02.txt
          -b --status wait
          SHOPSLIST
          /home/adbmanager/imp_file/imp_data_path02.txt
```

adbimport コマンドの実行時，--status wait オプションを指定してください。このオプションを指定すると，データをインポートするチャンクの状態が待機状態になります（新店舗情報は参照できない状態になります）。

> **メモ**
>
> バックグラウンドインポートの処理中，旧店舗情報は参照できます。

2. インポートした新店舗情報が格納されたチャンクのチャンク ID を確認する

   手順 1.のバックグラウンドインポート時にKFAA51242-I メッセージが出力されます。このメッセージにチャンク ID が表示されます。

```
KFAA51242-I A chunk was created. (chunk id = 5 )
```

   上記の例の場合，インポートした新店舗情報は，チャンク ID が 5 のチャンクに格納されています。

3. チャンクの状態を変更して，C 地区の旧店舗情報を参照できなくする（同時に C 地区の新店舗情報を参照できるようにする）

```
adbchgchunkstatus -u ADBUSER01 -p '#HelloHADB_01'
                  -n 5 -w 3 SHOPSLIST
```

   旧店舗情報を参照できなくし，新店舗情報を参照できるようにするために，adbchgchunkstatus コマンドでチャンクの状態を変更します。このとき，-n オプションには5 を，-w オプションには3 を指定します。新店舗情報を格納しているチャンク（チャンク ID が 5）が通常状態になり，チャンク内のデータが参照できるようになります。また，旧店舗情報を格納しているチャンク（チャンク ID が 3）が待機状態になり，チャンク内のデータは参照できなくなります。

4. C 地区の旧店舗情報を削除する

```
PURGE CHUNK "SHOPSLIST" WHERE CHUNKID=3
```

   旧店舗情報を格納しているチャンク（チャンク ID が 3）内のデータをPURGE CHUNK 文で削除します。

   なお，PURGE CHUNK 文が排他を取得する DB エリアを，参照または更新しているトランザクションがある場合，PURGE CHUNK 文がエラーになります。DB エリアを参照または更新しているトランザクションがあるかどうかを確認するには，adbls -d lock コマンドを実行して，DB エリアの排他を取得しているコネクションの有無を確認してください。adbls -d lock コマンドの詳細については，マニュアル『HADB　コマンドリファレンス』の『adbls -d lock（排他資源の状態表示)』を参照してください。

上記の操作が完了すると，この例では，チャンク ID が 5 のチャンクがカレントチャンクになります。カレントチャンクをデータの差し替え前と同じにしたい場合は，『カレントチャンクの変更手順』の操作を行ってください。

カレントチャンクの変更手順

1. adbimport コマンドで空のチャンク（データが 0 件のチャンク）を作成する

   adbimport コマンドの入力データファイルに空のファイルを指定して，-b オプションを指定してバックグラウンドインポートを実行してください。

2. チャンクをマージする

手順 1.で作成した空のチャンクと，『データの入れ替えを始める前に確認すること』で確認したカレントチャンクを，adbmergechunk コマンドでマージしてください。このとき，adbmergechunk コマンドの-m オプションには，『データの入れ替えを始める前に確認すること』で確認したチャンクのコメントを指定してください。

チャンクのマージが完了すると，データの入れ替え前にカレントチャンクだったチャンクが，カレントチャンクに戻ります。

11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法

マルチチャンク表の再編成が必要かどうかを確認する方法について説明します。

メモ

- シングルチャンク表の再編成が必要かどうかを確認する方法については，「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」を参照してください。
- アーカイブ状態のチャンクについては，データの再編成は不要です。

(1)　表の再編成が必要な理由

マルチチャンク表に対して，次の操作を繰り返し実行すると，表の検索性能やデータの格納効率が低下します。

- DELETE 文による行の削除
- UPDATE 文による行の更新
- INSERT 文による行の挿入
- adbimport コマンドによる少量データのバックグラウンドインポート

この場合，マルチチャンク表を再編成すると，データ用 DB エリア内のむだな空き領域が解放されて，表の検索性能やデータの格納効率が向上します。

マルチチャンク表の検索性能やデータの格納効率が低下する理由を次に示します。

マルチチャンク表がローストア表の場合

シングルチャンク表の場合と理由は同じです。理由については，「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」の「(1)　表の再編成が必要な理由」の「**シングルチャンク表がローストア表の場合**」を参照してください。

マルチチャンク表がカラムストア表の場合

シングルチャンク表の場合と理由は同じです。理由については，「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」の「(1)　表の再編成が必要な理由」の「**シングルチャンク表がカラムストア表の場合**」を参照してください。

ローストア表，カラムストア表共通の理由

マルチチャンク表に対して，adbimport コマンドによる少量データのバックグラウンドインポートを繰り返し実行すると，空き領域の一部が使用されなくなるおそれがあります。この使用されない空き領域は再利用されません（むだな空き領域になります）。その結果，データの格納効率が悪くなり，表の検索性能が低下します。

(2)　再編成が必要かどうかを確認する方法

マルチチャンク表の再編成が必要かどうかを確認する方法を説明します。

マルチチャンク表がローストア表で，SQL 文による行の更新，削除を繰り返した場合は，「(a)　マルチチャンク表がローストア表の場合」を参照して，表の再編成が必要かどうかを確認してください。

マルチチャンク表がカラムストア表で，SQL 文による行の追加，更新，削除を繰り返した場合は，「(b)　マルチチャンク表がカラムストア表の場合」を参照して，表の再編成が必要かどうかを確認してください。

表に対してadbimport コマンドによる少量データのバックグラウンドインポートを繰り返した場合は，「(c)　少量のデータをバックグラウンドインポートで繰り返し追加した場合」を参照して，表の再編成が必要かどうかを確認してください。

(a)　マルチチャンク表がローストア表の場合

データの格納効率を計算してください。データの格納効率が悪い場合は，マルチチャンク表の再編成を実行してください。データの格納効率はチャンク単位で確認します。手順を次に示します。

手順

1. 現在のチャンクが使用しているセグメント数を確認する

   adbdbstatus コマンドで，使用量情報を出力してください。そして，Chunk_ID（チャンク ID）をキーにして，Used_segments（使用中セグメント数）を確認してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

2. 現在のチャンク内のデータ件数を基に，データ用 DB エリアのセグメント数を求める

   現在のチャンク内のデータ件数を確認したあとで，データ用 DB エリアのセグメント数を求めてください。

   現在のチャンク内のデータ件数は，adbsql コマンドのサブコマンド#GETCOUNT で確認できます。手順 1.で確認したチャンク ID を使用して，adbsql サブコマンド#GETCOUNT を実行してください。adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行）』を参照してください。

データ用 DB エリアのセグメント数を求める場合は，「5.8.1　データ用 DB エリアの総ページ数の求め方」の「(2)　変数の説明」の「(f)　変数 SGROWTBL の求め方（マルチチャンク表の場合）」を参照してください。

なお，変数 *SGROWTBL* は，データ用 DB エリアに格納されている全マルチチャンク表の全チャンクの合計を求める計算式となっています。ここでは，対象表の対象チャンクだけを求めてください。

3. 求めた結果を基に，データの格納効率を求める

手順 1.および手順 2.で求めた結果を基に，チャンク内のデータの格納効率を求めてください。計算式を次に示します。

計算式

データの格納効率＝ 手順2. の結果÷手順1. の結果

上記の計算結果が 0 に近い場合，データの格納効率が悪いため，そのチャンクを再編成してください。再編成の運用方法は幾つかあるため，「(3)　再編成方法の選択」を参照して，再編成の運用方法を決めてください。

(b)　マルチチャンク表がカラムストア表の場合

adbdbstatus コマンドを実行して**再編成要否の情報**を確認し，マルチチャンク表の再編成をするかどうかを決めてください。確認手順を次に示します。

手順

1. -d reorginfo オプションを指定したadbdbstatus コマンドを実行して，再編成要否の情報を出力する
2. 再編成要否の情報のReorganization_necessity（再編成要否）の出力内容を確認する
   - Recommended が出力されている場合

     Recommended が出力されたチャンクの再編成を実行してください。再編成の運用方法は幾つかあるため，「(3)　再編成方法の選択」を参照して，再編成の運用方法を決めてください。
   - Not_recommended が出力されている場合

     Not_recommended が出力されたチャンクは再編成する必要はありません。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

(c)　少量のデータをバックグラウンドインポートで繰り返し追加した場合

データの格納効率を計算してください。データの格納効率が悪い場合は，マルチチャンク表の再編成を実行してください。格納効率を求める計算式を次に示します。

計算式

$$\text{少量のデータをバックグラウンドインポートで繰り返し追加した場合の格納効率} = \frac{A \times B}{C \times 4{,}096}$$

上記の計算式で求めた結果が 1 に近いほど，チャンク内のデータの格納効率が良いと判断できます。逆に，0 に近いほど，チャンク内のデータの格納効率が悪いと判断できます。0 に近い場合は，マルチチャンク表の再編成を実行してください。

変数の説明

A

再編成の方法によって，代入する値が異なります。

- チャンク単位で再編成をする場合

  チャンクの使用中ページ数を代入してください。

  チャンクの使用中ページ数を確認する場合は，-d used オプションを指定したadbdbstatus コマンドを実行してください。**DB エリア，表とインデクスの使用量情報**を出力できます。**DB エリア，表とインデクスの使用量情報**から，格納効率を確認したいチャンクのChunk_ID をキーにして，Used_pages の内容を確認してください。

- 表全体の再編成をする場合

  表の使用中ページ数を代入してください。

  表の使用中ページ数を確認する場合は，adbdbstatus コマンドで，**表のサマリ情報**を出力してください。**表のサマリ情報**のUsed_pages の内容を確認してください。

B

マルチチャンク表が格納されているデータ用 DB エリアのページサイズ（単位：キロバイト）

データ用 DB エリアのページサイズを確認する場合は，-c dbarea オプションを指定したadbdbstatus コマンドを実行してください。**DB エリアのサマリ情報**を出力できます。**DB エリアのサマリ情報**から，ページサイズを確認したいデータ用 DB エリアのPage_size の内容を確認してください。

C

再編成の方法によって，代入する値が異なります。

- チャンク単位で再編成をする場合

  チャンクの使用中セグメント数を代入してください。

  チャンクの使用中セグメント数を確認する場合は，-d used オプションを指定したadbdbstatus コマンドを実行してください。**DB エリア，表とインデクスの使用量情報**を出力できます。**DB エリア，表とインデクスの使用量情報**から，格納効率を確認したいチャンクのChunk_ID をキーにして，Used_segments の内容を確認してください。

> **重要**
>
> 使用中セグメントが 1 つだけのチャンクを再編成しても，データの格納効率は向上しません。再編成対象のチャンクの使用中セグメント数が 1，または 1 に近い場合は，次のどちらかの方法でデータを再編成してください。
>
> - 複数のチャンクをマージしたあとに，「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）」の手順に従って，マージ後のチャンクを再編成する

- 「11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合）」の「(2)　表の再編成後はデータを 1 つのチャンクにまとめてもよい場合」の手順に従って，マルチチャンク表全体を再編成する

- 表全体の再編成をする場合

  表の使用中セグメント数を代入してください。

  表の使用中セグメント数を確認する場合は，adbdbstatus コマンドで，**表のサマリ情報**を出力してください。**表のサマリ情報**のUsed_segments の内容を確認してください。

(3)　再編成方法の選択

マルチチャンク表の再編成方法には，次の表に示すとおり，幾つかの方法があります。条件に合った方法を選択してください。

重要

マルチチャンク表全体を再編成した場合（特にデータ件数が多い場合），再編成に時間が掛かるおそれがあります。そのため，マルチチャンク表全体を再編成するのではなく，チャンク単位での定期的な再編成を推奨します（次の表に示す項番 1 の方法を推奨します）。

表 11-7　マルチチャンク表の再編成方法

項番	再編成の実行単位	表の検索要否，チャンクの構成	再編成の実行時に必要な条件	再編成の方法
1	チャンク単位	(1)再編成中の表を検索できる※1 (2)チャンクの構成，状態，コメントを再編成前と同じにできる	1. データ用 DB エリアの空き容量 再編成前と再編成後の両方のデータを一時的に保持する必要があります。そのため，再編成対象のチャンクのデータ容量と同じ空き容量がデータ用 DB エリア内に必要です。 例えば，1TB のデータを格納するチャンクを再編成する場合，データ用 DB エリア内の空き容量が 1TB 必要です。※2 2. 出力データファイルの出力先の容量 チャンクのデータをエクスポートする出力データファイルを格納できるディスク容量が必要です。 3. 作業用一時ファイルの出力先の容量 データをインポートする際に，一時的に作成される作業用一時ファイルを格納できるディスク容量が必要です。	「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）」の「(1)　表の検索業務を継続した状態で再編成をする場合」で説明している手順で再編成してください。
2		(1)再編成中の表は検索できない (2)チャンクの構成，状態，コメントを再	項番 1 の，2.と 3.の条件を満たす必要があります。 なお，カレントチャンクだけが再編成対象となる場合，カレントチャンクに格納されているデータ	「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）」の「(2)　表の検索業務を停止して再編成をする場合」で説明

項番	再編成の実行単位	表の検索要否，チャンクの構成	再編成の実行時に必要な条件	再編成の方法
		編成前と同じにできる	容量分の空き容量がデータ用 DB エリアに必要になります。	している手順で再編成してください。
3	表全体	(1)再編成中の表を検索できる※1 (2)チャンクの構成，状態，コメントを再編成前と同じにできる	1. データ用 DB エリアの空き容量 再編成前と再編成後の両方のデータを一時的に保持する必要があります。そのため，再編成対象の表のチャンクの最大データ容量と同じ空き容量がデータ用 DB エリア内に必要です。 例えば，表にチャンクが 3 つあり，それぞれのデータ容量が 1TB，2TB，3TB の場合，データ用 DB エリア内に 3TB の空き容量が必要になります。 2. 出力データファイルの出力先の容量 チャンク単位でエクスポートするため，チャンクのデータをエクスポートする出力データファイルを格納できるディスク容量が必要です。 3. 作業用一時ファイルの出力先の容量 チャンク単位でデータをインポートするため，インポートの際に一時的に作成される作業用一時ファイルを格納できるディスク容量が必要です。	「11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合）」の「(1) 表の再編成後もチャンクの構成を同じにしたい場合」で説明している手順で再編成してください。
4		(1)再編成中の表は検索できない (2)チャンクを 1 つにマージできる	1. 出力データファイルの出力先の容量 表の全データを一度にエクスポートします。表の全データをエクスポートする出力データファイルを格納できるディスク容量が必要です。 2. 作業用一時ファイルの出力先の容量 表の全データを一度にインポートします。表の全データをインポートする際に一時的に作成される作業用一時ファイルを格納できるディスク容量が必要です。	「11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合）」の「(2) 表の再編成後はデータを 1 つのチャンクにまとめてもよい場合」で説明している手順で再編成してください。

注※1

再編成の手順の 1 つで，PURGE CHUNK 文を実行します。PURGE CHUNK 文の実行中は，表を検索することはできません。

注※2

再編成対象のチャンクが 2 つあり，それぞれのデータ容量が 1TB，2TB の場合，データ用 DB エリア内に 2TB の空き容量が必要になります。

11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）

マルチチャンク表をチャンク単位で再編成する方法を説明します。チャンク単位の再編成は，次のどちらかの方法で行います。通常は，1.の方法でチャンク単位の再編成を実行してください。

1. 表の検索業務を継続した状態で再編成をする方法

   この方法の場合，再編成中の表を検索することができます（更新はできません）。

   ただし，再編成前と再編成後の両方のデータを一時的に保持する必要があるため，再編成対象の表を格納しているデータ用 DB エリアにその分の空き容量が必要になります。空き容量が不足している場合は，この方法は実施できません。そのため，この方法を実施する場合は，adbdbstatus コマンドで，再編成を実行するチャンクのデータ容量と，表を格納しているデータ用 DB エリアの空き容量を比較してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

   この方法の再編成手順については，「(1)　表の検索業務を継続した状態で再編成をする場合」を参照してください。

2. 表の検索業務を停止して再編成をする方法

   再編成対象の表を格納しているデータ用 DB エリアの空き容量が不足しているために 1.の方法が実施できない場合は，この方法でチャンク単位の再編成を実行してください。この方法の場合，再編成中の表を検索，更新することはできません。

   この方法の再編成手順については，「(2)　表の検索業務を停止して再編成をする場合」を参照してください。

   なお，カレントチャンクだけが再編成対象となる場合，カレントチャンクに格納されているデータ容量分の空き容量がデータ用 DB エリアに必要になります。データ用 DB エリアの空き容量が不足している場合は，「11.10.6　データ用 DB エリアの空き容量を確保する方法」を参照して，空き容量不足を解消してください。

> **メモ**
>
> マルチチャンク表がカラムストア表の場合，チャンク単位で再編成を実行するサンプルシェルを提供しています。詳細については，「11.4.16　マルチチャンク表を再編成する方法（サンプルシェルを使用して再編成する場合)」を参照してください。

(1)　表の検索業務を継続した状態で再編成をする場合

チャンク単位の再編成手順を次に示します。

手順

1. AP やコマンドが再編成対象の表を更新できない状態にする

   次の操作を実行して，AP やコマンドが再編成対象の表を更新できない状態にしてください。

- HADB サーバマシン上で動作している，再編成対象の表を更新するコマンドやジョブ（定期的に実行しているadbimport コマンドやadbmergechunk コマンドなど）を停止してください。
- HADB サーバにアクセスして再編成対象の表を更新する AP や，HADB サーバにアクセスして再編成対象の表を更新するコマンドをすべて停止してください。

上記の操作を両方とも実行することを推奨します。

2. 再編成対象のチャンクの情報を取得する

システム表STATUS_CHUNKS 表を検索し，再編成対象のチャンクについて，次に示す情報を取得してください。STATUS_CHUNKS 表を検索する方法については，「付録 C.9　システム表の検索」の「(3)　表名から表内の全チャンクの情報を調べる場合」を参照してください。

- CHUNK_COMMENT（チャンクに設定されたコメント）
- CHUNK_STATUS（チャンクの状態）

3. カレントチャンクの情報を取得する

システム表STATUS_CHUNKS 表を検索し，カレントチャンクについて次に示す情報を取得してください。STATUS_CHUNKS 表を検索する方法については，「付録 C.9　システム表の検索」の「(16)　カレントチャンクの情報を調べる場合」を参照してください。

- CHUNK_ID（チャンク ID）

  再編成対象のチャンク ID と一致しているかどうか（再編成対象のチャンクがカレントチャンクかどうか）を確認してください。
- CHUNK_COMMENT（チャンクに設定されたコメント）

4. マルチチャンク表のデータをチャンク単位にエクスポートする

-c オプションを指定したadbexport コマンドで，マルチチャンク表のデータをチャンク単位にエクスポートしてください。チャンク単位にエクスポートする方法については，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

5. エクスポートしたデータを，待機状態のチャンクとしてバックグラウンドインポートする

adbimport コマンドに次に示すオプションを指定して，手順 4.でエクスポートしたデータをバックグラウンドインポートしてください。

- -b オプション
- --status wait オプション

  --status wait オプションを指定すると，待機状態のチャンクが作成されて，そのチャンクにデータが格納されます。
- -m オプション

  手順 2.で確認したチャンクのコメントを指定してください。

待機状態のチャンクを作成するバックグラウンドインポートについては，「11.4.4　マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成）」を参照してください。

重要

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，「(3) インポートオプションの見積もり」に示す計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

6. チャンクの状態を変更する

手順 2.で調べた再編成対象のチャンクの状態が待機状態の場合，この手順は不要です。手順 7.に進んでください。

手順 2.で調べた再編成対象のチャンクの状態が通常状態の場合，-w オプションと-n オプションを両方指定して，adbchgchunkstatus コマンドを実行してください。2 種類のチャンクの状態を同時に変更してください。

- -w オプションに，手順 4.でエクスポートしたデータを管理するチャンク（再編成対象のチャンク）を指定する

  通常状態のチャンクから，待機状態のチャンクに変更します。

- -n オプションに，手順 5.でバックグラウンドインポートした待機状態のチャンクを指定する

  待機状態のチャンクから，通常状態のチャンクに変更します。

チャンクの状態を変更する方法については，「11.4.12　チャンクの状態を変更する方法」を参照してください。

7. 再編成対象のチャンクを削除する

手順 4.でエクスポートしたチャンクを，PURGE CHUNK 文で削除してください。なお，PURGE CHUNK 文は，マルチチャンク表の検索と同時に実行できません。そのため，マルチチャンク表に対する操作が存在しないタイミングで，PURGE CHUNK 文を実行してください。

PURGE CHUNK 文でチャンクを削除する方法については，「11.4.6　チャンク単位にデータを削除する方法」を参照してください。

カレントチャンクに関する留意事項

- カレントチャンクを再編成した場合，次の手順 8.と手順 9.は不要です。手順 10.に進んでください。
- カレントチャンク以外のチャンクを再編成した場合，カレントチャンクが再編成前とは変わります。カレントチャンクを再編成前と同じにしたい場合は，次の手順 8.と手順 9.を実施して，カレントチャンクを変更してください。
- カレントチャンクを再編成前と同じにする必要がない場合は，次の手順 8.と手順 9.は不要です。手順 10.に進んでください。

8. adbimport コマンドで空のチャンク（データが 0 件のチャンク）を作成する

adbimport コマンドの入力データファイルに空のファイルを指定して，-b オプションを指定してバックグラウンドインポートを実行してください。

9. チャンクをマージする

手順 8.で作成した空のチャンクと，再編成前のカレントチャンクをadbmergechunk コマンドでマージしてください。このとき，adbmergechunk コマンドの-m オプションには，手順 3.で確認した，再編成前のカレントチャンクのコメントを指定してください。

チャンクのマージが完了すると，カレントチャンクが再編成前と同じになります。

10. AP やコマンドが再編成対象の表を更新できる状態にする

次に示す方法で，AP やコマンドが再編成対象の表を更新できる状態にしてください。

- 手順 1.で，HADB サーバマシン上で動作しているコマンドやジョブを停止した場合
  停止していたコマンドやジョブを開始してください。
- 手順 1.で，HADB サーバにアクセスする AP やコマンドを停止した場合
  停止していた AP やコマンドを開始してください。

重要

- ここで説明する手順に従ってチャンク単位の再編成を実行した場合，再編成前と再編成後で，再編成対象のチャンクのチャンク ID が変わります。チャンクの構成，状態，コメントは変わりません。
- 再編成中の表に対して，更新系 SQL または表を更新するコマンド（adbimport コマンドやadbmergechunk コマンドなど）を実行しないでください。実行した場合，SQL 文やコマンドで更新した内容が，再編成後に失われるおそれがあります。

(2) 表の検索業務を停止して再編成をする場合

チャンク単位の再編成手順を次に示します。

手順

1. AP やコマンドが再編成対象の表にアクセスできない状態にする

次の操作を実行して，AP やコマンドが再編成対象の表にアクセスできない状態にしてください。

- adbchgsrvmode コマンドに--offline オプションを指定して，HADB サーバの稼働モードをオフラインモードに変更してください。さらに，HADB サーバマシン上で動作している，対象表を更新するコマンドやジョブ（定期的に実行しているadbimport コマンドやadbmergechunk コマンドなど）を停止してください。
- HADB サーバにアクセスする AP やコマンドをすべて停止してください。

上記の操作を両方とも実行することを推奨します。

2. 再編成対象のチャンクの情報を取得する

システム表STATUS_CHUNKS 表を検索し，再編成対象のチャンクについて，次に示す情報を取得してください。STATUS_CHUNKS 表を検索する方法については，「付録 C.9 システム表の検索」の「(3) 表名から表内の全チャンクの情報を調べる場合」を参照してください。

- CHUNK_COMMENT（チャンクに設定されたコメント）

- CHUNK_STATUS（チャンクの状態）

複数のチャンクを再編成する場合は，再編成対象の全チャンクの情報を取得してください。

3. カレントチャンクの情報を取得する

システム表STATUS_CHUNKS 表を検索し，カレントチャンクについて次に示す情報を取得してください。STATUS_CHUNKS 表を検索する方法については，「付録 C.9　システム表の検索」の「(16)　カレントチャンクの情報を調べる場合」を参照してください。

- CHUNK_ID（チャンク ID）

  再編成対象のチャンク ID と一致しているかどうか（再編成対象のチャンクにカレントチャンクが含まれるかどうか）を確認してください。

- CHUNK_COMMENT（チャンクに設定されたコメント）

> **重要**
>
> **この先の手順について**
>
> - 再編成対象のチャンクがカレントチャンク以外の場合
>
> 手順 4.～手順 6.の操作を各チャンクに実施してください。そのあとで，手順 7.に進んでください。
>
> - 再編成対象のチャンクがカレントチャンクとカレントチャンク以外の両方の場合
>
> ・最初に，手順 4.～手順 6.の操作をカレントチャンク以外の各チャンクに実施してください。
>
> ・次に，手順 4.～手順 6.の操作をカレントチャンクに実施してください。
>
> ・上記が完了したあとに，手順 8.に進んでください。
>
> - 再編成対象のチャンクがカレントチャンクだけの場合
>
> 手順 4. → 手順 6. → 手順 5.の順序で操作してください。そのあとで，手順 8.に進んでください。

4. マルチチャンク表のデータをチャンク単位にエクスポートする

-c オプションを指定したadbexport コマンドで，マルチチャンク表のデータをチャンク単位にエクスポートしてください。チャンク単位にエクスポートする方法については，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

5. 再編成対象のチャンクを削除する

手順 4.でデータをエクスポートした再編成対象のチャンクを，PURGE CHUNK 文で削除してください。

PURGE CHUNK 文でチャンクを削除する方法については，「11.4.6　チャンク単位にデータを削除する方法」を参照してください。

6. エクスポートしたデータを，マルチチャンク表にバックグラウンドインポートする

- 手順 2.で調べた再作成対象のチャンクの状態が通常状態の場合

  adbimport コマンドに次に示すオプションを指定して，手順 4.でエクスポートしたデータをバックグラウンドインポートしてください。

- -b オプション
- -m オプション

  手順 2.で確認したチャンクのコメントを指定してください。

バックグラウンドインポートについては，「11.4.2　マルチチャンク表にデータを格納する方法（バックグラウンドインポート）」を参照してください。

- 手順 2.で調べた再作成対象のチャンクの状態が待機状態の場合

  adbimport コマンドに次に示すオプションを指定して，手順 4.でエクスポートしたデータをバックグラウンドインポートしてください。

  - -b オプション
  - --status wait オプション

    --status wait オプションを指定すると，待機状態のチャンクが作成されて，そのチャンクにデータが格納されます。
  - -m オプション

    手順 2.で確認したチャンクのコメントを指定してください。

  待機状態のチャンクを作成するバックグラウンドインポートについては，「11.4.4　マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成）」を参照してください。

> **重要**
>
> adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，「(3) インポートオプションの見積もり」に示す計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

7. カレントチャンクを変更する

   カレントチャンクが再編成前と再編成後で異なっている状態になっています。カレントチャンクを再編成前と同じにしたい場合は，次に示す操作を実施してください。カレントチャンクを再編成前と同じにする必要がない場合は，次に示す操作は不要です。手順 8.に進んでください。

   - adbimport コマンドで空のチャンク（データが 0 件のチャンク）を作成する

     adbimport コマンドの入力データファイルに空のファイルを指定して，-b オプションを指定してバックグラウンドインポートを実行してください。
   - チャンクをマージする

     上記の操作で作成した空のチャンクと，再編成前のカレントチャンクをadbmergechunk コマンドでマージしてください。このとき，adbmergechunk コマンドの-m オプションには，手順 3.で確認した，再編成前のカレントチャンクのコメントを指定してください。

   チャンクのマージが完了すると，カレントチャンクが再編成前と同じになります。

8. AP やコマンドが再編成対象の表にアクセスできる状態にする

   次に示す方法で，AP やコマンドが再編成対象の表にアクセスできる状態にしてください。

- 手順 1.で，HADB サーバの稼働モードをオフラインモードに変更した場合

  adbchgsrvmode コマンドに--normal オプションを指定して，HADB サーバの稼働モードを通常モードに変更してください。また，HADB サーバマシン上で動作しているコマンドやジョブを停止した場合は，停止していたコマンドやジョブを開始してください。

- 手順 1.で，HADB サーバにアクセスする AP やコマンドを停止した場合

  停止していた AP やコマンドを開始してください。

重要

- ここで説明する手順に従ってチャンク単位の再編成を実行した場合，再編成前と再編成後で，再編成対象のチャンクのチャンク ID が変わります。チャンクの構成，状態，コメントは変わりません。
- 再編成中の表に対して，更新系 SQL または表を更新するコマンド（adbimport コマンドや adbmergechunk コマンドなど）を実行しないでください。実行した場合，SQL 文やコマンドで更新した内容が，再編成後に失われるおそれがあります。

(3) インポートオプションの見積もり

再編成を実行した要因によって，インポートオプションadb_import_rthd_num の見積もり方法が異なります。該当する要因の見積もり方法を参照してください。

■行の更新および削除を繰り返したことによって再編成をした場合（かつマルチチャンク表がローストア表の場合）

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，次の計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

計算式

インポートオプションadb_import_rthd_numの指定値 ≦
↑ *再編成対象のセグメント数* × *データの格納効率* ↑＋1

再編成対象のセグメント数

再編成対象のチャンクの使用中セグメント数を代入してください。

-d used オプションを指定したadbdbstatus コマンドで，**DB エリア，表とインデクスの使用量情報**を出力してください。そして，Chunk_ID（チャンク ID）をキーにしてUsed_segments（チャンクごとの使用中セグメント数）の出力内容を確認してください。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus (データベースの状態解析)』を参照してください。

表全体を再編成する場合は，表内の全チャンクの使用中セグメント数を合計してください。

データの格納効率

再編成対象のチャンクのデータの格納効率を代入してください。「11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法」の「(2)　再編成が必要かどうかを確認する方法」の「(a)　マルチチャンク表がロースト表の場合」で求めた値を代入してください。

表全体を再編成する場合は，表内の全チャンクのデータの格納効率を計算してください。

■行の追加，更新，および削除を繰り返したことによって再編成をした場合（かつマルチチャンク表がカラムストア表の場合）

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，次の計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

計算式

インポートオプションadb_import_rthd_numの指定値 ≦
↑ *再編成対象のセグメント数* × *再編成対象の行数*
÷ *カラムデータ用セグメントに格納されている行数* ↑＋1

再編成対象のセグメント数

再編成対象のチャンクの使用中のカラムデータ用セグメント数を代入してください。-d used オプションを指定したadbdbstatus コマンドで，**DB エリア，表とインデクスの使用量情報**を出力してください。そして，Chunk_ID（チャンク ID）をキーにして，Segment_type（セグメントの種類）が Column_data の行のUsed_segments（使用中セグメント数）の出力内容を確認してください。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus (データベースの状態解析)』を参照してください。

表全体を再編成する場合は，表内の全チャンクの使用中のカラムデータ用セグメント数を合計してください。

再編成対象の行数

再編成対象のチャンク内のデータ件数を代入してください。

表全体を再編成する場合は，表のデータ件数を代入してください。

adbsql コマンドの#GETCOUNT コマンドを実行すると，チャンク内のデータ件数，または表のデータ件数を求めることができます。

カラムデータ用セグメントに格納されている行数

再編成対象のチャンクにadbimport コマンドでインポートした行数，または更新行のカラム化機能によってロースト形式からカラムストア形式に変換された行数を代入してください。

adbdbstatus コマンドを実行して，**再編成要否の情報**を取得してください。再編成対象のチャンクのColumn_data_num の値が，カラムデータ用セグメントに格納されている行数です。

表全体を再編成する場合は，表内の全チャンクのColumn_data_num の値を合計してください。

■少量のデータをバックグラウンドインポートで繰り返し追加したことによって再編成をした場合

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，次の計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

計算式

インポートオプションadb_import_rthd_numの指定値 ≦
↑ *再編成対象のセグメント数* × *データの格納効率* ↑＋1

再編成対象のセグメント数

- ローストア表を再編成する場合

  再編成対象のチャンクの使用中セグメント数を代入してください。

  -d used オプションを指定したadbdbstatus コマンドで，**DB エリア，表とインデクスの使用量情報**を出力してください。そして，Chunk_ID（チャンク ID）をキーにしてUsed_segments（チャンクごとの使用中セグメント数）の出力内容を確認してください。

  adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

  表全体を再編成する場合は，表内の全チャンクの使用中セグメント数を合計してください。

- カラムストア表を再編成する場合

  再編成対象のチャンクの使用中のカラムデータ用セグメント数を代入してください。-d used オプションを指定したadbdbstatus コマンドで，**DB エリア，表とインデクスの使用量情報**を出力してください。そして，Chunk_ID（チャンク ID）をキーにして，Segment_type（セグメントの種類）がColumn_data の行のUsed_segments（使用中セグメント数）の出力内容を確認してください。

  adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

  表全体を再編成する場合は，表内の全チャンクの使用中のカラムデータ用セグメント数を合計してください。

データの格納効率

再編成対象のチャンクのデータの格納効率を代入してください。「11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法」の「(2)　再編成が必要かどうかを確認する方法」の「(c)　少量のデータをバックグラウンドインポートで繰り返し追加した場合」で求めた値を代入してください。

表全体を再編成する場合は，表内の全チャンクのデータの格納効率を計算してください。

11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合）

マルチチャンク表全体を再編成する方法を説明します。

重要

マルチチャンク表全体を再編成する場合（特にデータ件数が多い場合），再編成に時間が掛かるおそれがあります。このため，マルチチャンク表全体を再編成するのではなく，再編成が必要なチャンクだけを定期的に再編成することを推奨します。

マルチチャンク表全体の再編成は，次のどちらかの方法で行います。通常は，1.の方法でマルチチャンク表全体の再編成を実行してください。

1. 表の再編成後もチャンクの構成を同じにする方法

   この方法の場合，再編成中の表を検索することができます（更新はできません）。また，チャンクの構成，状態，およびコメントを，再編成前の状態と同じにできます。

   ただし，再編成前と再編成後の両方のデータを一時的に保持する必要があるため，再編成対象の表を格納しているデータ用 DB エリアにその分の空き容量が必要になります。空き容量が不足している場合は，この方法は実施できません。そのため，この方法を実施する場合は，adbdbstatus コマンドで，再編成を実行するチャンクのデータ容量と，表を格納しているデータ用 DB エリアの空き容量を比較してください。adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

   この方法の再編成手順については，「(1)　表の再編成後もチャンクの構成を同じにしたい場合」を参照してください。

2. 表の再編成後はデータを 1 つのチャンクにまとめる方法

   再編成対象の表を格納しているデータ用 DB エリアの空き容量が不足しているために 1.の方法が実施できない場合は，この方法で表の再編成を実行してください。この方法の場合，再編成中の表を検索，更新することはできません。また，表中の全データを 1 つのチャンクにまとめて再編成するため，チャンクの構成，状態，コメントが変わります。

   この方法の再編成手順については，「(2)　表の再編成後はデータを 1 つのチャンクにまとめてもよい場合」を参照してください。

   なお，カレントチャンクだけが再編成対象となる場合，カレントチャンクに格納されているデータ容量分の空き容量がデータ用 DB エリアに必要になります。

(1)　表の再編成後もチャンクの構成を同じにしたい場合

ここで説明する方法で表を再編成した場合，チャンクの構成，状態，およびコメントを，再編成前の状態と同じにできます。チャンク ID は，表の再編成前とは異なるチャンク ID になります。

表の再編成手順を次に示します。

手順

1. AP やコマンドが再編成対象の表を更新できない状態にする

   次の操作を実行して，AP やコマンドが再編成対象の表を更新できない状態にしてください。

   - HADB サーバマシン上で動作している，再編成対象の表を更新するコマンドやジョブ（定期的に実行しているadbimport コマンドやadbmergechunk コマンドなど）を停止してください。
   - HADB サーバにアクセスして再編成対象の表を更新する AP や，HADB サーバにアクセスして再編成対象の表を更新するコマンドをすべて停止してください。

   上記の操作を両方とも実行することを推奨します。

2. 再編成対象の表の全チャンクの情報を取得する

システム表STATUS_CHUNKS 表を SQL 文で検索し，再編成対象の表の全チャンクの情報を取得してください。次の情報を取得します。

- CHUNK_ID（チャンク ID）
- CHUNK_COMMENT（チャンクに設定されたコメント）
- CHUNK_STATUS（チャンクの状態）
- CREATE_TIME（チャンクの作成日時）
- SWAP_TIME（カレントチャンクが切り替わった日時）

上記の情報を検索する SQL 文の例については，「付録 C.9　システム表の検索」の「(3)　表名から表内の全チャンクの情報を調べる場合」を参照してください。

重要

カレントチャンクを確認しておいてください。CREATE_TIME（チャンクの作成日時）が設定されていて，かつSWAP_TIME（カレントチャンクが切り替わった日時）がナル値のチャンクが，カレントチャンクです。

3. チャンク単位に再編成を実行する

再編成対象の表の全チャンクに対して，手順 a.～手順 d.を繰り返して，チャンク単位に再編成を実行してください。

重要

カレントチャンクは最後に再編成してください。

a. チャンク単位にデータをエクスポートする

-c オプションを指定したadbexport コマンドで，チャンク単位にデータをエクスポートしてください。チャンク単位にエクスポートする方法については，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照してください。

b. エクスポートしたデータを，待機状態のチャンクとしてバックグラウンドインポートする

adbimport コマンドに次に示すオプションを指定して，手順 a.でエクスポートしたデータをバックグラウンドインポートしてください。

- -b オプション
- --status wait オプション

  --status wait オプションを指定すると，待機状態のチャンクが作成されてデータが格納されます。
- -m オプション

  手順 2.で確認したチャンクに設定されたコメントを指定してください。

待機状態のチャンクを作成するバックグラウンドインポートについては，「11.4.4　マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成）」を参照してください。

重要

adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）」の「(3)　インポートオプションの見積もり」に示す計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

c. チャンクの状態を変更する

手順 2.で調べた再編成対象のチャンクの状態が待機状態だった場合，この手順は不要です。手順 d. に進んでください。

手順 2.で調べた再編成対象のチャンクの状態が通常状態だった場合，-w オプションと-n オプションの両方を指定して，adbchgchunkstatus コマンドを実行してください。2 種類のチャンクの状態を同時に変更します。

- -w オプションには，手順 a.でエクスポートしたデータを格納しているチャンク（再編成対象のチャンク）を指定します。このチャンクを通常状態から，待機状態に変更します。
- -n オプションには，手順 b.でバックグラウンドインポートした待機状態のチャンクを指定します。このチャンクを待機状態から，通常状態に変更します。

チャンクの状態を変更する方法については，「11.4.12　チャンクの状態を変更する方法」を参照してください。

d. 再編成対象のチャンクを削除する

手順 a.でエクスポートしたチャンクのデータを，PURGE CHUNK 文で削除してください。なお，PURGE CHUNK 文は，マルチチャンク表の検索と同時に実行できません。そのため，マルチチャンク表に対する操作が存在しないタイミングで，PURGE CHUNK 文を実行してください。

PURGE CHUNK 文でチャンクを削除する方法については，「11.4.6　チャンク単位にデータを削除する方法」を参照してください。

4. AP やコマンドが再編成対象の表を更新できる状態にする

すべてのチャンクに対して手順 3.の作業が完了したあとに，次の方法で AP やコマンドが再編成対象の表を更新できる状態にしてください。

- 手順 1.で，HADB サーバマシン上で動作しているコマンドやジョブを停止した場合

  停止していたコマンドやジョブを開始してください。
- 手順 1.で，HADB サーバにアクセスする AP やコマンドを停止した場合

  停止していた AP やコマンドを開始してください。

重要

再編成中の表に対して，更新系 SQL または表を更新するコマンド（adbimport コマンドや adbmergechunk コマンドなど）を実行しないでください。実行した場合，SQL 文やコマンドで更新した内容が，再編成後に失われるおそれがあります。

(2)　表の再編成後はデータを 1 つのチャンクにまとめてもよい場合

ここで説明する方法で表を再編成した場合，再編成後のチャンクは 1 つになるため，チャンクの構成，状態，コメント，チャンク ID が変わります。

表の再編成手順を次に示します。

手順

1. AP やコマンドが再編成対象の表にアクセスできない状態にする

   次の操作を実行して，AP やコマンドが再編成対象の表にアクセスできない状態にしてください。

   - adbchgsrvmode コマンドに--offline オプションを指定して，HADB サーバの稼働モードをオフラインモードに変更してください。さらに，HADB サーバマシン上で動作している，対象表を更新するコマンドやジョブ（定期的に実行しているadbimport コマンドやadbmergechunk コマンドなど）を停止してください。
   - HADB サーバにアクセスする AP やコマンドをすべて停止してください。

   上記の操作を両方とも実行することを推奨します。

2. 表のデータをエクスポートする

   adbexport コマンドの-n オプションに再編成対象の表を指定して，表のデータをエクスポートしてください。

   adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。

3. エクスポートしたデータを作成モードでインポートする

   adbimport コマンドに-d オプションを指定して，手順 2.でエクスポートしたデータをインポートしてください。-d オプションを指定すると，作成モードでインポートされます。表の既存データをすべて削除してから，手順 2.でエクスポートしたデータがインポートされます。

   adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

   > **重要**
   >
   > adbimport コマンドを実行する際，インポートオプションadb_import_rthd_num には，「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合)」の「(3) インポートオプションの見積もり」に示す計算式を満たす値を指定してください。計算式を満たさない値を指定すると，データの格納効率が悪くなるおそれがあります。

4. AP やコマンドが再編成対象の表にアクセスできる状態にする

   次の方法で，AP やコマンドが再編成対象の表にアクセスできる状態にしてください。

   - 手順 1.で，HADB サーバの稼働モードをオフラインモードに変更した場合

adbchgsrvmode コマンドに--normal オプションを指定して，HADB サーバの稼働モードを通常モードに変更してください。また，HADB サーバマシン上で動作しているコマンドやジョブを停止した場合は，停止していたコマンドやジョブを開始してください。

- 手順 1.で，HADB サーバにアクセスする AP やコマンドを停止した場合

  停止していた AP やコマンドを開始してください。

11.4.16　マルチチャンク表を再編成する方法（サンプルシェルを使用して再編成する場合）

HADB では，マルチチャンク表のチャンクのうち，再編成が必要なチャンクをチャンク単位で再編成するサンプルシェル（$ADBDIR/sample/reorg_column_sample.sh）を提供しています。ここでは，このサンプルシェルを使用した再編成の手順と，サンプルシェルが実行する処理について説明します。

重要

- マルチチャンク表がカラムストア表である場合に限り，このサンプルシェルを使用できます。マルチチャンク表がローストア表の場合は，このサンプルシェルは使用できません。
- このサンプルシェルは，あくまでサンプルであり，動作保証はしていません。使用する前に必ずサンプルシェルの内容を確認し，必要に応じて修正してください。サンプルシェルの内容を確認しないで使用し，エラーが発生した場合は，データベースが破壊されるおそれがあります。
- サンプルシェルを使用した場合，再編成中の表にアクセスすることはできません。

(1)　サンプルシェルを使用した表の再編成手順

サンプルシェルを使用した表の再編成手順を次に示します。

手順

1. データ用 DB エリアの空き容量を確認する

   サンプルシェルを実行するには，再編成対象の表を格納しているデータ用 DB エリアに，カレントチャンクに格納されているデータ容量分の空き容量が必要になります。

   データ用 DB エリアの空き容量が不足している場合は，「11.10.6　データ用 DB エリアの空き容量を確保する方法」を参照して，空き容量不足を解消してください。

2. 作業用ディレクトリを準備する

   必ず，空のディレクトリを準備してください。HADB サーバが，表の再編成処理中に作成するファイルがこのディレクトリに格納されます。

3. サンプルシェルを実行する

   サンプルシェルのコマンドの形式を次に示します。

■サンプルシェルのコマンドの形式

```
reorg_column_sample.sh user password schema table workdir
```

表 11-8　引数に指定する内容

項番	引数	指定する内容
1	*user*	HADB ユーザの認可識別子を指定します。
2	*password*	HADB ユーザのパスワードを指定します。
3	*schema*	再編成対象のカラムストア表のスキーマ名を指定します。
4	*table*	再編成対象のカラムストア表の表名を指定します。
5	*workdir*	作業用ディレクトリを指定します。必ず，空のディレクトリを指定してください。 サンプルシェルの実行時，作業用ディレクトリ下に次のファイルが生成されます。サンプルシェルが最後まで正常に実行されると，すべてのファイルが削除されます。 • adbdbstatus コマンドの実行結果 • 出力データパスファイル • 再編成するチャンクの出力データファイル • 空のチャンク作成用の入力データファイル • 空のチャンク作成用の入力データパスファイル

例題

サンプルシェルを使用して，カラムストア表のデータをチャンク単位に再編成します。表名などは次のとおりとします。

- 認可識別子，スキーマ名：ADBUSER01
- パスワード：ADBPASSWORD
- 処理対象のカラムストア表：T1
- 作業用ディレクトリ：/home/adbmanager/work_dir

■サンプルシェルのコマンドの指定例

```
reorg_column_sample.sh ADBUSER01 ADBPASSWORD ADBUSER01 T1 /home/adbmanager/work_dir
```

重要

- サンプルシェルの内部で呼び出す各コマンドでエラーが発生した場合は，エラーの内容を必ず確認してください。そして，必要に応じて対策してから，サンプルシェルを使用してください。
- サンプルシェルのコマンドの引数の指定（ユーザ名や，表名など）中に空白を含む場合は，サンプルシェルを使用しないでください。

(2)　サンプルシェルによって実行される処理

サンプルシェルを実行すると，再編成対象の表へのアクセスを停止し，再編成が必要なすべてのチャンクを再編成します。

サンプルシェルによって実行される処理の流れを次に示します。

■処理の流れ

1. adbchgsrvmode --offline コマンドが実行されて，HADB サーバの稼働モードがオフラインモードに変更されます。
   オフラインモードの詳細については，「10.2.3　HADB サーバの稼働モード」を参照してください。
2. 現在のカレントチャンクのチャンクに設定されたコメント，チャンク ID，およびチャンクの状態が取得されます。
3. 次のadbdbstatus コマンドが実行されて，再編成が必要なチャンクのチャンク ID が取得されます。

   ```
   adbdbstatus -d reorginfo -n 処理対象のカラムストア表 -t
   ```

4. 3.で取得したチャンクのうち，カレントチャンク以外のチャンクに対して，次に示す処理 a.～処理 d.が順に繰り返されます。
   a. 再編成対象のチャンクに設定されたコメント，およびチャンクの状態が取得されます。
   b. 再編成対象のチャンクのデータがエクスポートされます。
   c. 再編成対象のチャンクのデータが削除されます。
   d. 処理 b.でエクスポートされたデータがインポートされます。このとき，処理 a.で取得されたチャンクのコメント，およびチャンクの状態に合わせてインポートが実行されます。
5. 2.で取得されたカレントチャンクが再編成対象の場合は，カレントチャンクに対して 4.の処理 a.～処理 d.が実行されます。2.で取得されたカレントチャンクが再編成対象でない場合は，空のチャンクが作成されて，カレントチャンクとマージされます。
6. adbchgsrvmode --normal コマンドが実行されて，HADB サーバの稼働モードが通常モードに変更されます。

11.4.17　チャンクをアーカイブする方法（アーカイブマルチチャンク表使用時）

アーカイブマルチチャンク表のチャンクをアーカイブする場合（チャンクをアーカイブ状態にする場合）は，adbarchivechunk コマンドを実行してください。手順を次に示します。

手順

1. アーカイブ状態にするチャンクを特定する

アーカイブレンジ列の値を使用して，アーカイブ状態にするチャンクを特定してください。
adbarchivechunk コマンドに-r オプションと-t オプションを指定して実行すると，アーカイブレンジ列の値の範囲に含まれるチャンクのチャンク ID を確認できます。

指定例

アーカイブレンジ列中の日時情報に 2015 年 1 月 1 日～2015 年 1 月 31 日が含まれているチャンクを確認します。

```
adbarchivechunk -u ADBUSER01 -p '#HelloHADB_01'
                -r '2015/01/01-2015/01/31'
                -t
                SALESLIST
```

adbarchivechunk コマンドに-r オプションと-t オプションを指定して実行すると，KFAA80245-I メッセージが出力されます。-r オプションに指定した日付の範囲に含まれるチャンクのチャンク ID を KFAA80245-I メッセージで確認できます。

2. チャンクをアーカイブ状態にする

adbarchivechunk コマンドを実行して，チャンクをアーカイブ状態にしてください。このとき，手順 1. で調べたチャンク ID を-c オプションに指定します。

指定例

チャンク ID が 5 のチャンクをアーカイブ状態にします。

```
adbarchivechunk -u ADBUSER01 -p '#HelloHADB_01'
                -z /home/adbmanager/archive_file/archive_opt_file.txt
                -c 5
                SALESLIST
```

メモ

adbarchivechunk コマンドに-c オプションではなく，-r オプションでアーカイブレンジ列の値の範囲を指定して，アーカイブ状態にするチャンクを指定することもできます。

アーカイブ状態のチャンクの圧縮率や，アーカイブファイルのファイルサイズを確認したい場合は，adbdbstatus コマンドに-c archivechunk オプションと-n オプションを指定して，アーカイブ状態のチャンクのサマリ情報を出力してください。-n オプションには，アーカイブ状態にしたチャンクがあるアーカイブマルチチャンク表の表名を指定します。そして，次に示す情報を確認してください。

- Archive_file_size（アーカイブ状態のチャンクに対応している全アーカイブファイルのファイルサイズの合計値）
- Compression_ratio（アーカイブ状態のチャンクの圧縮率）

メモ

adbarchivechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbarchivechunk（チャンクのアーカイブ)』を参照してください。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

11.4.18　チャンクをアンアーカイブする方法（アーカイブマルチチャンク表使用時）

アーカイブマルチチャンク表のチャンクをアンアーカイブする場合（チャンクのアーカイブ状態を解除する場合）は，adbunarchivechunk コマンドを実行してください。手順を次に示します。

手順

1. アーカイブ状態を解除するチャンクを特定する

   アーカイブレンジ列の値を使用して，アーカイブ状態を解除するチャンクを特定してください。adbunarchivechunk コマンドに-r オプションと-t オプションを指定して実行すると，アーカイブレンジ列の値の範囲に含まれるチャンクのチャンク ID を確認できます。

   指定例

   アーカイブレンジ列中の日時情報に 2015 年 1 月 1 日～2015 年 1 月 31 日が含まれているチャンクを確認します。

   ```
   adbunarchivechunk -u ADBUSER01 -p '#HelloHADB_01'
                     -r '2015/01/01-2015/01/31'
                     -t
                     SALESLIST
   ```

   adbunarchivechunk コマンドに-r オプションと-t オプションを指定して実行すると，KFAA80245-I メッセージが出力されます。-r オプションに指定した日付の範囲に含まれるチャンクのチャンク ID を KFAA80245-I メッセージで確認できます。

2. アーカイブ状態を解除したあとのデータ容量を確認する

   チャンクのアーカイブ状態を解除するとデータ容量が増えます。データ容量が増えても問題がないかを確認してください。アーカイブ状態を解除したあとのチャンクのデータ容量を確認する場合は，adbdbstatus コマンドに-c archivechunk オプションと-n オプションを指定して，「アーカイブ状態のチャンクのサマリ情報」を出力してください。-n オプションには，アーカイブ状態を解除するチャンクがあるアーカイブマルチチャンク表の表名を指定します。

   「アーカイブ状態のチャンクのサマリ情報」で，アーカイブ状態を解除するチャンクについて，次に示す情報を確認してください。

   - Unarchive_table_size（アーカイブ状態になる前のチャンクで，表を格納するために使用していたセグメントの容量）
   - Unarchive_index_size（アーカイブ状態になる前のチャンクで，インデクスを格納するために使用していたセグメントの容量）

上記の値の合計が，アーカイブ状態を解除したあとのチャンクのデータ容量（表データとインデクスデータ）の目安になります。

3. チャンクのアーカイブ状態を解除する

アーカイブ状態を解除したあとのデータ容量に問題がない場合は，adbunarchivechunk コマンドでチャンクのアーカイブ状態を解除してください。このとき，手順 1.で調べたチャンク ID を-c オプションに指定してください。

指定例

チャンク ID が 5 のチャンクのアーカイブ状態を解除します。

```
adbunarchivechunk -u ADBUSER01 -p '#HelloHADB_01'
                  -w /home/adbmanager/tmp
                  -z /home/adbmanager/unarchive_file/unarchive_opt_file.txt
                  -c 5
                  SALESLIST
```

メモ

adbunarchivechunk コマンドに-c オプションではなく，-r オプションでアーカイブレンジ列の値の範囲を指定して，アーカイブ状態を解除するチャンクを指定することもできます。

アーカイブ状態を解除したチャンクについて，指定した範囲のチャンクのアーカイブ状態を解除できているかを確認したい場合は，adbdbstatus コマンドに-c archivechunk オプションと-n オプションを指定して，「アーカイブ状態のチャンクのサマリ情報」を出力してください。-n オプションには，アーカイブ状態を解除したチャンクがあるアーカイブマルチチャンク表の表名を指定します。そして，次に示す情報を確認してください。

- Range_min（アーカイブレンジ列の最小値）
- Range_max（アーカイブレンジ列の最大値）

また，アーカイブ状態を解除したチャンクについて，データ容量を確認したい場合は，adbdbstatus コマンドに-d used オプション，-c table オプションおよび-n オプションを指定して，「使用量情報」を出力してください。-n オプションには，アーカイブ状態を解除したチャンクがあるアーカイブマルチチャンク表の表名を指定します。そして，次に示す情報を確認してください。

- Used_segments（使用中セグメント数）
- Used_pages（使用中ページ数）

メモ

- adbunarchivechunk コマンドを実行すると，HADB サーバはアーカイブ状態を解除するために，システムが使用する一時的なチャンクを作成します。そのため，adbunarchivechunk コマンドを実行したあとに，バックグラウンドインポートやチャンクのマージを実行しても，作成されるチャンクのチャンク ID は「前回作成されたチャンクのチャンク ID に 1 が加算された番号」にはなりません。

- adbunarchivechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbunarchivechunk（チャンクのアンアーカイブ)』を参照してください。
- adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

11.4.19　アーカイブ状態のチャンクを確認する方法

adbdbstatus コマンドを実行すると，アーカイブ状態のチャンクを確認できます。手順を次に示します。

手順

1. adbdbstatus コマンドを実行する

   adbdbstatus コマンドを実行して，「アーカイブ状態のチャンクのサマリ情報」を出力してください。

   指定例

   ```
   adbdbstatus -c archivechunk -n ADBUSER01.SALESLIST
   ```

2. adbdbstatus コマンドの実行結果を確認する

   adbdbstatus コマンドの実行結果のArchive_status を確認してください。

   Archive_status に「Archived」が表示されているチャンクは，アーカイブ状態のチャンクです。
   Archive_status に何も表示されていないチャンクは，アーカイブされていないチャンクです。

メモ

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

11.4.20　レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する方法

レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更するには，ALTER TABLE 文を実行します。手順を次に示します。

重要

カラムストア表の場合は，レギュラーマルチチャンク表からアーカイブマルチチャンク表には変更できません。

手順

1. 無効化されるビュー表の有無を確認する

レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更すると，処理対象表に依存するビュー表は無効化されます。そのため，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する前に，無効化されるビュー表があるかどうかを確認してください。無効化されるビュー表（依存するビュー表）を調べる場合は，「11.2.11　依存するビュー表を確認する方法」を参照してください。

もし，無効化されるビュー表がある場合は，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更したあとで，ビュー表の無効化を解除する必要があります。

2. データ用 DB エリア内に作成できるチャンク数を確認する

adbdbstatus コマンドで，レギュラーマルチチャンク表が格納されているデータ用 DB エリアの「DB エリアのサマリ情報」を出力してください。そして，次の項目に出力される値が 10 以上であることを確認してください。

- Creatable_Chunks（DB エリア内に作成できるチャンク数）

10 未満の場合は，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更できません。アーカイブマルチチャンク表に変更した際に，HADB サーバが自動的に定義するロケーション表とロケーション表のインデクスによって，チャンク数が使用されます。

> **メモ**
>
> adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』を参照してください。

3. アーカイブレンジ列として指定する列を検討する

レギュラーマルチチャンク表の構成列から，アーカイブレンジ列として指定する列を検討してください。アーカイブレンジ列として指定する列を検討する場合は，「5.2.5　アーカイブマルチチャンク表を定義する場合の考慮点【ロースト ア表】」の「(5)　アーカイブレンジ列の指定」を参照してください。

4. アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを確認する

手順 3.で，アーカイブレンジ列として指定する列を決めたあとで，アーカイブレンジ列に指定する列にレンジインデクスが定義されているかどうかを確認してください。レンジインデクスが定義されているかどうかを確認する場合は，「付録 B.22　ディクショナリ表の検索」の「(27)　アーカイブマルチチャンク表のアーカイブレンジ列に定義されたレンジインデクスを調べる場合」を参照してください。

5. レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する

ALTER TABLE 文で，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更してください。なお，手順 4.の確認結果によってALTER TABLE 文の指定方法が異なります。

- アーカイブレンジ列として指定する列にレンジインデクスが定義されている場合

  ALTER TABLE 文のチャンクアーカイブ指定に，「IN *DB エリア名*」を指定しないでください。

  ALTER TABLE 文の実行が完了したら，手順 7.に進んでください。

- アーカイブレンジ列として指定する列にレンジインデクスが定義されていない場合

  ALTER TABLE 文のチャンクアーカイブ指定に，「IN *DB エリア名*」を指定してください。「IN *DB エリア名*」に指定した DB エリアに，HADB サーバが自動的に定義したレンジインデクスが格納されます。レンジインデクスは，アーカイブレンジ列として指定する列に定義されます。

ALTER TABLE 文の実行が完了したら，手順 6.に進んでください。

> **メモ**
>
> ALTER TABLE 文は，マニュアル『HADB　SQL リファレンス』の『ALTER TABLE（表定義の変更）』を参照してください。

6. HADB サーバが自動的に定義したレンジインデクスを再作成する

   手順 5.で，ALTER TABLE 文のチャンクアーカイブ指定に「IN *DB エリア名*」を指定した場合は，対象のアーカイブマルチチャンク表に対して，adbidxrebuild コマンドを実行してください。adbidxrebuild コマンドを実行して，HADB サーバが自動的に定義したレンジインデクスを再作成します。

   adbidxrebuild コマンドの実行が完了したら，手順 7.に進んでください。

7. ビュー表の無効化を解除する

   処理対象表に依存するビュー表がある場合は，ビュー表の無効化を解除してください。ビュー表の無効化を解除する場合は，「11.2.8　ビュー表の無効化を解除する方法」の「(1)　ALTER TABLE 文で表の種別を変更したことによってビュー表が無効化された場合」を参照してください。

これで，レギュラーマルチチャンク表から，アーカイブマルチチャンク表への変更が完了します。

11.4.21　アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する方法

アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更するには，ALTER TABLE 文を実行します。

> **重要**
>
> アーカイブマルチチャンク表にアーカイブ状態のチャンクが存在する場合は，レギュラーマルチチャンク表に変更できません。チャンクのアーカイブ状態を解除する必要があります。
>
> なお，「アーカイブ状態で，かつ削除仕掛中のチャンク」が存在する場合は，adbunarchivechunk コマンドでチャンクのアーカイブ状態を解除できません。PURGE CHUNK 文で，対象のチャンクを削除する必要があります。

手順

1. 無効化されるビュー表の有無を確認する

   アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更すると，処理対象表に依存するビュー表は無効化されます。そのため，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する前に，無効化されるビュー表があるかどうかを確認してください。無効化されるビュー表（依存するビュー表）を調べる場合は，「11.2.11　依存するビュー表を確認する方法」を参照してください。

   もし，無効化されるビュー表がある場合は，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更したあとで，ビュー表の無効化を解除する必要があります。

2.「アーカイブ状態のチャンク」と「アーカイブ状態で，かつ削除仕掛中のチャンク」の有無を確認する

adbdbstatus コマンドで，アーカイブマルチチャンク表の「表のサマリ情報」を出力してください。そして，次の項目を確認してください（最初に，Archive_chunks を確認してください）。

- Archive_chunks（アーカイブ状態のチャンク数）

  Archive_chunks の値が0 の場合は，手順 3.に進んでください。1 以上の場合は，アーカイブ状態のチャンクが存在します。adbunarchivechunk コマンドを実行して，全チャンクのアーカイブ状態を解除してください。

  その後，adbdbstatus コマンドを再度実行し，Archive_chunks の値が0 になっていることを確認してください。0 にならない場合は，次に示すPending_delete_chunks を確認してください。

- Pending_delete_chunks（表の削除仕掛中のチャンク数）

  Pending_delete_chunks の値が 1 以上で，かつArchive_chunks の値も 1 以上の場合，「アーカイブ状態で，かつ削除仕掛中のチャンク」が存在します。この場合，PURGE CHUNK 文を実行して，「アーカイブ状態で，かつ削除仕掛中のチャンク」をすべて削除してください。

「アーカイブ状態のチャンク」や「アーカイブ状態で，かつ削除仕掛中のチャンク」が存在していると，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更できません。

メモ

- adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。
- adbunarchivechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbunarchivechunk（チャンクのアンアーカイブ)』を参照してください。
- PURGE CHUNK 文については，マニュアル『HADB　SQL リファレンス』の『PURGE CHUNK（チャンク内の全行削除)』を参照してください。

3. アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する

ALTER TABLE 文で，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更してください。

メモ

ALTER TABLE 文は，マニュアル『HADB　SQL リファレンス』の『ALTER TABLE（表定義の変更)』を参照してください。

4. ビュー表の無効化を解除する

処理対象表に依存するビュー表がある場合は，ビュー表の無効化を解除してください。ビュー表の無効化を解除する場合は，「11.2.8　ビュー表の無効化を解除する方法」の「(1)　ALTER TABLE 文で表の種別を変更したことによってビュー表が無効化された場合」を参照してください。

これで，アーカイブマルチチャンク表から，レギュラーマルチチャンク表への変更が完了します。

なお，アーカイブマルチチャンク表のアーカイブレンジ列に自動的に定義されたレンジインデクスは，レギュラーマルチチャンク表に変更しても削除されません。レンジインデクスが不要な場合は，DROP INDEX

文で削除してください。レンジインデクスを削除する場合は，「11.3.9　インデクスの削除」を参照してください。

重要

adbunarchivechunk コマンドが中断している状態のときに，ALTER TABLE 文を実行しないでください。もし，ALTER TABLE 文を実行してしまった場合は，レギュラーマルチチャンク表をアーカイブマルチチャンク表に戻す必要があります。その後，adbunarchivechunk コマンドが中断した状態を解除する必要があります。手順を次に示します。

1. ALTER TABLE 文で，レギュラーマルチチャンク表をアーカイブマルチチャンク表に戻してください。
2. 手順 1.で元に戻したアーカイブマルチチャンク表に対して，--force オプションを指定した adbunarchivechunk コマンドを実行してください。adbunarchivechunk コマンドが中断した状態を解除できます。もし，KFAA50284-E メッセージが出力されてadbunarchivechunk コマンドがエラーになった場合でも，adbunarchivechunk コマンドが中断した状態は解除されます。
3. 手順 1.で元に戻したアーカイブマルチチャンク表を，ALTER TABLE 文でレギュラーマルチチャンク表に変更してください。

11.4.22　マルチチャンク表を削除する方法

マルチチャンク表を削除する場合，DROP TABLE 文を実行してください。DROP TABLE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『DROP TABLE（表の削除）』を参照してください。

なお，アーカイブマルチチャンク表を削除した場合，次に示すファイルなども一緒に削除されます。

- アーカイブファイル
- アーカイブレンジ列に定義されたレンジインデクス
- ロケーション表
- ロケーション表に定義されたインデクス

11.4.23　バックグラウンドインポートとチャンクを意識した運用例 1（定期的にデータの追加と削除を繰り返す運用）

バックグラウンドインポートとチャンクを意識した運用例について説明します。バックグラウンドインポートを使用すると，次に示すような定期的にデータの追加と削除を繰り返す運用ができます。

ここでは，A 社の運用例を基に説明します。

■A 社の運用例

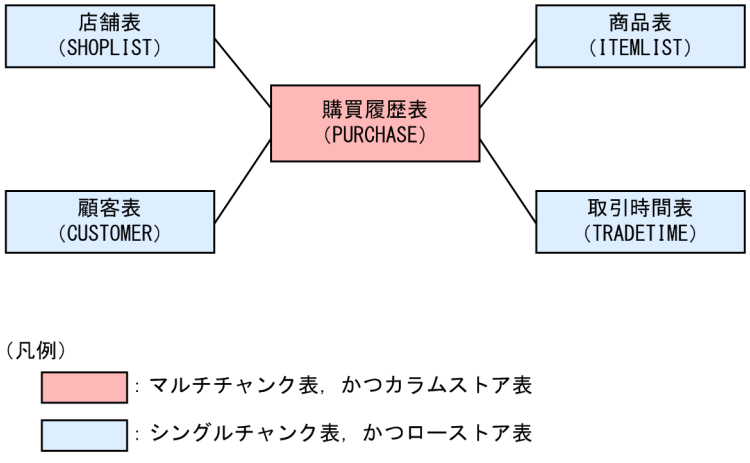
全国に約 1,000 の支店を持つ A 社では，商品の購買履歴を分析する分析システムを導入しています。A 社の分析システムのデータベースは，次のように運用されています。

- 全国にある約 1,000 の支店の購買履歴データは，分析システムのデータベースの購買履歴表で一元管理します。
- 分析システムのデータベースには，購買履歴表だけでなく，購買履歴データ以外のマスタデータを管理する店舗表，商品表なども存在します。
- 1 日に 1 回，全国の各支店から購買履歴データを収集し，まとめて購買履歴表にインポートします。
- 購買履歴表にデータをインポートしている間も，商品の購買履歴データの分析を実施します。
- 商品の購買履歴の分析では，6 か月ごと，3 か月ごと，1 か月ごと，1 週間ごと，1 日ごとの購買履歴をそれぞれ分析します。
- システムの検索性能を維持するために，直近 1 か月分の購買履歴データを毎月 1 つのチャンクにマージします（チャンク数の増加を抑えます）。
- A 社の分析業務では，過去 6 か月分の購買履歴データを必要とします。そのため，購買履歴データが格納されている購買履歴表には，過去 6 か月分のデータを格納する運用とします。
- 購買履歴表のデータ容量の増加を抑えるため，6 か月より前のデータはバックアップを取得してから削除します。
- 購買履歴表のデータのバックアップは，月単位で取得します。
- 購買履歴データの内容に誤りがある場合は，データを修正します。

■分析システムのデータベースのスキーマモデル

A 社で運用する分析システムのデータベースのスキーマモデルを，次の図に示します。

図 11-26　A 社で運用する分析システムのデータベースのスキーマモデル



[説明]

- 上記のスタースキーマの場合，購買履歴表（PURCHASE）がファクト表になります。購買履歴表は，データインポートが毎日実行され，その間も分析業務による検索で使用されます。データイ

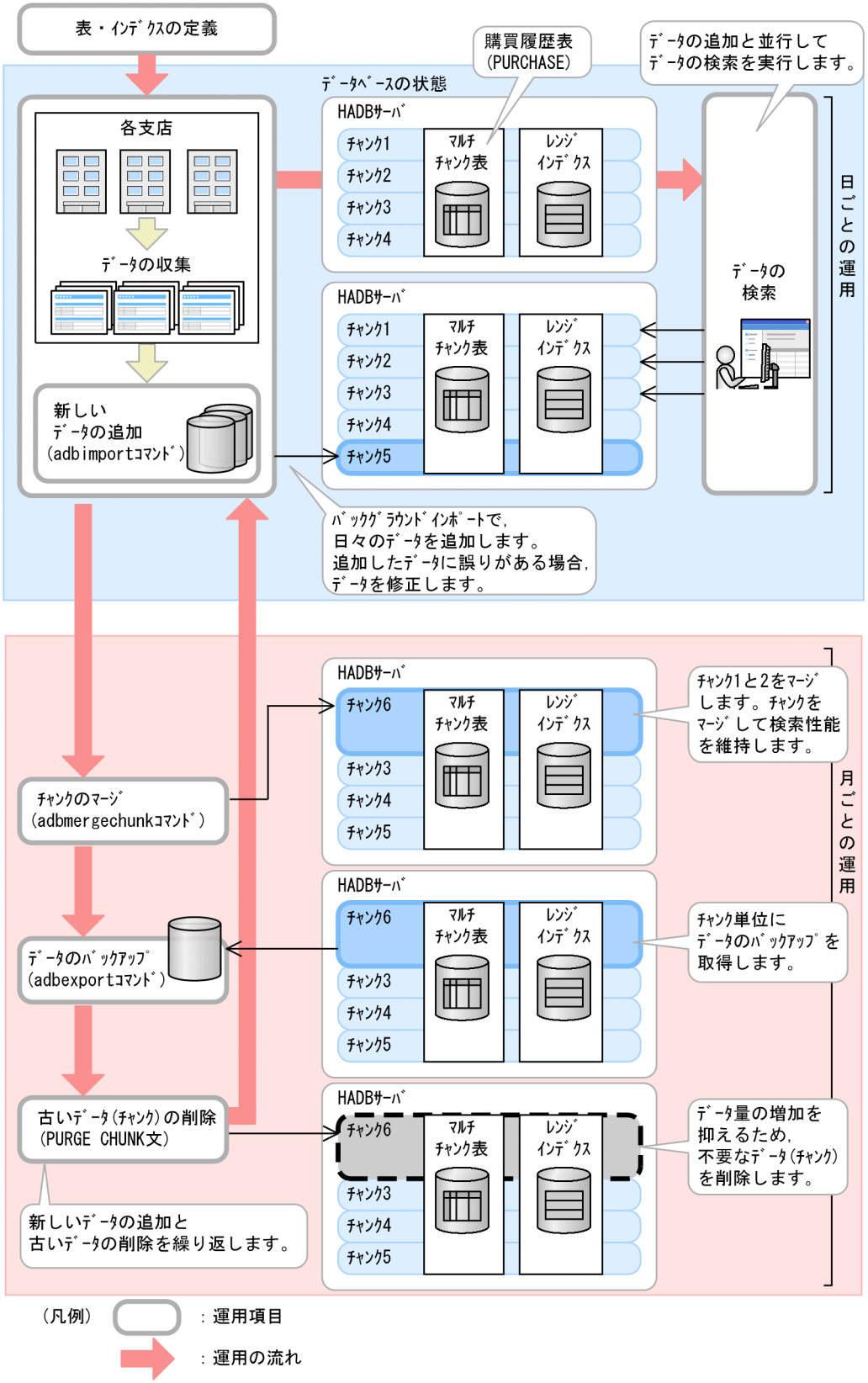
ンポートと表の検索を同時に実行するため，購買履歴表はマルチチャンク表として定義します。また，分析業務では，特定のデータに対する値の集計（平均や合計などを求める）が多いため，購買履歴表はカラムストア表として定義します。「(3)　新しいデータの追加（バックグラウンドインポート)」以降では，この購買履歴表の運用について説明しています。

- 購買履歴データ以外のマスタデータを管理するために，店舗表（SHOPLIST)，商品表（ITEMLIST)，顧客表（CUSTOMER)，および取引時間表（TRADETIME）を定義します。これらの表は，スタースキーマのディメンジョン表になります。これらの表は，シングルチャンク表かつローストア表として定義します。

■購買履歴表（PURCHASE）に関する運用の流れ

購買履歴表に関する運用の流れを，次の図に示します。

図 11-27　購買履歴表に関する運用の流れ



(1) 更新行のカラム化機能の有効化

カラムストア表である購買履歴表（PURCHASE）に対して，UPDATE 文でデータを更新することがあるため，adbcolumnize コマンドを実行して更新行のカラム化機能を有効にします。

■コマンドの実行例

```
adbcolumnize --start
```

メモ

カラムストア表に対してUPDATE 文またはINSERT 文を実行すると，更新または追加されたデータはローストア形式で格納されるため，カラムストア表の検索性能の低下の原因になります。更新行のカラム化機能を有効にすると，ローストア形式で格納されたデータをカラムストア形式に HADB が自動的に変換するため，カラムストア表の検索性能の低下を防止できます。

(2) 分析システムのデータベースの定義

各支店の購買履歴データを購買履歴表で一元管理するために，購買履歴表（PURCHASE）を定義します。また，購買履歴表にレンジインデクスを定義します。

購買履歴データ以外のマスタデータを管理するために，次の４つの表も定義します。

- 店舗表（SHOPLIST）
- 商品表（ITEMLIST）
- 顧客表（CUSTOMER）
- 取引時間表（TRADETIME）

(a) 購買履歴表（PURCHASE）の定義

購買履歴表（PURCHASE）は，次のように定義します。

- 購買履歴表に対してバックグラウンドインポートを実行するため，購買履歴表はマルチチャンク表として定義します。
- 購買履歴表はスタースキーマのファクト表とするため，購買履歴表はカラムストア表として定義します。
- 購買履歴表は，DB エリアTBLDAT_001 に格納します。
- 購買履歴表のチャンクの最大数は44 にします。見積もり方法を次に示します。

■購買履歴表のチャンクの最大数の見積もり方法

購買履歴表には，過去６か月分のデータを格納します。それを踏まえて，CREATE TABLE 文のチャンク指定に指定するチャンク数の最大値を見積もります。

1 か月を 31 日とすると，月に 31 回バックグラウンドインポートを実行するので，必要なチャンク数は 31 となります。残り 5 か月分については，検索性能を維持するために，毎月 1 つのチャンクにまとめることで，チャンク数を抑えるようにします。

安全率を 1.2 とすると，作成されるチャンク数の最大値は次の値となります。

作成されるチャンク数の最大値＝ ↑（31＋5）×1.2（安全率）↑＝ 44

メモ

チャンク数が増加すると検索性能に影響を与えます。そのため，使用するチャンク数をできるだけ抑えるように，設計・運用することを推奨します。

この見積もりによって，チャンク数の最大値には44 を設定します。この値をCREATE TABLE 文のチャンク指定のCHUNK に指定します。購買履歴表を定義するCREATE TABLE 文の指定例を次に示します。

■CREATE TABLE 文の指定例

```
CREATE TABLE "PURCHASE"
            ("SHOP_ID"     INTEGER,
             "TRADE_TIME"  TIMESTAMP,
             "ITEM_ID"     INTEGER,
             "ITEM_NAME"   VARCHAR(100),
             "CUSTOMER_ID" INTEGER,
             "PURCHASE_ID" CHAR(8))
    IN "TBLDAT_001"
    CHUNK=44                                ...1
    STORAGE FORMAT COLUMN                   ...2
```

[説明]

1. チャンク数の最大値を指定します。この指定をすると，購買履歴表がマルチチャンク表として定義されます。
2. この指定をすると，購買履歴表がカラムストア表として定義されます。

CREATE TABLE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE TABLE（表の定義)』を参照してください。

(b)　購買履歴表（PURCHASE）に対するレンジインデクスの定義

購買履歴表（PURCHASE）に対して，レンジインデクスを定義します。レンジインデクスは DB エリア RIXDAT_001 に格納するように定義します。

取引時間（TRADE_TIME 列）に対してレンジインデクスを定義します。レンジインデクスを定義するCREATE INDEX 文の指定例を次に示します。

■CREATE INDEX 文の指定例

```
CREATE INDEX "TRADE_TIME_RIX"
   ON "PURCHASE"("TRADE_TIME") IN "RIXDAT_001"
      EMPTY INDEXTYPE RANGE
```

CREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義)』を参照してください。

(c)　購買履歴表（PURCHASE）以外の表の定義

購買履歴データ以外のマスタデータを管理するために，次の表も定義します。

- 店舗表（SHOPLIST）
- 商品表（ITEMLIST）
- 顧客表（CUSTOMER）
- 取引時間表（TRADETIME）

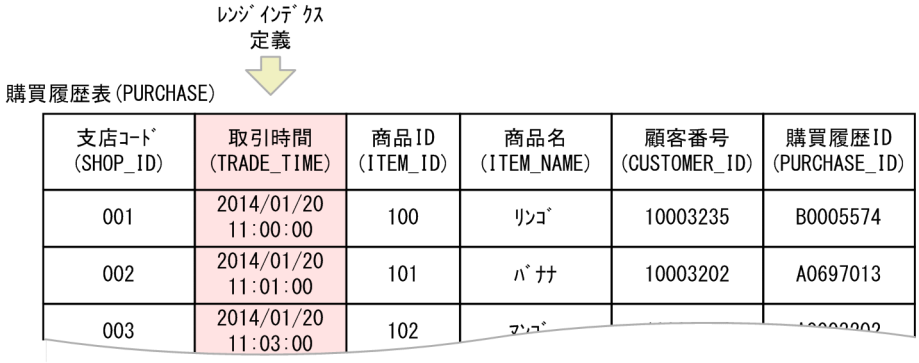
上記の表は，次のように定義します。

- 上記の表に対しては，バックグラウンドインポートを実行しないため，シングルチャンク表として定義します。
- 上記の表はスタースキーマのディメンジョン表とするため，ローストア表として定義します。
- 上記の表は，それぞれ DB エリアTBLDAT_002～TBLDAT_005 に格納するように定義します。
- 上記の表の次に示す列に対して，それぞれ B-tree インデクスを定義します。B-tree インデクスは，それぞれ DB エリアIDXDAT_002～IDXDAT_005 に格納するように定義します。
  - 店舗表（SHOPLIST）の支店コード（SHOP_ID 列）
  - 商品表（ITEMLIST）の商品 ID（ITEM_ID 列）
  - 顧客表（CUSTOMER）の顧客番号（CUSTOMER_ID 列）
  - 取引時間表（TRADETIME）の取引時間（TRADE_TIME 列）

(d)　この運用例で使用する表のイメージ

この運用例で使用する購買履歴表（PURCHASE）のイメージを，次の図に示します。

図 11-28　この運用例で使用する購買履歴表（PURCHASE）のイメージ



(3)　新しいデータの追加（バックグラウンドインポート）

1 日に 1 回，全国の各支店から収集した新しい購買履歴データを，購買履歴表（PURCHASE）に追加します。

新しいデータの追加は，バックグラウンドインポートで実行します。1 回のバックグラウンドインポートでマルチチャンク表に格納したデータ，およびそのインデクスは，1 つの固まり（チャンク）として管理されます。チャンクについては，「2.14.2　データインポート単位のデータ管理（チャンク）」を参照してください。

> **メモ**
>
> データをチャンク単位で管理することで，チャンク単位でのバックアップの取得や，データの削除ができます。バックグラウンドインポートやチャンクのマージの際に，チャンクに対してコメントを設定できます。コメントを設定しておくと，管理対象とするチャンクを特定しやすくなります。

バックグラウンドインポートには，-b オプションを指定したadbimport コマンドを使用します。adbimport コマンドの実行例を次に示します。

■コマンドの実行例

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file01.txt
          -b -m 'January 2014_daily'
          PURCHASE
          /home/adbmanager/imp_file/imp_data_path01.txt
```

[説明]

購買履歴表（PURCHASE）に対して，入力データパスファイル（/home/adbmanager/imp_file/imp_data_path01.txt）に設定した入力データファイルの内容をバックグラウンドインポートします。

adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。また，「11.4.2　マルチチャンク表にデータを格納する方法（バックグラウンドインポート)」も参照してください。

(4)　バックグラウンドインポートしたデータの修正

「(3)　新しいデータの追加（バックグラウンドインポート)」でバックグラウンドインポートしたデータの内容に誤りがあった場合，データの修正量が少ないときは，UPDATE 文またはDELETE 文でデータを修正してください。UPDATE 文でデータを 1 件修正する例を次に示します。

■SQL 文の実行例

```
UPDATE "PURCHASE" SET "ITEM_ID"=103 WHERE "PURCHASE_ID"='A0015309'
```

[説明]

購買履歴表（PURCHASE）に対して，購買履歴 ID（PURCHASE_ID）がA0015309 のデータの，商品 ID（ITEM_ID）を103 に更新します。

UPDATE 文については，マニュアル『HADB　SQL リファレンス』の『UPDATE（行の更新）』を参照してください。

なお，大量のデータ修正が必要な場合は，入力データファイルの内容を修正し，adbimport コマンドを実行して，修正後のデータに入れ替えることを推奨します。修正後のデータに入れ替える手順については，「11.4.12　チャンクの状態を変更する方法」の「(4)　運用例（チャンク内のデータを入れ替える）」を参照してください。

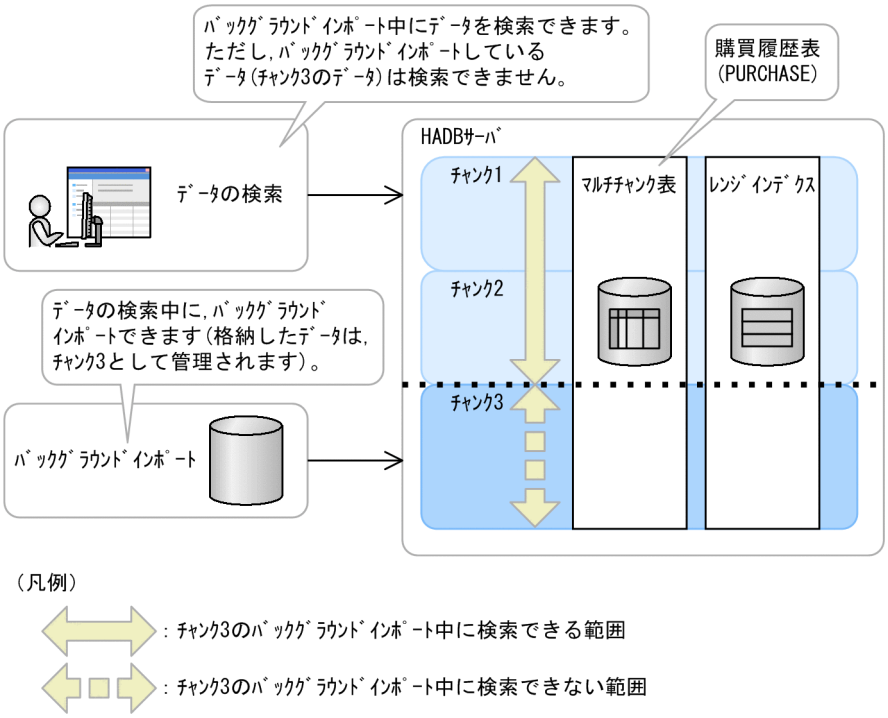
(5)　データの検索

購買履歴表（PURCHASE）に対するバックグラウンドインポートの実行中も，購買履歴表中のデータを検索できます。

バックグラウンドインポートによってデータを追加することで，インポートの対象となるデータベースに対して検索を実行しているユーザが存在しても，データのインポートを実行できます（検索と並行してデータのインポートを実行できます）。

バックグラウンドインポートとデータの検索の関係を次の図に示します。

図 11-29　バックグラウンドインポートとデータの検索の関係

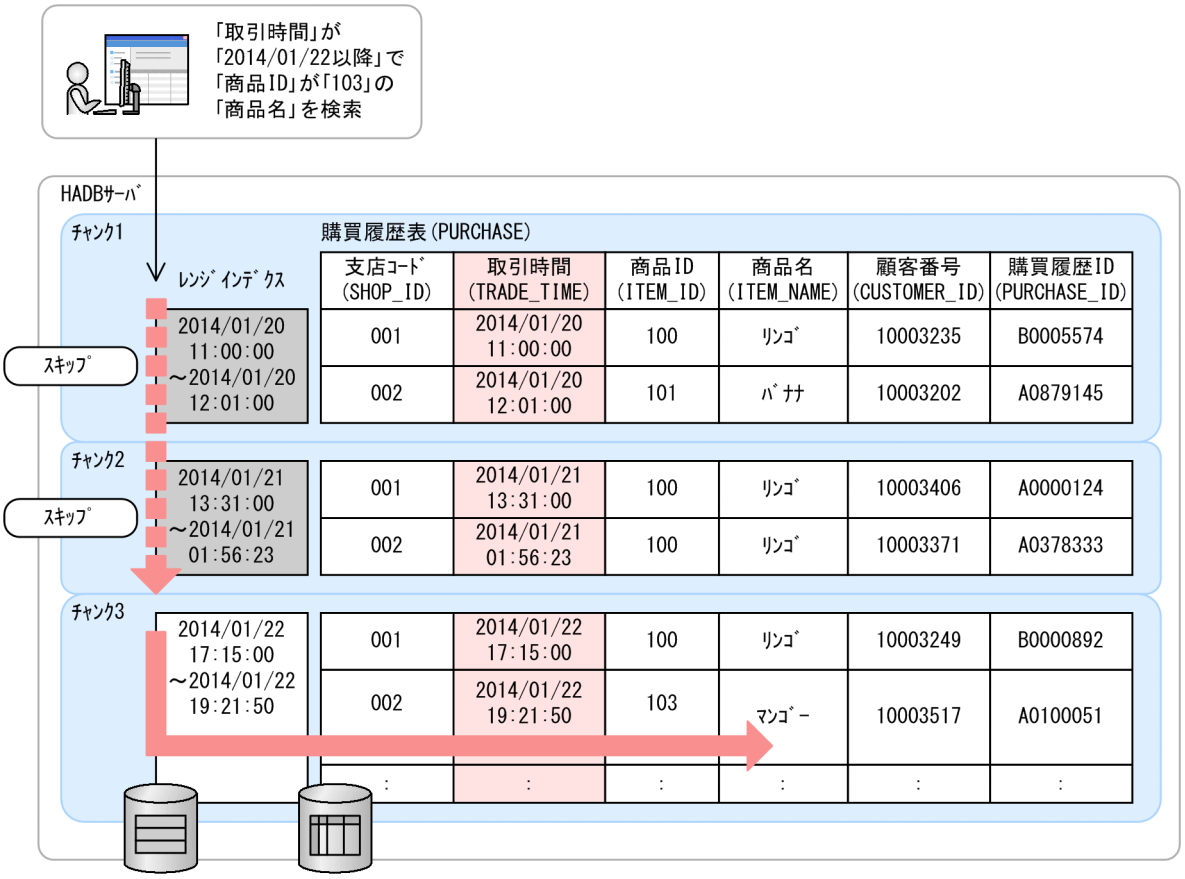


購買履歴表に格納されている購買履歴データを，半年間，3 か月間，1 か月間，1 週間，1 日単位で分析します。

マルチチャンク表である購買履歴表にレンジインデクスが定義されている場合，レンジインデクスはキーとなるデータの，チャンク内での最大値・最小値（値域）を保持しています。検索時には探索条件に含まれないデータを格納しているチャンクがスキップされるため，効率良く検索できます。

購買履歴表にレンジインデクスが定義されている場合の処理を次の図に示します。

図 11-30　購買履歴表にレンジインデクスが定義されている場合の処理



[説明]

この例の場合，取引時間（TRADE_TIME 列）に対してレンジインデクスが定義されています。そのため，取引時間で検索する際に，探索条件に含まれない取引時間のデータが格納されているチャンクの検索はスキップされます。

(6)　チャンクのマージ

購買履歴表（PURCHASE）中の月ごとのデータを 1 つのチャンクにまとめることで，チャンク数の増加による検索性能の低下を回避します。この例では，購買履歴表中の 1 か月分（2014 年 1 月分）のデータを 1 つのチャンクにまとめます。

(a)　マージ対象とするチャンクの特定

マージ対象とするチャンクのチャンク ID を特定するには，次に示すように，システム表STATUS_CHUNKS を検索します。

SQL 文の例を次に示します。

■SQL 文の例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='PURCHASE'
      AND "CHUNK_COMMENT"='January 2014_daily'
```

■検索結果

```
CHUNK_ID
--------------------
                   1
             （中略）
                  31
```

[説明]

この例の場合は，バックグラウンドインポート時に付加したチャンクのコメント（'January 2014_daily'）をキーにSTATUS_CHUNKS 表を検索することで，マージ対象とするチャンクのチャンク ID（1～31）を特定しています。

対象となるチャンクと，そのチャンク ID については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」を参照して確認してください。

(b)　チャンクの再編成要否の確認

「(a)　マージ対象とするチャンクの特定」で確認したマージ対象のチャンクに対して，チャンクの再編成が必要かどうかを確認します。確認するには，adbdbstatus コマンドで**再編成要否の情報**を取得します。

コマンドの実行例を次に示します。

■コマンドの実行例

```
adbdbstatus -d reorginfo -n ADBUSER01.PURCHASE -c 1-31
```

[説明]

この例の場合は，マージ対象とするチャンクのチャンク ID（1～31）の再編成要否の情報を取得しています。

再編成要否の情報（Reorganization_necessity）はチャンクごとに出力されます。
Reorganization_necessity がRecommended の場合，そのチャンクの再編成を実行してください。チャンクの再編成手順については，「(c)　チャンクの再編成」を参照してください。

チャンクの再編成後，「(d)　チャンクのマージ」に示す方法で 1 つのチャンクにマージします。

(c)　チャンクの再編成

「(b)　チャンクの再編成要否の確認」で，再編成要否の情報（Reorganization_necessity）がRecommended のチャンクがある場合，そのチャンクの再編成を実行します。ここでは，チャンク ID が2 のチャンクが再編成の対象となった場合を例に，チャンクを再編成する手順を説明します。

■チャンクを再編成する手順

1. 購買履歴表（PURCHASE）を更新するコマンド，ジョブ，および AP を停止する

2. システム表STATUS_CHUNK を検索し，チャンク ID が2 のチャンクの状態とコメントを確認する

■SQL 文の実行例

```
SELECT CHUNK_ID, CHUNK_COMMENT, CHUNK_STATUS
  FROM "MASTER"."STATUS_CHUNKS"
    WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='PURCHASE'
      AND "CHUNK_ID"=2
```

■SQL 文の実行結果

```
CHUNK_ID    CHUNK_COMMENT CHUNK_STATUS
----------- ------------- ----------------------
          2 daily         Normal
```

3. adbexport コマンドで，チャンク ID が2 のチャンクのデータを出力する

エクスポートオプションファイル（/home/adbmanager/exp_file/exp_opt_file01.txt），出力データパスファイル（/home/adbmanager/exp_file/exp_filepath.txt）を，それぞれ指定して実行します。

■コマンドの実行例

```
adbexport -u ADBUSER01 -p '#HelloHADB_01'
          -z /home/adbmanager/exp_file/exp_opt_file01.txt
          -n PURCHASE
          -c 2
          /home/adbmanager/exp_file/exp_filepath.txt
```

■出力データパスファイル（/home/adbmanager/exp_file/exp_filepath.txt）の内容

```
/home/adbmanager/exp_file/purchase_data0.csv
/home/adbmanager/exp_file/purchase_data1.csv
/home/adbmanager/exp_file/purchase_data2.csv
/home/adbmanager/exp_file/purchase_data3.csv
/home/adbmanager/exp_file/purchase_data4.csv
/home/adbmanager/exp_file/purchase_data5.csv
/home/adbmanager/exp_file/purchase_data6.csv
/home/adbmanager/exp_file/purchase_data7.csv
/home/adbmanager/exp_file/purchase_data8.csv
/home/adbmanager/exp_file/purchase_data9.csv
```

4. adbimport コマンドで，出力したデータをインポートする

インポートオプションファイル（/home/adbmanager/imp_file/imp_opt_file01.txt），入力データパスファイル（/home/adbmanager/exp_file/exp_filepath.txt）を，それぞれ指定して実行します。

入力データパスファイルには，adbexport コマンドの実行で使用した出力データパスファイルをそのまま使用します。また，-m オプションには，手順 2.で確認したチャンクのコメント（daily）を指定します。

■コマンドの実行例

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file01.txt
```

```
          -b --status wait
          -m 'daily'
          PURCHASE
          /home/adbmanager/exp_file/exp_filepath.txt
```

5. adbchgchunkstatus コマンドで，チャンクの状態を変更する

   再編成の対象のチャンク（チャンク ID は2）と，手順 4.で作成したチャンク（チャンク ID は32）の状態を変更します。

   ■コマンドの実行例

```
adbchgchunkstatus -u ADBUSER01 -p '#HelloHADB_01'
                  -w 2 -n 32 PURCHASE
```

   上記のコマンドを実行すると，チャンク ID が2 のチャンクは待機状態になり，チャンク ID が32 のチャンクは通常状態になります。

6. PURGE CHUNK 文で，再編成の対象のチャンク（チャンク ID は2）を削除する

   ■SQL 文の実行例

```
PURGE CHUNK PURCHASE WHERE CHUNKID=2
```

7. 購買履歴表（PURCHASE）を更新するコマンド，ジョブ，および AP を開始する

(d) チャンクのマージ

チャンクのマージには，adbmergechunk コマンドを使用します。adbmergechunk コマンドの実行例を次に示します。

■コマンドの実行例

```
adbmergechunk -u ADBUSER01 -p '#HelloHADB_01' -g 2
              -w /home/adbmanager/tmp
              -z /home/adbmanager/merge_file/merge_opt_file01.txt
              -m 'January 2014'
              -c 1-31
              PURCHASE
```

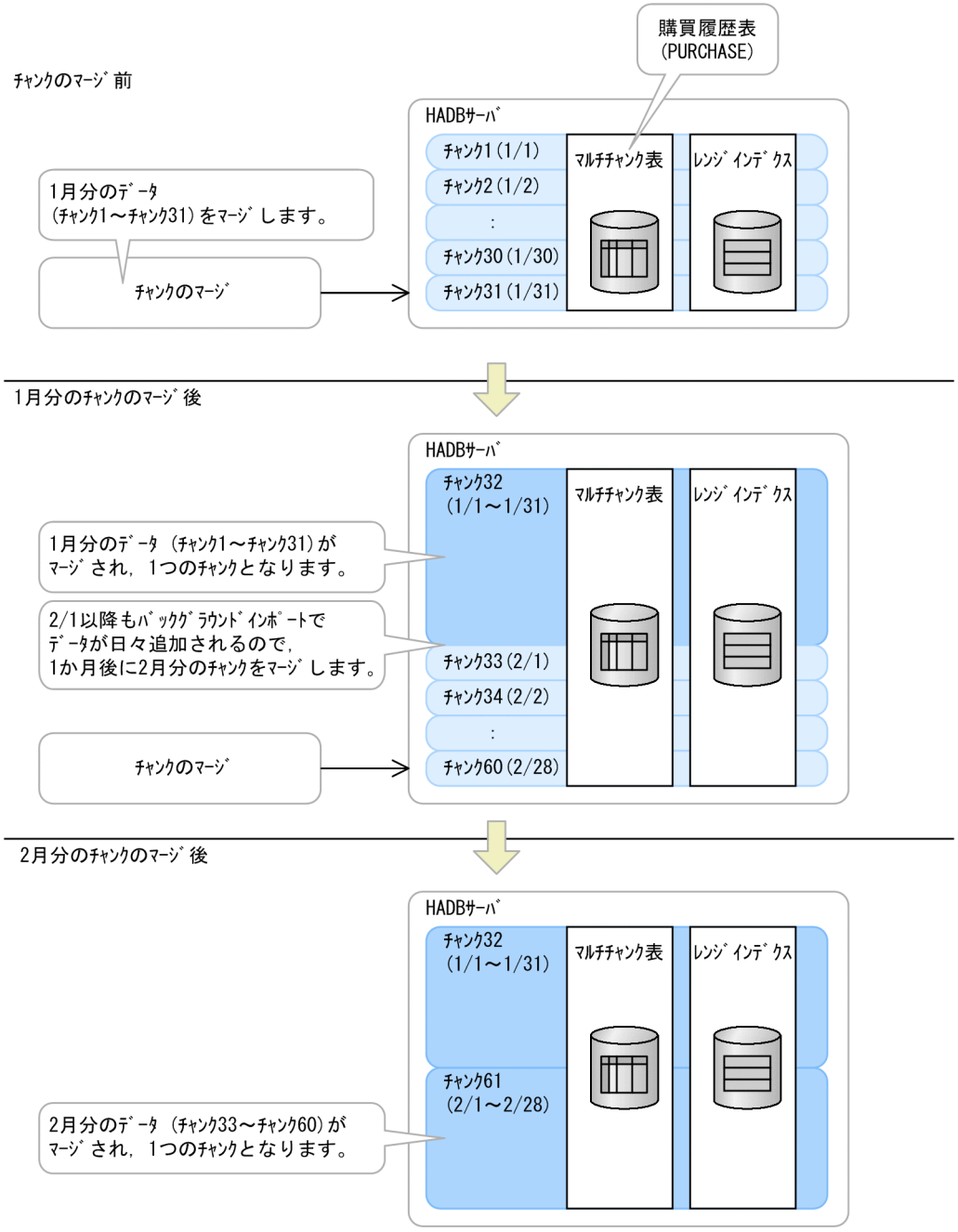
[説明]

購買履歴表（PURCHASE）に対してマージチャンクオプションファイル（/home/adbmanager/merge_file/merge_opt_file01.txt）に設定した値を入力情報として，-c オプションに指定したチャンクのマージを実行します。

この例では，1 から31 のチャンク ID のチャンクを 1 つのチャンクにマージすることで，2014 年 1 月分のデータとしてまとめています。

チャンクのマージの例を次の図に示します。

図 11-31　チャンクのマージの例



チャンク ID1 から31 のチャンクは，adbmergechunk コマンドの正常終了時に自動的に削除されます。

メモ

チャンクをマージすることで検索性能を向上させることができます。また，1 表，1DB エリアが管理できるチャンク数には限りがあります。チャンク数が足りない場合には，複数のチャンクを 1 つにマージすることで，使用するチャンク数を削減できます。

マージ対象のチャンクに対して検索を実行しているユーザが存在しても，マージチャンクは実行できます。

adbmergechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmergechunk (チャンクのマージ)』を参照してください。また，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法)」も参照してください。

(e)　マージしたチャンクの再編成要否の確認

2014 年 1 月分のデータを 1 つのチャンクにマージしたあとに，adbdbstatus コマンドを実行して，このチャンクに対する再編成要否の情報（Reorganization_necessity）を確認します。手順を次に示します。

手順

1. マージしたチャンクのチャンク ID を確認する

■SQL 文の実行例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
    WHERE "TABLE_SCHEMA"='ADBUSER' AND "TABLE_NAME"='PURCHASE'
      AND "CHUNK_COMMENT"='January 2014'
```

■SQL 文の実行結果

```
CHUNK_ID
----------------
              32
```

[説明]

2014 年 1 月分のデータとしてマージされたチャンクのチャンク ID は，上記の SQL 文を実行すると確認できます。この例の場合，マージされたチャンクのチャンク ID が 32 であることがわかります。

2. 再編成要否の情報を確認する

■コマンドの実行例

```
adbdbstatus -d reorginfo -n ADBUSER01.PURCHASE -c 32
```

チャンク ID が 32 のチャンクの再編成要否の情報（Reorganization_necessity）を確認します。Reorganization_necessity がRecommended の場合，このチャンクの再編成を実行してください。

(7)　データのバックアップ

購買履歴表（PURCHASE）のデータ容量の増加を抑えるために，購買履歴表には過去 6 か月分のデータを格納する運用にしています。6 か月より前のデータはバックアップを取得してから，削除します。バックアップの取得は，1 か月に 1 度実行します。

具体的な手順について，次に示します。

(a)　作成日時が最も古いチャンクの特定

購買履歴表（PURCHASE）で 6 か月より前のチャンク（作成した日時が最も古いチャンク）を特定します。購買履歴表の最もチャンクの作成日時が古いチャンクを特定するには，次に示すように，システム表 STATUS_CHUNKS を検索します。

SQL 文の例を次に示します。

■SQL 文の例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='PURCHASE'
      AND "CREATE_TIME"=(SELECT MIN("CREATE_TIME")
         FROM "MASTER"."STATUS_CHUNKS"
            WHERE TC."TABLE_SCHEMA"='ADBUSER01'
               AND TC."TABLE_NAME"='PURCHASE')
```

■検索結果

```
 CHUNK_ID
 --------------------
                   32
KFAA96404-I 1 rows were selected.
```

[説明]

STATUS_CHUNKS 表を検索した結果，この例の場合はチャンク ID が32 のチャンクが，作成日時が最も古いチャンク（6 か月より前のチャンク）であることがわかります。

> **メモ**
>
> チャンクをマージした場合は，STATUS_CHUNKS 表には，マージ元チャンクのうち，最も古い「チャンクの作成日時」が格納されます。

STATUS_CHUNKS 表の検索方法については，「付録 C.9　システム表の検索」を参照してください。

(b)　バックアップの取得

購買履歴表（PURCHASE）で最もチャンクの作成日時が古いチャンクのチャンク ID が32 であるとわかったので，そのチャンクに含まれるデータのバックアップを CSV 形式のファイルで取得します。

バックアップの取得には，adbexport コマンドを使用します。adbexport コマンドの実行例を次に示します。

■コマンドの実行例

```
adbexport -u ADBUSER01 -p '#HelloHADB_01'
          -z /home/adbmanager/exp_file/exp_opt_file01.txt
          -n PURCHASE
          -c 32
          /home/adbmanager/exp_file/exp_data_path01.txt
```

■出力データパスファイル（/home/adbmanager/exp_file/exp_data_path01.txt）の内容

```
/home/adbmanager/exp_backup/backup0001.csv
/home/adbmanager/exp_backup/backup0002.csv
/home/adbmanager/exp_backup/backup0003.csv
```

[説明]

チャンク ID が32 のチャンクに含まれるデータのバックアップを CSV 形式のファイル（/home/adbmanager/exp_backup/backup0001.csv，/home/adbmanager/exp_backup/backup0002.csv，/home/adbmanager/exp_backup/backup0003.csv）に取得します。

この例の場合，月ごとにチャンクがマージされているので，1 か月分のデータのバックアップになります。

> **メモ**
>
> バックアップを取得するファイルは 1 つにすることもできますが，複数のファイルに分散することで，出力にかかるオーバヘッドを分散させ性能を向上させることができます。

adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。また，「11.4.5　チャンク単位にデータをエクスポートする方法」も参照してください。

(8)　古いデータ（チャンク）の削除

購買履歴表（PURCHASE）のデータ容量の増加を抑えるために，6 か月より前のデータを削除します。

購買履歴表（PURCHASE）から作成日時が最も古いチャンクを削除します。チャンクを削除するには，PURGE CHUNK 文の探索条件にチャンク ID を指定して実行します。

■SQL 文の例

```
PURGE CHUNK ”PURCHASE” WHERE CHUNKID=32
```

[説明]

チャンク ID が32 のチャンクに含まれるデータを削除します。

この例の場合，月ごとにチャンクがマージされているので，1 か月分のデータが削除されます。

PURGE CHUNK 文については，マニュアル『HADB　SQL リファレンス』の『操作系 SQL』の『PURGE CHUNK（チャンク内の全行削除)』を参照してください。また，「11.4.6　チャンク単位にデータを削除する方法」も参照してください。

(9)　新しいデータの追加・マージと古いデータの削除の繰り返し

バックグラウンドインポートの実行とチャンクのマージ，およびPURGE CHUNK 文を利用したチャンクの削除を繰り返すことで，HADB が管理するデータ量を増やすことなく，購買履歴を分析する運用が継続できます。

11.4.24　バックグラウンドインポートとチャンクを意識した運用例 2（待機状態のチャンクを利用した運用）

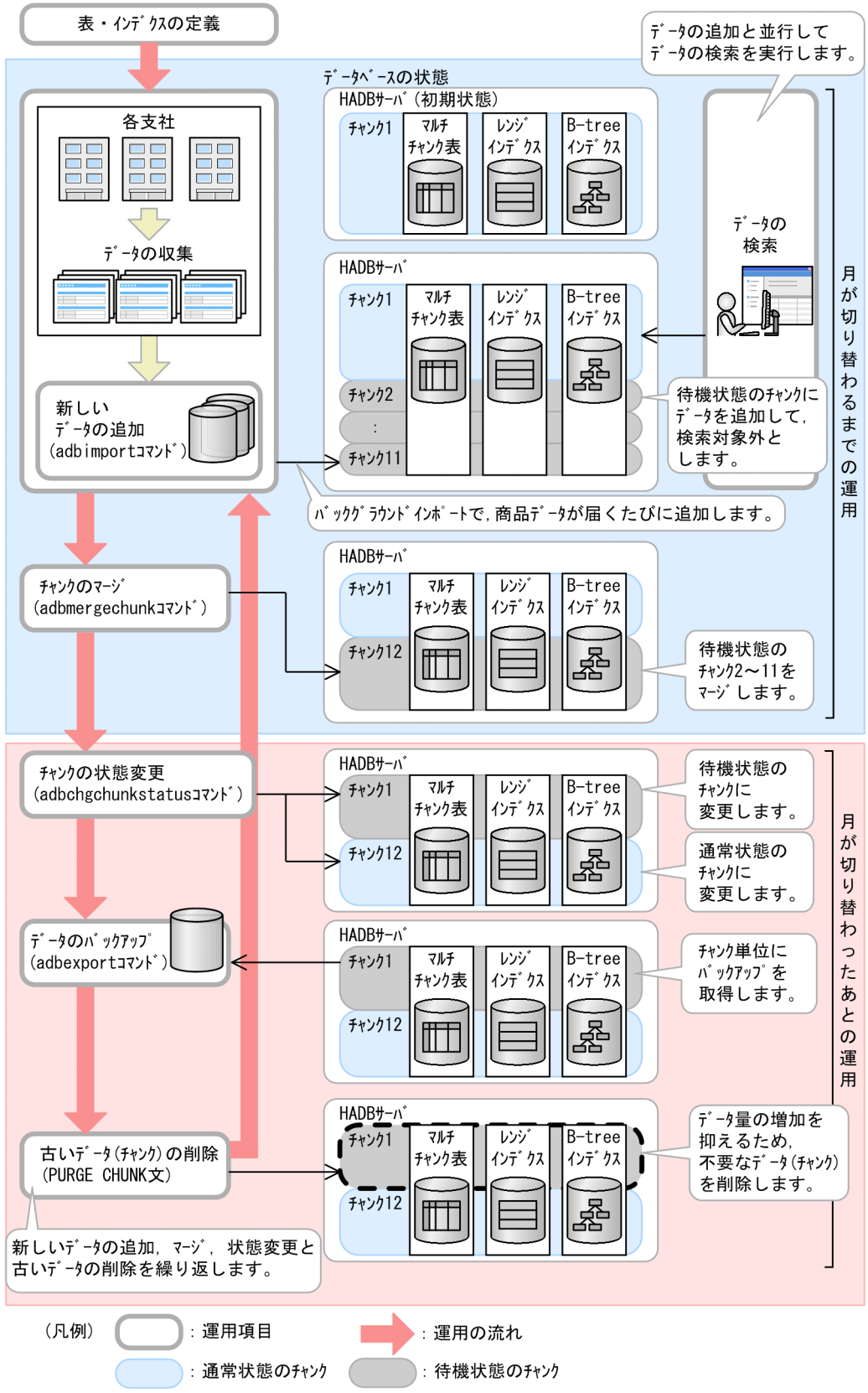
待機状態のチャンクを利用したバックグラウンドインポートの運用例について説明します。待機状態のチャンクを利用すると，次に示すような任意のタイミングでデータを検索対象にしたり，検索対象外にしたりする運用ができます。

ここでは次のようなシステムでの運用を例に説明します。

- 全国の各店舗が取り扱う商品データを 10 の支社から本社が収集し，取り扱う商品データを使用した分析をします。
- 本社が収集した商品データは，商品表で一元管理します。
- 商品データは毎月変更されるため，翌月の商品データは各支社から毎月 25 日までに本社に送付されます。
- 本社に翌月の商品データが届くたびに，データベースにインポートします。
- インポートした翌月の商品データは，月が切り替わるまで，分析対象外にする必要があります。
- データベースに翌月の商品データをインポートしている間も，今月の商品データを分析します。
- 月が切り替わると，分析対象外としていた商品データを分析対象にします。また，今まで分析対象としていた商品データを分析対象外とします。
- データベースには，分析対象外とした商品データを 6 か月間保持します。
- データ容量の増加を抑えるため，6 か月より前の商品データはバックアップを取得してから削除します。

また，このシステムでの運用の流れを次の図に示します。

図 11-32　このシステムでの運用の流れ（待機状態のチャンクの利用）



(1) マルチチャンク表とインデクスの定義

商品データを一元管理するために，商品表（ITEMLIST）およびインデクスを定義します。

(a) 商品表（ITEMLIST）の定義

商品表（ITEMLIST）は，DB エリアTBLDAT_002 に格納するように定義します。また，バックグラウンドインポートを実施するために，商品表をマルチチャンク表として定義します。

この運用例では，次に示す数のチャンクが必要となります。

- 分析対象となる今月の商品データを含むチャンク（1 個）
- 各支社から送付された商品データを含むチャンク（11 個）
  支社の数が 10 あるため，各支社のデータをインポートするチャンクが 10 個必要になります。また，それら 10 個のチャンクをマージするためのチャンクが 1 個必要になります。よって，合計 11 個のチャンクが必要になります。
- 分析対象外の商品データ 6 か月分を含むチャンク（6 個）

このため，安全率を 1.2 とすると，作成されるチャンク数の最大値は次の値となります。

作成されるチャンク数の最大値＝ ↑（1＋11＋6）×1.2（安全率）↑＝ 22

メモ

チャンク数が増加すると検索性能に影響を与えます。そのため，使用するチャンク数をできるだけ抑えるように，設計・運用することを推奨します。

この見積もりによって，チャンク数の最大値には22 を設定します。この値をCREATE TABLE 文のチャンク指定のCHUNK に指定します。SQL 文の例を次に示します。

■SQL 文の例

```
CREATE TABLE "ITEMLIST"
      ("SHOP_ID" INT,"DELIVERY_TIME" TIMESTAMP,
       "ITEM_ID" INT, "ITEM_NAME" VARCHAR(100))
    IN "TBLDAT002" CHUNK=22
```

CREATE TABLE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE TABLE（表の定義)』を参照してください。

(b) 商品表（ITEMLIST）に対するインデクスの定義

商品表に対して，B-tree インデクスとレンジインデクスを定義します。インデクスはそれぞれ DB エリア IDXDAT_002，RIXDAT_002 に格納するように定義します。

商品 ID（ITEM_ID 列）に対して B-tree インデクスを定義し，納品時間（DELIVERY_TIME 列）に対してレンジインデクスを定義します。SQL 文の例を次に示します。

■SQL 文の例

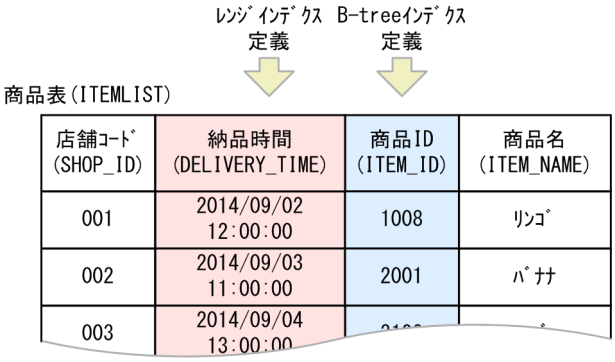
```
CREATE INDEX "ITEM_ID_IDX"
   ON "ITEMLIST"("ITEM_ID") IN "IDXDAT_002"
      EMPTY

CREATE INDEX "DELIVERY_TIME_RIX"
   ON "ITEMLIST"("DELIVERY_TIME") IN "RIXDAT_002"
      EMPTY INDEXTYPE RANGE
```

CREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義）』を参照してください。

(c)　この運用例で使用する表のイメージ

この運用例で使用する表のイメージを次の図に示します。

図 11-33　この運用例で使用する表のイメージ



(2)　翌月の商品データを待機状態のチャンクに追加（バックグラウンドインポート）

翌月の商品データは，各支社から毎月 25 日までに本社に送付されます。本社に翌月の商品データが届くたびに，商品表（ITEMLIST）に追加します。

翌月の商品データの追加は，バックグラウンドインポートで実行します。その際，翌月の商品データは，月が切り替わるまで分析対象外にする必要があります。そのため，待機状態のチャンクとして追加します。

1 回のバックグラウンドインポートでマルチチャンク表に格納したデータ，およびそのインデクスは，1 つの固まり（チャンク）として管理されます。チャンクについては，「2.14.2　データインポート単位のデータ管理（チャンク）」を参照してください。

> **メモ**
>
> データをチャンク単位で管理することで，チャンク単位でのバックアップの取得や，データの削除ができます。バックグラウンドインポートやチャンクのマージの際に，チャンクに対してコメントを設定できます。コメントを設定しておくと，管理対象とするチャンクを特定しやすくなります。

待機状態のチャンクを作成するバックグラウンドインポートの場合，-b オプションおよび--status wait オプションを指定したadbimport コマンドを使用します。adbimport コマンドの実行例を次に示します。

■コマンドの実行例

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file02.txt
          -b --status wait
          -m 'October 2014'
          ITEMLIST
          /home/adbmanager/imp_file/imp_data_path02.txt
```

[説明]

商品表（ITEMLIST）に対して，入力データパスファイル（/home/adbmanager/imp_file/imp_data_path02.txt）に設定した入力データファイルの内容をバックグラウンドインポートします。

10 の支社から翌月の商品データが送付されるたびに，バックグラウンドインポートを実行します。その結果，待機状態のチャンクが 10 個作成されます。

adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。また，「11.4.4　マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成)」も参照してください。

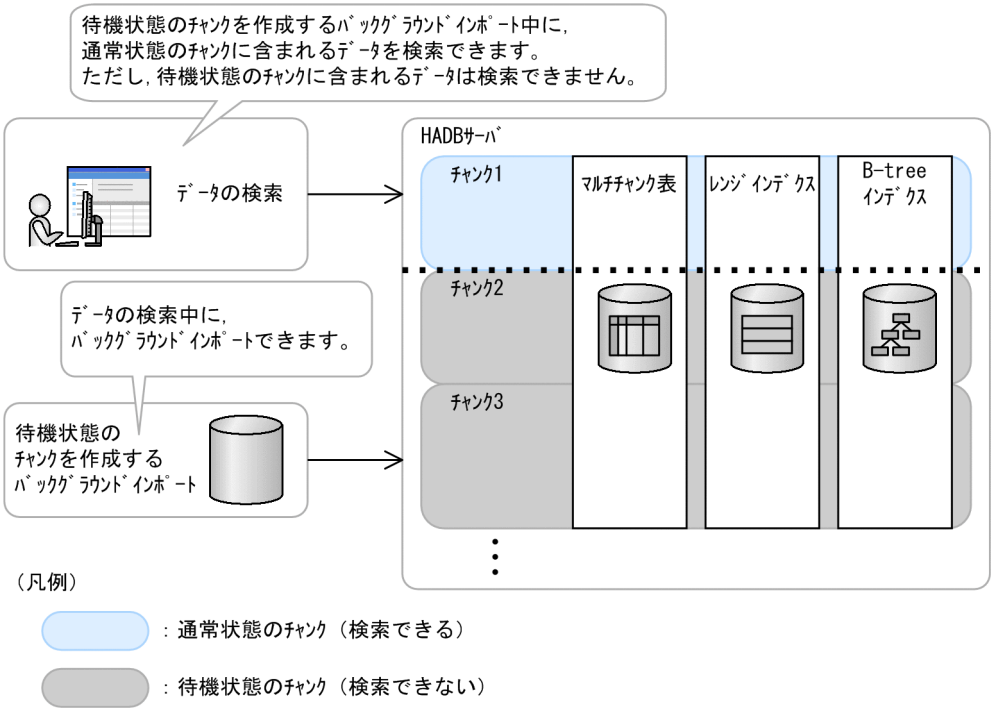
(3)　データの検索

翌月の商品データをデータベースに追加している間も，今月の商品データを検索・分析します。

バックグラウンドインポートによってデータを追加することで，インポートの対象となるデータベースに対して検索を実行しているユーザが存在しても，データのインポートを実行できます（検索と並行してデータのインポートを実行できます)。

バックグラウンドインポートとデータの検索の関係を次の図に示します。

図 11-34　待機状態のチャンクを作成するバックグラウンドインポートとデータの検索の関係



今月の商品データは分析できますが，翌月の商品データは分析対象外であるため，月が切り替わるまで分析できません（待機状態のチャンクは検索できません）。

(4)　待機状態のチャンクのマージ（チャンク数の削減）

各支社から翌月の商品データが送付されるたびに，バックグラウンドインポートすると，待機状態のチャンクが複数作成されます。これらの待機状態のチャンクを通常状態のチャンクに変更して分析対象に含めると，チャンク数の増加によって検索性能が低下するおそれがあります。

そのため，複数ある待機状態のチャンクを，1 つのチャンクにまとめることで，チャンク数の増加による検索性能の低下を回避します。この例では，各支社から送付されたすべての商品データを，データベースに追加したあとで，1 つのチャンクにまとめます。

(a)　マージ対象とする待機状態のチャンクの特定

マージ対象とする待機状態のチャンクのチャンク ID を特定するには，次に示すように，システム表 STATUS_CHUNKS を検索します。

SQL 文の例を次に示します。

■SQL 文の例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='ITEMLIST'
      AND "CHUNK_COMMENT"='October 2014'
```

■検索結果

```
CHUNK_ID
--------------------
                   2
            （中略）
                  11
```

[説明]

この例の場合は，バックグラウンドインポート時に付加したチャンクのコメント（'October 2014'）をキーにSTATUS_CHUNKS 表を検索することで，マージ対象とするチャンクのチャンク ID（2～11）を特定しています。

対象となるチャンクと，そのチャンク ID については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」を参照して確認してください。

(b)　待機状態のチャンクのマージ

待機状態のチャンクのマージには，adbmergechunk コマンドを使用します。adbmergechunk コマンドの実行例を次に示します。

■コマンドの実行例

```
adbmergechunk -u ADBUSER01 -p '#HelloHADB_01' -g 2
              -w /home/adbmanager/tmp
              -z /home/adbmanager/merge_file/merge_opt_file02.txt
              -m 'October 2014_merge'
              -c 2-11
              ITEMLIST
```

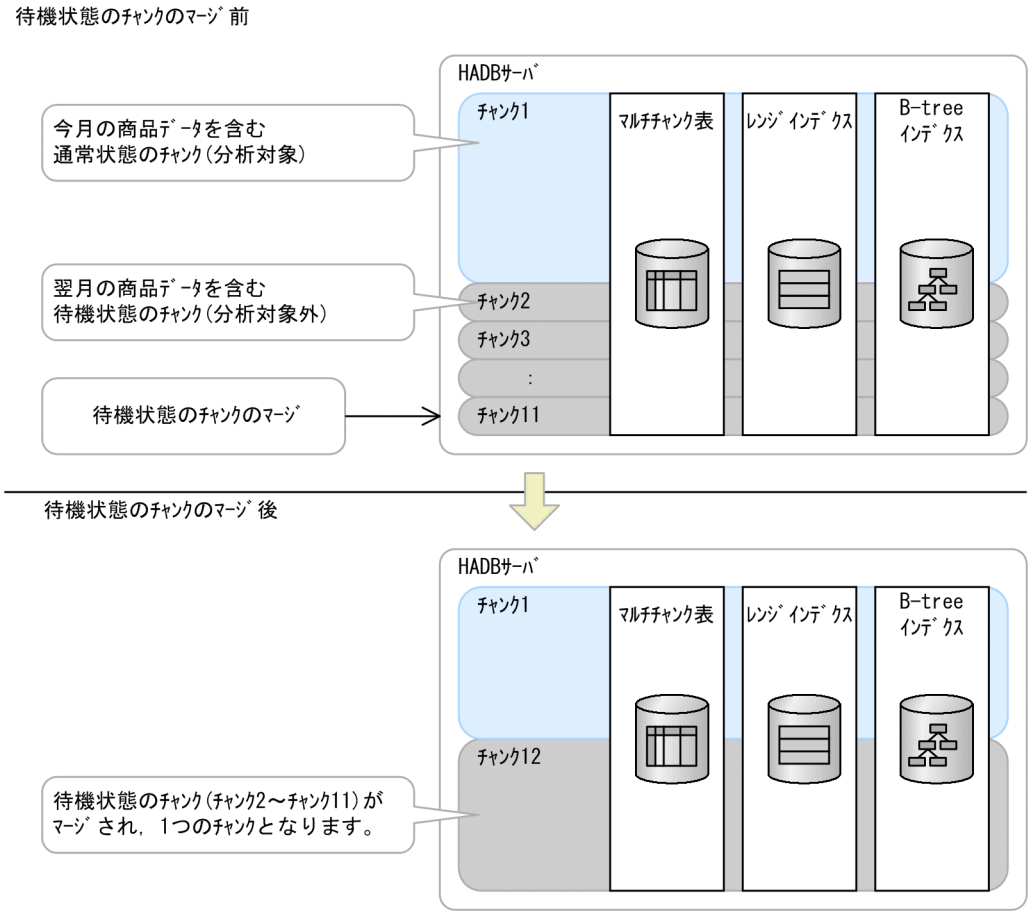
[説明]

商品表（ITEMLIST）に対してマージチャンクオプションファイル（/home/adbmanager/merge_file/merge_opt_file02.txt）に設定した値を入力情報として，-c オプションに指定した待機状態のチャンクのマージを実行します。

この例では，2 から11 のチャンク ID のチャンクを 1 つの待機状態のチャンクにマージすることで，翌月の商品データ（2014 年 10 月分）としてまとめています。

待機状態のチャンクのマージの例を次の図に示します。

図 11-35　待機状態のチャンクのマージの例



チャンク ID が2 から11 のチャンクは，adbmergechunk コマンドの正常終了時に自動的に削除されます。

> **メモ**
>
> 待機状態のチャンクをマージすることで，通常状態のチャンクに変更したときに検索性能の低下を防ぐことができます。また，1 表，1DB エリアが管理できるチャンク数には限りがあります。チャンク数が足りない場合には，複数のチャンクを 1 つにマージすることで，使用するチャンク数を削減できます。

adbmergechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmergechunk (チャンクのマージ)』を参照してください。また，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法)」も参照してください。

(5)　チャンクの状態変更（分析対象の商品データの切り替え）

月が切り替わったら，分析対象外としていた商品データを分析対象にします。また，今まで分析対象としていた商品データを分析対象外とします。

月が切り替わったら，次のようにチャンクの状態を変更します。

- 分析対象としていた商品データを含む通常状態のチャンクを，分析対象外とするため待機状態のチャンクに変更
- 分析対象外としていた商品データを含む待機状態のチャンクを，分析対象とするため通常状態のチャンクに変更

この例では，2014 年 9 月分の商品データを分析対象外に変更し，2014 年 10 月分の商品データを分析対象に変更します。

(a) 状態変更するチャンクのチャンク ID の特定

状態変更するチャンクのチャンク ID を特定するには，次に示すように，システム表STATUS_CHUNKS を検索します。

まず，2014 年 9 月分の商品データを含むチャンクのチャンク ID を特定します。

■SQL 文の例（通常状態のチャンクを待機状態のチャンクに変更する場合）

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='ITEMLIST'
      AND "CHUNK_COMMENT"='September 2014_merge'
```

■検索結果

```
 CHUNK_ID
--------------------
                   1
```

[説明]

この例の場合は，チャンクのマージ時に付加したチャンクのコメント（'September 2014_merge'）をキーにSTATUS_CHUNKS 表を検索することで，2014 年 9 月分の商品データを含むチャンクのチャンク ID を特定しています。

次に，2014 年 10 月分の商品データを含むチャンクのチャンク ID を特定します。

■SQL 文の例（待機状態のチャンクを通常状態のチャンクに変更する場合）

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='ITEMLIST'
      AND "CHUNK_COMMENT"='October 2014_merge'
```

■検索結果

```
 CHUNK_ID
--------------------
                  12
```

[説明]

この例の場合は，チャンクのマージ時に付加したチャンクのコメント（'October 2014_merge'）をキーにSTATUS_CHUNKS 表を検索することで，2014 年 10 月分の商品データを含むチャンクのチャンク ID を特定しています。

(b) チャンクの状態変更

チャンクの状態変更には，adbchgchunkstatus コマンドを使用します。次に示すように，チャンクの状態を変更します。

- 通常状態のチャンクを，待機状態のチャンクに変更
  2014 年 9 月分の商品データを含むチャンク（チャンク ID：1）が対象です。
- 待機状態のチャンクを，通常状態のチャンクに変更
  2014 年 10 月分の商品データを含むチャンク（チャンク ID：12）が対象です。

adbchgchunkstatus コマンドの実行例を次に示します。

■コマンドの実行例

```
adbchgchunkstatus -u ADBUSER01 -p '#HelloHADB_01'
                  -w 1 -n 12 ITEMLIST
```

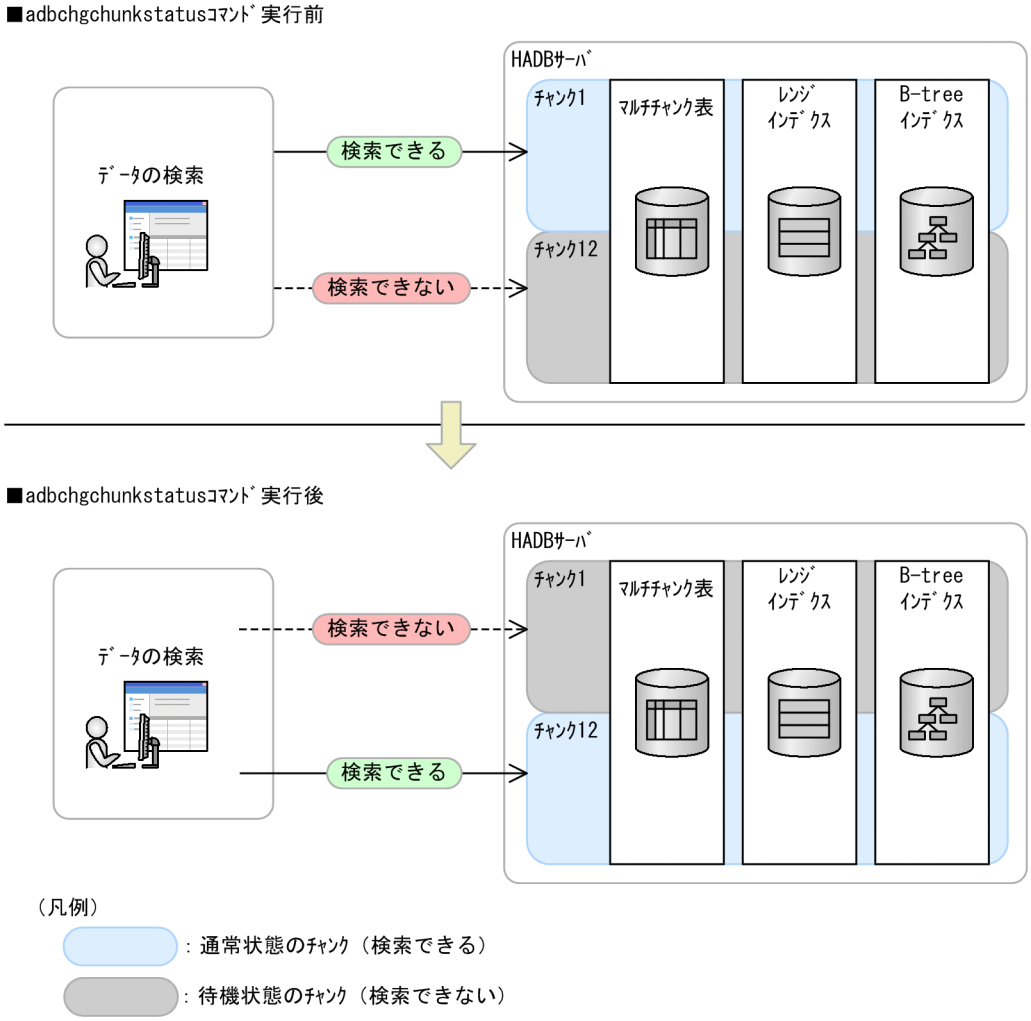
[説明]

商品表（ITEMLIST）に対して，-w オプションと-n オプションに従って，チャンク状態の変更を実行します。

この例では，チャンク ID が1 のチャンクを待機状態に変更し，チャンク ID が12 のチャンクを通常状態に，それぞれ変更します。

チャンクの状態変更の例を次の図に示します。

図 11-36　チャンクの状態変更の例



[説明]

2014 年 9 月分の商品データを含むチャンク（チャンク ID：1）が分析対象外となり，2014 年 10 月分の商品データを含むチャンク（チャンク ID：12）が分析対象となります。

adbchgchunkstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbchgchunkstatus（チャンクの状態変更）』を参照してください。また，「11.4.12　チャンクの状態を変更する方法」も参照してください。

(6)　データのバックアップ

データ容量の増加を抑えるため，6 か月より前の商品データはバックアップを取得してから，削除します。データベースのバックアップは月単位で取得します。

(a)　作成日時が最も古いチャンクの特定

商品表（ITEMLIST）で 6 か月より前のチャンク（作成した日時が最も古いチャンク）を特定します。商品表の最もチャンクの作成日時が古いチャンクを特定するには，次に示すように，システム表STATUS_CHUNKSを検索します。

SQL 文の例を次に示します。

■SQL 文の例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='ITEMLIST'
      AND "CREATE_TIME"=(SELECT MIN("CREATE_TIME")
         FROM "MASTER"."STATUS_CHUNKS"
            WHERE TC."TABLE_SCHEMA"='ADBUSER01'
               AND TC."TABLE_NAME"='ITEMLIST')
```

■検索結果

```
CHUNK_ID
--------------------
                   1
```

[説明]

STATUS_CHUNKS 表を検索した結果，この例の場合はチャンク ID が1 のチャンクが，作成日時が最も古いチャンク（6 か月より前のチャンク）であることがわかります。

> **メモ**
>
> チャンクをマージした場合は，STATUS_CHUNKS 表には，マージ元チャンクのうち，最も古い「チャンクの作成日時」が格納されます。

STATUS_CHUNKS 表の検索方法については，「付録 C.9　システム表の検索」を参照してください。

(b)　バックアップの取得

商品表（ITEMLIST）で最もチャンクの作成日時が古いチャンクのチャンク ID が1 であるとわかったので，そのチャンクに含まれるデータのバックアップを CSV 形式のファイルで取得します。

バックアップの取得には，adbexport コマンドを使用します。adbexport コマンドの実行例を次に示します。

■コマンドの実行例

```
adbexport -u ADBUSER01 -p '#HelloHADB_01'
          -z /home/adbmanager/exp_file/exp_opt_file02.txt
          -n ITEMLIST
          -c 1
          /home/adbmanager/exp_file/exp_data_path02.txt
```

■出力データパスファイル（/home/adbmanager/exp_file/exp_data_path02.txt）の内容

```
/home/adbmanager/exp_backup/backup0101.csv
/home/adbmanager/exp_backup/backup0102.csv
/home/adbmanager/exp_backup/backup0103.csv
```

[説明]

チャンク ID が1 のチャンクに含まれるデータのバックアップを CSV 形式のファイル（/home/adbmanager/exp_backup/backup0101.csv，/home/adbmanager/exp_backup/backup0102.csv，/home/adbmanager/exp_backup/backup0103.csv）に取得します。

この例の場合，月単位でチャンクがマージされているので，1 か月分のデータのバックアップになります。

> **メモ**
>
> バックアップを取得するファイルは 1 つにすることもできますが，複数のファイルに分散することで，出力にかかるオーバヘッドを分散させ性能を向上させることができます。

adbexport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbexport（データのエクスポート)』を参照してください。また，「11.4.5　チャンク単位にデータをエクスポートする方法」も参照してください。

(7)　古いデータ（チャンク）の削除

データ容量の増加を抑えるため，6 か月より前のデータを削除します。

商品表（ITEMLIST）から作成日時が最も古いチャンクを削除します。チャンクを削除するには，PURGE CHUNK 文の探索条件にチャンク ID を指定して実行します。

■SQL 文の例

```
PURGE CHUNK ”ITEMLIST” WHERE CHUNKID=1
```

[説明]

チャンク ID が1 のチャンクに含まれるデータを削除します。

この例の場合，月単位でチャンクがマージされているので，1 か月分のデータが削除されます。

PURGE CHUNK 文については，マニュアル『HADB　SQL リファレンス』の『操作系 SQL』の『PURGE CHUNK（チャンク内の全行削除)』を参照してください。また，「11.4.6　チャンク単位にデータを削除する方法」も参照してください。

(8)　新しいデータの追加・マージ・状態変更と古いデータの削除の繰り返し

待機状態のチャンクを作成するバックグラウンドインポートの実行，待機状態のチャンクのマージ，チャンクの状態変更およびPURGE CHUNK 文を利用したチャンクの削除を繰り返すことで，HADB が管理するデータ量を増やすことなく，対象月の商品データだけを分析する運用が継続できます。

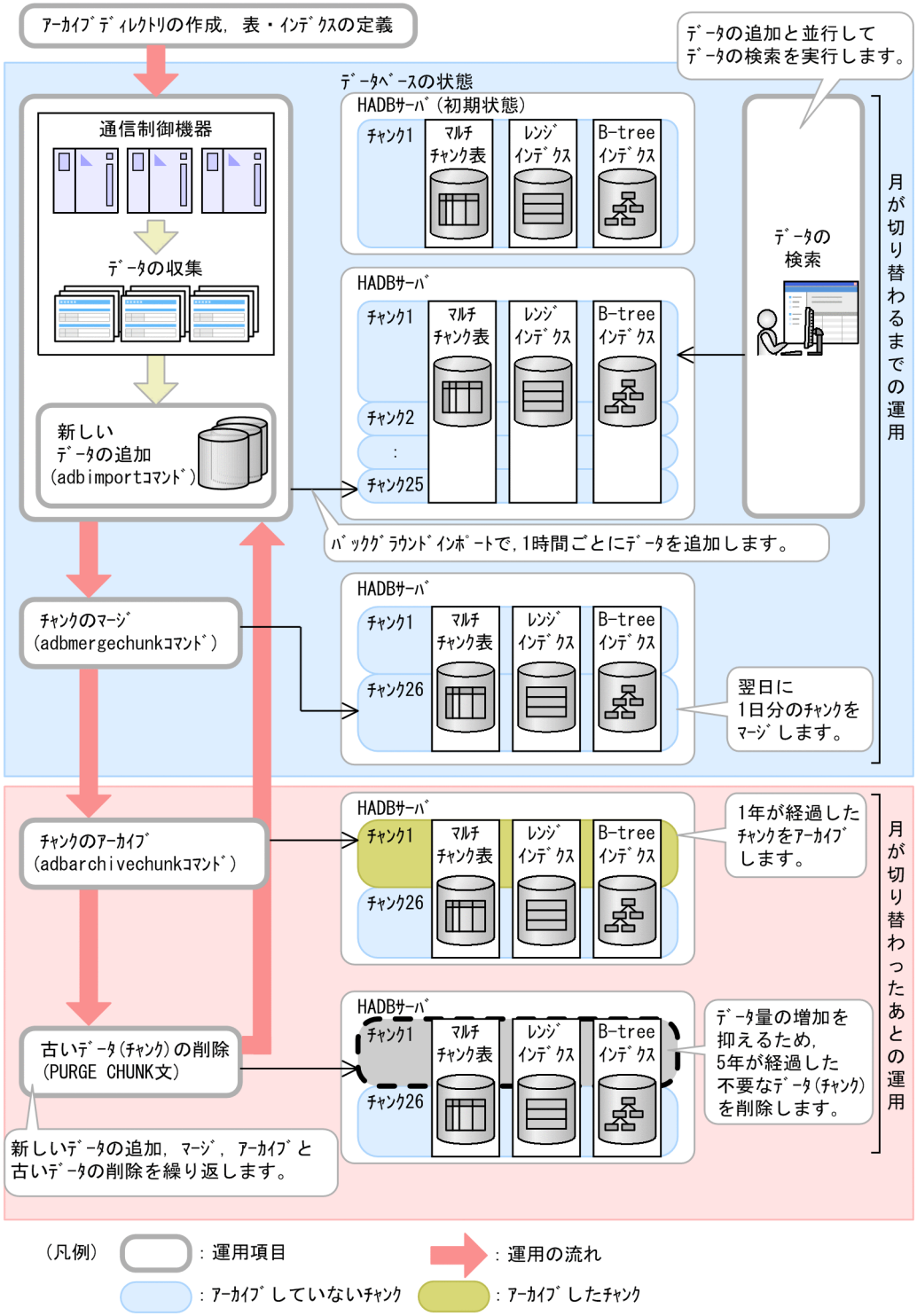
11.4.25　チャンクのアーカイブを意識した運用例

ここでは，検索頻度が少ない古いデータをアーカイブして，データベースの容量増加を抑える運用方法を，次に示す例を使って説明します。ここで説明する運用方法の前提は，次のとおりです。

- 全国の通信制御機器から発生したセンサデータ（通信量や障害予兆検知に関する情報）を本社が収集し，通信制御機器の交換や増設の要否を判断するために，センサデータを検索・分析します。
- 本社が収集したセンサデータは，通信制御機器ログ表で一元管理します。
- センサデータは，1 時間ごとにデータベースにインポートします。
- 1 時間ごとにインポートしたセンサデータは，日が切り替わったタイミングで，1 日分のチャンクとして，1 つのチャンクにマージします。また，月が切り替わったタイミングで，1 か月分のチャンクとして，1 つのチャンクにマージします。最終的には，1 つのチャンクを月単位で管理します。
- データベースにセンサデータをインポートしている間も，センサデータを検索します。直近 24 時間分（日単位），直近 7 日分（週単位）および直近 1 か月分（月単位）のセンサデータを検索します。検索結果を基に，通信制御機器の交換や増設の要否を判断します。
- センサデータを検索した結果，通信量がしきい値を超えていた場合は，該当する日と前後 1 日の日付をキーにして，過去 5 年間のデータを分析します。通信量がしきい値を超えるケースは，月 1 回程度です。
- データベースにインポートして 1 年が経過したセンサデータは，検索頻度が非常に少なくなるため，チャンク単位でセンサデータをアーカイブします。月が切り替わったタイミングで，センサデータをアーカイブして，データベースの容量を削減します。
- データベースには 5 年間のデータを保持します。5 年を超えたデータは，月が切り替わったタイミングで，チャンク単位で削除します。

このシステムでの運用の流れを次の図に示します。

図 11-37　このシステムでの運用の流れ（チャンクのアーカイブの利用）



(1)　アーカイブディレクトリの作成

この運用例では，データ容量削減のために，1 年経過したデータをチャンク単位にアーカイブします。チャンクをアーカイブする際，アーカイブファイルを格納するためのアーカイブディレクトリが必要になります。運用を開始する前に，アーカイブディレクトリを作成してください。

次に示す手順で，アーカイブディレクトリを作成します。

手順

1. スーパユーザで OS にログインする

   スーパユーザで，サーバマシンの OS にログインしてください。

2. アーカイブディレクトリ用のファイルシステムを作成する

   OS のコマンドを実行し，アーカイブディレクトリ用のファイルシステムを作成します。

   アーカイブディレクトリ用の LV として作成した/dev/vg_hadb/hadb_archivedir を，ext4 のファイルシステムで作成する場合の実行例を次に示します。

   ■コマンドの実行例

   ```
   mkfs -t ext4 /dev/vg_hadb/hadb_archivedir
   ```

3. アーカイブディレクトリ用のマウントポイントを作成する

   OS のコマンドを実行し，アーカイブディレクトリ用のマウントポイントを作成します。実行例を次に示します。

   ■コマンドの実行例

   ```
   mkdir -p /HADB/archive
   ```

4. アーカイブディレクトリ用のファイルシステムがある VG を活性化する

   OS のコマンドを実行し，アーカイブディレクトリ用のファイルシステムがある VG を活性化します。実行例を次に示します。

   ■コマンドの実行例

   ```
   vgchange -a y /dev/vg_hadb
   ```

5. アーカイブディレクトリ用ファイルシステムをマウントする

   OS のコマンドを実行し，アーカイブディレクトリ用ファイルシステムをマウントします。実行例を次に示します。

   ■コマンドの実行例

   ```
   mount /dev/vg_hadb/hadb_archivedir /HADB/archive -t ext4 -o defaults,noatime,_netdev4
   ```

6. アーカイブディレクトリのオーナーを変更する

   OS のコマンドを実行し，アーカイブディレクトリのオーナーを変更します。アーカイブディレクトリを使用する HADB 管理者と HADB 管理グループを指定してください。

   HADB 管理者のユーザ名がadbmanager，HADB 管理グループがadbgroup の場合の実行例を次に示します。

   ■コマンドの実行例

   ```
   chown adbmanager.adbgroup /HADB/archive
   ```

(2) アーカイブマルチチャンク表の定義

アーカイブディレクトリを作成したあとで，アーカイブマルチチャンク表とインデクスを定義します。

この運用例では，センサデータを一元管理するために，通信制御機器ログ表（INSTRUMENT_LOG）およびインデクスを定義します。

(a) 通信制御機器ログ表（INSTRUMENT_LOG）の定義

通信制御機器ログ表（INSTRUMENT_LOG）は，DB エリアTBLDAT_003 に格納するように定義します。また，バックグラウンドインポートおよびチャンクのアーカイブを実施するために，通信制御機器ログ表をアーカイブマルチチャンク表として定義します。

この運用例では，次に示す数のチャンクが必要となります。

- 1 時間ごとにバックグラウンドインポートで追加したセンサデータを管理するチャンク（24 個）
  1 時間ごとにバックグラウンドインポートでデータを追加するため，1 日分のチャンクは 24 個になります。
- 1 日ごとのセンサデータを管理するチャンク（32 個）
  1 か月分（31 日分）に加えて，それらをマージするマージ先チャンクが 1 個必要となるため，合計すると 32 個になります。
- 1 か月ごとのセンサデータを管理するチャンク（61 個）
  5 年分（60 か月分）に加えて，それらをマージするマージ先チャンクが 1 個必要となるため，合計すると 61 個になります。

このため，安全率を 1.2 とすると，作成されるチャンク数の最大値は次の値となります。

```
作成されるチャンク数の最大値＝ ↑(24＋32＋61)×1.2↑＝ 141
```

メモ

チャンク数が増加すると検索性能に影響を与えます。そのため，使用するチャンク数をできるだけ抑えるように，設計・運用することを推奨します。

この見積もりによって，チャンク数の最大値には141 を設定します。この値をCREATE TABLE 文のチャンク指定のCHUNK に指定します。

また，通信制御機器ログ表（INSTRUMENT_LOG）の収集日時列（C_TIME）は，データ分析の範囲を特定でき，かつチャンクごとにデータが重ならない列です。そのため，アーカイブレンジ列とします。アーカイブレンジ列に自動的に定義されるレンジインデクスは，通信制御機器ログ表（INSTRUMENT_LOG）に定義する，ほかのインデクスを格納する DB エリアIDXDAT_003 に格納します。

次に示す内容を，CREATE TABLE 文のチャンクアーカイブ指定に指定します。

- アーカイブレンジ列の列名（C_TIME）
- 自動的に定義されるレンジインデクスを格納する DB エリア名（IDXDAT_003）
- 作成したアーカイブディレクトリ（/HADB/archive）

CREATE TABLE 文の例を次に示します。

■CREATE TABLE 文の例

```
CREATE TABLE "INSTRUMENT_LOG"
            ("SID" INTEGER,
             "C_TIME" TIMESTAMP,
             "TRAFFIC" INTEGER,
             "THROUGH" INTEGER,
             "RETRY" INTEGER,
              ...)
  IN "TBLDAT_003"
  CHUNK= 141
  ARCHIVABLE RANGECOLUMN="C_TIME" IN "IDXDAT_003"
  ARCHIVEDIR='/HADB/archive'
```

CREATE TABLE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE TABLE（表の定義)』を参照してください。

(b)　通信制御機器ログ表（INSTRUMENT_LOG）に対するインデクスの定義

通信制御機器ログ表に対して，B-tree インデクスを定義します。B-tree インデクスは DB エリアIDXDAT_003に格納します。

機器 ID 列（SID）に対して B-tree インデクスを定義します。CREATE INDEX 文の例を次に示します。

■CREATE INDEX 文の例

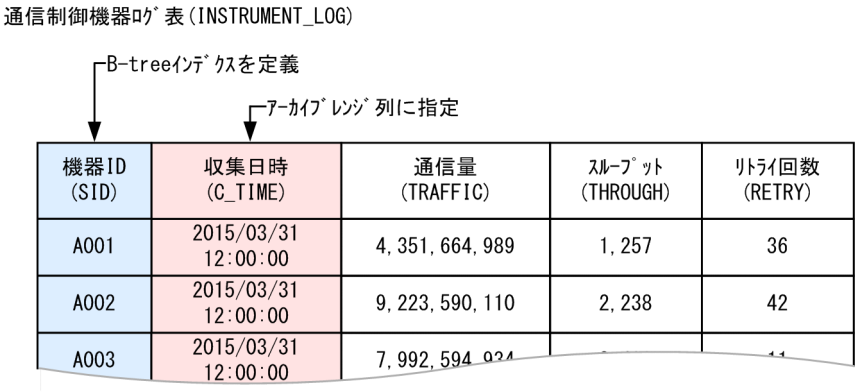
```
CREATE INDEX "SID_IDX"
   ON "INSTRUMENT_LOG"("SID") IN "IDXDAT_003"
   EMPTY
```

CREATE INDEX 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE INDEX（インデクスの定義)』を参照してください。

(c)　この運用例で使用する表のイメージ

この運用例で使用する表のイメージを次の図に示します。

図 11-38　この運用例で使用する表のイメージ



(3)　新しいセンサデータの追加（バックグラウンドインポート）

通信制御機器からリアルタイムにセンサデータが発生します。発生したセンサデータは，1 時間ごとに通信制御機器ログ表（INSTRUMENT_LOG）に追加します。

センサデータの追加は，バックグラウンドインポートで実行します。1 回のバックグラウンドインポートでアーカイブマルチチャンク表に格納したデータ，およびそのインデクスは，1 つの固まり（チャンク）として管理されます。チャンクについては，「2.14.2　データインポート単位のデータ管理（チャンク）」を参照してください。

> **メモ**
>
> データをチャンク単位で管理することで，チャンク単位でのバックアップの取得や，データの削除ができます。バックグラウンドインポートやチャンクをマージする際に，チャンクにコメントを設定できます。コメントを設定しておくと，操作対象のチャンクを特定しやすくなります。

-b オプションを指定したadbimport コマンドを実行して，バックグラウンドインポートを行います。adbimport コマンドの実行例を次に示します。

■コマンドの実行例

```
adbimport -u ADBUSER01 -p '#HelloHADB_01' -k "'" -s , -g 10
          -w /home/adbmanager/tmp
          -z /home/adbmanager/imp_file/imp_opt_file03.txt
          -b
          -m '2015/03/31'
          INSTRUMENT_LOG
          /home/adbmanager/imp_file/imp_data_path03.txt
```

[説明]

通信制御機器ログ表（INSTRUMENT_LOG）に対して，入力データパスファイル（/home/adbmanager/imp_file/imp_data_path03.txt）に指定した入力データファイルの内容をバックグラウンドインポートします。

「2015 年 3 月 31 日」に発生したセンサデータをバックグラウンドインポートで追加するため，-m オプションに「チャンクに設定するコメント」を指定します。

1 時間ごとにバックグラウンドインポートを実行します。その結果，1 日分（24 時間分）のチャンクが 24 個作成されます。

adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート）』を参照してください。また，「11.4.2　マルチチャンク表にデータを格納する方法（バックグラウンドインポート）」も参照してください。

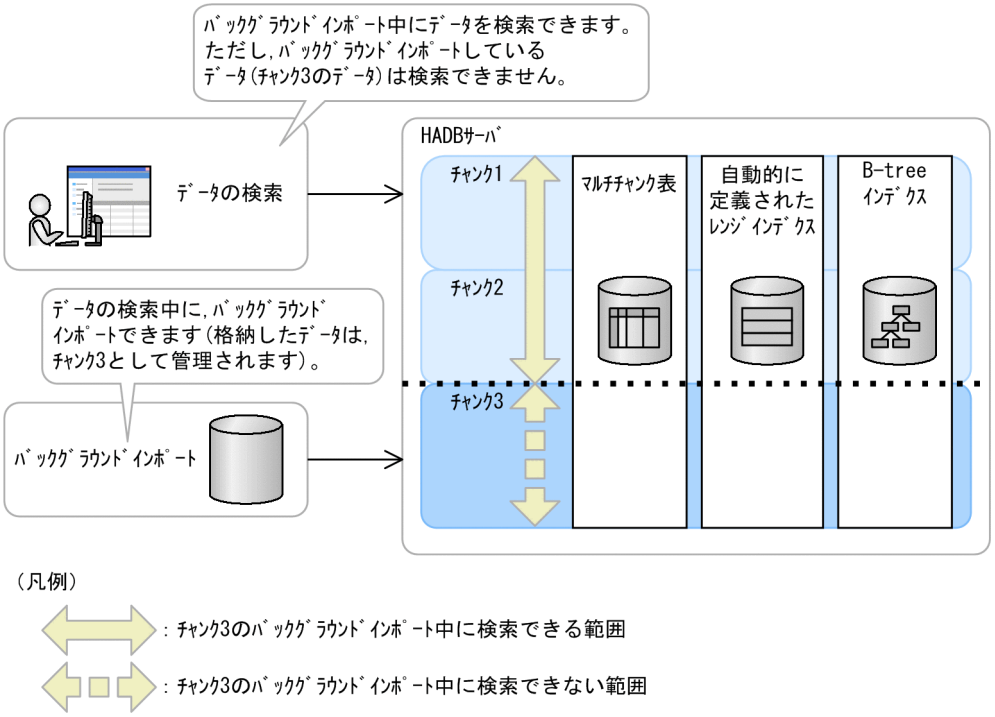
(4)　データの検索

バックグラウンドインポートを実行している間も，アーカイブマルチチャンク表のデータを検索できます。

バックグラウンドインポートによってセンサデータを追加することで，インポートの対象となるデータベースに対して検索を実行している HADB ユーザが存在しても，データのインポートを実行できます（検索と並行してデータのインポートを実行できます）。

バックグラウンドインポートとデータの検索の関係を次の図に示します。

図 11-39　バックグラウンドインポートとデータの検索の関係



(5)　チャンクのマージ（チャンク数の削減）

チャンク数が増加すると検索性能が低下するおそれがあるため，定期的にチャンクをマージする必要があります。この運用例では，センサデータを 1 時間ごとにバックグラウンドインポートしているので，1 日で 24 個のチャンクが作成されます。

この運用例では，次に示すタイミングで複数のチャンクをマージして，1 個のチャンクにマージしています。最終的には，月単位でチャンクを管理します。

- 1 日分（24 時間分）のチャンク 24 個を，1 個のチャンクにマージする
- 1 か月分（31 日分）のチャンク 31 個を，1 個のチャンクにマージする

ここでは，1 日分（2015 年 3 月 31 日分）のデータを，1 つのチャンクにマージする例を次に示します。

(a) マージ対象とするチャンクの特定

マージ対象とするチャンクのチャンク ID を特定するには，次に示すように，システム表STATUS_CHUNKS を検索します。SQL 文の例を次に示します。

■SQL 文の例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS" TC
   WHERE "TABLE_SCHEMA"='ADBUSER01' AND "TABLE_NAME"='INSTRUMENT_LOG'
      AND "CHUNK_COMMENT"='2015/03/31'
```

■検索結果

```
CHUNK_ID
--------------------
                  92
          （中略）
                 115
```

[説明]

この例の場合は，バックグラウンドインポート時に付加したチャンクのコメント（'2015/03/31'）をキーにSTATUS_CHUNKS 表を検索することで，マージ対象とするチャンクのチャンク ID（92～115）を特定しています。

対象となるチャンクと，そのチャンク ID については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」を参照して確認してください。

(b) チャンクのマージ

adbmergechunk コマンドを実行してチャンクをマージします。adbmergechunk コマンドの実行例を次に示します。

■コマンドの実行例

```
adbmergechunk -u ADBUSER01 -p '#HelloHADB_01' -g 2
              -w /home/adbmanager/tmp
              -z /home/adbmanager/merge_file/merge_opt_file03.txt
              -m '2015/03/31'
              -c 92-115
              INSTRUMENT_LOG
```

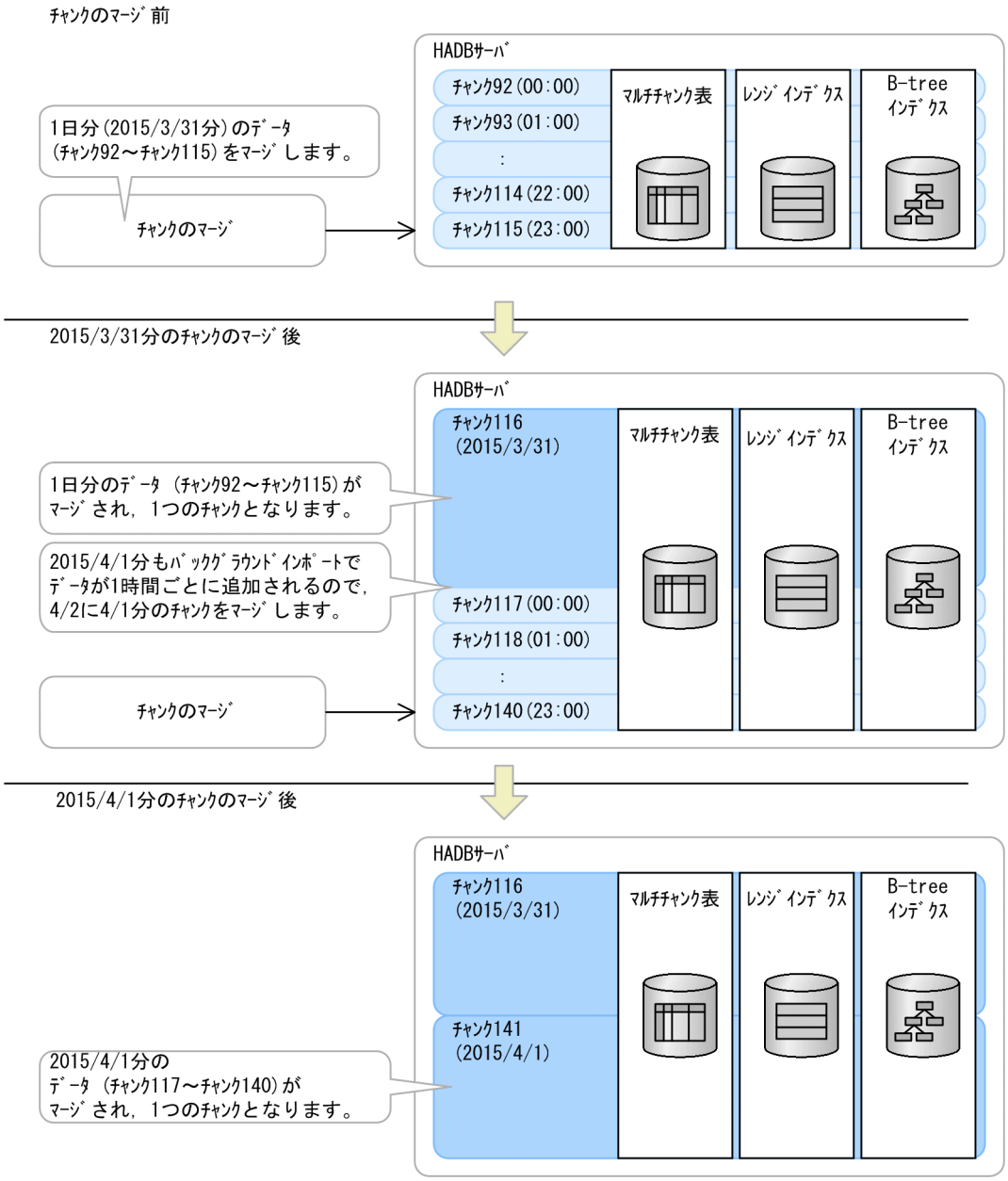
[説明]

通信制御機器ログ表（INSTRUMENT_LOG）に対してマージチャンクオプションファイル（/home/adbmanager/merge_file/merge_opt_file03.txt）に指定した値を入力情報として，-c オプションに指定したチャンクのマージを実行します。

この例では，92 から 115 のチャンク ID のチャンクを 1 つの待機状態のチャンクにマージすることで，1 日分のセンサデータ（2015 年 3 月 31 日分）としてまとめています。

チャンクのマージの例を次の図に示します。

図 11-40　チャンクのマージの例



チャンク ID が92 から115 のチャンクは，adbmergechunk コマンドの正常終了時に自動的に削除されます。

メモ

チャンクをマージすることで，検索性能の低下を防ぐことができます。また，1 表，1DB エリアが管理できるチャンク数には限りがあります。使用中のチャンク数が，1 表，1DB エリアが管理できる上限に達した場合には，複数のチャンクを 1 つにマージしてください。チャンクをマージすることで，使用中のチャンク数を削減できます。

adbmergechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmergechunk (チャンクのマージ)』を参照してください。また，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法)」も参照してください。

(6)　チャンクのアーカイブ（チャンク単位でのデータの圧縮）

この運用例では，データベースに追加してから 1 年が経過したデータは，検索頻度が非常に少なくなるため，チャンク単位でアーカイブ（データの圧縮）します。チャンクをアーカイブすることで，データベースの容量を削減できます。

重要

アーカイブしたチャンク内のデータを検索する場合，次に示す原因によって，アーカイブされていないチャンク内のデータを検索するときに比べて検索時間が長くなります。

- データの伸長に時間が掛かる
- インデクスを使用した検索ができない

この運用例のように，検索頻度が非常に少ないが，長期間保持する必要があるデータをアーカイブするようにしてください。詳細については，「2.15.2　チャンクアーカイブ機能を適用するケース」を参照してください。

(a)　チャンクのアーカイブの実行

adbarchivechunk コマンドを実行して，チャンクをアーカイブします。ここでは，2014 年 3 月のセンサデータを管理するチャンクをアーカイブします。adbarchivechunk コマンドの実行例を次に示します。

■コマンドの実行例

```
adbarchivechunk -u ADBUSER01 -p '#HelloHADB_01' -g 2
                -z /home/adbmanager/arc_file/archive_chunk03.def
                -r 2014/03/01-2014/03/31
                INSTRUMENT_LOG
```

[説明]

adbarchivechunk コマンドの-r オプションで，アーカイブするチャンクの日付の範囲（2014/03/01-2014/03/31）を指定します。-r オプションに指定した日付の範囲に，アーカイブレンジ列の値の範囲が含まれているチャンクがアーカイブされます。この運用例では，通信制御機器ログ表

（INSTRUMENT_LOG）の収集日時列（C_TIME）を，アーカイブレンジ列に指定しています。そのため，収集日時列（C_TIME）の値を基に，2014 年 3 月分のチャンクがアーカイブされます。

(b) アーカイブの実行結果の確認

adbdbstatus コマンドを実行して，アーカイブしたチャンクの状態を確認します。adbdbstatus コマンドで「アーカイブ状態のチャンクのサマリ情報」を出力すると，アーカイブしたチャンクの状態を確認できます。adbdbstatus コマンドの実行例を次に示します。

■コマンドの実行例

```
adbdbstatus -c archivechunk
            -n ADBUSER01.INSTRUMENT_LOG
            -r 2014/03/01-2014/03/31
```

[説明]

アーカイブ状態のチャンク（2014 年 3 月分）のサマリ情報を出力します。次の出力項目から，アーカイブしたチャンクに格納されている行数，アーカイブしたチャンクのデータ容量，およびアーカイブしたチャンクの圧縮率が確認できます。

- Rows（アーカイブ状態のチャンクに格納されている行数）
- Archive_file_size（アーカイブ状態のチャンクに対応している全アーカイブファイルのファイルサイズの合計値）
- Unarchive_table_size（アーカイブ状態になる前のチャンクで，表を格納するために使用していたセグメントの容量）
- Unarchive_index_size（アーカイブ状態になる前のチャンクで，インデクスを格納するために使用していたセグメントの容量）
- Compression_ratio（アーカイブ状態のチャンクの圧縮率）

(7) 古いデータ（チャンク）の削除

データ容量の増加を抑えるため，5 年が経過したデータ（アーカイブ状態のチャンク）を削除します。

通信制御機器ログ表（INSTRUMENT_LOG）から，データを追加して 5 年が経過した古いチャンクを削除します。削除対象のチャンクのチャンク ID を確認してから，PURGE CHUNK 文でチャンクを削除します。

ここでは，2010 年 4 月分のチャンクを削除します。

(a) チャンク ID の確認

adbdbstatus コマンドを実行して，削除対象のチャンクのチャンク ID を確認します。adbdbstatus コマンドで「アーカイブ状態のチャンクのサマリ情報」を出力すると，削除対象のチャンクのチャンク ID を確認できます。adbdbstatus コマンドの実行例を次に示します。

■コマンドの実行例

```
adbdbstatus -c archivechunk
            -n ADBUSER01.INSTRUMENT_LOG
            -r 2010/04/01-2010/04/30
```

[説明]

アーカイブ状態のチャンク（2010 年 4 月分）のサマリ情報を出力します。次の出力項目から，削除対象のチャンクのチャンク ID，およびチャンク削除後にアーカイブディレクトリに確保される空き容量が確認できます。

- Chunk_ID（チャンク ID）
- Archive_file_size（アーカイブ状態のチャンクに対応している全アーカイブファイルのファイルサイズの合計値）

(b) チャンクの削除

adbdbstatus コマンドで確認したチャンク ID を基に，チャンクを削除します。チャンクを削除するには，PURGE CHUNK 文の探索条件にチャンク ID を指定して実行します。PURGE CHUNK 文の例を次に示します。

■SQL 文の例

```
PURGE CHUNK ”INSTRUMENT_LOG” WHERE CHUNKID=4
```

[説明]

チャンク ID が4 のチャンクに含まれるデータを削除します。

この例の場合，月単位でチャンクがマージされているので，1 か月分のデータが削除されます。

PURGE CHUNK 文については，マニュアル『HADB　SQL リファレンス』の『操作系 SQL』の『PURGE CHUNK（チャンク内の全行削除)』を参照してください。また，「11.4.6　チャンク単位にデータを削除する方法」も参照してください。

(8) 新しいデータの追加・マージ・アーカイブと古いデータの削除の繰り返し

次に示す運用を繰り返すことで，長期間保持するデータの容量を抑えながら，センサデータを分析する運用を継続できます。

- バックグラウンドインポートの実行
- チャンクのマージ
- チャンクのアーカイブ
- PURGE CHUNK 文を利用したチャンクの削除

11.5　コスト情報の収集

コスト情報は，データを効率良く検索するために，adbgetcst コマンドを使用して実表，B-tree インデクスおよびテキストインデクスから収集した検索方法に関する情報のことです。収集したコスト情報は，システム表に格納され，検索処理時に利用されます。

adbgetcst コマンドを使用して，コスト情報を収集するタイミングについて説明します。

■コスト情報を収集するタイミング

次に示す作業を実行したら，コスト情報を収集することを推奨します。

- adbimport コマンドを実行して実表にデータを格納した場合

  作成モードおよび追加モードに関係なく，adbimport コマンドを使用して実表にデータを格納した場合は，コスト情報を収集することを推奨します。

- 大量のデータを追加，更新または削除した場合

  INSERT 文，UPDATE 文，およびDELETE 文を使用して，大量のデータを追加，更新または削除した場合は，コスト情報を収集することを推奨します。

- adbidxrebuild コマンドを実行してインデクスを再作成した場合

  adbidxrebuild コマンドを使用して B-tree インデクス，またはテキストインデクスを再作成した場合は，コスト情報を収集することを推奨します。

一度に大量のデータではなく，複数回に分けて少量のデータを追加，更新または削除した場合も，任意のタイミングでコスト情報を収集することを推奨します。

なお，コスト情報を収集する場合，対象表のすべての行を検索するため，収集完了までに時間が掛かります。

adbgetcst コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbgetcst（コスト情報の収集)』を参照してください。

11.6　HADB ユーザの管理

ここでは，HADB ユーザの管理方法について説明します。

11.6.1　HADB ユーザを作成する方法

HADB ユーザは，定義系 SQL のCREATE USER 文で作成します。

DBA 権限およびCONNECT 権限を持っている HADB ユーザが，CREATE USER 文を実行してください。

HADB ユーザを作成する例を次に示します。

指定例

次に示す認可識別子（ユーザ ID）およびパスワードの HADB ユーザを作成します。

- 認可識別子：ADBUSER02
- パスワード：#HelloHADB_02

```
CREATE USER ”ADBUSER02” IDENTIFIED BY ’#HelloHADB_02’
```

なお，HADB ユーザを作成する場合に指定するパスワードの文字数ですが，最低 8 文字以上とすることを推奨します。

HADB ユーザの認可識別子およびパスワードの指定規則については，「9.4.2　認可識別子の指定規則」および「9.4.3　パスワードの指定規則」を参照してください。

重要

作成した HADB ユーザには，権限は付与されていません。「11.7.1　HADB ユーザにユーザ権限とスキーマ操作権限を付与する方法」を参照して，必要な権限を付与してください。

11.6.2　HADB ユーザのパスワードを変更する方法

HADB ユーザのパスワードは，定義系 SQL のALTER USER 文で変更します。HADB ユーザのパスワードを変更する例を次に示します。

指定例

HADB ユーザ「ADBUSER02」のパスワードを#HelloHADB_03 に変更します。

```
ALTER USER ”ADBUSER02” IDENTIFIED BY ’#HelloHADB_03’
```

重要

- DBA 権限を持っていない HADB ユーザは，自分（HADB サーバに接続中の認可識別子の HADB ユーザ）のパスワードだけを変更できます。ほかの HADB ユーザのパスワードは変更できません。
- DBA 権限を持っている HADB ユーザは，自分のパスワードを変更できます。さらに，ほかの HADB ユーザのパスワードも変更できます。ただし，監査権限を持っている HADB ユーザのパスワードは変更できません。

メモ

- HADB ユーザのパスワードを変更する場合に指定するパスワードの文字数は，最低 8 文字以上とすることを推奨します。
- パスワードは定期的に変更することを推奨します。
- HADB ユーザのパスワードの指定規則については，「9.4.3　パスワードの指定規則」を参照してください。

11.6.3　HADB ユーザを削除する方法

HADB ユーザは，定義系 SQL のDROP USER 文で削除します。HADB ユーザを削除する例を次に示します。

指定例

HADB ユーザ「ADBUSER02」を削除します。

```
DROP USER ”ADBUSER02” CASCADE
```

DBA 権限およびCONNECT 権限を持っている HADB ユーザが，DROP USER 文を実行して HADB ユーザを削除できます。

メモ

- HADB ユーザを削除する場合，事前にadbchgsrvmode --offline コマンドを実行して，HADB サーバの稼働モードをオフラインモードに変更しておくことを推奨します。HADB クライアントからの接続を受け付けないようにしてから，HADB ユーザを削除してください。
- 監査権限を持っている HADB ユーザを削除することはできません。監査権限を取り消してもらったあとに，その HADB ユーザを削除してください。監査権限の取り消し方法については，「12.4.1　監査人の追加，削除，変更（監査権限の付与または取り消し）」の「(2) 監査人の削除（監査権限の取り消し）」を参照してください。

DROP USER 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『DROP USER (HADB ユーザの削除)』を参照してください。

重要

削除対象の HADB ユーザが所有しているスキーマおよび表も削除されるため，次に示す影響があります。

- DROP USER 文の実行によって削除される表に依存するビュー表（ほかのスキーマのビュー表）が，削除または無効化されます。
- DROP USER 文の実行によって削除される表を被参照表とする外部キー（ほかのスキーマの外部キー）も削除されます。

スキーマの削除については，「11.9.2　スキーマを削除する方法」を参照してください。スキーマに定義されている実表が更新不可状態だった場合，中断したコマンドによって作成された作業用一時ファイルが残ることがあります。

また，削除対象の HADB ユーザが，ほかの HADB ユーザにアクセス権限を付与していた場合，そのアクセス権限は取り消されます。

11.7　ユーザ権限とスキーマ操作権限の管理

ここでは，ユーザ権限とスキーマ操作権限の管理方法について説明します。

監査権限の管理方法については，「12.4.1　監査人の追加，削除，変更（監査権限の付与または取り消し)」を参照してください。

11.7.1　HADB ユーザにユーザ権限とスキーマ操作権限を付与する方法

ユーザ権限およびスキーマ操作権限は，定義系 SQL のGRANT 文で付与します。

DBA 権限およびCONNECT 権限を持っている HADB ユーザが，GRANT 文を実行してください。

HADB ユーザに，ユーザ権限およびスキーマ操作権限を付与する例を次に示します。

指定例 1

HADB ユーザ「ADBUSER02」に，DBA 権限，CONNECT 権限およびスキーマ定義権限を付与します。

```
GRANT DBA,CONNECT,SCHEMA TO "ADBUSER02"
```

指定例 2

HADB ユーザ「ADBUSER03」と「ADBUSER04」に，CONNECT 権限およびスキーマ定義権限を付与します。

```
GRANT CONNECT,SCHEMA TO "ADBUSER03","ADBUSER04"
```

GRANT 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『GRANT（権限の付与)』を参照してください。

重要

同一の HADB ユーザが監査管理権限とDBA 権限の両方を持つことはできません。そのため，監査管理権限を持っている HADB ユーザにDBA 権限を付与することはできません。

11.7.2　HADB ユーザが持っているユーザ権限とスキーマ操作権限を確認する方法

HADB ユーザが持っているユーザ権限およびスキーマ操作権限は，ディクショナリ表SQL_USERS を検索して確認します。

DBA 権限を持っている HADB ユーザは，すべての HADB ユーザが持っているユーザ権限およびスキーマ操作権限を確認できます。DBA 権限を持っていない HADB ユーザは，自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が持っているユーザ権限およびスキーマ操作権限だけしか確認できません。

HADB ユーザに付与したユーザ権限およびスキーマ操作権限を確認する例を次に示します。

指定例

DBA 権限およびCONNECT 権限を持った HADB ユーザが，作成されたすべての HADB ユーザに付与されたユーザ権限およびスキーマ操作権限を確認します。

```
SELECT "USER_NAME","DBA_PRIVILEGE","CONNECT_PRIVILEGE","SCHEMA_PRIVILEGE"
  FROM "MASTER"."SQL_USERS"
```

検索結果のうち，"DBA_PRIVILEGE"，"CONNECT_PRIVILEGE"，"SCHEMA_PRIVILEGE"が'Y'の場合は，対象のユーザ権限およびスキーマ操作権限を持っています。'N'の場合は，対象のユーザ権限およびスキーマ操作権限を持っていません。

ディクショナリ表SQL_USERS については，「付録 B.13　SQL_USERS の内容」を参照してください。

11.7.3　HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法

HADB ユーザに付与したユーザ権限とスキーマ操作権限は，定義系 SQL のREVOKE 文で取り消せます。

DBA 権限およびCONNECT 権限を持っている HADB ユーザが，REVOKE 文を実行してください。

HADB ユーザに付与したユーザ権限とスキーマ操作権限を取り消す例を次に示します。

指定例 1

HADB ユーザ「ADBUSER02」に付与したDBA 権限，CONNECT 権限およびスキーマ定義権限を取り消します。

```
REVOKE DBA,CONNECT,SCHEMA FROM "ADBUSER02" CASCADE
```

重要

スキーマ定義権限を取り消した場合，取り消し対象の HADB ユーザが所有しているスキーマおよび表も削除されるため，次に示す影響があります。

- REVOKE 文の実行によって削除される表に依存するビュー表（ほかのスキーマのビュー表）が，削除または無効化されます。
- REVOKE 文の実行によって削除される表を被参照表とする外部キー（ほかのスキーマの外部キー）も削除されます。

スキーマの削除については，「11.9.2　スキーマを削除する方法」を参照してください。スキーマに定義されている実表が更新不可状態だった場合，中断したコマンドによって作成された作業用一時ファイルが残ることがあります。

指定例 2

HADB ユーザ「ADBUSER03」と「ADBUSER04」に付与したCONNECT 権限およびスキーマ定義権限を取り消します。ただし，「ADBUSER03」と「ADBUSER04」がスキーマを所有している場合は，REVOKE 文の実行を中止します。

```
REVOKE CONNECT,SCHEMA FROM ”ADBUSER03”,”ADBUSER04” RESTRICT
```

ADBUSER03 はスキーマを所有していて，ADBUSER04 はスキーマを所有していない場合に上記のREVOKE 文を実行すると，ADBUSER03 およびADBUSER04 の両方に対して，REVOKE 文の実行を中止します。

REVOKE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『REVOKE（権限の取り消し)』を参照してください。

重要

監査権限を持っている HADB ユーザのCONNECT 権限およびスキーマ定義権限を取り消すことはできません。

11.8　アクセス権限の管理

ここでは，アクセス権限の管理方法について説明します。

11.8.1　HADB ユーザにアクセス権限を付与する方法

スキーマオブジェクトに対するアクセス権限をほかの HADB ユーザに付与するには，GRANT 文を実行します。

> **重要**
>
> アクセス権限を付与する HADB ユーザ（GRANT 文を実行する HADB ユーザ）は，そのアクセス権限の付与権を持っている必要があります。

ほかの HADB ユーザにアクセス権限を付与する方法を，例を使って説明します。

例題 1

HADB ユーザADBUSER01 とADBUSER02 に，表A.T1 に対するSELECT 権限，INSERT 権限，およびEXPORT TABLE 権限を付与します。

```
GRANT SELECT,INSERT,EXPORT TABLE ON "A"."T1" TO "ADBUSER01","ADBUSER02"
```

GRANT 文を実行する HADB ユーザは，表A.T1 に対するSELECT 権限，INSERT 権限，およびEXPORT TABLE 権限を付与権付きで持っている必要があります。

例題 2

HADB ユーザADBUSER01 に，表A.T1 に対するアクセス権限を付与します。

```
GRANT ALL PRIVILEGES ON "A"."T1" TO "ADBUSER01"
```

GRANT 文を実行する HADB ユーザは，表A.T1 に対するアクセス権限を付与権付きで持っている必要があります。

> **メモ**
>
> GRANT 文を実行する HADB ユーザが，表A.T1 に対する一部の種類のアクセス権限の付与権だけを持っている場合に，上記のGRANT 文を実行すると，付与権を持っているアクセス権限だけが付与されます。例えば，GRANT 文を実行する HADB ユーザが，SELECT 権限とINSERT 権限の付与権だけを持っている場合に，上記のGRANT 文を実行すると，HADB ユーザADBUSER01 には，SELECT 権限とINSERT 権限だけが付与されます。

例題 3

表A.T1 に対するSELECT 権限を全 HADB ユーザに許可します。

```
GRANT SELECT ON "A"."T1" TO PUBLIC
```

GRANT 文を実行する HADB ユーザは，表A.T1 の所有者である必要があります。

例題 4

HADB ユーザADBUSER01 に，表A.T1 に対するSELECT 権限を付与権付きで付与します。

```
GRANT SELECT ON "A"."T1" TO "ADBUSER01" WITH GRANT OPTION
```

アクセス権限を付与権付きで付与する場合は，WITH GRANT OPTION を指定します。

GRANT 文を実行する HADB ユーザは，表A.T1 に対するSELECT 権限を付与権付きで持っている必要があります。

GRANT 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『GRANT（権限の付与)』を参照してください。

11.8.2　HADB ユーザが持っているアクセス権限を確認する方法

HADB ユーザが持っているアクセス権限の種類は，ディクショナリ表SQL_TABLE_PRIVILEGES を検索して確認します。

なお，HADB ユーザが持っている権限によって，確認できる範囲が異なります。HADB ユーザが参照できるディクショナリ表の範囲については，「2.7.7　HADB ユーザが参照できるディクショナリ表とシステム表の範囲」を参照してください。

HADB ユーザが持っているアクセス権限の種類を確認する例を次に示します。

指定例

DBA 権限およびCONNECT 権限を持った HADB ユーザが，作成されたすべての HADB ユーザに付与されたアクセス権限の種類を確認します。

```
SELECT "GRANTOR","GRANTEE","TABLE_SCHEMA","TABLE_NAME",
       "SELECT_PRIVILEGE","INSERT_PRIVILEGE","UPDATE_PRIVILEGE",
       "DELETE_PRIVILEGE","TRUNCATE_PRIVILEGE",
       "REFERENCES_PRIVILEGE","IMPORT_TABLE_PRIVILEGE",
       "REBUILD_INDEX_PRIVILEGE","GET_COSTINFO_PRIVILEGE",
       "EXPORT_TABLE_PRIVILEGE","MERGE_CHUNK_PRIVILEGE",
       "CHANGE_CHUNK_COMMENT_PRIVILEGE","CHANGE_CHUNK_STATUS_PRIVILEGE",
       "ARCHIVE_CHUNK_PRIVILEGE","UNARCHIVE_CHUNK_PRIVILEGE"
  FROM "MASTER"."SQL_TABLE_PRIVILEGES"
```

[説明]

例えば，SELECT 権限を持っているかどうかは，SELECT_PRIVILEGE 列に表示される情報で確認できます。

- 'G' が表示された場合は，スキーマオブジェクトに対するSELECT 権限を付与権付きで持っています。
- 'Y' が表示された場合は，スキーマオブジェクトに対するSELECT 権限（付与権なし）を持っています。

- 'N'が表示された場合は，スキーマオブジェクトに対するSELECT 権限を持っていません。

スキーマオブジェクトのスキーマ名と表識別子は，TABLE_SCHEMA 列とTABLE_NAME 列に表示されます。

ディクショナリ表SQL_TABLE_PRIVILEGES については，「付録 B.18　SQL_TABLE_PRIVILEGES の内容」を参照してください。

11.8.3　HADB ユーザに付与したアクセス権限を取り消す方法

HADB ユーザに付与したアクセス権限を取り消すには，REVOKE 文を実行します。

> **重要**
>
> アクセス権限を付与した HADB ユーザだけが，そのアクセス権限を取り消すことができます。

HADB ユーザに付与したアクセス権限を取り消す方法を，例を使って説明します。

例題 1

HADB ユーザADBUSER01 に付与した，表A.T1 に対するSELECT 権限を取り消します。

```
REVOKE SELECT ON "A"."T1" FROM "ADBUSER01"
```

例題 2

HADB ユーザADBUSER01 に付与した，表A.T1 に対する全種類のアクセス権限を取り消します。

```
REVOKE ALL PRIVILEGES ON "A"."T1" FROM "ADBUSER01" RESTRICT
```

上記の例の場合，RESTRICT を指定しているため，次のどれかの条件を満たしているときは，REVOKE 文がエラーになります。

- ADBUSER01 が，表A.T1 を基表としたビュー表を定義している場合
- ADBUSER01 が，表A.T1 に対するアクセス権限をほかの HADB ユーザに付与している場合（ADBUSER01 が持っている表A.T1 に対するアクセス権限に依存権限がある場合）
- ADBUSER01 が，表A.T1 を被参照表にして定義した参照制約がある場合

例題 3

GRANT 文のPUBLIC 指定で全 HADB ユーザに許可した，表A.T1 に対するSELECT 権限の許可を取り消します。

```
REVOKE SELECT ON "A"."T1" FROM PUBLIC
```

REVOKE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『REVOKE（権限の取り消し)』を参照してください。

重要

- 取り消したアクセス権限に依存権限がある場合，依存権限も一緒に取り消されます。詳細については，「2.7.5　アクセス権限」の「(4)　アクセス権限の取り消し」を参照してください。
- ビュー表の基表に対するSELECT 権限が取り消された場合，その基表に依存するビュー表が無効化されます。また，付与権付きのSELECT 権限が取り消された場合，その付与権を使用して付与されたSELECT 権限も取り消されます。それに伴いビュー表が無効化されることがあります。詳細については，「2.7.6　ビュー表に対するアクセス権限」を参照してください。
- 表に対するREFERENCES 権限が取り消された場合，そのREFERENCES 権限を使用して定義した参照制約が削除されます。

11.8.4　HADB ユーザに付与したアクセス権限の付与権だけを取り消す方法

HADB ユーザに付与したアクセス権限の付与権だけを取り消すには，GRANT OPTION FOR を指定してREVOKE 文を実行します。

重要

付与権付きのアクセス権限を付与した HADB ユーザだけが，そのアクセス権限の付与権を取り消すことができます。

HADB ユーザに付与したアクセス権限の付与権だけを取り消す方法を，例を使って説明します。

例題

HADB ユーザADBUSER01 に付与した，表A.T1 に対するSELECT 権限の付与権だけを取り消します。

```
REVOKE GRANT OPTION FOR SELECT ON "A"."T1" FROM "ADBUSER01"
```

付与権だけを取り消す場合は，GRANT OPTION FOR を指定してREVOKE 文を実行します。

この場合，HADB ユーザADBUSER01 が持っている，表A.T1 に対するSELECT 権限は取り消されません。

REVOKE 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『REVOKE（権限の取り消し)』を参照してください。

重要

- アクセス権限の付与権を取り消した場合，その付与権を使用して付与したアクセス権限が取り消されます。詳細については，「2.7.5　アクセス権限」の「(4)　アクセス権限の取り消し」を参照してください。

- SELECT 権限の付与権が取り消された場合，その付与権を使用して付与されたSELECT 権限も取り消されます。それに伴いビュー表が無効化されることがあります。詳細については，「2.7.6　ビュー表に対するアクセス権限」を参照してください。

11.8.5　HADB ユーザが持っているアクセス権限を変更する方法

HADB ユーザが持っているアクセス権限を変更する場合は，不要なアクセス権限を取り消したあとで，必要なアクセス権限を付与する必要があります。

次の手順に従って，アクセス権限を変更してください。

手順

1. HADB ユーザが持っているアクセス権限を確認する
   「11.8.2　HADB ユーザが持っているアクセス権限を確認する方法」を参照して，持っているアクセス権限を確認してください。
2. HADB ユーザが持っている不要なアクセス権限を取り消す
   「11.8.3　HADB ユーザに付与したアクセス権限を取り消す方法」を参照して，REVOKE 文を実行してください。
3. HADB ユーザに必要なアクセス権限を付与する
   「11.8.1　HADB ユーザにアクセス権限を付与する方法」を参照して，GRANT 文を実行してください。

11.9　スキーマの運用

ここでは，スキーマの運用について説明します。

11.9.1　スキーマを定義する方法

スキーマを定義する場合は，CREATE SCHEMA 文を実行します。

CREATE SCHEMA 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『CREATE SCHEMA（スキーマの定義)』を参照してください。

11.9.2　スキーマを削除する方法

スキーマを削除する場合は，DROP SCHEMA 文を実行します。

DROP SCHEMA 文については，マニュアル『HADB　SQL リファレンス』の『定義系 SQL』の『DROP SCHEMA（スキーマの削除)』を参照してください。

重要

スキーマを削除すると，スキーマに定義されたスキーマオブジェクトもすべて削除されます。

削除するスキーマに実表が定義されている場合は，スキーマを削除する前に「11.1.17　実表を削除する方法」を確認してください。実表が更新不可状態のままスキーマを削除すると，中断したコマンドによって作成された作業用一時ファイルが残るおそれがあります。

11.10　データ用 DB エリアの運用

ここでは，データ用 DB エリア，およびデータ用 DB エリアファイルの運用について説明します。

> **メモ**
>
> 実表が格納されているデータ用 DB エリアを変更したい場合は，「11.1.16　実表が格納されているデータ用 DB エリアを変更する方法」を参照してください。
>
> インデクスが格納されているデータ用 DB エリアを変更したい場合は，「11.3.8　インデクスが格納されているデータ用 DB エリアを変更する方法」を参照してください。

11.10.1　データ用 DB エリアを追加する方法

ここでは，adbmodarea コマンドを使用して，新規にデータ用 DB エリアを追加する方法を説明します。adbmodarea コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』を参照してください。

データ用 DB エリアを追加する場合の指定例を次に示します。HADB 管理者が実行してください。

■指定例

データ用 DB エリアを追加します。

```
adbmodarea /HADB/server/conf/adbmodarea.opt
```

adbmodarea コマンドを実行する場合，DB エリア追加・変更オプションのadbaddarea オペランドを指定した DB エリア追加・変更オプションファイルを作成する必要があります。

DB エリア追加・変更オプションのadbaddarea オペランドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』の『adbmodarea コマンドの指定形式』を参照してください。

実表，B-tree インデクス，テキストインデクスまたはレンジインデクスを新たに定義する場合で，かつ次に示すケースに該当する場合は，データ用 DB エリアを追加することを検討してください。

- レンジインデクスを定義するとき
  レンジインデクス専用のデータ用 DB エリアを追加することを推奨します。
- グローバルバッファやデータ用 DB エリアファイルを，既存のデータ用 DB エリアと分けたいとき
  既存の実表，B-tree インデクス，テキストインデクスまたはレンジインデクスへの影響を分散するために，データ用 DB エリアを追加してください。
- 新たに実表を格納するのに適当なページサイズのデータ用 DB エリアがないとき

メモ

■追加したデータ用 DB エリアとグローバルバッファの関係

サーバ定義adbbuff オペランドの-o オプションを指定していた場合，追加したデータ用 DB エリアには，-o オプションの指定に従ってグローバルバッファが自動的に割り当てられます。

-o オプションを指定していなかった場合，追加したデータ用 DB エリアには，-o オプションを指定したときと同様のグローバルバッファが自動的に割り当てられます。

どちらの場合も必要な処理性能が得られないときは，追加したデータ用 DB エリアに割り当てるグローバルバッファを，adbbuff オペランドに指定してください。詳細については，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」，および「8.5.2　サーバ定義の変更方法」を参照してください。

また，必要に応じて，adbbuff オペランドの指定値を見直してください。詳細については，「13.2.2　SQL 文の実行時間を短縮する方法」を参照してください。

11.10.2　データ用 DB エリアを削除する方法

ここでは，adbmodarea コマンドを使用して，データ用 DB エリアを削除する方法を説明します。adbmodarea コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』を参照してください。

データ用 DB エリアを削除する場合の指定例を次に示します。HADB 管理者が実行してください。

■指定例

データ用 DB エリアを削除します。

```
adbmodarea /HADB/server/conf/adbmodarea.opt
```

adbmodarea コマンドを実行する場合，DB エリア追加・変更オプションのadbrmarea オペランドを指定した DB エリア追加・変更オプションファイルを作成する必要があります。

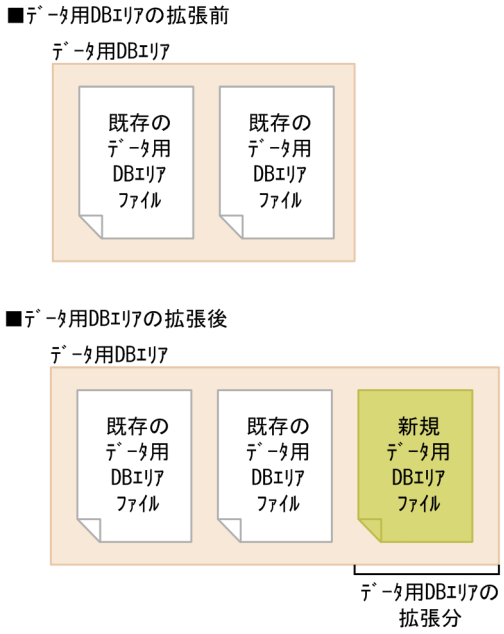
DB エリア追加・変更オプションのadbrmarea オペランドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』の『adbmodarea コマンドの指定形式』を参照してください。

11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）

ここでは，adbmodarea コマンドを使用して，既存のデータ用 DB エリアを拡張する方法について説明します。adbmodarea コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea (DB エリアの追加・変更)』を参照してください。

新たにデータ用 DB エリアファイルを追加して，既存のデータ用 DB エリアの容量を拡張したい場合に，adbmodarea コマンドを実行してください。データ用 DB エリアの拡張について，次の図に示します。

図 11-41　データ用 DB エリアの拡張の概要



データ用 DB エリアを拡張する場合の指定例を次に示します。HADB 管理者が実行してください。

■指定例

データ用 DB エリアを拡張します。

```
adbmodarea /HADB/server/conf/adbmodarea.opt
```

adbmodarea コマンドを実行する場合，DB エリア追加・変更オプションのadbexpandarea オペランドを指定した DB エリア追加・変更オプションファイルを作成する必要があります。

DB エリア追加・変更オプションのadbexpandarea オペランドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』の『adbmodarea コマンドの指定形式』を参照してください。

11.10.4　データ用 DB エリアを再初期化する方法

ここでは，特定のデータ用 DB エリアを再初期化する方法について説明します。

メモ

データベースの初期設定でデータ用 DB エリアを作成（初期化）したあとで，特定のデータ用 DB エリアをいったん削除し，同名のデータ用 DB エリアとして再度追加することを，**データ用 DB エリアの再初期化**といいます。

特定のデータ用 DB エリアを再初期化したい場合は，adbmodarea コマンドを使用します。ただし，adbmodarea コマンドは一度の操作で再初期化を実施できません。そのため，次の手順に従って，データ用 DB エリアの再初期化を行ってください。

手順

1. 再初期化したいデータ用 DB エリアをいったん削除する

   データ用 DB エリアを削除する場合，「11.10.2　データ用 DB エリアを削除する方法」を参照してください。

2. 同名のデータ用 DB エリアとして再度追加する

   データ用 DB エリアを追加する場合，「11.10.1　データ用 DB エリアを追加する方法」を参照してください。

adbmodarea コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodarea（DB エリアの追加・変更)』を参照してください。

> **重要**
>
> 特定のデータ用 DB エリアを再初期化したい場合，adbinit コマンドは使用しないでください。
>
> adbinit コマンドは，データベースの初期設定を行うために，すべての DB エリアを初期化するコマンドです。すでにデータ用 DB エリアが存在する状態で，adbinit コマンドを実行してしまうと，すべての DB エリア（データ用 DB エリア以外の DB エリアも含む）が削除されてしまいます。

11.10.5　データ用 DB エリアファイルの格納場所を変更する方法

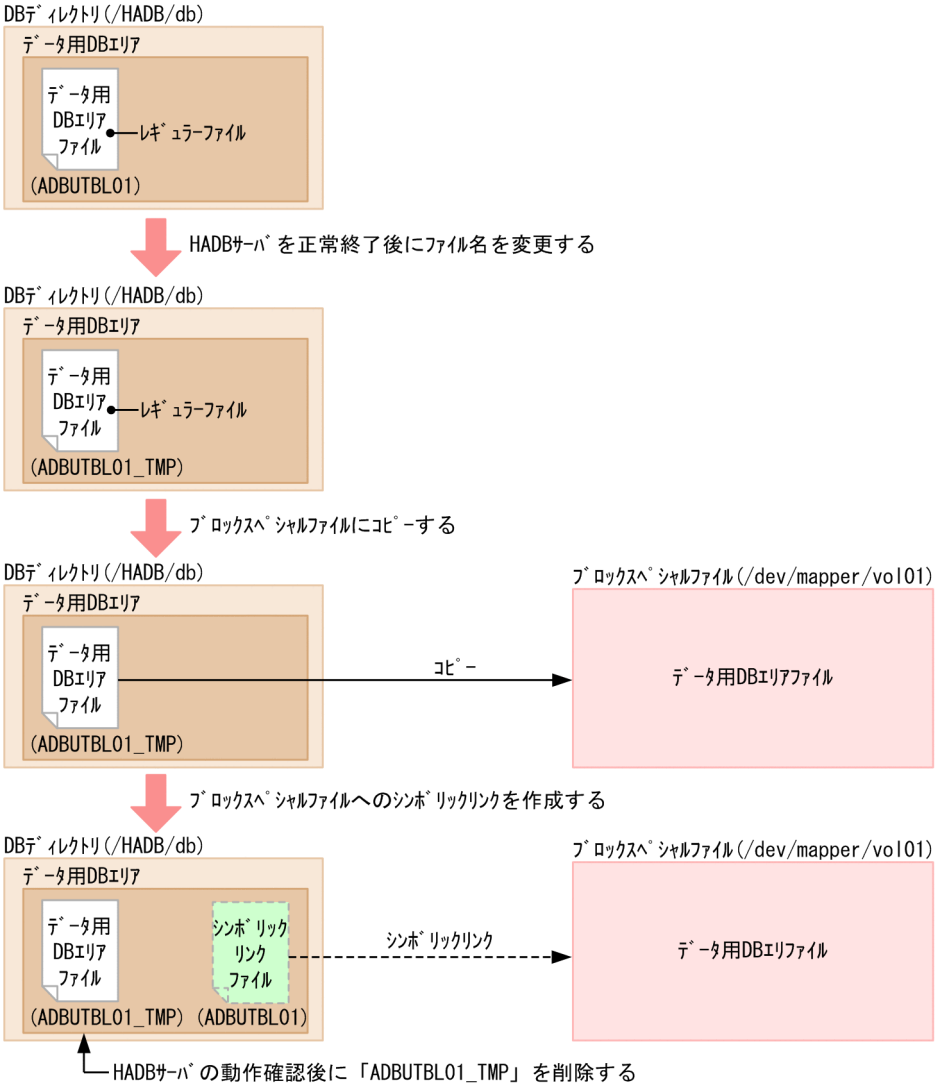
ここでは，データ用 DB エリアを構成するデータ用 DB エリアファイルの格納場所を変更する方法について説明します。

> **重要**
>
> 1 つのデータ用 DB エリアが複数のデータ用 DB エリアファイルで構成されている場合，データ用 DB エリアファイルの格納場所を変更するときには，変更先のファイルにレギュラーファイルとブロックスペシャルファイルが混在しないようにしてください。どちらかのファイルに統一してください。

(1)　データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する

必要なサイズを割り当てたブロックスペシャルファイルを準備したあとで，データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更します。

図 11-42　データ用 DB エリアファイルをブロックスペシャルファイルに変更する方法



手順

1. adbstop コマンドで HADB サーバを正常終了する

   adbstop コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstop（HADB サーバの終了)』を参照してください。

2. データ用 DB エリアファイルのファイル名を変更する

   レギュラーファイルから，ブロックスペシャルファイルに変更したいデータ用 DB エリアファイルのファイル名を変更します。OS のmv コマンドで，ファイル名を変更してください。実行例を次に示します。

   ■コマンドの実行例

   ```
   mv /HADB/db/ADBUTBL01 /HADB/db/ADBUTBL01_TMP
   ```

   この例では，ファイル名をADBUTBL01 からADBUTBL01_TMP に変更しています。

3. ファイル名を変更したデータ用 DB エリアファイルをブロックスペシャルファイルにコピーする

手順 2.でファイル名を変更したデータ用 DB エリアファイルを，ブロックスペシャルファイルにコピーします。OS のdd コマンドで，データ用 DB エリアファイルをコピーしてください。実行例を次に示します。

■コマンドの実行例

```
dd if=/HADB/db/ADBUTBL01_TMP of=/dev/mapper/vol01 bs=524288
```

この例では，レギュラーファイルのデータ用 DB エリアファイル（/HADB/db/ADBUTBL01_TMP）を，ブロックスペシャルファイル（/dev/mapper/vol01）にコピーしています。

4. 変更前のファイル名でブロックスペシャルファイルへのシンボリックリンクを作成する

手順 3.でデータ用 DB エリアファイルをコピーしたブロックスペシャルファイルに対して，シンボリックリンクを作成します。その際，手順 2.で変更する前のデータ用 DB エリアファイル名で，ブロックスペシャルファイルへのシンボリックリンクを作成する必要があります。OS のln コマンドで，シンボリックリンクを作成してください。実行例を次に示します。

■コマンドの実行例

```
ln -s /dev/mapper/vol01 /HADB/db/ADBUTBL01
```

この例では，変更前のファイル名（/HADB/db/ADBUTBL01）で，ブロックスペシャルファイル（/dev/mapper/vol01）へのシンボリックリンクを作成しています。

5. adbstart コマンドで HADB サーバを開始する

adbstart コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

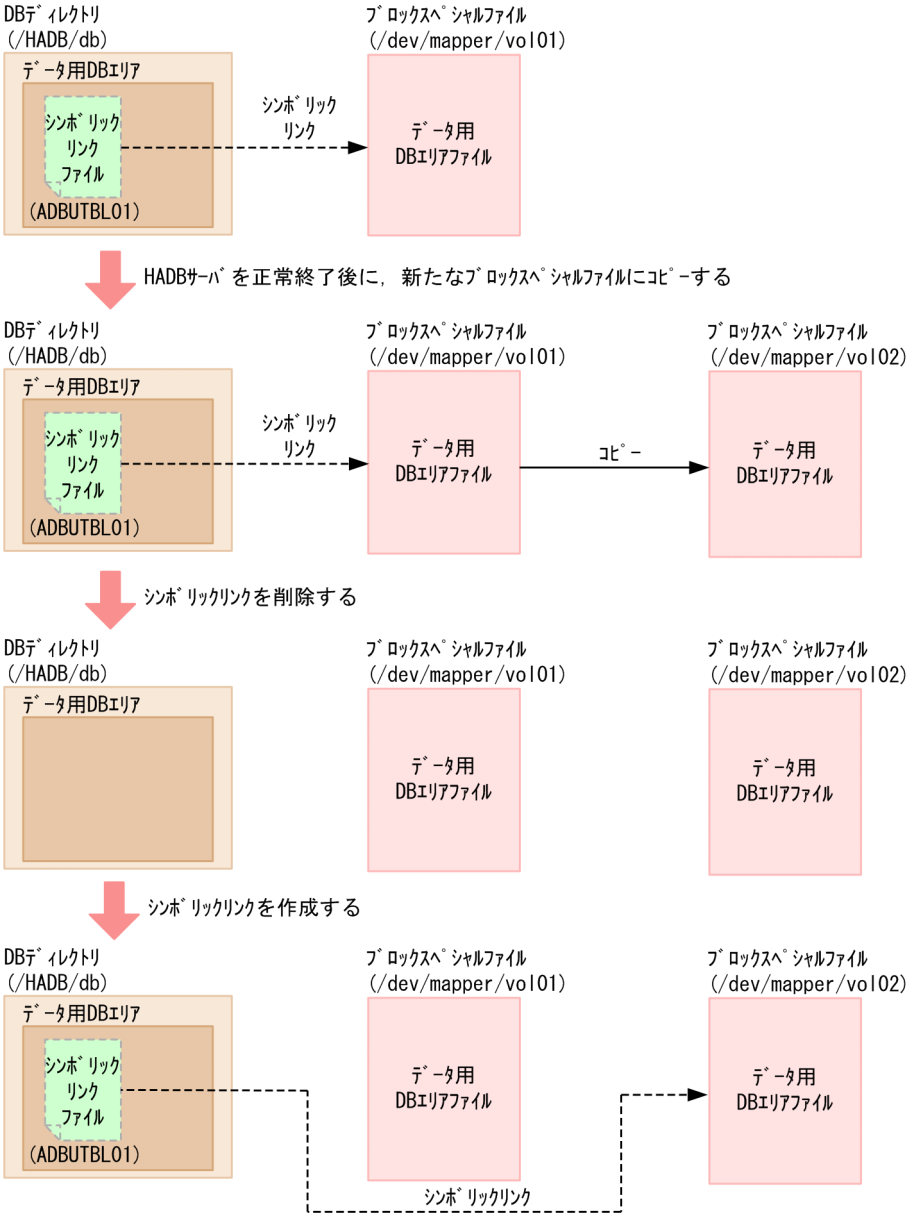
6. HADB サーバの動作確認後に不要なレギュラーファイルを削除する

データ用 DB エリアファイルを，レギュラーファイルからブロックスペシャルファイルに変更して，問題がないことを確認してください。問題がない場合は，手順 2.でファイル名を変更したデータ用 DB エリアファイル（レギュラーファイル）は不要となります。OS のコマンドで削除してください。

(2) データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する

必要なサイズを割り当てたブロックスペシャルファイルを準備したあとで，データ用 DB エリアファイルの格納場所を変更します（新たなブロックスペシャルファイルに変更します)。

図 11-43　データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する方法



手順

1. adbstop コマンドで HADB サーバを正常終了する

   adbstop コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstop（HADB サーバの終了)』を参照してください。

2. データ用 DB エリアファイルの実体を新たなブロックスペシャルファイルにコピーする（格納場所を変更する）

   格納場所を変更したいデータ用 DB エリアファイルの実体（シンボリックリンクのリンク先にあるブロックスペシャルファイル）を，新たなブロックスペシャルファイルにコピーします。OS のdd コマンドで，データ用 DB エリアファイルの実体をコピーしてください。実行例を次に示します。

   ■コマンドの実行例

```
dd if=/dev/mapper/vol01 of=/dev/mapper/vol02 bs=524288
```

この例では，データ用 DB エリアファイルの実体（/dev/mapper/vol01）を，別のブロックスペシャルファイル（/dev/mapper/vol02）にコピーしています。

3. 変更前の格納場所にあるブロックスペシャルファイルへのシンボリックリンクを削除する

データ用 DB エリアファイルの格納場所を変更するために，手順 2.でコピー元となったブロックスペシャルファイルへのシンボリックリンクを削除します。OS の rm コマンドで，シンボリックリンク元のシンボリックリンクファイルを削除してください。実行例を次に示します。

■コマンドの実行例

```
rm /HADB/db/ADBUTBL01
```

この例では，変更前の格納場所にあるブロックスペシャルファイル（/dev/mapper/vol01）へのシンボリックリンクを削除するため，シンボリックリンク元のシンボリックリンクファイル（/HADB/db/ADBUTBL01）を削除しています。

4. 変更後の格納場所にあるブロックスペシャルファイルへのシンボリックリンクを作成する

データ用 DB エリアファイルの格納場所を変更するために，手順 2.でコピー先となったブロックスペシャルファイルへのシンボリックリンクを作成します。その際，手順 3.で削除したシンボリックリンクファイル名で，ブロックスペシャルファイルへのシンボリックリンクを作成する必要があります。OS の ln コマンドで，シンボリックリンクを作成してください。実行例を次に示します。

■コマンドの実行例

```
ln -s /dev/mapper/vol02 /HADB/db/ADBUTBL01
```

この例では，削除したシンボリックリンクファイル名（/HADB/db/ADBUTBL01）で，ブロックスペシャルファイル（/dev/mapper/vol02）へのシンボリックリンクを作成しています。

5. adbstart コマンドで HADB サーバを開始する

adbstart コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

6. HADB サーバの動作を確認する

データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更して，問題がないことを確認してください。

11.10.6　データ用 DB エリアの空き容量を確保する方法

ここでは，データ用 DB エリアの空き容量を確保する方法について説明します。

重要

データ用 DB エリアの空き容量を確保する前に，データ用 DB エリアを拡張できるかどうかを検討してください。データ用 DB エリアの拡張については，「11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照してください。

データ用 DB エリアの拡張ができない場合は，次に示す対処方法 1，または対処方法 2 を実施して，データ用 DB エリアの空き容量を確保してください。対処方法 1，または対処方法 2 を実施してもデータ用 DB エリアの空き容量を確保できないときは，対処方法 3 を実施してください。

対処方法 1

チャンクのアーカイブを行います。チャンク内のデータを圧縮して，データ用 DB エリアの空き容量を確保します。

詳細については，「(1)　データ用 DB エリアの空き容量を確保する方法（データの圧縮）」を参照してください。

対処方法 2

実表の再編成を行います。無効な領域を削除したり，データを圧縮したりして，データ用 DB エリアの空き容量を確保します。

詳細については，「(2)　データ用 DB エリアの空き容量を確保する方法（実表の再編成）」を参照してください。

対処方法 3

不要なデータを削除します。不要なデータを削除して，データ用 DB エリアの空き容量を確保します。

詳細については，「(3)　データ用 DB エリアの空き容量を確保する方法（不要なデータの削除）」を参照してください。

(1)　データ用 DB エリアの空き容量を確保する方法（データの圧縮）

空き容量を確保したいデータ用 DB エリアにアーカイブマルチチャンク表がある場合は，チャンクをアーカイブすることを検討してください。adbarchivechunk コマンドで，チャンクをアーカイブ状態にしてチャンク内のデータを圧縮すると，データ用 DB エリアの空き容量を増やすことができます。

メモ

チャンクのアーカイブについては，「2.15.1　チャンクアーカイブ機能とは」を参照してください。

チャンク内のデータを圧縮する手順を次に示します。

手順

1. アーカイブマルチチャンク表のスキーマ名と表識別子を調査する

次に示すSELECT 文を実行して，データ用 DB エリアに格納されているアーカイブマルチチャンク表のスキーマ名と表識別子を調査してください。

指定例

```
SELECT "MASTER"."SQL_DIV_TABLE"."TABLE_SCHEMA",
       "MASTER"."SQL_DIV_TABLE"."TABLE_NAME"
  FROM "MASTER"."SQL_DIV_TABLE","MASTER"."SQL_TABLES"
    WHERE "DBAREA_ID" = (SELECT "DBAREA_ID"
                           FROM "MASTER"."SQL_DBAREAS"
                             WHERE "DBAREA_NAME" = ?)
      AND "MASTER"."SQL_DIV_TABLE"."TABLE_SCHEMA"
            = "MASTER"."SQL_TABLES"."TABLE_SCHEMA"
      AND "MASTER"."SQL_DIV_TABLE"."TABLE_NAME"
            = "MASTER"."SQL_TABLES"."TABLE_NAME"
      AND "IS_ARCHIVABLE" = 'Y'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。入力要求には，空き容量を確保したいデータ用 DB エリアの DB エリア名を入力してください。

2. アーカイブ状態にするチャンクを検討する

チャンクをアーカイブ状態にして，データ用 DB エリアの空き容量を確保します。アーカイブマルチチャンク表内の**アーカイブ状態ではないチャンク**の中から，アーカイブするチャンクを検討してください。

チャンクをアーカイブすると，アーカイブされていないチャンク内のデータを検索するときに比べて，検索時間が長くなります。そのため，検索する頻度が少ないチャンクをアーカイブすることを推奨します。例えば，作成日時が新しいチャンク内のデータを検索する頻度が多い場合は，作成日時が古いチャンクをアーカイブしてください。

アーカイブするチャンクの情報を確認する場合は，-d used オプションを指定したadbdbstatus コマンドを実行して，**DB エリア，表とインデクスの使用量情報**を確認してください。または，システム表の STATUS_CHUNKS 表をSELECT 文で検索してください。詳細については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」の「(1)　adbdbstatus コマンドを実行する場合」または「(3)　システム表 STATUS_CHUNKS を検索する場合」を参照してください。

3. チャンクをアーカイブ状態にして確保できる空き容量を調査する

チャンクをアーカイブ状態にすると，そのチャンクの使用量分だけ，データ用 DB エリアの空き容量を確保できます。チャンクをアーカイブ状態にして確保できる空き容量を確認する場合は，-d used オプションを指定したadbdbstatus コマンドを実行してください。**DB エリア，表とインデクスの使用量情報**が出力されたら，Chunk_ID をキーにして，Used_segments とUsed_pages を確認してください。

詳細については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」の「(1)　adbdbstatus コマンドを実行する場合」を参照してください。

4. チャンクをアーカイブ状態にしてデータを圧縮する

adbarchivechunk コマンドを実行して，チャンクをアーカイブします（チャンク内のデータを圧縮します）。チャンクをアーカイブする方法については，「11.4.17　チャンクをアーカイブする方法（アーカイブマルチチャンク表使用時）」を参照してください。

(2)　データ用 DB エリアの空き容量を確保する方法（実表の再編成）

実表に対して次の作業を繰り返し実行した場合，実表を再編成することを検討してください。データ用 DB エリアの空き容量を確保できる可能性があります。

- ローストア表に対する SQL 文を使用した行の更新，および行の削除
- カラムストア表に対する SQL 文を使用した行の追加，行の更新，および行の削除
- マルチチャンク表に対するadbimport コマンドを使用したバックグラウンドインポート

メモ

実表を再編成すると，データ用 DB エリアの空き容量を確保できる理由を次に示します。

- SQL 文による行の更新，行の削除，およびadbimport コマンドによるバックグラウンドインポートを繰り返し実行すると，無効な行データや再利用されない空き領域が増加することがあります。実表を再編成すると，無効な行データや再利用されない空き領域が解放されます。そのため，データ用 DB エリアの空き容量を確保できる可能性があります。
- カラムストア表に対して SQL 文を使用した行の追加を実行すると，カラムストア形式ではなくローストア形式で追加されます。ローストア形式で追加されたデータは圧縮されません。実表やインデクスを再編成すると，カラムストア形式でデータが追加されます。カラムストア形式で追加されたデータは圧縮されるため，データ用 DB エリアの空き容量を確保できる可能性があります。

実表を再編成する手順を次に示します。

手順

1. 確保できる空き容量を調査する

   実表を再編成すると確保できる空き容量を，次の計算式から見積もってください。

   計算式（実表の再編成で確保できる空き容量）（単位：メガバイト）

   実表の再編成で確保できる空き容量 ＝
   再編成前の実表の使用量 － *再編成後の実表の使用量*

   再編成前の実表の使用量（単位：メガバイト）

   「10.9.2　実表の状態や使用量を確認したい場合」の「(2)　実表の使用量を確認する場合」を参照して，*再編成前の実表の使用量*を確認してください。

   再編成後の実表の使用量（単位：メガバイト）

   「5.8　データ用 DB エリアの容量見積もり」を参照して見積もってください。なお，参照先の計算式の単位はキロバイトです。そのため，メガバイトに換算してください。

   また，実表を再編成すると，実表に定義されたインデクスも再編成されます。インデクスを再編成すると確保できる空き容量を，次の計算式から見積もってください。

計算式（インデクスの再編成で確保できる空き容量）（単位：メガバイト）

```
インデクスの再編成で確保できる空き容量 ＝
    再編成前のインデクスの使用量 － 再編成後のインデクスの使用量
```

再編成前のインデクスの使用量（単位：メガバイト）

次の個所を参照して，*再編成前のインデクスの使用量*を確認してください。

- 「10.9.3　B-tree インデクスの状態や使用量を確認したい場合」の「(3)　B-tree インデクスの使用量を確認する場合」
- 「10.9.4　テキストインデクスの状態や使用量を確認したい場合」の「(2)　テキストインデクスの使用量を確認する場合」
- 「10.9.5　レンジインデクスの状態や使用量を確認したい場合」の「(2)　レンジインデクスの使用量を確認する場合」

再編成後のインデクスの使用量（単位：メガバイト）

「5.8　データ用 DB エリアの容量見積もり」を参照して見積もってください。なお，参照先の計算式の単位はキロバイトです。そのため，メガバイトに換算してください。

2. 実表を再編成する

次の個所を参照して，実表を再編成してください。実表を再編成すると，定義されているインデクスも再編成されます。なお，データの格納効率を求める必要はありません。

- 「11.1.11　シングルチャンク表を再編成する方法」
- 「11.4.14　マルチチャンク表を再編成する方法（チャンク単位で再編成する場合）」
- 「11.4.15　マルチチャンク表を再編成する方法（表全体を再編成する場合）」
- 「11.4.16　マルチチャンク表を再編成する方法（サンプルシェルを使用して再編成する場合）」

重要

- マルチチャンク表の場合，不要なチャンクがあるときは，再編成する前に不要なチャンクの削除を検討してください。不要なチャンクの削除については，「(3)　データ用 DB エリアの空き容量を確保する方法（不要なデータの削除）」を参照してください。
- 実表の再編成では，一時的にデータを削除します。そのため，再編成中は，実表のデータを参照できません。
- 大量のデータを格納している実表の場合，再編成のときにエクスポートするデータサイズが大きくなります。また，再編成にも時間が掛かります。データをエクスポートするための領域がないときや，再編成に必要な時間を確保できないときは，別の方法でデータ用 DB エリアの空き容量を確保することを検討してください。

(3)　データ用 DB エリアの空き容量を確保する方法（不要なデータの削除）

ここでは，不要なデータを削除して，データ用 DB エリアの空き容量を確保する方法について説明します。

次に示す方法ではデータ用 DB エリアの空き容量を確保できない場合，不要なデータを削除することを検討してください。

- 「(1)　データ用 DB エリアの空き容量を確保する方法（データの圧縮）」
- 「(2)　データ用 DB エリアの空き容量を確保する方法（実表の再編成）」

次に示す不要なデータを削除することで，データ用 DB エリアの空き容量を確保できます。

- 不要なインデクス
- マルチチャンク表にある不要なチャンク

> **重要**
>
> 不要なデータの削除には次のリスクが伴います。
>
> - 運用誤りによるデータの損失
> - SQL 文やコマンドの性能の低下
>
> そのため，各削除処理の留意事項を確認した上で，適用を検討してください。

■空き容量を確保したいデータ用 DB エリアに不要なインデクスがある場合

不要なインデクスを削除すると空き容量を確保できます。手順を次に示します。

手順

1. 削除するインデクスを検討する

   削除するインデクスを検討する場合は，**アクセスパスの統計情報**から，インデクスの使用有無を確認することを推奨します。

   SQL トレース機能を有効にしている場合，実行した SQL 文の終了時に出力される**アクセスパスの統計情報**を確認すると，SQL 文がインデクスを使用しているかどうかを確認できます。実行している SQL 文が使用していないインデクスを削除するようにしてください。

   SQL トレース機能については，「10.11　SQL トレース機能の運用」を参照してください。アクセスパスの統計情報については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」を参照してください。

2. 確保できる空き容量を調査する

   削除するインデクスの使用量分だけ，データ用 DB エリアの空き容量を確保できます。インデクスの使用量の確認については，次の個所を参照してください。

   - 「10.9.3　B-tree インデクスの状態や使用量を確認したい場合」の「(3)　B-tree インデクスの使用量を確認する場合」
   - 「10.9.4　テキストインデクスの状態や使用量を確認したい場合」の「(2)　テキストインデクスの使用量を確認する場合」
   - 「10.9.5　レンジインデクスの状態や使用量を確認したい場合」の「(2)　レンジインデクスの使用量を確認する場合」

3. インデクスを削除する

「11.3.9　インデクスの削除」を参照して，不要なインデクスを削除してください。

留意事項

- インデクスを削除した場合，削除したインデクスを使用していたSQL文の性能が低下するおそれがあります。
- インデクスの削除後に同じインデクスを再定義する必要がある場合，インデクスを削除する前にインデクスの再定義に必要な定義情報を取得しておいてください。定義情報の取得については，「付録B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。

■空き容量を確保したいデータ用DBエリアに不要なチャンクを含むマルチチャンク表がある場合

PURGE CHUNK文を使用してマルチチャンク表の不要なチャンクを削除すると空き容量を確保できます。手順を次に示します。

手順

1. 削除するチャンクを検討する

   マルチチャンク表内のチャンクの状態を確認して，削除するチャンクを検討してください。例えば，次に示すチャンクは不要である可能性が高いため，優先的に削除するようにしてください。

   - 削除仕掛中のチャンク
   - アーカイブ状態ではない待機状態のチャンク
   - 検索頻度が極端に少ないチャンク（作成日時が古いチャンクなど）

   チャンクの情報を確認する場合は，-d usedオプションを指定したadbdbstatusコマンドを実行して，**DBエリア，表とインデクスの使用量情報**を確認してください。または，システム表のSTATUS_CHUNKS表をSELECT文で検索してください。詳細については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」の「(1)　adbdbstatusコマンドを実行する場合」または「(3)　システム表STATUS_CHUNKSを検索する場合」を参照してください。

2. 確保できる空き容量を調査する

   削除するチャンクの使用量分だけ，データ用DBエリアの空き容量を確保できます。チャンクの使用量を確認する場合は，-d usedオプションを指定したadbdbstatusコマンドを実行してください。**DBエリア，表とインデクスの使用量情報**が出力されたら，Chunk_IDをキーにして，Used_segmentsとUsed_pagesを確認してください。

   詳細については，「11.4.8　チャンクの状態や作成されているチャンク数を確認する方法」の「(1) adbdbstatusコマンドを実行する場合」を参照してください。

3. チャンクを削除する

   「11.4.6　チャンク単位にデータを削除する方法」を参照して，不要なチャンクを削除してください。

留意事項

- 削除後に参照する可能性があるチャンク内のデータが存在する場合は，チャンクを削除する前にエクスポートしておいてください。チャンク内のデータをエクスポートする方法については，「11.4.5 チャンク単位にデータをエクスポートする方法」を参照してください。

- アーカイブ状態のチャンクを削除してもデータ用 DB エリアの空き容量を確保できません。

11.10.7　データ用 DB エリアの使用率を監視する方法

データ用 DB エリアの使用率を監視することができます。サーバ定義のadb_dbarea_wrn_pnt_use_rate オペランドを指定すると，データ用 DB エリアの使用率がこのオペランドの指定値に達したときに，警告メッセージKFAA41230-W がメッセージログファイルに出力されます。

（例）

```
set adb_dbarea_wrn_pnt_use_rate = 30,60,90
```

この場合，データ用 DB エリアの使用率が 30%，60%，90%に達したときに，KFAA41230-W メッセージが出力されます。

重要

- レギュラーファイルで構成されているデータ用 DB エリアは，使用率の監視対象にはなりません。
- データ用 DB エリア以外の DB エリア（ディクショナリ用 DB エリアなど）は，使用率の監視対象にはなりません。

■監視対象から外したいデータ用 DB エリアがある場合

adb_dbarea_wrn_pnt_use_rate オペランドを指定すると，ブロックスペシャルファイルで構成されているすべてのデータ用 DB エリアが使用率の監視対象になります。この場合，KFAA41230-W メッセージが頻繁に出力されることがあります。出力されるKFAA41230-W メッセージの量を減らしたい場合は，サーバ定義のadb_dbarea_wrn_pnt_except_name オペランドを指定してください。このオペランドには，使用率の監視対象から外したいデータ用 DB エリアを指定します。このオペランドに指定したデータ用 DB エリアの使用率が，adb_dbarea_wrn_pnt_use_rate オペランドの指定値に達しても，KFAA41230-W メッセージは出力されません。

メモ

データ用 DB エリアの使用率は，新規にセグメントが確保されて割り当てられるときに計算されます。このとき，データ用 DB エリアの使用率がadb_dbarea_wrn_pnt_use_rate オペランドの指定値に達したときに，KFAA41230-W メッセージが出力されます。

11.11　作業表用 DB エリアの運用

ここでは，作業表用 DB エリア，および作業表用 DB エリアファイルの運用について説明します。

11.11.1　作業表用 DB エリアファイルの格納場所を変更する方法

作業表用 DB エリアファイルがブロックスペシャルファイルの場合は，作業表用 DB エリアファイルの格納場所を，サーバ定義adb_blk_path_wrk オペランドで変更できます。また，作業表用 DB エリアファイルを，レギュラーファイルからブロックスペシャルファイルに変更することもできます。手順を次に示します。

手順

1. adbstop コマンドで HADB サーバを正常終了する

   adbstop コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstop（HADB サーバの終了)』を参照してください。

2. サーバ定義adb_blk_path_wrk オペランドに作業表用 DB エリアファイルの変更後のブロックスペシャルファイルを指定する

   サーバ定義adb_blk_path_wrk オペランドについては，「7.2.1　システム構成に関するオペランド（set 形式)」の adb_blk_path_wrk オペランドを参照してください。

3. adbstart コマンドで HADB サーバを開始する

   adbstart コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

11.11.2　作業表用 DB エリアファイルの容量を小さくする方法

作業表を作成する SQL 文を実行する場合，SQL 文の内容によっては，作業表の容量が非常に大きくなることがあります。作業表用 DB エリアファイルがレギュラーファイルの場合は，作業表の容量によって，レギュラーファイルの容量が非常に大きくなり，ディスクの空き容量が少なくなるおそれがあります。

作業表用 DB エリアファイルの容量が大きい場合は，作業表用 DB エリアファイルを削除してください。作業表用 DB エリアファイルを削除する方法は，「15.3.1　DB エリアファイルの容量増加が原因の場合」の「(3)　作業表用 DB エリアファイルのサイズを小さくする」を参照してください。

作業表用 DB エリアファイルの格納先は，DB ディレクトリ下（$DBDIR/ADBWRK）になります。

> **重要**
>
> 作業表用 DB エリアファイルの容量が大き過ぎて問題がある場合は，SQL 文を実行する前に，SQL 文の内容を見直すことを検討してください。

11.12　バッファの運用

ここでは，バッファの運用について説明します。

11.12.1　ローカル作業表用バッファを変更する方法

HADB サーバの稼働中に，ローカル作業表用バッファのページ数を変更する方法について説明します。

サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドで指定したローカル作業表用バッファのページ数は，adbmodbuff コマンドで変更できます。

ローカル作業表用バッファのページ数を，adbmodbuff コマンドで変更する方法を次に示します。

手順

1. サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値を確認する

   サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドで指定したローカル作業表用バッファのページ数を，adbls -d lbuf コマンドで確認してください。

   adbls -d lbuf コマンドの実行例を次に示します。

   コマンドの指定例

   ```
   adbls -d lbuf
   ```

   adbls -d lbuf コマンドを実行すると，次のような出力結果が表示されます。

   実行したコマンドの出力結果の例

   ```
   CNUMBER    PAGE_SIZE PAGE_NUM
   0          256       128 S
   1          256       128 S
   ```

   サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドで指定したローカル作業表用バッファのページ数は，CNUMBER が0 の行にあるPAGE_NUM に出力された内容になります。

2. ローカル作業表用バッファのページ数を変更する

   手順 1.で確認した内容を踏まえて，サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドで指定したローカル作業表用バッファのページ数を，adbmodbuff コマンドで変更してください。

これで，HADB サーバの稼働中に，サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドで指定したローカル作業表用バッファのページ数を変更できます。

なお，adbmodbuff コマンドで変更した内容は，HADB サーバ終了時に失われます。変更した内容を次回起動時も引き継ぐ場合は，変更した内容をサーバ定義に反映してください。サーバ定義の変更方法については，「8.5.2　サーバ定義の変更方法」を参照してください。

また，次に該当する場合は，ローカル作業表用バッファのページ数をadbmodbuff コマンドで変更できません。

- **クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドを指定した場合**

  サーバ定義よりもクライアント定義の指定値が優先されるため，adbmodbuff コマンドを実行しても，使用されるローカル作業表用バッファのページ数を変更できません。

- **エクスポートオプションadb_export_wrktbl_blk_num を指定した場合**

  サーバ定義よりもエクスポートオプションの指定値が優先されるため，adbmodbuff コマンドを実行しても，使用されるローカル作業表用バッファのページ数を変更できません。

メモ

- サーバ定義adb_dbbuff_wrktbl_clt_blk_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_dbbuff_wrktbl_clt_blk_num オペランドを参照してください。
- adbmodbuff コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbmodbuff（バッファの変更）』を参照してください。
- adbls -d lbuf コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d lbuf（ローカル作業表用バッファの情報表示）』を参照してください。

11.13　クライアントグループの設定内容の確認

クライアントグループ機能を適用して設定した，クライアントグループおよびコマンドグループに関する情報を確認する方法について説明します。

11.13.1　AP を実行した HADB クライアントが所属しているグループを確認したい場合

AP を実行した HADB クライアントが所属しているグループを確認するには，adbls -d cnct コマンドを実行してください。

実行結果の「CLIENT_GROUP」列に表示された内容から，AP を実行した HADB クライアントが所属しているグループを確認できます。なお，AP を実行した HADB クライアントがどのグループにも所属していない場合は，「CLIENT_GROUP」列には何も表示されません。

adbls -d cnct コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示)』を参照してください。

11.13.2　リアルスレッドを使用している HADB クライアントが所属しているグループを確認したい場合

リアルスレッドを使用している HADB クライアントが所属しているグループを確認するには，adbls -d thd コマンドを実行してください。

実行結果の「CLIENT_GROUP」列に表示された内容から，リアルスレッドを使用している HADB クライアントが所属しているグループを確認できます。なお，HADB クライアントがどのグループにも所属していない場合は，「CLIENT_GROUP」列には何も表示されません。

adbls -d thd コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d thd（スレッドの状態表示)』を参照してください。

11.13.3　各グループに設定した接続数や処理リアルスレッド数を確認したい場合

adbls -d cltgrp コマンドを実行すると，次のことが確認できます。

- 各グループに設定されている最大同時接続数
- 各グループに設定されている接続保証数
- 各グループの HADB サーバへの現在の接続数

- 各グループに設定されている処理リアルスレッドの最大数
- 各グループに設定されている処理リアルスレッドの保証数
- 各グループで使用中の処理リアルスレッド数
- 各グループに設定されている最大同時接続数に関する警告メッセージの出力契機
- 各グループに設定されている最大同時接続数に関する警告メッセージの出力済み状態のリセット契機

`adbls -d cltgrp` コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cltgrp (クライアントグループ・コマンドグループの情報表示)』を参照してください。

11.14　クライアント定義の集中管理の運用

クライアント定義の集中管理機能を使用して，各 HADB クライアントが管理しているクライアント定義を，HADB サーバで一括して管理する方法について説明します。

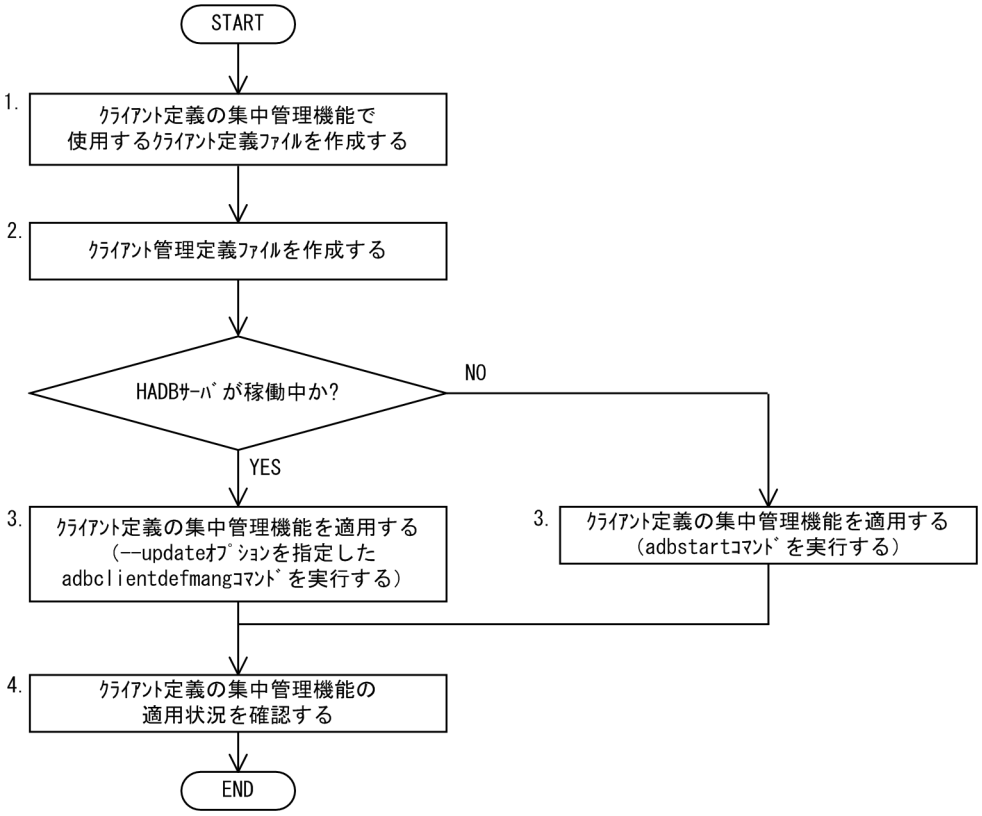
11.14.1　クライアント定義の集中管理機能を使用する流れ

クライアント定義の集中管理機能を使用する流れについて説明します。

(1)　新たにクライアント定義の集中管理機能を使用する場合の流れ

新たにクライアント定義の集中管理機能を使用する場合の流れを，次の図に示します。

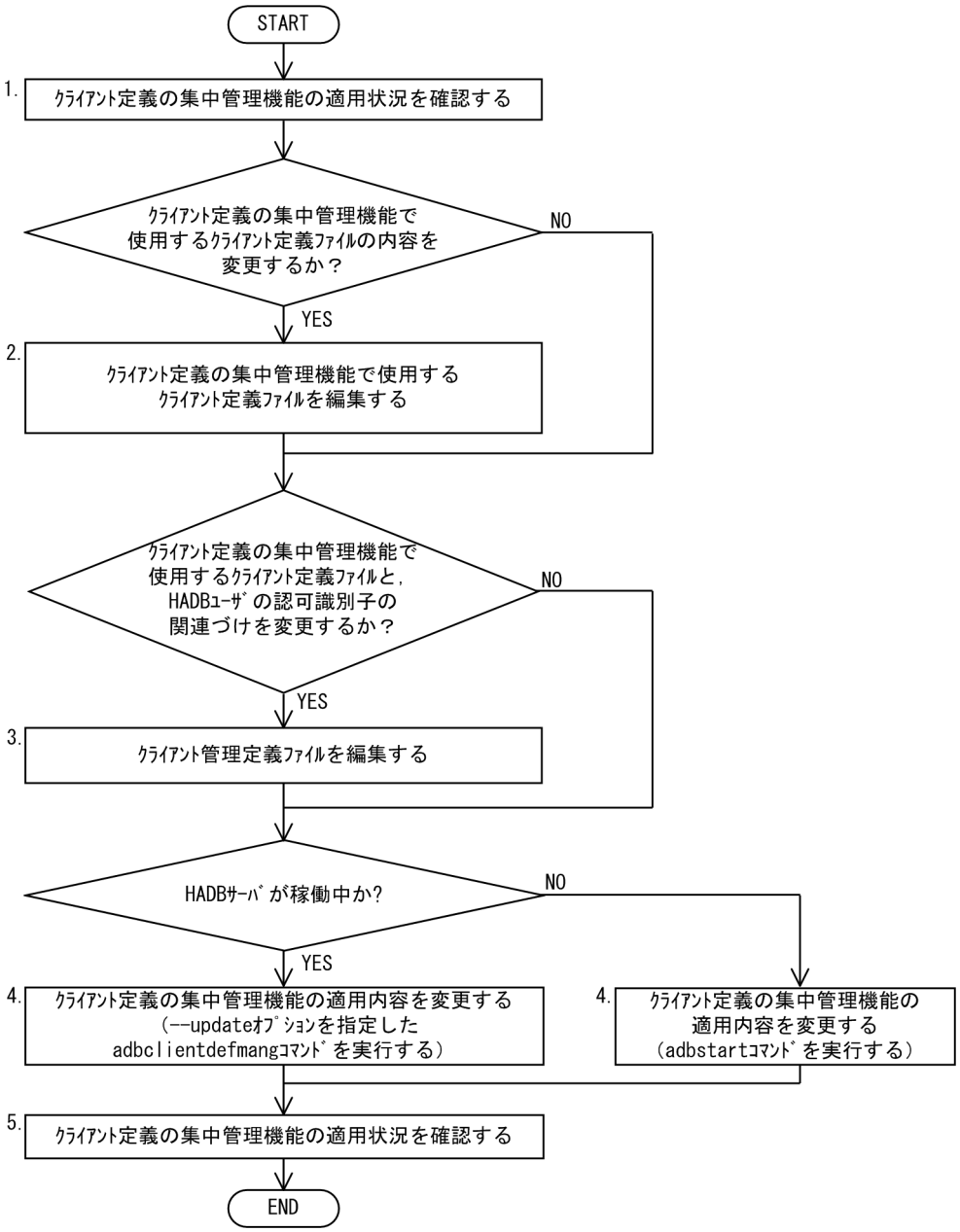
図 11-44　新たにクライアント定義の集中管理機能を使用する場合の流れ



(2)　クライアント定義の集中管理機能の適用内容を変更する場合の流れ

クライアント定義の集中管理機能の適用内容を変更する場合の流れを，次の図に示します。

図 11-45　クライアント定義の集中管理機能の適用内容を変更する場合の流れ



11.14.2　クライアント定義の集中管理機能で必要なファイル

クライアント定義の集中管理機能を使用する場合，次に示すファイルを作成する必要があります。

HADB サーバで作成が必要なファイル

- クライアント定義の集中管理機能で使用するクライアント定義ファイル
- クライアント管理定義ファイル

HADB クライアントで作成が必要なファイル

- HADB クライアントで管理するクライアント定義ファイル

(1)　クライアント定義の集中管理機能で使用するクライアント定義ファイル

クライアント定義の集中管理機能で使用するクライアント定義ファイルには，クライアント定義の集中管理機能で管理する認可識別子に適用したいクライアント定義のオペランドの値を指定します。クライアント定義の集中管理機能で使用するクライアント定義ファイルの内容に従って，対象の認可識別子は HADB サーバに接続します。

クライアント定義の集中管理機能で使用するクライアント定義ファイルのひな形（client.def）は，サーバディレクトリの$ADBDIR/sample/conf ディレクトリに格納されています。ひな形をコピーして，クライアント定義の各オペランドを指定してください。クライアント定義の指定規則は，サーバ定義と同じです。サーバ定義の指定規則については，「7.3　サーバ定義の文法規則」を参照してください。

メモ

- クライアント定義の各オペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義のオペランドの内容』を参照してください。
- クライアント定義の集中管理機能の適用対象外となるクライアント定義のオペランドについては，指定不要です。適用対象外となるクライアント定義のオペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の集中管理機能を使用する場合の留意事項』を参照してください。

重要

クライアント定義のオペランドのうち，指定を省略したオペランドには，デフォルト値が仮定されます。そのため，HADB クライアントが管理しているクライアント定義ファイルの指定内容を適用したい場合は，クライアント定義の集中管理機能で使用するクライアント定義ファイルにも同じ内容を指定してください。

クライアント定義の集中管理機能で使用するクライアント定義ファイルを作成したら，サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納してください。クライアント定義ファイルのファイル名は，ひな形（client.def）から変更できます。

なお，複数の認可識別子に対して，別々のクライアント定義ファイルの内容を適用したい場合は，クライアント定義の集中管理機能で使用するクライアント定義ファイルを必要な分だけ作成してください。

(2)　クライアント管理定義ファイル

クライアント管理定義ファイルには，クライアント管理定義のadbclientmang オペランドを指定します。adbclientmang オペランドには，次の 2 つを指定して関連づけます。

クライアント管理定義のadbclientmang オペランドに指定する内容

- クライアント定義の集中管理機能で使用するクライアント定義ファイルのファイル名
- クライアント定義ファイルの指定内容を適用したい HADB ユーザの認可識別子

クライアント管理定義ファイルのひな形（adbclientdefmang.def）は，サーバディレクトリの$ADBDIR/sample/conf ディレクトリに格納されています。ひな形をコピーして，クライアント管理定義のadbclientmang オペランドを指定してください。

クライアント管理定義ファイルを作成したら，サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納してください。ファイル名は，ひな形（adbclientdefmang.def）のままとしてください。

なお，次に示す場合は，クライアント定義の集中管理機能は適用されません。

- クライアント管理定義ファイルに何も指定していない（空のファイルである）
- クライアント管理定義ファイル内のすべての内容をコメントアウトしている

上記に該当する場合は，HADB クライアントが管理するクライアント定義ファイルの指定内容が適用されます。

(a) クライアント管理定義の指定形式

```
 { {
   [adbclientmang
     -f クライアント定義の集中管理機能で使用するクライアント定義ファイル
     -i 認可識別子 〔,認可識別子〕…]
} }
```

クライアント管理定義の指定規則は，サーバ定義と同じです。サーバ定義の指定規則については，「7.3 サーバ定義の文法規則」を参照してください。

(b) クライアント管理定義の指定形式の説明

●-f *クライアント定義の集中管理機能で使用するクライアント定義ファイル*

～〈文字列〉((1～255 バイト))

クライアント定義の集中管理機能で使用するクライアント定義ファイルのファイル名を指定します。

このオプションには，サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納されているクライアント定義ファイルのファイル名だけを指定してください。指定例を次に示します。

指定例（クライアント定義ファイル名が client01.def の場合）

```
-f client01.def
```

また，複数のadbclientmang オペランドの-f オプションに，同じファイル名を指定することもできます。指定例を次に示します。指定例 1 と指定例 2 は，どちらも同じ指定内容として扱われます。

指定例 1

```
adbclientmang -f client01.def -i ADBUSER01,ADBUSER02
```

指定例 2

```
adbclientmang -f client01.def -i ADBUSER01
adbclientmang -f client01.def -i ADBUSER02
```

なお，-f オプションの指定内容について，次のどれかに該当する場合は，エラーとなります。

- $ADBDIR/conf ディレクトリ下に存在しないファイル名を指定した場合
- ファイル名以外にディレクトリ名も指定した場合
- このオプションに指定したクライアント定義ファイルの内容に誤りがある場合

●-i *認可識別子*

～〈文字列〉((1～100 バイト))

-f オプションに指定したクライアント定義ファイルの内容を適用して，HADB サーバに接続する HADB ユーザの認可識別子を指定します。

このオプションに指定する認可識別子は，1～100 バイトの範囲で指定してください。ただし，認可識別子を囲む二重引用符は，1～100 バイトには含みません。

このオプションには，コンマで区切ることで，複数の認可識別子を指定できます。指定例を次に示します。

指定例（HADB ユーザの認可識別子が ADBUSER01 と ADBUSER02 の場合）

```
-i ADBUSER01,ADBUSER02
```

重要

認可識別子の文字列中に英小文字がある場合は，認可識別子の指定規則を必ず確認してください。認可識別子の指定規則については，「9.4.2　認可識別子の指定規則」を参照してください。

なお，-i オプションの指定内容について，次のどれかに該当する場合は，エラーとなります。

- -i オプションに，同じ認可識別子が複数指定されている場合
- 複数のadbclientmang オペランドの-i オプションに，同じ認可識別子がそれぞれ指定されている場合
- すべての-i オプションに指定した認可識別子の総数が，30,000 を超えた場合

(3)　HADB クライアントで管理するクライアント定義ファイル

クライアント定義の集中管理機能を使用する場合でも，HADB クライアントで管理するクライアント定義ファイルを作成する必要があります。

メモ

HADB クライアントで管理するクライアント定義ファイルの作成方法については，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の作成方法』を参照してください。

HADB クライアントで管理するクライアント定義ファイルには，クライアント定義の集中管理機能の適用対象外となるクライアント定義のオペランドを指定します。接続先の HADB サーバのホスト名やポート番号などです。クライアント定義の集中管理機能の適用対象外となるクライアント定義のオペランドについ

ては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の集中管理機能を使用する場合の留意事項』を参照してください。

なお，HADB クライアントで管理するクライアント定義ファイルには，クライアント定義の集中管理機能の適用対象となるクライアント定義のオペランドは，指定不要です。

クライアント定義の集中管理機能の適用対象となるクライアント定義のオペランドは，クライアント定義の集中管理機能で使用するクライアント定義ファイルに指定してください。オペランドの指定を省略した場合は，デフォルト値が仮定されます。

11.14.3　新たにクライアント定義の集中管理機能を使用する方法

新たにクライアント定義の集中管理機能を使用する場合の手順を次に示します。

新たにクライアント定義の集中管理機能を使用する場合の流れについては，「11.14.1　クライアント定義の集中管理機能を使用する流れ」の「(1)　新たにクライアント定義の集中管理機能を使用する場合の流れ」を参照してください。

手順

1. クライアント定義の集中管理機能で使用するクライアント定義ファイルを作成する

   クライアント定義の集中管理機能で使用するクライアント定義ファイルを作成してください。作成したクライアント定義ファイルは，サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納してください。詳細については，「11.14.2　クライアント定義の集中管理機能で必要なファイル」の「(1)　クライアント定義の集中管理機能で使用するクライアント定義ファイル」を参照してください。

2. クライアント管理定義ファイルを作成する

   クライアント管理定義ファイルを作成してください。作成したクライアント管理定義ファイルは，サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納してください。ファイル名は，ひな形（adbclientdefmang.def）のままとしてください。詳細については，「11.14.2　クライアント定義の集中管理機能で必要なファイル」の「(2)　クライアント管理定義ファイル」を参照してください。

3. クライアント定義の集中管理機能を適用する

   クライアント定義の集中管理機能で使用するクライアント定義ファイル，およびクライアント管理定義ファイルを作成したら，クライアント定義の集中管理機能を適用してください。

   HADB サーバが稼働中かどうかで，クライアント定義の集中管理機能を適用する方法が異なります。

   ■HADB サーバが停止中の場合

   adbstart コマンドを実行して，HADB サーバを開始してください。HADB サーバを開始すると，作成したクライアント定義ファイルとクライアント管理定義ファイルに従って，クライアント定義の集中管理機能が適用されます。

   adbstart コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

■HADB サーバが稼働中の場合

--update オプションを指定したadbclientdefmang コマンドを実行してください。adbclientdefmang コマンドを実行すると，作成したクライアント定義ファイルとクライアント管理定義ファイルに従って，クライアント定義の集中管理機能が適用されます。

adbclientdefmang コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbclientdefmang（クライアント定義の集中管理)』を参照してください。

> **メモ**
>
> クライアント定義の集中管理機能を適用した時点で，adbclientmang オペランドの-i オプションに指定した HADB ユーザの認可識別子で，HADB サーバに接続している HADB クライアントについては，クライアント定義の集中管理機能は適用されません。一度 HADB サーバから切り離して，次に接続したタイミングで，クライアント定義の集中管理機能が適用されます。

4. クライアント定義の集中管理機能の適用状況を確認する

adbclientdefmang コマンドを実行して，クライアント定義の集中管理機能の適用状況を確認してください。

adbclientdefmang コマンドの実行結果から，手順 2.で作成したクライアント管理定義ファイルの内容が適用されているかどうかを確認できます。

adbclientdefmang コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbclientdefmang（クライアント定義の集中管理)』を参照してください。

11.14.4　クライアント定義の集中管理機能の適用内容を変更する方法

クライアント定義の集中管理機能の適用内容を変更する場合の手順を次に示します。

クライアント定義の集中管理機能の適用内容を変更する場合の流れについては，「11.14.1　クライアント定義の集中管理機能を使用する流れ」の「(2)　クライアント定義の集中管理機能の適用内容を変更する場合の流れ」を参照してください。

手順

1. クライアント定義の集中管理機能の適用状況を確認する

adbclientdefmang コマンドを実行して，クライアント定義の集中管理機能の適用状況を確認してください。adbclientdefmang コマンドの実行結果から，現在適用されているクライアント管理定義ファイルの内容が確認できます。

adbclientdefmang コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbclientdefmang（クライアント定義の集中管理)』を参照してください。

クライアント定義の集中管理機能に使用するクライアント定義ファイルの内容を変更する場合は，手順 2.に進んでください。

クライアント定義の集中管理機能に使用するクライアント定義ファイルの内容を変更しない場合は，手順 3.に進んでください。

2. クライアント定義の集中管理機能で使用するクライアント定義ファイルを編集する

   サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納している，クライアント定義の集中管理機能で使用するクライアント定義ファイルを編集してください。詳細については，「11.14.2　クライアント定義の集中管理機能で必要なファイル」の「(1)　クライアント定義の集中管理機能で使用するクライアント定義ファイル」を参照してください。

   クライアント定義の集中管理機能で使用するクライアント定義ファイルと，HADB ユーザの認可識別子の関連づけを変更する場合は，手順 3.に進んでください。

   クライアント定義の集中管理機能で使用するクライアント定義ファイルと，HADB ユーザの認可識別子の関連づけを変更しない場合は，手順 4.に進んでください。

3. クライアント管理定義ファイルを編集する

   サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納している，クライアント管理定義ファイル（adbclientdefmang.def）を編集してください。詳細については，「11.14.2　クライアント定義の集中管理機能で必要なファイル」の「(2)　クライアント管理定義ファイル」を参照してください。

4. クライアント定義の集中管理機能の適用内容を変更する

   クライアント定義の集中管理機能の適用内容を変更してください。

   HADB サーバが稼働中かどうかで，クライアント定義の集中管理機能の適用内容を変更する方法が異なります。

   ■HADB サーバが停止中の場合

   adbstart コマンドを実行して，HADB サーバを開始してください。HADB サーバを開始すると，編集したクライアント定義ファイルとクライアント管理定義ファイルに従って，クライアント定義の集中管理機能の適用内容が変更されます。

   adbstart コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

   ■HADB サーバが稼働中の場合

   --update オプションを指定したadbclientdefmang コマンドを実行してください。adbclientdefmang コマンドを実行すると，編集したクライアント定義ファイルとクライアント管理定義ファイルに従って，クライアント定義の集中管理機能の適用内容が変更されます。

   adbclientdefmang コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbclientdefmang（クライアント定義の集中管理)』を参照してください。

   > **メモ**
   >
   > クライアント定義の集中管理機能を適用した時点で，adbclientmang オペランドの-i オプションに指定した HADB ユーザの認可識別子で，HADB サーバに接続している HADB クライアントについては，クライアント定義の集中管理機能は適用されません。一度 HADB サーバから切り離して，次に接続したタイミングで，クライアント定義の集中管理機能が適用されます。

5. クライアント定義の集中管理機能の適用状況を確認する

adbclientdefmang コマンドを実行して，クライアント定義の集中管理機能の適用状況を確認してください。

adbclientdefmang コマンドの実行結果から，手順 3.で編集したクライアント管理定義ファイルの内容が適用されているかどうかを確認できます。

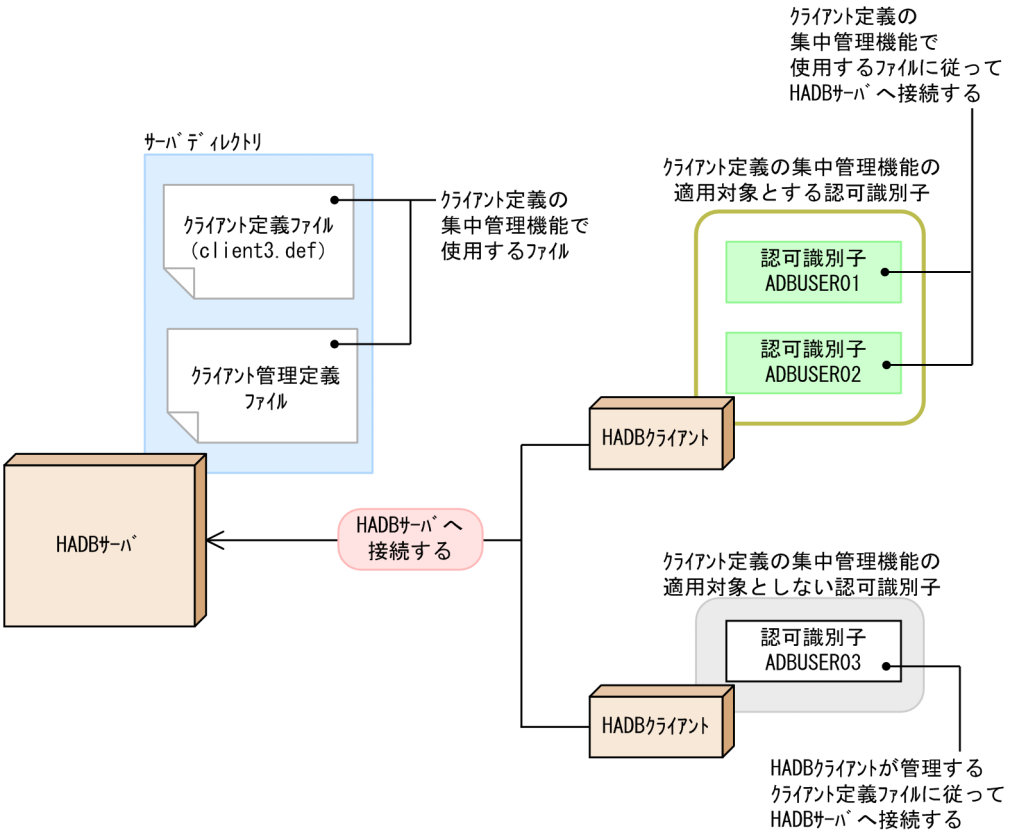
11.14.5　クライアント定義の集中管理機能の使用例

「11.14.3　新たにクライアント定義の集中管理機能を使用する方法」で示した手順に従ったクライアント定義の集中管理機能の使用例を説明します。

■条件

- HADB サーバが停止している
- クライアント定義の集中管理機能で，次に示すクライアント定義の指定値を管理する
  - adb_sql_exe_max_rthd_num オペランド
  - adb_sql_exe_hashgrp_area_size オペランド
  - adb_sql_exe_hashtbl_area_size オペランド
  - adb_sql_exe_hashflt_area_size オペランド
- 次に示す認可識別子を，クライアント定義の集中管理機能の適用対象とする
  - ADBUSER01
  - ADBUSER02
- 次に示す認可識別子については，クライアント定義の集中管理機能の適用対象としない
  - ADBUSER03

図 11-46　クライアント定義の集中管理機能の使用例



(1)　クライアント定義の集中管理機能で使用するクライアント定義ファイルを作成する

クライアント定義の集中管理機能で使用するクライアント定義ファイルを作成して，各クライアント定義を指定します。

サーバディレクトリの$ADBDIR/conf ディレクトリ下に，クライアント定義ファイル（client3.def）を作成します。

クライアント定義の集中管理機能で使用するクライアント定義ファイルの指定例

```
set adb_sql_exe_max_rthd_num = 12
set adb_sql_exe_hashgrp_area_size = 2400
set adb_sql_exe_hashtbl_area_size = 1000
set adb_sql_exe_hashflt_area_size = 100
```

(2)　クライアント管理定義ファイルを作成する

サーバディレクトリの$ADBDIR/conf ディレクトリ下に，クライアント管理定義ファイル（adbclientdefmang.def）を作成して，クライアント管理定義のadbclientmang オペランドを指定します。

クライアント管理定義ファイルの指定例

```
adbclientmang -f client3.def              ...1
              -i ADBUSER01,ADBUSER02      ...2
```

[説明]

1.「(1)　クライアント定義の集中管理機能で使用するクライアント定義ファイルを作成する」で作成したクライアント定義ファイルのファイル名（client3.def）を指定します。

2.クライアント定義の集中管理機能を適用する認可識別子（ADBUSER01，ADBUSER02）を指定します。

(3)　adbstart コマンドを実行する

adbstart コマンドを実行して，HADB サーバを開始します。HADB サーバを開始すると，次に示す 2 つのファイルの内容に従って，クライアント定義の集中管理機能が適用されます。

- クライアント定義の集中管理機能で使用するクライアント定義ファイル
- クライアント管理定義ファイル

> **メモ**
>
> HADB サーバが稼働している場合は，--update オプションを指定したadbclientdefmang コマンドを実行することで，クライアント管理定義ファイルの内容が適用されます。

(4)　adbclientdefmang コマンドを実行する

adbclientdefmang コマンドを実行して，クライアント定義の集中管理機能の適用状況を確認します。

adbclientdefmang コマンドの実行例

```
adbclientdefmang -u ADBUSER01        ...1
                 -p '#HelloHADB_01'  ...2
```

[説明]

1.adbclientdefmang コマンドを実行する HADB ユーザの認可識別子を指定します。

2.-u オプションに指定した認可識別子のパスワードを指定します。

adbclientdefmang コマンドの実行結果から，「(2)　クライアント管理定義ファイルを作成する」で作成したクライアント管理定義ファイルの内容が適用されているかどうかを確認できます。

adbclientdefmang コマンドの実行結果の例

```
authorization-identifier client-definition-file
ADBUSER01                client3.def
ADBUSER02                client3.def
```

[説明]

認可識別子ADBUSER01 とADBUSER02 に対して，クライアント定義の集中管理機能で使用するクライアント定義ファイル（client3.def）が適用されていることが確認できます。

11.14.6　クライアント定義の集中管理機能の使用をやめる方法

クライアント定義の集中管理機能の使用をやめる場合の手順を次に示します。

手順

1. クライアント管理定義ファイルを無効化する

   サーバディレクトリの$ADBDIR/conf ディレクトリ下に格納している，クライアント管理定義ファイル（adbclientdefmang.def）に対して，次のどれかを行ってください。

   - クライアント管理定義ファイルを削除する
   - クライアント管理定義ファイル内を空にする（何も指定していない状態にする）
   - クライアント管理定義ファイル内のすべての内容をコメントアウトする

2. クライアント定義の集中管理機能の使用をやめる

   無効化したクライアント管理定義ファイルを基に，クライアント定義の集中管理機能の使用をやめます。

   HADB サーバが稼働中かどうかで，クライアント定義の集中管理機能の使用をやめる方法が異なります。

   ■HADB サーバが停止中の場合

   adbstart コマンドを実行して，HADB サーバを開始してください。HADB サーバを開始すると，無効化したクライアント管理定義ファイルに従って，クライアント定義の集中管理機能の使用をやめます。

   adbstart コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstart（HADB サーバの開始)』を参照してください。

   ■HADB サーバが稼働中の場合

   --update オプションを指定したadbclientdefmang コマンドを実行してください。adbclientdefmang コマンドを実行すると，無効化したクライアント管理定義ファイルに従って，クライアント定義の集中管理機能の使用をやめます。

   adbclientdefmang コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbclientdefmang（クライアント定義の集中管理)』を参照してください。

   > **メモ**
   >
   > クライアント定義の集中管理機能の使用をやめた時点で，adbclientmang オペランドの-i オプションに指定した HADB ユーザの認可識別子で，HADB サーバに接続している場合は，クライアント定義の集中管理機能は適用されたままです。一度 HADB サーバから切り離して，次に接続したタイミングで，クライアント定義の集中管理機能の使用をやめることができます。

11.15　CSV ファイル中のデータを検索するときの運用

ここでは，CSV ファイル中のデータを検索する方法について説明します。

11.15.1　準備作業

次に示す準備作業を実施してください。

(1)　CSV ファイルの準備

次に示すことを実施してください。

- CSV ファイルは，次のどれかの条件を満たす必要があります。条件を満たしていることを確認してください。
  - 圧縮されていない CSV ファイル
  - OS の`gzip` コマンドで GZIP 形式に圧縮された CSV ファイル
  - `adbexport` コマンドでエクスポートした，GZIP 形式に圧縮された CSV ファイル
- CSV ファイルのデータ形式が，規則に従っていることを確認してください。CSV ファイルのデータ形式の規則については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『システム定義関数』の『ADB_CSVREAD 関数』にある『規則』の『CSV ファイルの形式に関する規則』を参照してください。CSV ファイルのデータ形式が正しくない場合，データの検索時にエラーとなります。
- HADB サーバが動作するサーバマシンのファイルシステムに，CSV ファイルを格納してください。
- CSV ファイルの絶対パス名の長さが，510 バイト以下であることを確認してください。

(2)　CSV ファイルおよびディレクトリへの読み取り権限の付与

HADB サーバが CSV ファイルをオープンできるように，HADB 管理者に次の権限を付与してください。

- CSV ファイルへの読み取り権限
- CSV ファイルを格納するディレクトリへの読み取り権限と実行権限

11.15.2　データの検索方法

`SELECT` 文に`ADB_CSVREAD` 関数を指定して，CSV ファイル中のデータを検索します。

(例)

GZIP 形式で圧縮した次の CSV ファイル中のデータを検索します。

- `/dir/file1.csv.gz`

- /dir/file2.csv.gz
- /dir/file3.csv.gz

```
SELECT * FROM TABLE(ADB_CSVREAD(
              MULTISET['/dir/file1.csv.gz','/dir/file2.csv.gz','/dir/file3.csv.gz'],
             'COMPRESSION_FORMAT=GZIP;
              FIELD_NUM=1,2;
              ENCLOSING_CHAR=";
              DELIMITER_CHAR=,;'))
         AS "T1" ("C1" INTEGER, "C2" VARCHAR(32))
```

[説明]

ADB_CSVREAD 関数の引数MULTISET（下線部分）に，CSV ファイルの絶対パス名を指定します。そのほかの引数や，ADB_CSVREAD 関数の指定方法については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『システム定義関数』の『ADB_CSVREAD 関数』を参照してください。

重要

ADB_CSVREAD 関数を指定した SQL 文の実行時，HADB サーバは CSV ファイルをオープンし，CSV ファイル中のデータを読み込みます。そのため，SQL 文の実行中は，CSV ファイルを編集しないでください。

メモ

- CSV ファイル中のデータを読み込む際，1 つの CSV ファイルに対して 1 つの処理リアルスレッドが割り当てられます。
- ADB_CSVREAD 関数中に指定する CSV ファイルを，表副問合せを使用して指定することもできます。詳細については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『システム定義関数』の『ADB_CSVREAD 関数』を参照してください。

11.16　同義語検索を行う際の運用

ここでは，同義語検索を行う際の準備作業，同義語検索の方法，および同義語辞書の管理方法について説明します。

同義語検索の概要については，「2.17.3　同義語検索」を参照してください。

マルチノード機能を使用している場合に同義語検索を行う場合は，「16.26　同義語検索を行う際の運用(マルチノード機能使用時)」の説明もお読みください。

11.16.1　同義語検索を行うための準備をする

同義語検索を行う場合，ここで説明している準備作業を実施してください。

(1)　テキストインデクスの定義を検討する

同義語検索を行う場合，テキストインデクスを使用すると性能向上が見込まれるため，テキストデータ（文書データ）を格納している列にテキストインデクスを定義することを推奨します。

■テキストインデクスを定義しても効果が見込めないケース

テキストデータ中の文字の種類が少ない場合は，テキストインデクスを定義しても性能向上が見込まれません。例えば，テキストデータが数字だけの場合が該当します。

また，同義語検索を行う際に指定する検索対象文字列が「a」や「0」などの単純で短い文字列の場合は，テキストインデクスを定義しても性能向上が見込まれません。場合によっては，テキストインデクスが使用されないこともあります。

テキストインデクスを定義する際の考慮点については，「5.4.1　テキストインデクスを定義する列を決めるときの考慮点」を参照してください。

■表記ゆれ補正検索との関係

同義語検索は，表記ゆれ補正検索と合わせて実行できます。同義語検索と表記ゆれ補正検索を合わせて実行する場合は，CREATE INDEX 文でテキストインデクスを定義する際にCORRECTIONRULE（テキストインデクス表記ゆれ補正指定）を指定してください。

表記ゆれ補正検索については，「2.17.1　表記ゆれ補正検索」を参照してください。

表記ゆれ補正検索を適用するかどうかは，同義語検索を実行する際にスカラ関数CONTAINS の指定で選択できます。そのため，準備の段階で表記ゆれ補正検索を適用するかどうかが不明な場合は，表記ゆれ補正検索に対応するテキストインデクスを定義することを推奨します。

> **重要**
>
> HADB サーバで使用する文字コードがShift-JIS の場合（環境変数ADBLANG の指定値がSJIS の場合），表記ゆれ補正検索は実行できません。

なお，テキストインデクスを定義しなくても，同義語検索と表記ゆれ補正検索は実行できますが，性能の面から，テキストインデクスを定義することを推奨します。

■ワード検索との関係

同義語検索は，ワード検索と合わせて実行できます。同義語検索とワード検索を合わせて実行する場合は，CREATE INDEX 文でテキストインデクスを定義する際，INDEXTYPE にTEXT WORDCONTEXT を指定してください（ワード検索用のテキストインデクスを定義してください）。

ワード検索については，「2.17.4　ワード検索」を参照してください。

ワード検索を適用するかどうかは，同義語検索を実行する際にスカラ関数CONTAINS の指定で選択できます。そのため，準備の段階でワード検索を適用するかどうかが不明な場合は，ワード検索用のテキストインデクスを定義することを推奨します。

なお，テキストインデクスを定義しなくても，同義語検索とワード検索は実行できますが，性能の面から，テキストインデクスを定義することを推奨します。

(2)　テキストインデクスを定義する

テキストインデクスを定義する場合は，次に示す手順でテキストインデクスを定義してください。

手順

1. CREATE INDEX 文を実行して，テキストインデクスを定義する
2. adbidxrebuild コマンドを実行して，テキストインデクスのインデクスデータを作成する

(3)　同義語辞書ファイルの格納ディレクトリに必要な容量を見積もる

同義語辞書を登録するadbsyndict コマンドを実行すると，同義語辞書の情報を格納した同義語辞書ファイルが作成されます。この同義語辞書ファイルを格納するディレクトリに必要な容量を見積もってください。同義語辞書ファイルの格納ディレクトリに必要な概算容量については，「11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する」を参照してください。

同義語辞書ファイルの格納ディレクトリに必要な容量を詳細に見積もる場合は，次の個所を参照してください。

- 「6.14　同義語辞書ファイルの格納ディレクトリの容量見積もり」
- 「6.21.6　adbsyndict コマンド実行時の作業用一時ファイルの容量見積もり」

上記の見積もり結果の合計サイズが，同義語辞書ファイルの格納ディレクトリに必要な容量になります。

> **重要**
>
> 同義語辞書ファイルの格納ディレクトリの空き容量が不足した場合，同義語辞書ファイルの格納ディレクトリを変更する必要があります。同義語辞書ファイルの格納ディレクトリを変更すると，すべての同義語辞書を再登録する必要があるため，ディレクトリを作成しているディスクの容量不足などが発生しないように，見積もりを行ってください。

(4) 同義語辞書ファイルの格納ディレクトリを作成する

同義語辞書ファイルを格納するディレクトリを作成してください。性能の観点から，同義語辞書ファイルの格納ディレクトリだけを作成するファイルシステムを準備することを推奨します。

なお，同義語辞書ファイルの格納ディレクトリを作成する際，次の点に注意してください。

- 次のディレクトリは，同義語辞書ファイルの格納ディレクトリにできません。
  - サーバディレクトリ
  - サーバディレクトリの下位のディレクトリ
  - 下位のディレクトリにサーバディレクトリがあるディレクトリ
  - DB ディレクトリ
  - DB ディレクトリの下位のディレクトリ
  - 下位のディレクトリに DB ディレクトリがあるディレクトリ
  - ルートディレクトリ

  DB ディレクトリが/HADB/db の場合，同義語辞書ファイルの格納ディレクトリにできるディレクトリの例と，できないディレクトリの例を次に示します。

同義語辞書ファイルの格納ディレクトリの例		理由
同義語辞書ファイルの格納ディレクトリにできるディレクトリの例	/HADB/syndict	なし。
同義語辞書ファイルの格納ディレクトリにできないディレクトリの例	/HADB/db	左記のディレクトリは，DB ディレクトリと同一のため，同義語辞書ファイルの格納ディレクトリにできません。
	/HADB/db/synonym	左記のディレクトリは，DB ディレクトリの下位のディレクトリのため，同義語辞書ファイルの格納ディレクトリにできません。
	/HADB	左記のディレクトリは，下位のディレクトリに DB ディレクトリがあるため，同義語辞書ファイルの格納ディレクトリにできません。

- HADB サーバのインストール時にインストールデータを格納したディレクトリを，同義語辞書ファイルの格納ディレクトリにしないでください。

> **メモ**
>
> データベースのバックアップを取得する際，同義語辞書ファイルの格納ディレクトリもバックアップの取得対象になります。バックアップの取得方法については，「10.3　データベースのバックアップ」を参照してください。

(5)　同義語辞書ファイルの格納ディレクトリに対する権限を付与する

次に示す手順で，同義語辞書ファイルの格納ディレクトリに対して必要な権限を付与してください。

手順

1. 同義語辞書ファイルの格納ディレクトリに，HADB 管理者に対する読み取り権限，書き込み権限，および実行権限を付与する
2. 同義語辞書ファイルの格納ディレクトリのパスに含まれるすべてのディレクトリに，HADB 管理者に対する実行権限を付与する

（例）

同義語辞書ファイルの格納ディレクトリが/HADB/syndict の場合

- /HADB/syndict には，読み取り権限，書き込み権限，および実行権限が必要です。
- /，および/HADB には，実行権限が必要です。

(6)　サーバ定義を修正する

サーバ定義のadb_syndict_storage_path オペランドに，同義語辞書ファイルの格納ディレクトリ名を指定してください。adb_syndict_storage_path オペランドについては，「7.2.8　同義語検索に関するオペランド（set 形式）」の adb_syndict_storage_path を参照してください。

adb_syndict_storage_path オペランドに同義語辞書ファイルの格納ディレクトリのシンボリックリンクを指定した場合，シンボリックリンクを解決したあとの絶対パス名が，「(4)　同義語辞書ファイルの格納ディレクトリを作成する」で説明している規則に従っているかどうかがチェックされます。

> **メモ**
>
> サーバ定義の変更手順については，「8.5.2　サーバ定義の変更方法」を参照してください。

(7)　同義語一覧定義ファイルを作成する

同義語辞書に登録する同義語の一覧を記述したファイルを作成します。このファイルを同義語一覧定義ファイルといいます。同義語検索を行った場合，同義語一覧定義ファイルに記述された同義語が一括検索されます。

■同義語一覧定義ファイルの指定例

```
データベース,データバンク,database↓
アプリケーションサーバ,APサーバ,application server↓
```

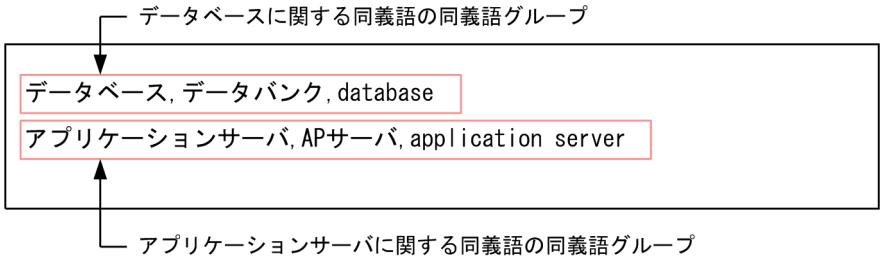
（凡例）↓：改行

上記のように同義語一覧定義ファイルを作成した場合，「データベース」，「データバンク」，「database」のどれか 1 つを検索対象文字列に指定すると，「データベース」，「データバンク」，および「database」のすべての語句が検索対象文字列になります。

「アプリケーションサーバ」,「APサーバ」,「application server」のどれか 1 つを検索対象文字列に指定すると,「アプリケーションサーバ」,「APサーバ」, および「application server」のすべての語句が検索対象文字列になります。

■同義語一覧定義ファイルの指定規則

- 各同義語を「, 」で区切って記述します。
- 一括検索の対象とする同義語を 1 行に記述します。1 行に書いた同義語の集まりを同義語グループといいます。下記の例の場合, 2 つの同義語グループを指定しています。



> **メモ**
>
> ここでは, 同義語一覧定義ファイルの基本的な指定規則だけを説明しています。同義語一覧定義ファイルの指定規則の詳細については,「11.16.15　同義語一覧定義ファイルの指定規則」を参照してください。

この例では, 同義語一覧定義ファイルを次のファイル名で保存したとします。

- /home/adbmanager/dictionary1_synonym.txt

> **重要**
>
> ここで作成した同義語一覧定義ファイルは, 同義語辞書に同義語を追加するときなどに使用するため, 必ず保管しておいてください。

(8)　辞書情報登録ファイルを作成する

同義語辞書の辞書情報を指定したファイルを作成します。このファイルを辞書情報登録ファイルといいます。ここで作成した辞書情報登録ファイルを, 同義語辞書を登録する際に実行するadbsyndict コマンドのオプション引数に指定します。

■辞書情報登録ファイルの指定例

```
Dictionary1,/home/adbmanager/dictionary1_synonym.txt,CORRECTIONRULE,DBとAPサーバの関連用語↓
```

(凡例) ↓ : 改行

［説明］

辞書情報登録ファイル中には，次に示す４つの項目を指定します。各項目は，コンマ（，）で区切って指定してください。

- １列目

  同義語辞書名を指定します。

- ２列目

  「(7)　同義語一覧定義ファイルを作成する」で作成した同義語一覧定義ファイルの名称を指定します。

- ３列目

  表記ゆれ補正オプションを指定します。表記ゆれ補正オプションには，表記ゆれ補正検索に対応する同義語辞書を作成するかどうかを指定します。次のどちらかを指定します。

  ・CORRECTIONRULE：表記ゆれ補正検索に対応する同義語辞書を作成する場合に指定します。

  ・CASESENSITIVE：表記ゆれ補正検索に対応しない同義語辞書を作成する場合に指定します。

- ４列目

  同義語辞書に対するコメントを指定します。

メモ

ここでは，辞書情報登録ファイルの基本的な指定規則だけを説明しています。辞書情報登録ファイルの指定規則の詳細については，「11.16.16　辞書情報登録ファイルの指定規則」を参照してください。

この例では，辞書情報登録ファイルを次のファイル名で保存したとします。

- /home/adbmanager/dictionary1_information.txt

重要

ここで作成した辞書情報登録ファイルは，同義語辞書に同義語を追加するときなどに使用するため，必ず保管しておいてください。

(9)　同義語辞書を登録する

adbsyndict コマンドを実行して同義語辞書を登録します。

コマンドの実行例

```
adbsyndict -m /home/adbmanager/dictionary1_information.txt
```

［説明］

-m オプションには，「(8)　辞書情報登録ファイルを作成する」で作成した辞書情報登録ファイルの絶対パス名を指定します。

メモ

- 同義語辞書の登録が完了した場合，KFAA51509-I メッセージが出力されます（登録を完了した同義語辞書の数だけメッセージが出力されます)。メッセージ中の同義語辞書名を確認し，同義語辞書の登録が完了したことを確認してください。
- adbsyndict コマンドを実行すると，同義語辞書ファイルが作成されます。表記ゆれ補正検索に対応している同義語辞書の場合は，同義語辞書ファイルが 3 つ作成されます。表記ゆれ補正検索に対応していない同義語辞書の場合は，同義語辞書ファイルが 1 つだけ作成されます。
- 同義語辞書ファイルの格納ディレクトリ下に作成される同義語辞書ファイルを，adbsyndict コマンド以外の方法で削除または更新しないでください。

(10)　同義語辞書ファイルの格納ディレクトリのバックアップを取得する

同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バックアップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndict コマンドで再作成する必要があります。

同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

11.16.2　同義語検索を行う

スカラ関数CONTAINS を使用して同義語検索を行います。スカラ関数CONTAINS については，マニュアル『HADB　SQL リファレンス』の『CONTAINS』を参照してください。

(1)　基本的な検索例

同義語辞書Dictionary1 を使用して，同義語検索を行います。同義語辞書Dictionary1 には，次に示す語句が同義語として登録されています。

- データベース，データバンク，database

例題

DOCUMENTS 列には文書データが格納されています。その文書データ中に「データベース，データバンク，database」のどれかの語句が含まれている文書データ名（TITLE 列）を検索します。

実行する SQL 文

```
SELECT "TITLE" FROM "REPORTS"
    WHERE CONTAINS("DOCUMENTS",'SYNONYM("Dictionary1","データバンク")')>0
```

[説明]

スカラ関数CONTAINS 中に同義語辞書名（Dictionary1）と，Dictionary1 に登録されている同義語の 1 つ（データバンク）を指定します。上記の SQL 文を実行すると，同義語辞書の同義語グループに登録されている同義語（データベース，データバンク，database）のどれかが文章情報中に含まれている文書データ名（TITLE 列）が検索結果として返されます。

メモ

同義語辞書に登録されている同義語を調べる方法については，「11.16.3　同義語辞書に登録されている同義語を確認する」を参照してください。

(2)　表記ゆれ補正検索を適用した検索例

表記ゆれ補正検索に対応している同義語辞書Dictionary1 を使用して，同義語検索を行います。同義語辞書Dictionary1 には，次に示す語句が同義語として登録されています。

- データベース，データバンク，database

例題

DOCUMENTS 列には文書データが格納されています。その文書データ中に「データベース，データバンク，database，ﾃﾞｰﾀﾍﾞｰｽ，ﾃﾞｰﾀﾊﾞﾝｸ，DATABASE」などの語句が含まれている文書データ名（TITLE 列）を検索します。

実行する SQL 文

```
SELECT "TITLE" FROM "REPORTS"
    WHERE CONTAINS("DOCUMENTS",'SYNONYM("Dictionary1",SORTCODE("データバンク"))')>0
```

[説明]

スカラ関数CONTAINS 中に同義語辞書名（Dictionary1）と，Dictionary1 に登録されている同義語の 1 つ（データバンク）を指定します。上記の SQL 文を実行すると，同義語辞書の同義語グループに登録されている同義語（データベース，データバンク，database）のどれかが文章情報中に含まれている文書データ名（TITLE 列）が検索結果として返されます。

さらに，表記ゆれ補正検索が働くため，ﾃﾞｰﾀﾍﾞｰｽ，ﾃﾞｰﾀﾊﾞﾝｸ，DATABASE などの語句が含まれている文書データの文書データ名（TITLE 列）も検索結果として返されます。

(3)　２つの同義語辞書を使用した検索例

次に示す 2 つの同義語辞書を使用した検索例を説明します。

- 同義語辞書名：Dictionary2

  Dictionary2 には，次に示す語句が同義語として登録されています。

- リレーショナルデータベース，RDB，relational database，NoSQLデータベース

- 同義語辞書名：Dictionary3

  Dictionary3 には，次に示す語句が同義語として登録されています。

  - リレーショナルデータベース，RDB，relational database，NoSQLデータベース，クラウドデータベース，ビッグデータデータベース

例題

次の条件をすべて満たす文書データ名（TITLE 列）を検索します。

- DOCUMENTS 列に格納されている文書データ中に，「リレーショナルデータベース，RDB，relational database，NoSQLデータベース」のどれかの語句が含まれている
- TITLE 列に格納されている文書データ名中に，「リレーショナルデータベース，RDB，relational database，NoSQLデータベース，クラウドデータベース，ビッグデータデータベース」のどれかの語句が含まれている

```
SELECT "TITLE" FROM "REPORTS"
    WHERE CONTAINS("DOCUMENTS",'SYNONYM("Dictionary2","RDB")')>0
      AND CONTAINS("TITLE",'SYNONYM("Dictionary3","RDB")')>0
```

(4) 注意事項

- スカラ関数CONTAINS に指定した検索対象文字列が，同義語辞書に登録されていない場合，同義語検索は行われません。

  (例)

  同義語辞書に登録されている同義語が「データベース，データバンク，database」の場合に，次のSELECT 文を実行しても，同義語検索は行われません。この場合，「RDB」だけが検索対象文字列になります。

```
SELECT "TITLE" FROM "REPORTS"
    WHERE CONTAINS("DOCUMENTS",'SYNONYM("Dictionary1",SORTCODE("RDB"))')>0
```

- 同義語辞書の更新中（adbsyndict コマンドの実行中）に，更新中の同義語辞書を使用した同義語検索を実行できます。ただし，更新前の同義語辞書を使用した検索になるか，または更新後の同義語辞書を使用した検索になるかは，タイミングによって異なります。

11.16.3 同義語辞書に登録されている同義語を確認する

adbsyndict コマンドを実行すると，同義語辞書に登録されている同義語の一覧を出力できます。

例題

同義語辞書Dictionary1 に登録されている同義語の一覧を出力します。

手順

1. adbsyndict コマンドを実行して，同義語辞書Dictionary1 に登録されている同義語の一覧を出力する

```
adbsyndict -n Dictionary1 -o /home/adbmanager/dictionary1_output.txt
```

-n オプションには，同義語辞書名を指定します。

-o オプションには，同義語辞書Dictionary1 に登録されている同義語の一覧を出力するファイルを指定します。

2. 出力されたファイルを開いて同義語の一覧を確認する

同義語の一覧の出力例

```
”データベース”,”データバンク”,”database”↓
”アプリケーションサーバ”,”APサーバ”,”application server”↓
```

（凡例）↓：改行

同義語の一覧は，同義語一覧定義ファイルと同じ形式で出力されます。上記の例のように，1 行が 1 同義語グループの同義語になります。

メモ

手順の 1.で同義語の一覧の出力が完了した場合，KFAA51529-I メッセージが出力されます。メッセージ中の同義語辞書名を確認し，同義語の一覧の出力が完了したことを確認してください。

11.16.4 同義語辞書の情報を確認する

同義語辞書の名称を確認したい場合や，同義語辞書が表記ゆれ補正検索に対応しているかどうかを確認したい場合は，システム表を検索します。

(1) 特定の同義語辞書の情報を確認する

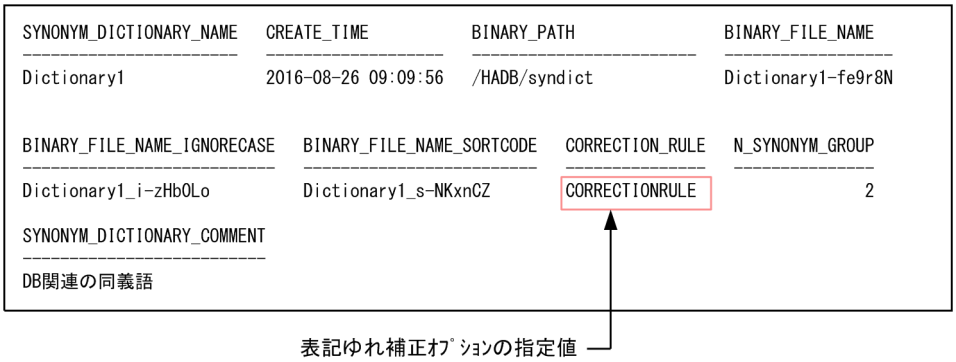
例題

同義語辞書Dictionary1 が表記ゆれ補正検索に対応しているかどうかを確認します。

実行する SQL 文

```
SELECT * FROM ”MASTER”.”STATUS_SYNONYM_DICTIONARIES”
    WHERE ”SYNONYM_DICTIONARY_NAME”='Dictionary1'
```

実行結果の例



[説明]

CORRECTION_RULE 欄にCORRECTIONRULE が表示されているため，同義語辞書Dictionary1 は表記ゆれ補正検索に対応しています。

CORRECTION_RULE 欄にCASESENSITIVE が表示されている場合は，その同義語辞書は表記ゆれ補正検索に対応していません。

説明のない出力項目については，「付録 C.6　STATUS_SYNONYM_DICTIONARIES の内容」で説明されているSTATUS_SYNONYM_DICTIONARIES の列情報の説明を参照してください。

上記の検索結果の表示形式は，実際に出力される表示形式とは異なるところがあります。

(2)　すべての同義語辞書の情報を確認する

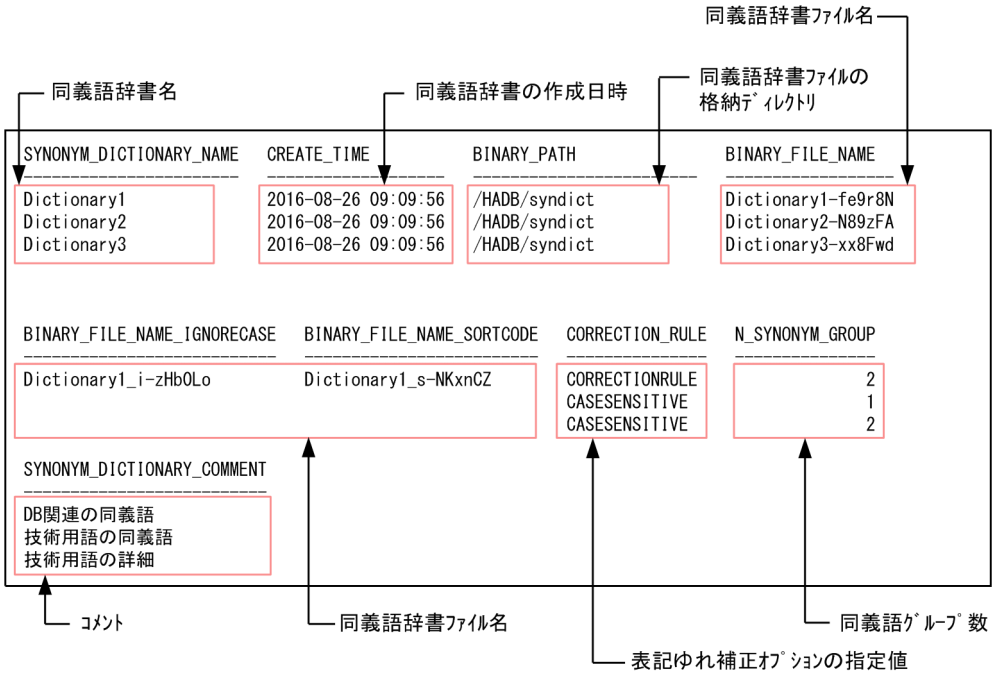
例題

同義語辞書の一覧と各同義語辞書に関する情報を確認します。

実行する SQL 文

```
SELECT * FROM "MASTER"."STATUS_SYNONYM_DICTIONARIES"
```

実行結果の例



例えば，SQL 文中に指定する同義語辞書名を確認するために，同義語辞書の一覧を表示した場合は，コメントから該当する同義語辞書を推定するようにしてください。

上記の検索結果の表示形式は，実際に出力される表示形式とは異なるところがあります。

11.16.5　同義語を追加する

同義語辞書に同義語を追加する方法を説明します。

例題

同義語辞書Dictionary1 に登録されている同義語グループに同義語を追加します。この例では，「リレーショナルデータベース」と「RDB」を追加しています。

手順

1. 保管している同義語一覧定義ファイルの指定を修正する

   ＜同義語一覧定義ファイルの指定内容＞

   ```
   データベース,データバンク,database,リレーショナルデータベース,RDB↓
   アプリケーションサーバ,APサーバ,application server↓
   ```

   （凡例）↓：改行

   保管しているDictionary1 用の同義語一覧定義ファイルを修正します。この例では，下線部分の同義語を追加しています。

重要

変更がない同義語グループはそのまま指定してください。上記の例の 2 行目（アプリケーションサーバの行）が該当します。上記の例の 2 行目を指定しないと，アプリケーションサーバの行の同義語グループが削除されてしまいます。

この例では，Dictionary1 用の同義語一覧定義ファイルを次のファイル名で保存しているとします。

- /home/adbmanager/dictionary1_synonym.txt

2. 保管している辞書情報登録ファイルの指定内容を確認する

＜辞書情報登録ファイルの指定内容＞

```
Dictionary1,/home/adbmanager/dictionary1_synonym.txt,CORRECTIONRULE,DBとAPサーバの関連用語↓
```

（凡例）↓：改行

保管しているDictionary1 用の辞書情報登録ファイルの指定内容を確認してください。同義語の追加によってコメントを変更する必要がある場合は，コメントを変更してください。

この例では，辞書情報登録ファイルを次のファイル名で保存しているとします。

- /home/adbmanager/dictionary1_information.txt

3. 同義語辞書ファイルの格納ディレクトリの空き容量を確認する

空き容量の確認方法については，「11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する」を参照してください。

4. adbsyndict コマンドを実行して同義語辞書を更新する

```
adbsyndict -m /home/adbmanager/dictionary1_information.txt
```

-m オプションには，手順の 2.で確認した辞書情報登録ファイルの絶対パス名を指定します。

5. 同義語辞書ファイルの格納ディレクトリのバックアップを取得する

同義語辞書を更新した場合，同義語辞書ファイルが更新されます。同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バックアップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndict コマンドで再作成する必要があります。

同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

メモ

- 同義語一覧定義ファイルの指定規則については，「11.16.15　同義語一覧定義ファイルの指定規則」を参照してください。
- 辞書情報登録ファイルの指定規則については，「11.16.16　辞書情報登録ファイルの指定規則」を参照してください。

- 同義語一覧定義ファイルを紛失してしまった場合は，「11.16.14　同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合）」を参照して，同義語一覧定義ファイルを再作成してください。
- 手順の 4.で同義語辞書の更新が完了した場合，KFAA51509-I メッセージが出力されます（更新が完了した同義語辞書の数だけメッセージが出力されます）。メッセージ中の同義語辞書名を確認し，同義語辞書の更新が完了したことを確認してください。

11.16.6　同義語を削除する

同義語辞書から同義語を削除する方法を説明します。

例題

同義語辞書Dictionary1 に登録されている同義語を削除します。この例では，「データバンク」を削除しています。

手順

1. 保管している同義語一覧定義ファイルの指定を修正する

＜変更前の同義語一覧定義ファイルの指定内容＞

```
データベース,データバンク,database↓
アプリケーションサーバ,APサーバ,application server↓
```

＜変更後の同義語一覧定義ファイルの指定内容＞

```
データベース,database↓
アプリケーションサーバ,APサーバ,application server↓
```

（凡例）↓：改行

保管しているDictionary1 用の同義語一覧定義ファイルを修正します。この例では，「データバンク」を削除しています。

> **重要**
>
> 変更がない同義語グループはそのまま指定してください。上記の例の 2 行目（アプリケーションサーバの行）が該当します。上記の例の 2 行目を指定しないと，アプリケーションサーバの行の同義語グループが削除されてしまいます。

この例では，Dictionary1 用の同義語一覧定義ファイルを次のファイル名で保存しているとします。

- /home/adbmanager/dictionary1_synonym.txt

2. 保管している辞書情報登録ファイルの指定内容を確認する

＜辞書情報登録ファイルの指定内容＞

```
Dictionary1,/home/adbmanager/dictionary1_synonym.txt,CORRECTIONRULE,DBとAPサーバの関連用語↓
```

（凡例）↓：改行

保管している辞書情報登録ファイルの指定内容を確認してください。同義語の削除によってコメントを変更する必要がある場合は，コメントを変更してください。

この例では，辞書情報登録ファイルを次のファイル名で保存しているとします。

- /home/adbmanager/dictionary1_information.txt

3. 同義語辞書ファイルの格納ディレクトリの空き容量を確認する

   空き容量の確認方法については，「11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する」を参照してください。

4. adbsyndict コマンドを実行して同義語辞書を更新する

```
adbsyndict -m /home/adbmanager/dictionary1_information.txt
```

   -m オプションには，手順の 2.で確認した辞書情報登録ファイルの絶対パス名を指定します。

5. 同義語辞書ファイルの格納ディレクトリのバックアップを取得する

   同義語辞書を更新した場合，同義語辞書ファイルが更新されます。同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バックアップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndictコマンドで再作成する必要があります。

   同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

メモ

- 同義語一覧定義ファイルの指定規則については，「11.16.15　同義語一覧定義ファイルの指定規則」を参照してください。
- 辞書情報登録ファイルの指定規則については，「11.16.16　辞書情報登録ファイルの指定規則」を参照してください。
- 同義語一覧定義ファイルを紛失してしまった場合は，「11.16.14　同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合）」を参照して，同義語一覧定義ファイルを再作成してください。
- 手順の 4.で同義語辞書の更新が完了した場合，KFAA51509-I メッセージが出力されます（更新を完了した同義語辞書の数だけメッセージが出力されます）。メッセージ中の同義語辞書名を確認し，同義語辞書の更新が完了したことを確認してください。

11.16.7　同義語グループを追加する

同義語辞書に同義語グループを追加する方法を説明します。

例題

同義語辞書Dictionary1に同義語グループを追加します。追加する同義語グループには，次に示す同義語を登録します。

- ウェブサーバ，WWWサーバ，Webサーバ

手順

1. 保管している同義語一覧定義ファイルの指定を修正する

   ＜同義語一覧定義ファイルの指定内容＞

   ```
   データベース,データバンク,database↓
   アプリケーションサーバ,APサーバ,application server↓
   ウェブサーバ,WWWサーバ,Webサーバ↓
   ```

   （凡例）↓：改行

   保管しているDictionary1用の同義語一覧定義ファイルを修正します。この例では，下線部分の同義語グループを追加しています。

   > **重要**
   >
   > 変更がない同義語グループはそのまま指定してください。上記の例の1行目（データベースの行）と2行目（アプリケーションサーバの行）が該当します。上記の例の1行目と2行目を指定しないと，データベースの行の同義語グループとアプリケーションサーバの行の同義語グループが削除されてしまいます。

   この例では，Dictionary1用の同義語一覧定義ファイルを次のファイル名で保存しているとします。

   - /home/adbmanager/dictionary1_synonym.txt

2. 保管している辞書情報登録ファイルの指定内容を確認する

   ＜辞書情報登録ファイルの指定内容＞

   ```
   Dictionary1,/home/adbmanager/dictionary1_synonym.txt,CORRECTIONRULE,"DB,APサーバ,Webサーバの関連用語"↓
   ```

   （凡例）↓：改行

   保管している辞書情報登録ファイルの指定内容を確認してください。同義語グループの追加によってコメントを変更する必要がある場合は，コメントを変更してください。この例では，コメント（下線部分）を変更しています。

   > **メモ**
   >
   > コメント中に「,」を指定しているため，囲み文字の「"」でコメントを囲んでいます。

   この例では，辞書情報登録ファイルを次のファイル名で保存しているとします。

   - /home/adbmanager/dictionary1_information.txt

3. 同義語辞書ファイルの格納ディレクトリの空き容量を確認する

空き容量の確認方法については，「11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する」を参照してください。

4. adbsyndict コマンドを実行して同義語辞書を更新する

```
adbsyndict -m /home/adbmanager/dictionary1_information.txt
```

-m オプションには，手順の 2.で確認した辞書情報登録ファイルの絶対パス名を指定します。

5. 同義語辞書ファイルの格納ディレクトリのバックアップを取得する

同義語辞書を更新した場合，同義語辞書ファイルが更新されます。同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バックアップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndict コマンドで再作成する必要があります。

同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

メモ

- 同義語一覧定義ファイルの指定規則については，「11.16.15　同義語一覧定義ファイルの指定規則」を参照してください。
- 辞書情報登録ファイルの指定規則については，「11.16.16　辞書情報登録ファイルの指定規則」を参照してください。
- 同義語一覧定義ファイルを紛失してしまった場合は，「11.16.14　同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合）」を参照して，同義語一覧定義ファイルを再作成してください。
- 手順の 4.で同義語辞書の更新が完了した場合，KFAA51509-I メッセージが出力されます（更新を完了した同義語辞書の数だけメッセージが出力されます）。メッセージ中の同義語辞書名を確認し，同義語辞書の更新が完了したことを確認してください。

■同義語グループを削除する場合

同義語グループを削除する手順は，ここで説明している同義語グループを追加する手順と同じです。手順の 1.の『保管している同義語一覧定義ファイルの指定を修正する』で，不要となった同義語グループを削除してください。

11.16.8　同義語辞書を登録する

同義語辞書を登録する方法を説明します。

例題

同義語辞書Dictionary2 を登録します。Dictionary2 には，次の同義語を登録します（同義語グループを 1 つだけ登録します）。

- 日立，日立製作所，Hitachi

手順

1. Dictionary2 用の同義語一覧定義ファイルを作成する

   ＜同義語一覧定義ファイルの指定内容＞

   ```
   日立,日立製作所,Hitachi↓
   ```

   （凡例）↓：改行

   この例では，作成した同義語一覧定義ファイルを次のファイル名で保存したとします。

   - /home/adbmanager/dictionary2_synonym.txt

2. 辞書情報登録ファイルを作成する

   ＜辞書情報登録ファイルの指定内容＞

   ```
   Dictionary2,/home/adbmanager/dictionary2_synonym.txt,CORRECTIONRULE,日立の同義語↓
   ```

   （凡例）↓：改行

   この例では，作成した辞書情報登録ファイルを次のファイル名で保存したとします。

   - /home/adbmanager/dictionary2_information.txt

3. 同義語辞書ファイルの格納ディレクトリの空き容量を確認する

   空き容量の確認方法については，「11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する」を参照してください。

4. adbsyndict コマンドを実行して同義語辞書を登録する

   ```
   adbsyndict -m /home/adbmanager/dictionary2_information.txt
   ```

   -m オプションには，手順の 2.で作成した辞書情報登録ファイルの絶対パス名を指定します。

5. 同義語辞書ファイルの格納ディレクトリのバックアップを取得する

   同義語辞書を登録した場合，同義語辞書ファイルが追加されます。同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バックアップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndict コマンドで再作成する必要があります。

   同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

メモ

- 同義語一覧定義ファイルの指定規則については，「11.16.15　同義語一覧定義ファイルの指定規則」を参照してください。
- 辞書情報登録ファイルの指定規則については，「11.16.16　辞書情報登録ファイルの指定規則」を参照してください。

- 手順の 4.で同義語辞書の登録が完了した場合，KFAA51509-I メッセージが出力されます（登録を完了した同義語辞書の数だけメッセージが出力されます)。メッセージ中の同義語辞書名を確認し，同義語辞書の登録が完了したことを確認してください。

11.16.9　同義語辞書を削除する

同義語辞書を削除する方法を説明します。

例題

同義語辞書Dictionary1 およびDictionary2 を削除します。

手順

1. 辞書情報削除ファイルを作成する

＜辞書情報削除ファイルの指定内容＞

```
Dictionary1↓
Dictionary2↓
```

（凡例）↓：改行

削除する同義語辞書の名称を指定します。1 行に 1 つの同義語辞書名を指定します。

この例では，作成した辞書情報削除ファイルを次のファイル名で保存したとします。

- /home/adbmanager/dictionary_delete.txt

2. adbsyndict コマンドを実行して同義語辞書を削除する

```
adbsyndict -d /home/adbmanager/dictionary_delete.txt
```

-d オプションには，辞書情報削除ファイルの絶対パス名を指定します。

■辞書情報削除ファイルの指定規則

- 各行の終わりに改行を入力してください。また，辞書情報削除ファイルの最後の行にも改行を入力してください。改行コードは，X'0A'（LF），X'0D0A'（CRLF)，またはX'00'のどれかにしてください。
- 環境変数ADBLANG に指定した文字コードで辞書情報削除ファイルを作成してください。
- 改行だけの行がある場合，その行を読み飛ばして処理が実行されます。

メモ

手順の 2.で同義語辞書の削除が完了した場合，KFAA51509-I メッセージが出力されます（削除を完了した同義語辞書の数だけメッセージが出力されます)。メッセージ中の同義語辞書名を確認し，同義語辞書の削除が完了したことを確認してください。

11.16.10　同義語辞書を表記ゆれ補正検索対応に変更する

表記ゆれ補正検索に対応していない同義語辞書を，表記ゆれ補正検索に対応する同義語辞書に変更する方法を説明します。

例題

表記ゆれ補正検索に対応していない同義語辞書Dictionary3 を，表記ゆれ補正検索に対応する同義語辞書に変更します。その際，コメントも変更します。

手順

1. 保管している辞書情報登録ファイルの指定内容を変更する

   ＜変更前の辞書情報登録ファイルの指定内容＞

   ```
   Dictionary3,/home/adbmanager/dictionary3_synonym.txt,CASESENSITIVE,詳細用語↓
   ```

   ＜変更後の辞書情報登録ファイルの指定内容＞

   ```
   Dictionary3,/home/adbmanager/dictionary3_synonym.txt,CORRECTIONRULE,"詳細用語,表記ゆれ対応"↓
   ```

   （凡例）↓：改行

   保管しているDictionary3 用の辞書情報登録ファイルの指定内容を変更します。この例では，次の変更をしています。

   - 3 列目

     表記ゆれ補正オプションの指定を，CASESENSITIVE からCORRECTIONRULE に変更します。

   - 4 列目

     コメントを変更します。コメント中に「,」を指定しているため，囲み文字の「"」でコメントを囲みます。

   この例では，辞書情報登録ファイルを次のファイル名で保存しているとします。

   - /home/adbmanager/dictionary3_information.txt

2. 同義語辞書ファイルの格納ディレクトリの空き容量を確認する

   空き容量の確認方法については，「11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する」を参照してください。

3. adbsyndict コマンドを実行して同義語辞書を更新する

   ```
   adbsyndict -m /home/adbmanager/dictionary3_information.txt
   ```

   -m オプションには，手順の 1.で変更した辞書情報登録ファイルの絶対パス名を指定します。

4. 同義語辞書ファイルの格納ディレクトリのバックアップを取得する

   同義語辞書を更新した場合，同義語辞書ファイルが更新されます。同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バック

アップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndictコマンドで再作成する必要があります。

同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

メモ

手順の 3.で同義語辞書の更新が完了した場合，KFAA51509-I メッセージが出力されます（更新を完了した同義語辞書の数だけメッセージが出力されます)。メッセージ中の同義語辞書名を確認し，同義語辞書の更新が完了したことを確認してください。

11.16.11　同義語検索のチューニングを行う

同義語検索のチューニングを行う際は，ここで説明することを検討および確認してください。

(1)　テキストインデクスの定義を検討する

文書データを格納している列にテキストインデクスを定義していない場合は，テキストインデクスの定義を検討してください。テキストインデクスを定義すると，アクセスするページ数を減らすことができるため，検索性能の向上が見込まれます。

(2)　テキストインデクスが効果的に使用されているかを確認する

アクセスパスの統計情報に出力されるData_tidx_all_search_cnt の値を確認してください。この値が 1 以上で値が大きいほど，テキストインデクスが効果的に使用されていません。Data_tidx_all_search_cnt には，テキストインデクスを使用してチャンク内のデータを検索した際，チャンク内のデータを全件検索した回数が出力されます。

次に示すどちらかの条件を満たす SQL 文を指定している場合は，SQL 文を変更することで同義語検索の性能が向上することがあります。

- 同義語検索指定を指定したスカラ関数CONTAINS，LIKE 述語，LIKE_REGEX 述語を使用した条件を，論理演算子OR を使って複数指定している場合
- 同義語検索指定を指定したスカラ関数CONTAINS を，論理演算子OR を使って複数指定している場合

この場合，UNION を指定してそれぞれの検索結果を統合するか，またはテキストインデクスを使用しないで検索することを検討してください。

SQL 文の変更例を次に示します。

＜変更前＞

```
SELECT ”TITLE” FROM ”REPORTS”
    WHERE CONTAINS(”DOCUMENTS”,'SYNONYM(”Dictionary1”,”データベース”)')>0
```

```
    OR CONTAINS(”DOCUMENTS”,’SYNONYM(”Dictionary1”,”アプリケーションサーバ”)’)>0
    OR ～
```

＜変更後＞

- UNION を使用する場合

```
SELECT ”TITLE” FROM ”REPORTS”
    WHERE CONTAINS(”DOCUMENTS”,’SYNONYM(”Dictionary1”,”データベース”)’)>0
UNION DISTINCT
SELECT ”TITLE” FROM ”REPORTS”
    WHERE CONTAINS(”DOCUMENTS”,’SYNONYM(”Dictionary1”,”アプリケーションサーバ”)’)>0
UNION DISTINCT ～
```

- テキストインデクスを使用しない場合

```
SELECT ”TITLE” FROM ”REPORTS” /*>> WITHOUT INDEX <<*/
    WHERE CONTAINS(”DOCUMENTS”,’SYNONYM(”Dictionary1”,”データベース”)’)>0
       OR CONTAINS(”DOCUMENTS”,’SYNONYM(”Dictionary1”,”アプリケーションサーバ”)’)>0
       OR ～
```

(3)　同義語辞書ファイルへのアクセス時間を確認する

同義語検索が行われた場合，HADB サーバは同義語辞書ファイルにアクセスします。そのため，専用のファイルシステムに同義語辞書ファイルの格納ディレクトリだけを作成すると，同義語検索の性能向上が見込まれます。

SQL 文の統計情報に出力されるSyndict_file_access_time の値を確認してください。
Syndict_file_access_time には，HADB サーバが同義語辞書ファイルにアクセスしている時間が出力されます。SQL 文の実行時間のうち，Syndict_file_access_time の時間が占める割合が大きい場合は，専用のファイルシステムに同義語辞書ファイルの格納ディレクトリを作成することを検討してください。

同義語辞書ファイルの格納ディレクトリを変更する方法については，「11.16.13　同義語辞書ファイルの格納ディレクトリを変更する」を参照してください。

11.16.12　同義語辞書ファイルの格納ディレクトリに必要な空き容量を確認する

同義語辞書を登録または更新する際，同義語辞書ファイルの格納ディレクトリに必要な空き容量があるかどうかを確認してください。空き容量が不足している場合，adbsyndict コマンドがエラーになります。

必要となる空き容量の概算値は，次に示す計算式で求められます。

計算式（単位：キロバイト）

$5 \times A + 513 \times C + 27 \times B + 1{,}539 \times D$

A：

表記ゆれ補正オプションにCASESENSITIVE を指定する同義語辞書の同義語一覧定義ファイルの合計サイズ（単位：キロバイト）

B：

表記ゆれ補正オプションにCORRECTIONRULE を指定する同義語辞書の同義語一覧定義ファイルの合計サイズ（単位：キロバイト）

C：

表記ゆれ補正オプションにCASESENSITIVE を指定する同義語辞書の数

D：

表記ゆれ補正オプションにCORRECTIONRULE を指定する同義語辞書の数

計算の結果，空き容量が不足している場合は，adbsyndict コマンドで一度に登録または更新する同義語辞書の数を減らしてください。上記の計算式では，一時的に必要となる容量を計算しています。そのため，一度に登録または更新する同義語辞書の数を減らすと，一時的に必要となる容量が減るため，adbsyndict コマンドを実行できることがあります。

メモ

例えば，10 個の同義語辞書を更新する場合，更新後の同義語辞書ファイルを 10 個作成したあとに，更新前の同義語辞書ファイルを削除します。そのため，同義語辞書ファイルの格納ディレクトリ下には，一時的に同義語辞書ファイルが 20 個存在することになります。

11.16.13　同義語辞書ファイルの格納ディレクトリを変更する

同義語辞書ファイルの格納ディレクトリを変更する方法を説明します。

手順

1. 変更後の同義語辞書ファイルの格納ディレクトリを作成する

   作成方法の詳細については，「11.16.1　同義語検索を行うための準備をする」の次の個所を参照してください。

   - 「(3)　同義語辞書ファイルの格納ディレクトリに必要な容量を見積もる」
   - 「(4)　同義語辞書ファイルの格納ディレクトリを作成する」
   - 「(5)　同義語辞書ファイルの格納ディレクトリに対する権限を付与する」

2. adbstop コマンドを実行して HADB サーバを正常終了する
3. サーバ定義のadb_syndict_storage_path オペランドの指定を変更する

   手順の 1.で作成したディレクトリをadb_syndict_storage_path オペランドに指定します。

4. adbstart コマンドを実行して HADB サーバを正常開始する

5. adbsyndict コマンドを実行してすべての同義語辞書を再登録する

   保管している同義語一覧定義ファイルと辞書情報登録ファイルを使用して，すべての同義語辞書を再登録してください。

6. 同義語辞書ファイルの格納ディレクトリのバックアップを取得する

   同義語辞書を登録した場合，同義語辞書ファイルが追加されます。同義語辞書ファイルに障害が発生した場合に備えて，同義語辞書ファイルの格納ディレクトリのバックアップを取得してください。バックアップを取得しないと，同義語辞書ファイルに障害が発生した場合，同義語辞書ファイルをadbsyndictコマンドで再作成する必要があります。

   同義語辞書ファイルの格納ディレクトリは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリです。

メモ

手順の 5.で同義語辞書の登録が完了した場合，KFAA51509-I メッセージが出力されます（登録を完了した同義語辞書の数だけメッセージが出力されます)。メッセージ中の同義語辞書名を確認し，同義語辞書の登録が完了したことを確認してください。

11.16.14 同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合）

adbsyndict コマンドを実行すると，同義語一覧定義ファイルを再作成できます。同義語一覧定義ファイルを紛失したときなどに，同義語一覧定義ファイルを再作成してください。

例題

同義語辞書Dictionary1 の同義語一覧定義ファイルを再作成します。

手順

1. adbsyndict コマンドを実行して，同義語辞書Dictionary1 の同義語一覧定義ファイルを再作成する

```
adbsyndict -n Dictionary1 -o /home/adbmanager/dictionary1_output.txt
```

   -n オプションには，同義語辞書名を指定します。

   -o オプションには，出力先のファイル名を指定します。

2. 再作成された同義語一覧定義ファイルを開いて内容を確認する

```
”データベース”,”データバンク”,”database”↓
”アプリケーションサーバ”,”APサーバ”,”application server”↓
```

   （凡例）↓：改行

メモ

- 自分が作成した同義語一覧定義ファイルと，adbsyndict コマンドで再作成した同義語一覧定義ファイルでは，次の点が異なることがあります。
  ・各同義語に囲み文字（二重引用符）が付きます。
  ・同義語グループ内で同じ同義語を 2 つ以上指定していた場合，1 つだけが出力されます。
  ・同義語グループ内の同義語の指定順が異なることがあります。
  ・改行コードはX'0A'（LF）になります。
- 手順の 1.で同義語一覧定義ファイルの再作成が完了した場合，KFAA51529-I メッセージが出力されます。メッセージ中の同義語辞書名を確認し，同義語一覧定義ファイルの再作成が完了したことを確認してください。

11.16.15　同義語一覧定義ファイルの指定規則

同義語一覧定義ファイルの指定規則について説明します。

(1)　同義語一覧定義ファイルの指定例

同義語辞書に次の 2 つの同義語グループを登録します。このときの同義語一覧定義ファイルの指定例を説明します。

- データベース，データバンク，database
- アプリケーションサーバ，APサーバ，application server

＜同義語一覧定義ファイルの指定例＞

```
データベース,データバンク,database↓
アプリケーションサーバ,APサーバ,application server↓
```

（凡例）↓：改行

[説明]

- 「データベース」,「データバンク」,「database」のどれか 1 つを検索対象文字列に指定して検索を実行すると,「データベース」,「データバンク」，および「database」のすべての語句が同義語として一括検索されます。
- 「アプリケーションサーバ」,「APサーバ」,「application server」のどれか 1 つを検索対象文字列に指定して検索を実行すると,「アプリケーションサーバ」,「APサーバ」，および「application server」のすべての語句が同義語として一括検索されます。

(2) 同義語一覧定義ファイルの指定規則

- 1 行に 1 つの同義語グループを指定します。
- 各同義語をコンマ（ , ）で区切って指定します。
- 各行の終わり（同義語グループの終わり）に改行を入力してください。また，同義語一覧定義ファイルの最後の行にも改行を入力してください。前記の例の場合，application server の後ろに改行を入力する必要があります。
- 改行コードは，X'0A'（LF），X'0D0A'（CRLF），またはX'00'のどれかにしてください。
- 同義語の指定がない行（改行だけの行）があると，adbsyndict コマンドがエラーになります。
- 環境変数ADBLANG に指定している文字コードで同義語一覧定義ファイルを作成してください。
- 1 つの同義語には，1～1,000 文字の文字列を指定できます。
- 1 つの同義語グループには，2～1,000 の同義語を指定できます。
- 同義語グループは，1,000,000 個まで指定できます。
- 水平タブX'09'（HT），改行文字X'0A'（LF），またはX'00'を含む同義語は登録できません。
- 二重引用符（"）を囲み文字として指定することができます。同義語中に「,」がある場合は，囲み文字を指定してください。
- 同義語の文字列の前後の半角空白は削除されません。「△ab△c△d△△」と記述した場合，「△ab△c△d△△」が同義語として登録されます。△は，半角空白を示しています。

(3) 同義語一覧定義ファイルと同義語辞書の関係

検索条件や，検索対象の文書データの内容によっては，同義語辞書を複数作成して同義語辞書を使い分けることもできます。同義語辞書を複数作成する場合，同義語一覧定義ファイルは同義語辞書ごとに作成します。

例えば，次のような検索をする場合は，2 つの同義語一覧定義ファイルを作成し，2 つの同義語辞書を作成します。

1. データベースの一般的な用語が含まれている文章データを検索したい場合
2. データベースの種類を絞り込んで文章データを検索したい場合（例えば，リレーショナルデータベースに関する文章データを検索したい，NoSQL データベースに関する文章データを検索したいなど）

■上記の 1.の場合の同義語一覧定義ファイル（Terminology.txt）の指定

```
データベース,データバンク,database,リレーショナルデータベース,RDB,relational database,NoSQLデータベース,クラウドデータベース,ビッグデータデータベース↓
```

（凡例）↓：改行

この同義語一覧定義ファイルを使用して，同義語辞書Terminology を作成します。データベースの一般的な用語が含まれている文章データを検索する際に，同義語辞書Terminology を使用します。

■上記の 2.の場合の同義語一覧定義ファイル（Terminology_detail.txt）の指定

```
リレーショナルデータベース,RDB,relational database↓
NoSQLデータベース,クラウドデータベース,ビッグデータデータベース↓
```

（凡例）↓：改行

この同義語一覧定義ファイルを使用して，同義語辞書Terminology_detail を作成します。リレーショナルデータベースに関する文章データを検索する場合や，NoSQL データベースに関する文章データを検索する場合などに，同義語辞書Terminology_detail を使用します。

11.16.16　辞書情報登録ファイルの指定規則

辞書情報登録ファイルの指定規則について説明します。

(1)　辞書情報登録ファイルの指定例

同義語辞書Dictionary2 を登録するときの辞書情報登録ファイルの指定例を説明します。

＜辞書情報登録ファイルの指定例＞

```
Dictionary2,/home/adbmanager/dictionary2_synonym.txt,CORRECTIONRULE,一般用語↓
```

（凡例）↓：改行

[説明]

辞書情報登録ファイル中には，次に示す 4 つの項目を指定します。各項目は，コンマ（ , ）で区切って指定します。

- *同義語辞書名*（1 列目に指定します）
- *同義語一覧定義ファイル名*（2 列目に指定します）
- *表記ゆれ補正オプション*（3 列目に指定します）
- *コメント*（4 列目に指定します）

(2)　辞書情報登録ファイル中の各項目に指定する内容

■1 列目

同義語辞書名：～〈文字列〉((1～120 バイト))

同義語辞書名を指定します。同義語辞書名は SQL 文中に指定するため，わかりやすい名称を指定することを推奨します。

同義語辞書名に使用できる文字列は，半角英字（A～Z，a～z），半角数字（0～9），および下線文字(_)になります。

重要

すでに登録されている同義語辞書名を指定した場合，同義語辞書が再登録されてしまうため，同義語辞書を新規に登録するときは，すでに登録されている辞書名を指定しないように注意してください。登録済みの同義語辞書があるかどうかは，システム表を検索すると，確認できます。システム表の検索方法については，「11.16.4　同義語辞書の情報を確認する」の「(2)　すべての同義語辞書の情報を確認する」を参照してください。

■2 列目

同義語一覧定義ファイル名：～〈OS パス名〉((2～510 バイト))

同義語一覧定義ファイル名を絶対パス名で指定します。

同義語一覧定義ファイル名の指定規則を次に示します。

- 制御文字（0x00～0x1f，0x7f）は指定できません。

■3 列目

表記ゆれ補正オプション：

表記ゆれ補正検索に対応する同義語辞書を作成するかどうかを指定します。次のどちらかを指定します。

- CORRECTIONRULE またはCR

  表記ゆれ補正検索に対応する同義語辞書を作成する場合に指定します。

- CASESENSITIVE またはCS

  表記ゆれ補正検索に対応しない同義語辞書を作成する場合に指定します。

CORRECTIONRULE を指定して作成した同義語辞書は，表記ゆれ補正検索をしない場合にも使用できます。そのため，表記ゆれ補正検索を行うかどうかが，同義語辞書の作成時に不明な場合は，CORRECTIONRULE を指定することを推奨します。

メモ

- 表記ゆれ補正検索を行うかどうかは，同義語検索の実行時にスカラ関数CONTAINS の指定で選択できます。
- 表記ゆれ補正検索に対応しない同義語辞書を，表記ゆれ補正検索に対応する同義語辞書に変更する場合，同義語辞書を再登録する必要があります。

重要

HADB サーバで使用する文字コードが Shift-JIS の場合（環境変数ADBLANG の指定値がSJIS の場合），表記ゆれ補正検索は実行できません。そのため，表記ゆれ補正オプションに CORRECTIONRULE を指定することはできません。

表記ゆれ補正オプションの指定を省略した場合，CASESENSITIVE が仮定されます。

（例）表記ゆれ補正オプションの指定を省略した場合の例

```
Dictionary3,/home/adbmanager/dictionary3_synonym.txt,,詳細用語
```

■4 列目

コメント：～〈文字列〉((0～1,024 バイト))

同義語辞書に対するコメントを指定します。

同義語検索時に使用する同義語辞書がわからなくなってしまった場合，同義語辞書の一覧を表示して，使用する同義語辞書を確認します。その際，ここで指定したコメントから同義語辞書が特定できるように，同義語辞書の内容と関連があるコメントを指定するようにしてください。

> **メモ**
>
> - 同義語辞書の一覧を表示するには，システム表を検索します。詳細については，「11.16.4　同義語辞書の情報を確認する」の「(2)　すべての同義語辞書の情報を確認する」を参照してください。
> - 同義語辞書の一覧には，同義語辞書名とここで指定したコメントが表示されます。

コメントの指定規則を次に示します。

- 制御文字（0x00～0x1f，0x7f）は指定できません。
- コメントを省略することもできます。

（例）コメントの指定を省略した場合の例

```
Dictionary2,/home/adbmanager/dictionary2_synonym.txt,CORRECTIONRULE,
```

(3)　辞書情報登録ファイルの指定規則

- 辞書情報登録ファイルには，複数の同義語辞書に関する情報を指定できます。詳細は，「(4)　留意事項」を参照してください。
- 登録できる同義語辞書の数は最大 50 個です。
- 1 つの同義語辞書に関する情報を 1 行に記述します。
- 各行の終わりに改行を入力してください。また，辞書情報登録ファイルの最後の行にも改行を入力してください。改行コードは，X'0A'（LF），X'0D0A'（CRLF），またはX'00'のどれかにしてください。
- 改行だけの行がある場合，その行を読み飛ばして処理が実行されます。
- 環境変数ADBLANG に指定した文字コードで辞書情報登録ファイルを作成してください。
- 二重引用符（"）を囲み文字として指定することができます。コメントなど，指定する文字列中に「,」がある場合は，囲み文字を指定してください。

  （例）

```
Dictionary3,/home/adbmanager/dictionary2_synonym.txt,CORRECTIONRULE,"一般用語,法律関連"
```

(4)　留意事項

辞書情報登録ファイルには，次に示すように複数の同義語辞書の辞書情報を指定できます。この場合，一度のadbsyndict コマンドの実行で，複数の同義語辞書を登録することができます。

（例）

```
Dictionary1,/home/adbmanager/dictionary2_synonym.txt,CORRECTIONRULE,一般用語↓
Dictionary2,/home/adbmanager/dictionary3_synonym.txt,CASESENSITIVE,詳細用語↓
```

（凡例）↓：改行

例えば，次のような運用をする場合は，上記のように複数の同義語辞書を指定してください。

- Dictionary1 の同義語を追加，変更，または削除する場合，Dictionary2 の同義語も同様に追加，変更，または削除する必要がある場合

このような運用をしない場合は，1 つの辞書情報登録ファイルには，1 つの同義語辞書の辞書情報だけを指定することを推奨します。

11.17　システム表の再編成

ここでは，システム表の再編成について説明します。

システム表の詳細については，「付録 C　システム表」を参照してください。

11.17.1　システム表を再編成する理由

HADB ユーザが，実表に対して SQL 文やコマンドを実行した場合，HADB サーバによって自動的にシステム表（実表）も更新されることがあります。その際，システム表（実表）に無効な行データが増加することがあります。無効となった行データは，ディスクから削除されません。

- システム表（実表）の行が削除されると，削除対象の行データは無効となります。
- システム表（実表）の行が更新されると，更新後の行データが新たな行として追加されます。更新前の行データは無効となります。

システム表（実表）に無効な行データが増加し続けた場合，システム表（実表）を格納するディスクの空き容量が不足し，最終的にシステム表（実表）の更新が発生するような運用は実行できなくなります。例えば，バックグラウンドインポートによるデータインポートやチャンクのマージなどは実行できなくなります。

そのため，定期的にシステム表（実表）を再編成して，無効な行データを削除する必要があります。システム表（実表）の再編成は，adbreorgsystemdata コマンドで行います。

システム表（実表）に無効な行データが増加するタイミングを，次の表に示します。

表 11-9　システム表（実表）に無効な行データが増加するタイミング

項番	システム表（実表）の種類	システム表（実表）に無効な行データが増加する操作
1	• STATUS_TABLES • STATUS_COLUMNS • STATUS_INDEXES	• 定義系 SQL による表やインデクスの削除※ • adbgetcst コマンドによるコスト情報の収集や削除 • ALTER TABLE 文による表名の変更※ • ALTER TABLE 文による列名の変更※
2	STATUS_CHUNKS	• 定義系 SQL によるマルチチャンク表の削除 • ALTER TABLE 文によるマルチチャンク表の表名変更 • PURGE CHUNK 文によるチャンクの削除 • TRUNCATE TABLE 文によるマルチチャンク表に格納されているデータの一括削除 • 作成モードによるマルチチャンク表へのデータインポート（adbimport コマンドに-d オプションを指定して実行したとき） • バックグラウンドインポートによるマルチチャンク表へのデータインポート（adbimport コマンドに-b オプションを指定して，かつ--status wait オプションを指定しないで実行したとき）

項番	システム表（実表）の種類	システム表（実表）に無効な行データが増加する操作
		• 追加モードによるマルチチャンク表へのデータインポート，かつチャンクコメントの変更（adbimport コマンドに-m オプションを指定して実行したとき） • adbidxrebuild コマンドによるインデクスの再作成（バックグラウンドインポートによるマルチチャンク表へのデータインポートが中断したあとに実行したとき） • adbmergechunk コマンドによるチャンクのマージ • adbchgchunkcomment コマンドによるチャンクのコメントの設定，変更および削除 • adbchgchunkstatus コマンドによるチャンクの状態変更 • adbarchivechunk コマンドによるチャンクのアーカイブ • adbunarchivechunk コマンドよるチャンクのアンアーカイブ
3	STATUS_SYNONYM_DICTIONARIES	adbsyndict コマンドによる同義語辞書の登録，更新および削除

注※

処理対象の表やインデクスに対して，コスト情報を収集している場合が対象です。

11.17.2　システム表を再編成するタイミング

無効な行データの増加によるディスクの空き容量の不足を防ぐために，adbreorgsystemdata コマンドを実行して，システム表（実表）を再編成する必要があります。

システム表（実表）を再編成するタイミングは，システム表用 DB エリアファイルを，レギュラーファイルとブロックスペシャルファイルのどちらに作成したかによって異なります。

■レギュラーファイルの場合

無効な行データが増加して，システム表（実表）を格納するディスクの空き容量が不足しても，HADB サーバから警告メッセージは出力されません。

そのため，運用を開始する前に，次に示す手順でシステム表（実表）を再編成するタイミングを検討してください。そして，運用を開始したあとに，検討したタイミングで，システム表（実表）を再編成するようにしてください。

> **メモ**
>
> ブロックスペシャルファイルの場合は，再編成が必要なタイミングで，HADB サーバから警告メッセージが出力されます。そのため，システム表用 DB エリアファイルは，ブロックスペシャルファイルに作成することを推奨します。レギュラーファイルからブロックスペシャルファイルに変更する方法は，「11.10.5　データ用 DB エリアファイルの格納場所を変更する方法」の「(1)　データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する」を参照してください。

手順

1. システム表（実表）を再編成する頻度を決める

adbreorgsystemdata コマンドで，システム表（実表）を再編成する頻度を決めてください。基本的には，半年に 1 回を目安としてください。

2. システム表用 DB エリアの容量を見積もる

   手順 1.で決めた再編成の頻度を踏まえて，システム表用 DB エリアの容量を見積もってください。見積もりを行う場合は，「5.12　システム表用 DB エリアの容量見積もり」を参照してください。

3. 見積もり結果を基にデータベースを作成する

   手順 2.で求めたシステム表用 DB エリアの容量を基に，ディスクを準備して，データベースを作成してください。

4. 定期的にシステム表（実表）を再編成する

   運用を開始したら，手順 1.で決めた再編成の頻度に従って，adbreorgsystemdata コマンドを実行してください。

■ブロックスペシャルファイルの場合

無効な行データが増加して，システム表（実表）を格納するディスクの空き容量が不足すると，HADB サーバから次の警告メッセージが出力されます。

- KFAA61213-W メッセージ
- KFAA61214-W メッセージ

警告メッセージが出力されたら，adbreorgsystemdata コマンドを実行して，システム表（実表）を再編成してください。

なお，警告メッセージが出力されなくても，運用に合わせてシステム表（実表）を再編成しても問題ありません。その場合は，半年に 1 回を目安に，adbreorgsystemdata コマンドを実行することを推奨します。

11.17.3　システム表を再編成する方法

システム表（実表）を再編成する場合は，adbreorgsystemdata コマンドを実行します。

adbreorgsystemdata コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbreorgsystemdata（システム表の再編成）』を参照してください。

> **重要**
>
> システム表（実表）の再編成処理中には，システム表（実表）を格納するシステム用 DB エリアには，次に示す 2 つのデータが一時的に存在します。
>
> - 再編成前のデータ
> - 再編成後のデータ
>
> そのため，ディスクに 2 つのデータを格納できるだけの空き容量がない場合は，システム表（実表）を再編成できません。

■adbreorgsystemdata コマンドの実行例

システム表（実表）のSTATUS_CHUNKS 表に対して，再編成を行います。また，adbreorgsystemdata コマンドがタイムアウトになるまでの時間として，300 秒を指定します。

```
adbreorgsystemdata --timeout 300
                   -c table
                   -f /home/adbmanager/reorg_file/unldpath.txt
                   -n STATUS_CHUNKS
```

メモ

adbreorgsystemdata コマンドを実行する際，--timeout オプションを指定することを推奨します。

adbreorgsystemdata コマンドを実行すると，通常は数十秒から数分程度で終了します。ただし，特定の SQL 文やコマンドを実行している場合は，adbreorgsystemdata コマンドが待ち状態となります。そのため，adbreorgsystemdata コマンドの実行時間が長くなるおそれがあります。adbreorgsystemdata コマンドを実行している間は，adbimport コマンドや adbmergechunk コマンドなどは実行できません。adbreorgsystemdata コマンドの待ち状態については，「11.17.4　システム表の再編成と排他制御」を参照してください。

adbreorgsystemdata コマンドの実行時間が長くなることで，システムの運用に支障が出る場合は，--timeout オプションを指定してください。--timeout オプションに指定した時間内でadbreorgsystemdata コマンドが終了しなかった場合は，再編成処理がキャンセルされます。

adbreorgsystemdata コマンドがキャンセルされた場合は，出力されたメッセージを確認したあとで，再度adbreorgsystemdata コマンドを実行してください。

11.17.4　システム表の再編成と排他制御

ここでは，adbreorgsystemdata コマンドと排他制御の関係，およびadbreorgsystemdata コマンドの待ち状態について説明します。

(1)　adbreorgsystemdata コマンドと排他制御の関係

adbreorgsystemdata コマンドを実行すると，HADB サーバは「2.10.4　確保される排他資源と排他制御モード」に従って，排他資源を確保します。

adbreorgsystemdata コマンドの実行中は，排他資源の「システム表用 DB エリア」を占有モード（EX）で確保します。そのため，adbimport コマンドやadbmergechunk コマンドなどは実行できません。

また，次に示すSQL文やコマンドを実行している場合，adbreorgsystemdataコマンドは待ち状態になります（KFAA80257-Iメッセージが出力されたあとで，再編成の処理が中断します）。実行しているSQL文やコマンドが終了したあとで，adbreorgsystemdataコマンドの処理が再開されます。

■adbreorgsystemdataコマンドを待ち状態にするSQL文とコマンド

- 検索系SQL（システム表（実表），またはアーカイブマルチチャンク表※に実行する場合）
- 更新系SQL（アーカイブマルチチャンク表※に実行する場合）
- adbexportコマンド（システム表（実表），またはアーカイブマルチチャンク表※に実行する場合）
- --shared-lockオプションを指定したadbdbstatusコマンド（システム表用DBエリアに実行する場合）
- --shared-lockオプションを指定したadbdbstatusコマンド（adbreorgsystemdataコマンドの処理対象であるシステム表（実表）に実行する場合）
- --shared-lockオプションを指定したadbdbstatusコマンド（adbreorgsystemdataコマンドの処理対象であるシステム表（実表）のインデクスに実行する場合）

注※

adbreorgsystemdataコマンドを，システム表（実表）のSTATUS_CHUNKS表に実行するときが該当します。STATUS_CHUNKS表以外のシステム表（実表）に対してadbreorgsystemdataコマンドを実行するときは，アーカイブマルチチャンク表に対象のSQL文とコマンドを実行していても，adbreorgsystemdataコマンドは待ち状態にはなりません。

SQL文やコマンドの終了を待たないで，adbreorgsystemdataコマンドの待ち状態を解消したい場合は，「11.17.4　システム表の再編成と排他制御」の「(2)　adbreorgsystemdataコマンドの待ち状態を解消する方法」を参照してください。

(2)　adbreorgsystemdataコマンドの待ち状態を解消する方法

待ち状態の原因となっているSQL文やコマンドの終了を待たないで，adbreorgsystemdataコマンドの待ち状態を解消したい場合は，対象のSQL文やコマンドを終了する必要があります。

adbreorgsystemdataコマンドが待ち状態の原因となっているSQL文やコマンドを終了したい場合は，次に示す手順を実施してください。

手順

1. adbls -d lockコマンドを実行する

   次の条件に該当する項目「CID」を確認してください。

   - 項目「RESOURCEID」に，adbreorgsystemdataコマンドの処理対象表の表IDが出力されている行

   adbreorgsystemdataコマンドの処理対象表であるシステム表（実表）の表IDについては，「付録C.1　システム表の概要」の「(1)　システム表の一覧」を参照してください。

2. adbls -d cnctコマンドを実行する

次のどちらかの条件に該当する項目「CID」を確認してください。

- 項目「CLIENT_TYPE」にcommand が出力されている行，かつ項目「CONNECT_TIME」に出力されている時刻が，前回実施したadbreorgsystemdata コマンドで出力されたKFAA80256-I メッセージの出力時刻よりも前である行
- 項目「CLIENT_TYPE」にcommand が出力されていない行，かつ項目「SQL_ELAPSED_TIME」に出力されているマイクロ秒から求めた SQL 文の経過時間が，前回実施したadbreorgsystemdata コマンドで出力されたKFAA80256-I メッセージの出力時刻よりも前である行

3. adbcancel コマンドを実行する

手順 1.および手順 2.の両方で確認した項目「CID」が，adbreorgsystemdata コマンドが待ち状態の原因となっている SQL 文またはコマンドのコネクション ID です。adbcancel コマンドにコネクション ID を指定して実行すると，adbreorgsystemdata コマンドの待ち状態が解消されます。

各コマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

11.17.5　システム表の状態や使用量を確認する方法

システム表（実表）の状態，使用量および格納効率を確認する方法について説明します。

(1)　システム表（実表）の状態や使用量を確認する

システム表（実表）の状態や使用量を確認する場合は，adbdbstatus コマンドで「表のサマリ情報」を出力してください。

adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析)』を参照してください。

実行するコマンドの例

ここでは，システム表（実表）のSTATUS_CHUNKS 表について，サマリ情報を出力します。

```
adbdbstatus -d summary -c table -n ”HADB”.”STATUS_CHUNKS” -S M --shared-lock
```

コマンドを実行すると，標準出力にシステム表HADB.STATUS_CHUNKS のサマリ情報が出力されます。
adbdbstatus コマンドによる出力結果の確認項目を次に示します。

再編成前のデータが残っているかどうかを確認する場合

Pending_delete_chunks の出力内容を確認してください。Pending_delete_chunks に出力されている値が1 の場合，再編成前のデータ（削除仕掛中のチャンク）が残っています。

adbreorgsystemdata コマンドを再実行すると，再編成前のデータを削除できます。

システム表（実表）の使用量を確認する場合

MB_Used_pages の出力内容を確認してください。システム表用 DB エリア内に格納されているシステム表（実表）のデータが使用している領域を，メガバイト単位で確認できます。

なお，--shared-lock オプションを指定した場合は，MB_Used_pages の出力内容に，削除仕掛中のチャンクが使用している領域は含まれません。MB_Used_pages の出力内容に，削除仕掛中のチャンクが使用している領域を含めたい場合は，--shared-lock オプションを指定しないでください。

(2) システム表（実表）の格納効率を確認する

システム表（実表）の格納効率を確認する場合は，SQL トレース機能を使用する必要があります。

手順

1. SQL トレース情報が出力されるようにする

   SQL トレース機能を使用して，次に示す SQL トレース情報が出力されるようにしてください。

   - アクセスパス情報
   - アクセスパスの統計情報

   SQL トレース情報の出力については，「10.11.5　SQL トレース情報を出力するための準備」を参照してください。

   > **メモ**
   >
   > 次に示すサーバ定義のオペランドにY を指定すれば，アクセスパス情報およびアクセスパスの統計情報が出力されます。
   >
   > - adb_sql_trc_out オペランド
   > - adb_sql_trc_accesspath オペランド
   >
   > サーバ定義のオペランドについては，「7.2.5　SQL 文に関するオペランド（set 形式）」を参照してください。
   >
   > また，HADB サーバを開始している場合は，adbchgsqltrc コマンドを使用してください。adbchgsqltrc コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbchgsqltrc（SQL トレース情報の出力開始・停止）』を参照してください。

2. SQL 文でシステム表の行数を確認する

   格納効率を確認するシステム表に対して，DBA 権限を持つ HADB ユーザで，行数を出力する SQL 文を実行してください。SQL 文を実行して出力された行数が，システム表の行数となります。

   各システム表に対応する SQL 文を次に示します。

   ■STATUS_TABLES 表の行数を出力する SQL 文

   ```
   SELECT COUNT(COLLECT_TIME)
     FROM MASTER.STATUS_TABLES
   ```

   ■STATUS_COLUMNS 表の行数を出力する SQL 文

   ```
   SELECT COUNT(DATA_TYPE_CODE)
     FROM MASTER.STATUS_COLUMNS
   ```

■STATUS_INDEXES 表の行数を出力する SQL 文

```
SELECT COUNT(COLLECT_TIME)
  FROM MASTER.STATUS_INDEXES
```

■STATUS_CHUNKS 表の行数を出力する SQL 文

```
SELECT COUNT(NVL(CREATE_TIME,CURRENT_TIMESTAMP))
  FROM MASTER.STATUS_CHUNKS
```

■STATUS_SYNONYM_DICTIONARIES 表の行数を出力する SQL 文

```
SELECT COUNT(CREATE_TIME)
  FROM MASTER.STATUS_SYNONYM_DICTIONARIES
```

3. 無効な行データの行数を確認する

手順 2.で実行した SQL 文によって出力された SQL トレース情報のアクセスパス情報から，システム表内の「無効な行データの行数」を確認してください。次の条件に該当するアクセスパス情報の「Data_deleted_rows_cnt」の値を確認してください。

- スキーマ名が「HADB」である。
- 表名が「手順 2.で実行した SQL 文に該当するシステム表」である。

条件に該当する「Data_deleted_rows_cnt」の値が，対象のシステム表内にある無効な行データの行数となります。

4. システム表（実表）の格納効率を求める

手順 2.と手順 3.で求めた値を基に，対象のシステム表（実表）の格納効率を求めてください。計算式を次に示します。

計算式

```
システム表(実表)の格納効率＝
    手順2. の結果÷(手順2. の結果＋手順3. の結果)
```

計算式で求めた値が 0 に近い場合は，システム表（実表）の格納効率が悪いと判断できます。adbreorgsystemdata コマンドで，システム表を再編成してください。

11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）

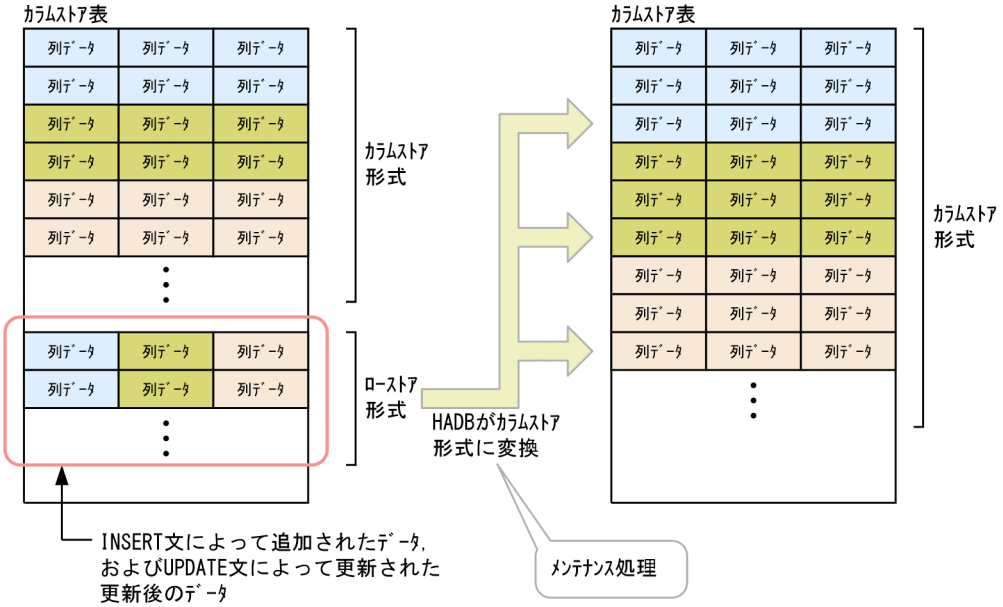
カラムストア表の検索性能を維持するための機能（更新行のカラム化機能）の使用方法について説明します。

11.18.1　更新行のカラム化機能とは

カラムストア表に対してINSERT 文またはUPDATE 文を実行すると，追加または更新されたデータはローストア形式でカラムストア表に格納されます。カラムストア表にローストア形式のデータが格納された場合，カラムストア表の特性を生かした検索（特定の列データの検索や，特定の列データに対する値の集計など）の性能が低下するおそれがあります。また，データの圧縮率も低下するおそれがあります。**更新行のカラム化機能**を使用すると，HADB が，カラムストア表に格納されたローストア形式のデータをカラムストア形式に自動的に変換します。これによって，カラムストア表の検索性能の低下，およびデータの圧縮率の低下を防ぐことができます。カラムストア表に対してINSERT 文またはUPDATE 文を実行することがある場合は，更新行のカラム化機能を使用してください。

更新行のカラム化機能の概要を次の図に示します。

図 11-47　更新行のカラム化機能の概要



メモ

更新行のカラム化機能を使用するには，adbcolumnize コマンドを実行して更新行のカラム化機能を有効にする必要があります。デフォルトの設定では，更新行のカラム化機能は動作しません。詳細については，「11.18.2　準備作業」を参照してください。

■メンテナンス処理

カラムストア表に格納されたローストア形式のデータをカラムストア形式に変換する処理を**メンテナンス処理**といいます。メンテナンス処理は，解析処理，再配置処理，および再利用処理の 3 つの処理から構成されています。メンテナンス処理の詳細については，「11.18.7　メンテナンス処理の詳細」を参照してください。

重要

- トランザクションおよびコマンドの処理性能に影響を及ぼさないように，トランザクションまたはコマンドが実行されている間は，メンテナンス処理は実行されません。また，メンテナンス処理の実行中にトランザクションまたはコマンドが開始された場合は，メンテナンス処理を中断し，トランザクションまたはコマンドの処理が完了したあとにメンテナンス処理を再開します。

  なお，ここでいうコマンドとは，HADB サーバに接続するコマンドのことです。HADB サーバに接続するコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧と共通規則』の『コマンドの一覧』を参照してください。
- トランザクションまたはコマンドが常に実行中であるようなシステムの場合は，メンテナンス処理を実行する時間を確保できないため，更新行のカラム化機能の使用を見合わせてください。

■表またはチャンクの再編成との関係

更新行のカラム化機能を使用すると，ローストア形式のデータがカラムストア形式に変換されますが，次に示す無効となった行データは削除されません。

- UPDATE 文によって更新された更新前の行データ
- DELETE 文によって削除された行データ

無効となった行データを削除するには，表またはチャンクの再編成を実行する必要があります。表またはチャンクの再編成を実行すると，無効となった行データが格納されていたページを再利用できます。

また，一度に大量のデータをINSERT 文またはUPDATE 文で追加・更新した場合，大量のローデータ用セグメントが作成されます。そのため，更新行のカラム化機能を使用しても，検索性能の低下やデータベース容量の増加を防止できません。したがって，一度に大量のデータをINSERT 文またはUPDATE 文で追加・更新した場合は，そのあとに表またはチャンクの再編成の実行を検討してください。

メモ

更新行のカラム化機能と，表またはチャンクの再編成を組み合わせた運用を推奨します。運用例については，「11.4.23　バックグラウンドインポートとチャンクを意識した運用例 1（定期的にデータの追加と削除を繰り返す運用）」を参照してください。

■制限事項

B-tree インデクスが定義されているカラムストア表は，更新行のカラム化機能の適用対象外となります。

11.18.2　準備作業

更新行のカラム化機能を使用する際の準備作業について説明します。

(1)　機能の適用対象となるカラムストア表を確認する

B-tree インデクスが定義されているカラムストア表は，更新行のカラム化機能の適用対象外となります。そのため，更新行のカラム化機能を有効にする前に，「付録 B.22　ディクショナリ表の検索」の「(38) 更新行のカラム化機能の対象となる表を調べる方法（B-tree インデクスを定義していないカラムストア表を調べる方法)」を参照して，更新行のカラム化機能の適用対象となるカラムストア表を確認してください。

更新行のカラム化機能の適用対象となるカラムストア表に対して，INSERT 文またはUPDATE 文を実行することがある場合は，更新行のカラム化機能を使用します。

(2)　HADB サーバが使用するメモリ所要量を見積もる

更新行のカラム化機能を使用する場合，次の個所を参照して HADB サーバが使用するメモリ所要量を再見積もりしてください。

- 「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時)」
- 「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時)」

(3)　監視対象のメッセージを追加する

メンテナンス処理が，何らかの要因によって停止していないかを，次のメッセージで確認できます。次のメッセージを監視対象に追加してください。

- KFAA41220-W

  更新行のカラム化機能で使用するファイルの入出力制御の開始処理に失敗した場合に出力されるメッセージです。

- KFAA51277-I

  メンテナンス処理が 24 時間動作していない場合に出力されるメッセージです。

- KFAA51280-W

  メンテナンス処理でエラーが発生した場合に出力されるメッセージです。

■メッセージの出力例

更新行のカラム化機能に関するエラーメッセージがメッセージログファイルに出力された場合，エラーの原因を示すメッセージと，更新行のカラム化機能でエラーが発生したことを示すメッセージを一緒に出力します。

（例）

```
2019/06/24 15:13:28 … 0000000000999900001 KFAA40007-E メッセージテキスト     ...1
2019/06/24 15:13:28 … 0000000000999900001 KFAA51280-W メッセージテキスト     ...2
```

［説明］

1. エラーの原因を示すメッセージです。メモリ不足が発生したことを示すメッセージです。このメッセージで説明されている対処方法に従います。
2. メンテナンス処理でエラーが発生したことを示すメッセージです。

更新行のカラム化機能に関連するメッセージのコネクション通番には，9999000001（上記の例の下線部分）が出力されます。

(4) 更新行のカラム化機能を有効にする

adbcolumnize コマンドを実行して，更新行のカラム化機能を有効にしてください。更新行のカラム化機能を有効にすると，メンテナンス処理が実行されます。

コマンドの実行例

```
adbcolumnize --start
```

更新行のカラム化機能が有効になったことを「11.18.3　更新行のカラム化機能の状態（有効または無効）を確認する方法」で説明している方法で確認してください。

メモ

- 更新行のカラム化機能の有効または無効の状態は，HADB サーバを再起動しても引き継がれます。
- マルチノード機能のプライマリノードの切り替え，コールドスタンバイ構成の系切り替えが発生した場合でも，更新行のカラム化機能の有効または無効の状態は引き継がれます。

11.18.3　更新行のカラム化機能の状態（有効または無効）を確認する方法

更新行のカラム化機能の状態（有効または無効）を確認したい場合は，adbcolumnize コマンドを実行してください。

コマンドの実行例

```
adbcolumnize -d
```

コマンドの実行結果の例

```
STATUS
ACTIVE
```

STATUS 欄に更新行のカラム化機能の状態が表示されます。ACTIVE が表示された場合は，更新行のカラム化機能は有効な状態です。INACTIVE が表示された場合は，更新行のカラム化機能は無効な状態です。

11.18.4　更新行のカラム化機能を一時的に無効にする場合

次に示す場合は，更新行のカラム化機能を一時的に無効にしてください。

- バックアップを取得する場合

  更新行のカラム化機能によってローストア形式のデータがカラムストア形式に変換される場合，データベースの更新処理が発生します。バックアップを取得する際に，データベースの更新を抑止したい場合は，更新行のカラム化機能を一時的に無効にしてください。

- SQL 文やコマンドの実行時間にまったく影響を与えたくない場合

  更新行のカラム化機能を有効にしている場合，SQL 文やコマンドの実行時間に多少の影響（0.5 秒程度）を与える可能性があります。影響をまったく与えたくない場合は，更新行のカラム化機能を一時的に無効にしてください。

  例えば，次のような処理を実行する場合に，更新行のカラム化機能を無効にするかどうかを検討してください。

  - 短時間（数秒～数分）で完了する SQL 文またはコマンドを実行する場合（実行時間への影響が大きいと判断した場合）
  - 実行時間に影響を与えてはいけない SQL 文またはコマンドを実行する場合
  - バッチ処理など，決められた時間に完了する必要がある処理を実行する場合

> **メモ**
>
> - ここでいうコマンドとは，「HADB サーバに接続するコマンド」のことです。
> - 実行時間の影響が 0.5 秒程度というのは目安です。処理を実行する際の条件によって異なります。メンテナンス処理の実行中に，SQL 文またはコマンドが実行された場合，メンテナンス処理の中断処理が発生し，その中断処理が完了するまで SQL 文またはコマンドの実行が待たされます。その時間が 0.5 秒程度掛かります。
> - DB エリアにフラッシュストレージではなくハードディスクを使用している場合，またはブロックスペシャルファイルではなくレギュラーファイルを使用している場合は，メンテナンス処理の中断処理に数秒以上掛かる可能性があります。

更新行のカラム化機能を無効にする場合は，adbcolumnize コマンドを実行してください。

コマンドの実行例

```
adbcolumnize --stop
```

更新行のカラム化機能が無効になったことを「11.18.3　更新行のカラム化機能の状態（有効または無効）を確認する方法」で説明している方法で確認してください。

11.18.5　更新行のカラム化機能に関するエラーが発生した場合の対処方法

更新行のカラム化機能に関するエラーが発生した場合，次のどちらかのメッセージが出力されます。それぞれの対処方法について説明します。

- KFAA51280-W

  メンテナンス処理中にエラーが発生した場合，このメッセージが出力されます。

- KFAA41220-W

  更新行のカラム化機能で使用するファイルの入出力制御の開始処理でエラーが発生した場合，このメッセージが出力されます。

(1)　メンテナンス処理中にエラーが発生した場合（KFAA51280-W）

メンテナンス処理中にエラーが発生した場合，KFAA51280-W メッセージがメッセージログファイルに出力されます。この場合，KFAA51280-W メッセージの対策欄を参照して，エラーの対処をしてください。

エラーの対処をしないで，更新行のカラム化機能を有効な状態のままにしておくと，メンテナンス処理が 6 時間ごとに再実行されて同じエラーが発生します（KFAA51280-W メッセージが再度出力されます）。

(2)　更新行のカラム化機能で使用するファイルの入出力制御の開始処理でエラーが発生した場合（KFAA41220-W）

HADB サーバの開始時に，更新行のカラム化機能で使用するファイルの入出力制御の開始処理が実行されます。更新行のカラム化機能で使用するファイルの入出力制御の開始処理でエラーが発生した場合，KFAA41220-W メッセージがメッセージログファイルに出力されます。この場合，KFAA41220-W メッセージの対策欄の説明に従って，エラーの対処をしてください。エラーの対処をしないで，更新行のカラム化機能を有効な状態のままにしておくと，メンテナンス処理が 6 時間ごとに再実行されます。このとき，更新行のカラム化機能で使用するファイルの入出力制御の開始処理が実行されて，同じエラーが発生します（KFAA41220-W メッセージが再度出力されます）。

11.18.6　メンテナンス処理が実行されない場合の対処方法

次のどれかの条件に該当する場合は，メンテナンス処理が実行されません。

- トランザクションまたは HADB サーバに接続するコマンドが実行中の場合
- HADB サーバの稼働モードが静止モードの場合
- ノードの復帰処理中の場合（マルチノード機能使用時）

メンテナンス処理が 24 時間実行されなかった場合，KFAA51277-I メッセージが出力されます。そのあとも，メンテナンス処理が実行されない状況が続くと，KFAA51277-I メッセージが 24 時間ごとに出力されます。

上記の条件ごとに対処方法を説明します。

(1)　トランザクションまたは HADB サーバに接続するコマンドが実行中の場合

トランザクションまたは HADB サーバに接続するコマンドが実行中の場合，トランザクションおよびコマンドの処理性能に影響を及ぼさないように，メンテナンス処理は実行されません。また，メンテナンス処理の実行中にトランザクションまたはコマンドが開始された場合は，メンテナンス処理が中断されます。

メンテナンス処理が長時間実行されない場合，または長時間実行されないと予想される場合は，定期的にカラムストア表の再編成要否を確認してください。再編成が必要な場合は，再編成を実行してください。カラムストア表の再編成要否の確認方法については，次の個所を参照してください。

- 「11.1.10　シングルチャンク表の再編成が必要かどうかを確認する方法」の「(3)　シングルチャンク表がカラムストア表の場合」
- 「11.4.13　マルチチャンク表の再編成が必要かどうかを確認する方法」の「(2)　再編成が必要かどうかを確認する方法」の「(b)　マルチチャンク表がカラムストア表の場合」

(2)　HADB サーバの稼働モードが静止モードの場合

HADB サーバの稼働モードが静止モードの場合，データベースの更新が抑止されるため，メンテナンス処理が実行されません。HADB サーバの稼働モードを静止モードからほかのモードに変更すると，メンテナンス処理が実行されます。

稼働モードの確認方法，および変更方法については，「10.2.3　HADB サーバの稼働モード」を参照してください。

(3)　ノードの復帰処理中の場合（マルチノード機能使用時）

ノードの復帰処理中の間は，メンテナンス処理が実行されません。ノードの復帰処理が完了すると，メンテナンス処理が実行されます。ノードの復帰については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

11.18.7　メンテナンス処理の詳細

更新行のカラム化機能のメンテナンス処理について説明します。

(1)　メンテナンス処理とは

メンテナンス処理は，次の 3 つの処理で構成されています。

- 解析処理

  カラムストア表内のローストア形式のデータの格納状況を解析します。HADB によって再配置処理が必要と判断された場合，再配置処理と再利用処理が実行されます。

解析処理の開始タイミングは，HADB によって制御されています。

- 再配置処理

  カラムストア表内のローストア形式のデータをカラムストア形式に変換します。変換したデータをカラムストア形式のデータを格納するセグメント（カラムデータ用セグメント）に格納します。

- 再利用処理

  カラムストア形式に変換したローストア形式のデータを削除し，ローストア形式のデータを格納していたページを再利用可能なページにします。再利用可能となったページは，INSERT 文またはUPDATE 文によって追加または更新されたデータをローストア形式で格納する際に再利用されます

メモ

- 更新不可状態のカラムストア表は，メンテナンス処理の対象になりません。
- レンジインデクスが未完状態のカラムストア表は，メンテナンス処理の対象になりません。

(2) メンテナンス処理が中断されるケース

次の場合，メンテナンス処理が中断されます。

- トランザクションが開始された場合，または HADB サーバに接続するコマンドが実行された場合

  トランザクションが開始された場合，または HADB サーバに接続するコマンドが実行された場合，実行中のメンテナンス処理が中断されます。トランザクションまたはコマンドの終了後に，メンテナンス処理が再開されます。

- HADB サーバとのコネクションが確立された場合

  HADB サーバとのコネクションを確立する場合，ユーザ認証と権限チェックのためにトランザクションが一時的に開始されます。そのため，実行中のメンテナンス処理が中断されます。HADB サーバとのコネクションが確立すると，このトランザクションは終了します。トランザクションの終了後，メンテナンス処理が再開されます。

- adbcolumnize --stop コマンドを実行した場合

  adbcolumnize --stop コマンドを実行して更新行のカラム化機能を無効にした場合，実行中のメンテナンス処理が中断されます。adbcolumnize --start コマンドを実行して更新行のカラム化機能を有効にすると，メンテナンス処理が再開されます。

- adbstop コマンドを実行した場合

  adbstop コマンドを実行して HADB サーバを終了した場合，実行中のメンテナンス処理が中断されます。adbstart コマンドを実行して HADB サーバを開始すると，メンテナンス処理が再開されます。

- adbchgsrvmode --quiescence コマンドを実行した場合

  adbchgsrvmode --quiescence コマンドを実行して，HADB サーバの稼働モードを静止モードに変更した場合，実行中のメンテナンス処理が中断されます。adbchgsrvmode コマンドを実行して，HADB サーバの稼働モードを静止モード以外に変更すると，メンテナンス処理が再開されます。

- ノードの復帰処理が開始された場合（マルチノード機能の使用時）

ノードの復帰処理が開始された場合，実行中のメンテナンス処理が中断されます。ノードの復帰処理の完了後に，メンテナンス処理が再開されます。

12 監査証跡機能の運用

この章では，監査証跡機能の運用について説明します。

12.1　監査証跡機能を使用する際に検討する項目

ここでは，監査証跡機能を使用する際に，事前に検討が必要な項目について説明します。

12.1.1　監査対象定義の検討（監査証跡を出力するイベントの選定）

監査証跡機能では，監査対象イベントごとに監査証跡を出力できます。しかし，すべてのイベントで監査証跡を出力した場合，監査証跡の出力量が膨大になり，監査の効率が低下するおそれがあります。そのため，監査の目的に応じて必要となる監査証跡を検討し，監査の対象とするイベントを決めてください。監査の目的と監査の対象とするイベントの例を次の表に示します。

表 12-1　監査の目的と監査の対象とするイベントの例

監査の目的	監査の対象とするイベントの例	監査対象イベントの種類
HADB サーバの設定が不正に変更されていないことを監査で確認する場合	HADB サーバの開始，終了	必須監査イベント
監査証跡機能に対して不正な操作が行われていないことを監査で確認する場合	監査証跡機能に対する操作	必須監査イベント
HADB サーバに対する不審な接続がないことを監査で確認する場合	HADB サーバへの接続	選択監査イベント
データの改ざんや，持ち出しがないことを監査で確認する場合	表の作成，検索，更新，および削除	選択監査イベント
セキュリティポリシーに従った運用が行われていることを監査で確認する場合（パスワードが定期的に変更されているかなど）	HADB ユーザのパスワードの変更	選択監査イベント

必須監査イベントとは，監査証跡機能が有効な場合に監査証跡が必ず出力されるイベントです。選択監査イベントとは，監査証跡を出力するかどうかを選択できるイベントです。選択監査イベントで監査証跡を出力したい場合は，CREATE AUDIT 文で監査対象を定義する必要があります。

監査の目的に応じて，必須監査イベントだけを監査対象とするか，選択監査イベントも監査対象に含めるかを決めてください。

監査対象イベントの一覧（必須監査イベントの一覧および選択監査イベントの一覧）については，「12.9.1 監査対象イベントの一覧と出力項目」を参照してください。

12.1.2　監査証跡機能で使用するディスクの設計

監査証跡機能で使用するディスク（記憶領域）の設計について説明します。

監査証跡機能で使用するディレクトリを次に示します。

- 監査証跡の出力先ディレクトリ
- 監査証跡の保存先ディレクトリ
- 監査証跡の長期保存先ディレクトリ

各ディレクトリを作成するディスク（記憶領域）は，それぞれ専用に用意することを推奨します。すべてのディレクトリを 1 つのディスク（記憶領域）に作成した場合，ディスク（記憶領域）に対する I/O が集中するため，HADB サーバの処理性能が低下するおそれがあります。

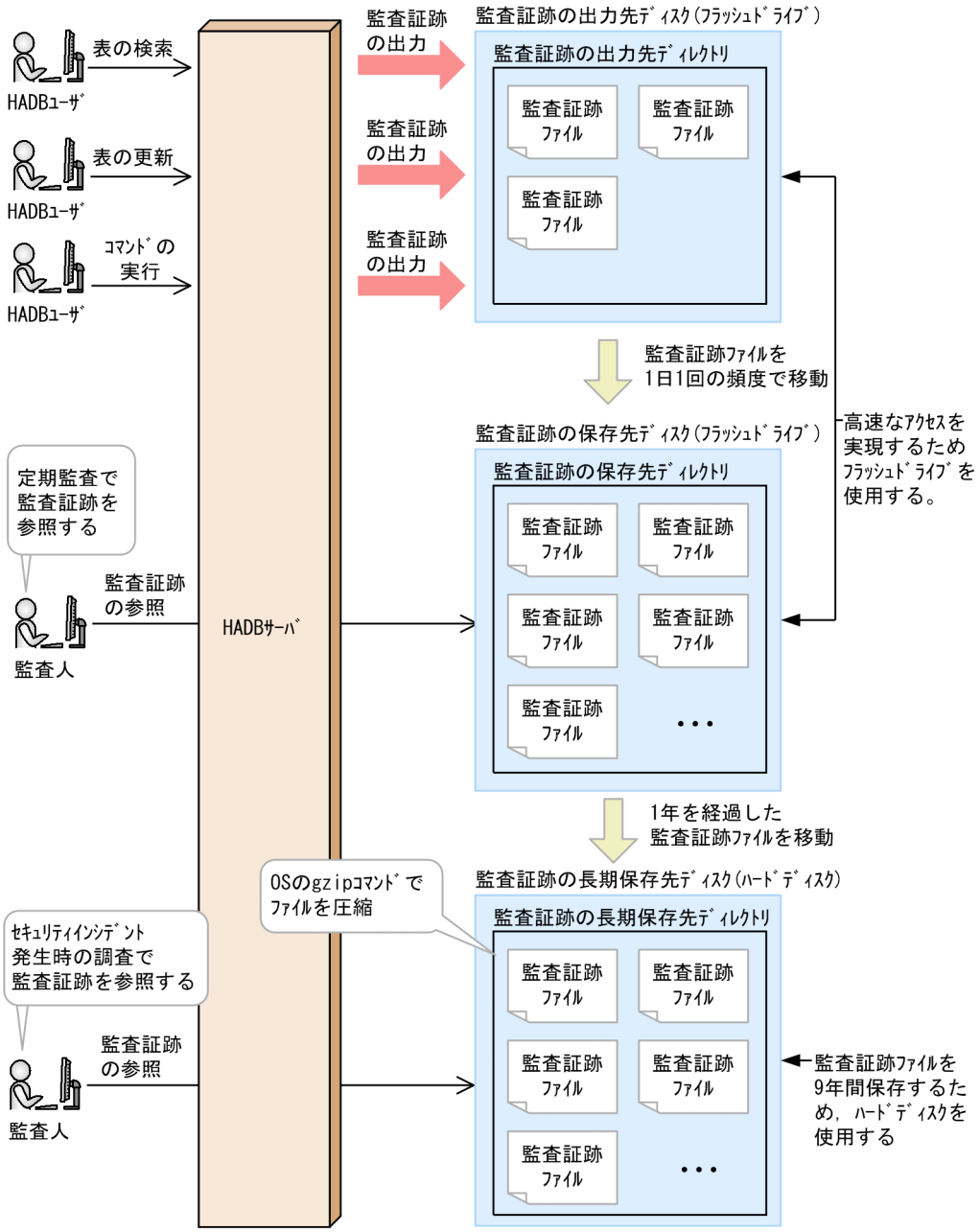
また，監査証跡を参照する際，監査証跡ファイルに対する書き込みは発生しません。そのため，次に示す 2 つのディレクトリは，ライトワンス型のディスク（記憶領域）に作成することを推奨します。ライトワンス型のディスク（記憶領域）に作成することで，監査証跡ファイルに対する改ざんを防止する効果が期待できます。

- 監査証跡の保存先ディレクトリ
- 監査証跡の長期保存先ディレクトリ

上記のディレクトリを作成するディスク（記憶領域）は，監査証跡の参照頻度や保存期間によって，使い分けることを推奨します。例えば，監査証跡の保存先ディレクトリに存在する監査証跡ファイルは，頻繁に参照することが考えられます。そのため，高速なアクセスを実現できるフラッシュドライブを使用することを検討してください。また，監査証跡の長期保存先ディレクトリに存在する監査証跡ファイルは，参照頻度が低く，かつ長期間保存しておくことが考えられるため，フラッシュドライブよりもハードディスクや磁気テープを使用することを検討してください。

監査証跡機能で使用するディスク（記憶領域）の構成例を，次の図に示します。この構成例では，監査証跡の保存先ディレクトリで，監査証跡ファイルを 1 年間保存します。その後，監査証跡の長期保存先ディレクトリで，監査証跡ファイルを 9 年間保存します。

図 12-1　監査証跡機能で使用するディスク（記憶領域）の構成例



監査証跡機能で使用するディスク（記憶領域）に必要な容量の求め方については，次の個所を参照してください。

- 「12.1.3　監査証跡のデータ容量の見積もり」
- 「12.1.4　監査証跡ファイルの圧縮率の見積もり」
- 「12.1.5　監査証跡機能で使用するディレクトリの容量見積もり」

12.1.3　監査証跡のデータ容量の見積もり

監査証跡機能に必要なディスク容量を見積もるためには，HADB サーバが出力する監査証跡のデータ容量を求める必要があります。

次のどちらかの方法で，監査証跡のデータ容量を求めてください。

1. 監査証跡機能を適用した小規模のテスト環境で，出力される監査証跡のデータ容量を確認する
2. 想定される HADB サーバの運用，およびデータベースへの操作を基に監査証跡のデータ容量を見積もる

監査証跡のデータ容量は，HADB サーバの環境，実行する SQL 文やコマンドなどによって大きく異なります。そのため，1.の方法で求めることを推奨します。テスト環境を準備できない場合は，2.の方法で求めてください。2.の方法で求める場合は，次の表に従って監査証跡のデータ容量を求めてください。

表 12-2　監査証跡の概算データ容量

項番	監査対象イベントの例	監査証跡の概算データ容量（単位：キロバイト）
1	HADB サーバの開始	4
2	HADB サーバの終了	1
3	HADB サーバへの接続※	1
4	HADB サーバからの切り離し※	1
5	SQL 文の実行	2 + SQL 文のサイズ
6	コマンドの実行	1

注※

コマンドの実行時にも発生します。

2.の方法で監査証跡のデータ容量を求める例を，次に示します。

■2.の方法で監査証跡のデータ容量を求める例

次の条件で，HADB サーバを運用することを想定します。

- HADB サーバを 1 日 1 回再起動する（HADB サーバの終了と HADB サーバの開始を行う）
- 1 コネクション当たり 2 キロバイトの SQL 文を 2 回実行する（その処理が毎秒 1 コネクション発生する）
- 必須監査イベントと選択監査イベントの両方について，監査証跡を出力する

この条件の場合，1 日（86,400 秒）当たりの監査証跡の概算データ容量を次に示します。

計算式

```
監査証跡の概算データ容量 =
    4キロバイト＋1キロバイト＋(1キロバイト＋(2キロバイト＋2キロバイト)×2＋1キロバイト)×
86,400秒
  = 864,005キロバイト/日
```

上記の条件で出力される監査証跡の概算データ容量は，1 日当たり 864,005 キロバイト（約 850 メガバイト）となります。

12.1.4　監査証跡ファイルの圧縮率の見積もり

監査証跡ファイルは，OS のgzip コマンドで圧縮できます。そのため，次に示すディレクトリで監査証跡ファイルを保存する場合，圧縮することでファイルサイズを削減できます。

- 監査証跡の保存先ディレクトリ
- 監査証跡の長期保存先ディレクトリ

監査証跡ファイルをgzip コマンドで圧縮する場合，監査証跡機能に必要なディスク容量を見積もるために，監査証跡ファイルの圧縮率を求める必要があります。

次のどちらかの方法で，監査証跡ファイルの圧縮率を求めてください。

1. 監査証跡機能を適用した小規模のテスト環境で作成された監査証跡ファイルをgzip コマンドで圧縮して，圧縮率を確認する
2. 圧縮率を 0.25 と仮定する（圧縮前の監査証跡ファイルが，1/4 のファイルサイズに圧縮されると仮定する）

監査証跡ファイルの圧縮率は，HADB サーバの環境，実行する SQL 文やコマンドなどによって大きく異なります。そのため，1.の方法で求めることを推奨します。テスト環境を準備できない場合は，2.の方法で求めてください。

12.1.5　監査証跡機能で使用するディレクトリの容量見積もり

「12.1.3　監査証跡のデータ容量の見積もり」および「12.1.4　監査証跡ファイルの圧縮率の見積もり」の見積もり結果を基に，監査証跡機能で使用するディレクトリの容量を見積もってください。そして，監査証跡機能で使用するディレクトリを作成するディスクを用意してください。容量の見積もりが必要なディレクトリは，次の 3 つです。

- 監査証跡の出力先ディレクトリ
- 監査証跡の保存先ディレクトリ
- 監査証跡の長期保存先ディレクトリ

出力される監査証跡のデータ容量や監査証跡ファイルの圧縮率は，HADB サーバの環境，実行する SQL 文やコマンドなどによって大きく異なります。そのため，見積もり結果よりも余裕を持って，ディスクの容量を準備してください。また，大容量のディスクを準備した場合でも，監査証跡機能の使用中は，ディスクの空き容量を必ず監視してください。

特に，監査証跡の出力先ディレクトリを作成するディスクが満杯にならないように，注意してください。監査証跡の出力先ディレクトリを作成するディスクが満杯になった場合，「12.1.7　監査証跡ファイルへの書き込みに失敗した場合の処理方式の検討」の選択に従って，HADB サーバは動作します。そのため，通常運用時だけでなく突発的な作業も考慮して，余裕を持ってディスクの容量を確保してください。

監査証跡機能で使用するディレクトリ容量の見積もり例を，次の表に示します。

■見積もり例の条件

- 監査証跡ファイルに出力される監査証跡のデータ容量は，1 日当たり 864,005 キロバイト（約 850 メガバイト）とする
- 監査証跡の出力先ディレクトリにある監査証跡ファイルを，1 日 1 回の頻度で監査証跡の保存先ディレクトリに移動させる
- 監査証跡ファイルは，合計で 10 年間保存する
  監査証跡の保存先ディレクトリで，監査証跡ファイルを 1 年間保存します。
  その後，監査証跡の長期保存先ディレクトリで，監査証跡ファイルを 9 年間保存します。
- 監査証跡の長期保存先ディレクトリで保存する監査証跡ファイルは，OS の gzip コマンドで圧縮する
  このときの監査証跡ファイルの圧縮率は 0.25 とします。
- 各ディレクトリ容量は，最低限必要な容量（必須容量）と余裕を持たせた容量（予備容量）を合計して求めることとする

表 12-3　監査証跡機能で使用するディレクトリ容量の見積もり例

項番	ディレクトリの種類	監査証跡ファイルの格納期間	必須容量（単位：ギガバイト）	予備容量（単位：ギガバイト）	合計容量（単位：ギガバイト）
1	監査証跡の出力先ディレクトリ	1 日	0.9	1.8（2 日分）	2.7
2	監査証跡の保存先ディレクトリ	1 年（365 日）	329	54（2 か月分）	383
3	監査証跡の長期保存先ディレクトリ	9 年（3,285 日）	740※1，※2	83（1 年分）※1	823

注※1

圧縮後の監査証跡ファイルの総容量です。

注※2

圧縮前の監査証跡ファイルの総容量は，約 2,957 ギガバイトです。

この見積もり例の場合，各ディレクトリを作成するために，上記の表で見積もった「合計容量」以上の容量があるディスクをそれぞれ用意する必要があります。

12.1.6　監査人の選定

監査証跡機能を使用する場合は，次の担当者（監査人）が必要になります。

- 監査証跡機能の管理担当者

  監査証跡機能の運用を行う担当者です。監査証跡機能の運用時に使用するadbaudittrailコマンドを実行するため，HADB 管理グループに所属する OS ユーザの OS アカウントが必要になります。

  また，監査管理権限を持っている HADB ユーザが必要になります。

- 監査担当者

  監査証跡を参照して，データベースの利用状況の監査を行う担当者です。

  監査証跡は，サーバマシン以外のマシンから参照できるため，監査担当者はサーバマシンの OS アカウントを必ず持つ必要はありません。サーバマシン上で監査証跡を参照する場合に限り，HADB 管理グループに所属する OS ユーザの OS アカウントが必要になります。

  また，監査参照権限を持っている HADB ユーザが必要になります。

  なお，監査担当者の操作に対しても監査を行う場合は，監査担当者を複数人用意して，監査担当者の操作については相互に監査するようにしてください。

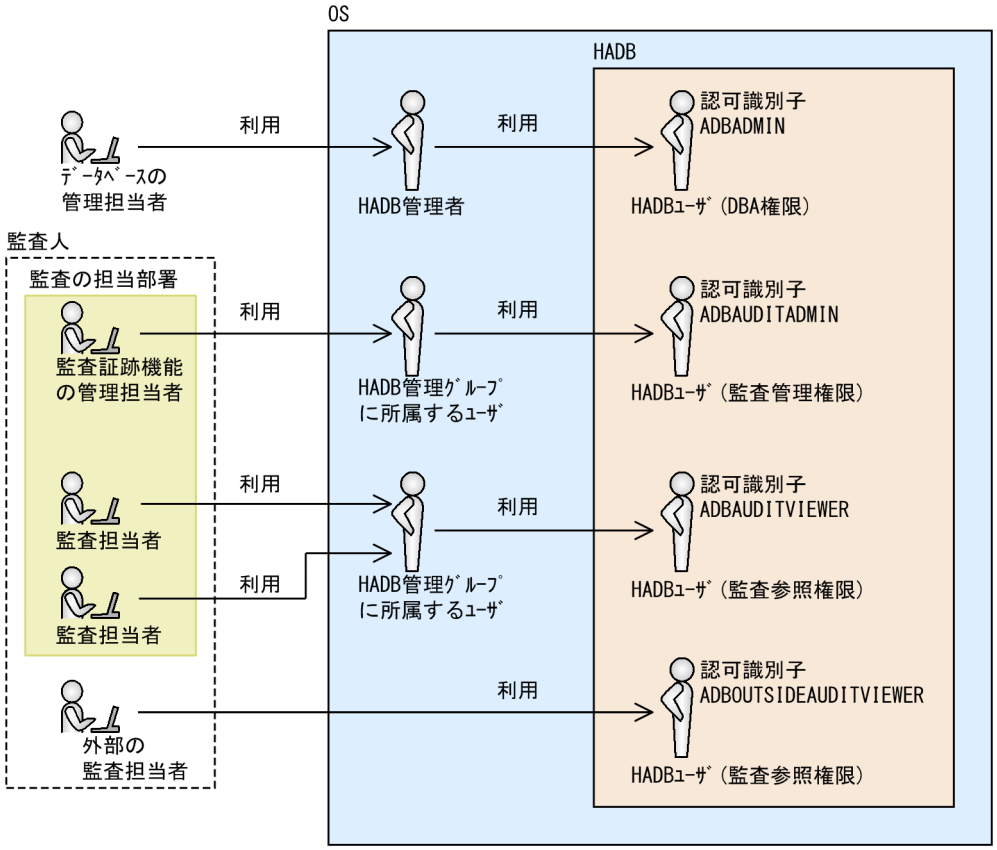
> **メモ**
>
> 外部の監査担当者による外部監査を行う場合は，外部の監査担当者には，監査参照権限を持っている HADB ユーザを用意することを推奨します。

監査証跡機能の管理担当者と監査担当者を 1 人で兼ねることもできますが，それぞれの担当者を選定することを推奨します。

> **重要**
>
> - データベースの管理担当者（DBA 権限を持っている HADB ユーザ）とは別に，監査人を選定してください。
> - 監査人は，監査に関する業務だけを行うことを推奨します。監査に関する業務で使用する HADB ユーザは，ほかの業務で使用する HADB ユーザとは兼ねないで，監査に関する業務だけで使用する HADB ユーザとして作成することを推奨します。

監査証跡機能を使用する場合のユーザ構成例を次の図に示します。

図 12-2　監査証跡機能を使用する場合のユーザ構成例



[説明]

- この例の場合，監査の担当部署に所属する監査証跡機能の管理担当者は，監査証跡機能の運用時に使用する OS ユーザと HADB ユーザ（ADBAUDITADMIN）を専用で持っています。

  監査の担当部署に所属する監査担当者は，監査時に使用する OS ユーザと HADB ユーザ（ADBAUDITVIEWER）を共用しています。
- 外部の監査担当者の操作は，監査証跡の参照だけに限定します。そのため，サーバマシンの OS アカウントは付与しないで，サーバマシン以外のマシンから監査証跡を参照するようにします。また，外部の監査担当者だけが使用する，監査参照権限を持っている HADB ユーザ（ADBOUTSIDEAUDITVIEWER）を作成します。

12.1.7　監査証跡ファイルへの書き込みに失敗した場合の処理方式の検討

次に示す原因で，監査証跡を監査証跡ファイルに書き込めない場合，HADB サーバを停止するかどうかを検討する必要があります。

- 監査証跡の出力先ディレクトリを作成したディスクが満杯になったとき
- 監査証跡の出力先ディレクトリを作成したディスクで障害が発生したとき

監査証跡ファイルへの書き込みに失敗した場合の処理方式として，次のどちらかを選択してください。選択した内容を，adbaudittrail --start コマンドの--write-error オプションに指定してください。

■HADB サーバの停止

監査証跡ファイルに監査証跡を書き込めない場合，HADB サーバを停止します。このときの HADB サーバの終了モードは，異常終了となります。

HADB サーバの動作継続よりも，HADB ユーザが行った操作を監査証跡として残すことを優先したい場合は，こちらを選択してください。監査証跡の取得漏れを最小限にできます。

HADB サーバを停止する場合は，--write-error オプションにDOWN を指定してください。または，--write-error オプションを省略してください。

■HADB サーバの動作継続

監査証跡ファイルに監査証跡を書き込めない場合でも，HADB サーバを停止しません。監査証跡ファイルに書き込めなかった監査証跡は破棄されますが，HADB サーバは動作を継続します。

HADB ユーザが行った操作を監査証跡として残すことよりも，HADB サーバの動作継続を優先したい場合は，こちらを選択してください。

HADB サーバの動作を継続する場合は，--write-error オプションにFAILSOFT を指定してください。なお，FAILSOFT を指定した場合，監査証跡ファイルに監査証跡を書き込めない原因の対処が完了すると，次回の監査証跡の書き込みタイミングから，監査証跡ファイルへの出力が自動的に再開されます。

監査証跡ファイルへの出力が再開されたことを確認したい場合，次のどちらかの方法で確認してください。

- 次回の監査証跡の書き込みタイミングまで待ってから，adbaudittrail -d コマンドを実行する
- adbaudittrail --swap コマンドで監査証跡の書き込みを発生させてから，adbaudittrail -d コマンドを実行する

adbaudittrail -d コマンドを実行したら，audit の出力内容にACTIVE が表示されていることを確認してください。ACTIVE の場合，監査証跡ファイルへの出力が再開されています。

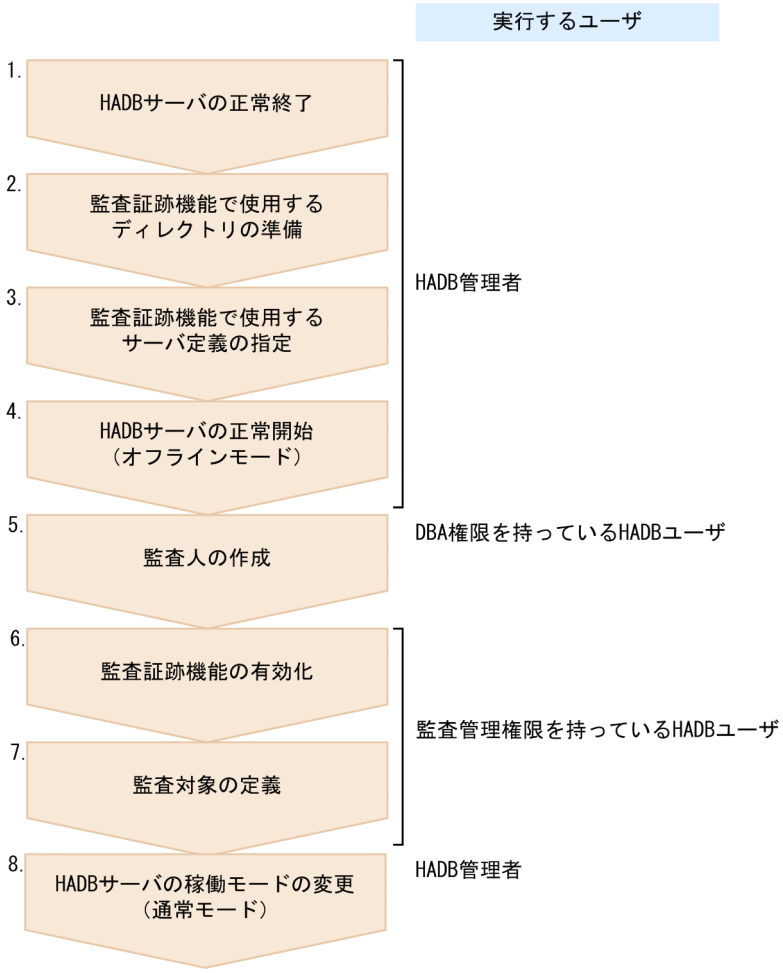
メモ

adbaudittrail コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbaudittrail（監査証跡機能の管理)』を参照してください。

12.2　監査証跡機能の環境設定

ここでは，監査証跡機能の環境設定を行う方法について説明します。次の図に示す手順に従って，監査証跡機能の環境設定を行います。

図 12-3　監査証跡機能の環境設定手順



上記の各手順を次項以降で順次説明します。

12.2.1　HADB サーバの正常終了

監査証跡機能の環境設定は，HADB サーバが正常終了している状態で実施する必要があります。

HADB サーバを正常終了するために，HADB 管理者はadbstop コマンドを実行してください。

すでに HADB サーバが終了している場合は，次に示す手順で HADB サーバが正常終了していることを確認してください。

1. adbls -d srv コマンドを実行する
   出力項目STATUS に出力される HADB サーバの状態が，"STOP"であることを確認してください。

"STOP"である場合は，手順 2.または手順 3.のどちらかに進んでください。

2. サーバメッセージログファイルを確認する

   サーバメッセージログファイル（$ADBDIR/spool/adbmessage*XX*.log※）を開き，KFAA91154-I メッセージに出力されている終了モードが，"normally"であることを確認してください。

   注※

   XX は通番であり，01～04 のどれかになります。

3. syslog を確認する

   syslog を開き，KFAA91154-I メッセージに出力されている終了モードが，"normally"であることを確認してください。

手順 1.の確認結果が"STOP"であり，かつ手順 2.または手順 3.の確認結果が"normally"である場合は，HADB サーバは正常終了しています。

もし，HADB サーバが正常終了していない場合は，adbstart コマンドで HADB サーバを開始したあとで，adbstop コマンドで正常終了してください。

12.2.2　監査証跡機能で使用するディレクトリの準備

監査証跡機能で使用するディレクトリを準備します。

HADB 管理者は，「12.1.2　監査証跡機能で使用するディスクの設計」～「12.1.5　監査証跡機能で使用するディレクトリの容量見積もり」の検討結果に従って，次に示す監査証跡機能で使用するディレクトリを準備してください。

- 監査証跡の出力先ディレクトリ
- 監査証跡の保存先ディレクトリ
- 監査証跡の長期保存先ディレクトリ

HADB 管理者は，監査証跡機能で使用するディレクトリを，次の手順で準備してください。各手順の詳細は，OS のマニュアルを参照してください。

手順

1. 監査証跡で使用するディレクトリを作成するディスクを用意する

   監査証跡で使用するディレクトリの容量を見積もってください。見積もり結果以上の容量があるディスクを用意してください。

2. ファイルシステムを構築する

   手順 1.で用意したディスク上に，ファイルシステムを構築してください。

3. マウント用のディレクトリを作成する

手順 2.で構築したファイルシステムをマウントするディレクトリを作成してください。そして，マウントしてください。

4. 監査証跡で使用するディレクトリを作成する

   監査証跡で使用するディレクトリを，各ディレクトリ用に構築したファイルシステム上に作成してください。

5. 監査証跡で使用するディレクトリにパーミッションを設定する

   監査証跡で使用するディレクトリには，HADB 管理者に対して書き込み権限，読み取り権限，および実行権限を設定してください。

12.2.3　監査証跡機能で使用するサーバ定義の指定

監査証跡機能で使用するサーバ定義の指定について説明します。

監査証跡機能を使用する場合，HADB 管理者は，次に示すオペランドをサーバ定義に指定する必要があります。サーバ定義の指定方法については，「8.5　サーバ定義の作成および変更方法」を参照してください。

■監査証跡機能に関するオペランド

各オペランドの詳細は，「7.2.9　監査証跡機能に関するオペランド（set 形式）」を参照してください。

- adb_audit_log_path オペランド

  監査証跡の出力先ディレクトリを絶対パス名で指定します。監査証跡機能を使用する場合，このオペランドは必ず指定してください。

- adb_audit_log_max_size オペランド

  通常，このオペランドの指定は不要です。監査証跡ファイルの最大容量を大きくしたい場合に，このオペランドを指定します。

- adb_audit_log_max_num オペランド

  通常，このオペランドの指定は不要です。作成される監査証跡ファイル数の上限を設定したい場合に，このオペランドを指定します。

12.2.4　HADB サーバの正常開始（オフラインモード）

HADB サーバをオフラインモードで開始します。HADB 管理者は，--offline オプションを指定して adbstart コマンドを実行してください。

HADB サーバをオフラインモードで開始することで，監査証跡機能の環境設定を実施している間に，不正にアクセスされることを防げます。

オフラインモードの詳細については，「10.2.3　HADB サーバの稼働モード」を参照してください。

12.2.5　監査人の作成

DBA 権限を持っている HADB ユーザが，監査人（監査権限を持っている HADB ユーザ）を作成します。監査人の作成手順を次に示します。

例題

次の監査人を作成します。

- 監査管理権限を持つ HADB ユーザ（認可識別子がADBAUDITADMIN）
- 監査参照権限を持つ HADB ユーザ（認可識別子がADBAUDITOR）

手順

1. HADB ユーザを追加する

```
CREATE USER ”ADBAUDITADMIN” IDENTIFIED BY ’#HelloHADB_AUD01’
CREATE USER ”ADBAUDITOR” IDENTIFIED BY ’#HelloHADB_AUD02’
```

認可識別子がADBAUDITADMIN と，認可識別子がADBAUDITOR の HADB ユーザを追加します。

2. 追加した HADB ユーザに，CONNECT 権限と監査権限を付与する

```
GRANT CONNECT,AUDIT ADMIN TO ”ADBAUDITADMIN”
GRANT CONNECT,AUDIT VIEWER TO ”ADBAUDITOR”
```

ADBAUDITADMIN にはCONNECT 権限と監査管理権限を，ADBAUDITOR にはCONNECT 権限と監査参照権限を付与します。

以上の操作で監査人が作成されます。

重要

- 監査証跡機能の管理担当者と監査担当者は，作成された監査人の初期パスワード（#HelloHADB_AUD01，#HelloHADB_AUD02）をすぐに変更してください。パスワードの変更方法については，「11.6.2　HADB ユーザのパスワードを変更する方法」を参照してください。

  監査人を作成したデータベースの管理担当者は，監査人の初期パスワードを知っています。そのため，初期パスワードを変更しないと，データベースの管理担当者が，作成した監査人を使用して監査の不正などを行うおそれがあります。
- 監査証跡機能を有効にするには，監査管理権限を持っている HADB ユーザが 1 人以上必要になります。

既存の HADB ユーザを監査人にする場合は，上記の手順の 2.だけを実施してください。ただし，DBA 権限を持っている HADB ユーザには，監査管理権限を付与できません。

12.2.6　監査証跡機能の有効化

監査証跡機能を有効にする方法について説明します。

監査証跡機能を有効にする場合，adbaudittrail コマンドを実行する必要があります。監査管理権限を持っている HADB ユーザが，--start オプションを指定してadbaudittrail コマンドを実行してください。

また，監査証跡機能を有効にする際，--write-error オプションによって，監査証跡ファイルに監査証跡を書き込めなくなったときの処理方式を指定できます。「12.1.7　監査証跡ファイルへの書き込みに失敗した場合の処理方式の検討」で検討した結果に従って，--write-error オプションを指定してください。

■--write-error オプションにDOWN を指定する場合

--write-error オプションにDOWN を指定すると，監査証跡ファイルに監査証跡を書き込めない場合，HADB サーバを停止します。

指定例を次に示します。

```
adbaudittrail -u ADBAUDITADMIN
              -p '#HelloHADB_ADMIN'
              --start --write-error DOWN
```

■--write-error オプションにFAILSOFT を指定する場合

--write-error オプションにFAILSOFT を指定すると，監査証跡ファイルに監査証跡を書き込めない場合でも，HADB サーバを停止しません。監査証跡ファイルに書き込めなかった監査証跡は破棄されますが，HADB サーバは動作を継続します。

指定例を次に示します。

```
adbaudittrail -u ADBAUDITADMIN
              -p '#HelloHADB_ADMIN'
              --start --write-error FAILSOFT
```

メモ

adbaudittrail コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbaudittrail（監査証跡機能の管理)』を参照してください。

12.2.7　監査対象の定義

監査対象を定義する場合は，CREATE AUDIT 文を実行します。

重要

- 監査管理権限を持っている HADB ユーザだけが，監査対象を定義できます。
- 監査証跡機能が有効なときだけ，監査対象を定義できます。

監査対象を定義する例を次に示します。

例題

選択監査イベントでも監査証跡を出力するために，監査対象を定義します。

```
CREATE AUDIT AUDITTYPE EVENT FOR ANY OPERATION
```

なお，定義済みの監査対象定義と同一の監査対象を定義することはできません。

12.2.8　HADB サーバの稼働モードの変更（通常モード）

監査証跡機能の環境設定が完了したら，HADB サーバの稼働モードを通常モードに変更します。

HADB 管理者は，--normal オプションを指定してadbchgsrvmode コマンドを実行してください。HADB サーバの稼働モードが，オフラインモードから通常モードに変更されます。

メモ

adbchgsrvmode コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbchgsrvmode（HADB サーバの稼働モードの変更)』を参照してください。

12.3　監査証跡機能の定期運用

ここでは，監査証跡機能で定期的に行う運用項目について説明します。

■監査証跡ファイルの移動

次に示す方法で，監査証跡ファイルを保存用のディレクトリに移動させます。

- 「12.3.1　監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）」
- 「12.3.2　監査証跡ファイルの移動（監査証跡の長期保存先ディレクトリへの移動）」

■監査証跡ファイルの参照

次に示す方法で，監査証跡ファイルに出力された監査証跡を確認します。

- 「12.3.3　監査証跡の参照（SELECT 文で監査証跡を参照する場合）」
- 「12.3.4　監査証跡の参照（CSV 形式に変換した監査証跡を参照する場合）」

12.3.1　監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）

監査証跡ファイルを，**監査証跡の出力先ディレクトリ**から**監査証跡の保存先ディレクトリ**に移動させることについて説明します。

HADB 管理者は，定期的に監査証跡ファイルを，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリに移動させてください。定期的に監査証跡ファイルを移動させることで，次のような事象が発生することを防いでください。

- 監査証跡の出力先ディレクトリを作成したディスクが満杯になること

  サーバ定義のadb_audit_log_max_num オペランドに 0 を指定している場合，上限なく監査証跡ファイルが作成されます。そのため，監査証跡ファイルを移動させないと，最終的にはディスクが満杯になります。

- 古い監査証跡ファイルが削除されること

  サーバ定義のadb_audit_log_max_num オペランドに 4 以上を指定している場合，作成される監査証跡ファイル数の上限が設定されます。監査証跡ファイル数が上限を超えた場合，いちばん古い監査証跡ファイルが削除されます。そのため，監査証跡ファイルを移動させないと，古い監査証跡ファイルが削除されてしまいます。

監査証跡ファイルを移動させるタイミングには，次の 2 つがあります。監査証跡の出力先ディレクトリを作成したディスクが満杯になる前，かつ古い監査証跡ファイルが削除される前に，監査証跡ファイルを移動させるようにしてください。

■監査証跡ファイルを移動させるタイミング

- 事前に決めたタイミングで移動させる

  事前に決めたタイミングで，監査証跡ファイルを移動させます。例えば，1 日 1 回のタイミングで，監査証跡ファイルを移動させます。

- 監査証跡ファイルが切り替わったときに移動させる

  監査証跡ファイルが切り替わった直後に，監査証跡ファイルを移動させます。監査証跡ファイルの切り替え時に出力されるKFAA81401-I メッセージ，およびKFAA81402-I メッセージを監視してください。

> **重要**
>
> サーバ定義のadb_audit_log_max_num オペランドに 4 以上を指定している場合，事前に決めたタイミングで監査証跡ファイルを移動させる前に，古い監査証跡ファイルが削除されるおそれがあります。そのため，メッセージを監視して，監査証跡ファイルが切り替わった直後に，監査証跡ファイルを移動させるようにしてください。

監査証跡ファイルを移動させる場合，次の手順で実施してください。

■監査証跡ファイルを移動させる手順

1. 監査証跡ファイルを，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリにコピーする
   ただし，現用の監査証跡ファイルはコピーしないでください。
2. 監査証跡ファイルのコピーが完了したら，コピー元の監査証跡ファイルを削除する
   ただし，次のファイルは削除しないでください。
   - 現用の監査証跡ファイル
   - ADB_AUDITREAD 関数で読み込み対象になっている監査証跡ファイル

> **重要**
>
> 現用の監査証跡ファイルを削除した場合，現用の監査証跡ファイルに出力された監査証跡が失われます。そのため，現用の監査証跡ファイルは削除しないように注意してください。

監査証跡ファイルを移動させる例を，次に示します。

■監査証跡ファイルを移動させる例

監査証跡ファイルを，監査証跡の出力先ディレクトリ（/mnt/auditttrail/outputarea/audit）から監査証跡の保存先ディレクトリ（/mnt/auditttrail/shorttimesavearea/audit_bak）に移動させます。

1. 移動が必要な監査証跡ファイルの一覧を作成する

   OS のfind コマンドで，監査証跡の出力先ディレクトリに存在する監査証跡ファイルを検索して，移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を作成します。なお，次のコマンドでは，現用の監査証跡ファイルは検索対象外となっています。

```
find /mnt/auditttrail/outputarea/audit ¥
    -name ”adbaud-????????-??????-???*.aud” ¥
    > /home/adbmanager/tmp/auditfilelist.txt
```

2. 移動が必要な監査証跡ファイルをコピーする

   移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）に存在する全ファイルを，OS のcp コマンドで，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリにコピーします。

```
while read filename ; do
  cp ${filename} /mnt/audittrail/shorttimesavearea/audit_bak
done < /home/adbmanager/tmp/auditfilelist.txt
```

3. コピーが完了した監査証跡ファイルを削除する

   監査証跡ファイルのコピーが完了したら，OS のrm コマンドで，コピー元の監査証跡ファイルを監査証跡の出力先ディレクトリから削除します。

```
while read filename ; do
  rm ${filename}
done < /home/adbmanager/tmp/auditfilelist.txt
```

4. 移動が必要な監査証跡ファイルの一覧を削除する

   OS のrm コマンドで，手順 1.で作成した移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を削除します。

```
rm /home/adbmanager/tmp/auditfilelist.txt
```

12.3.2 監査証跡ファイルの移動（監査証跡の長期保存先ディレクトリへの移動）

監査証跡ファイルを，**監査証跡の保存先ディレクトリ**から**監査証跡の長期保存先ディレクトリ**に移動させることについて説明します。

HADB 管理者は，監査期間が過ぎた監査証跡ファイルを，監査証跡の保存先ディレクトリから監査証跡の長期保存先ディレクトリに移動させてください。

監査証跡ファイルを移動させる場合，次の手順で実施してください。

■監査証跡ファイルを移動させる手順

1. 監査証跡ファイルを，監査証跡の保存先ディレクトリから監査証跡の長期保存先ディレクトリにコピーする
2. 監査証跡ファイルのコピーが完了したら，コピー元の監査証跡ファイルを削除する

なお，必要に応じて，監査証跡の長期保存先ディレクトリに移動させた監査証跡ファイルを，OS のgzip コマンドで圧縮してください。

監査証跡ファイルを移動させる例を，次に示します。

■監査証跡ファイルを移動させる例

監査証跡の保存先ディレクトリ（/mnt/audittrail/shorttimesavearea/audit_bak）に 1 年間格納した監査証跡ファイルを，監査証跡の長期保存先ディレクトリ（/mnt/audittrail/longtimesavearea/audit_bak）に移動させます。

1. 移動が必要な監査証跡ファイルの一覧を作成する

   OS のfind コマンドで，監査証跡の保存先ディレクトリに 1 年間格納した監査証跡ファイルを検索して，移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を作成します。

```
find /mnt/audittrail/shorttimesavearea/audit_bak ¥
    -name ”adbaud-????????-??????-???*.aud” -and -mtime 365 ¥
    > /home/adbmanager/tmp/auditfilelist.txt
```

2. 移動が必要な監査証跡ファイルをコピーする

   移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）に存在する全ファイルを，OS のcp コマンドで，監査証跡の保存先ディレクトリから監査証跡の長期保存先ディレクトリにコピーします。

```
while read filename ; do
  cp ${filename} /mnt/audittrail/longtimesavearea/audit_bak
done < /home/adbmanager/tmp/auditfilelist.txt
```

3. コピーが完了した監査証跡ファイルを削除する

   監査証跡ファイルのコピーが完了したら，OS のrm コマンドで，コピー元の監査証跡ファイルを監査証跡の保存先ディレクトリから削除します。

```
while read filename ; do
  rm ${filename}
done < /home/adbmanager/tmp/auditfilelist.txt
```

4. 移動させた監査証跡ファイルを圧縮する

   OS のgzip コマンドで，監査証跡の長期保存先ディレクトリに移動させた監査証跡ファイルを圧縮します。

```
gzip /mnt/audittrail/longtimesavearea/audit_bak/*.aud
```

5. 移動が必要な監査証跡ファイルの一覧を削除する

   OS のrm コマンドで，手順 1.で作成した移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を削除します。

```
rm /home/adbmanager/tmp/auditfilelist.txt
```

12.3.3 監査証跡の参照（SELECT 文で監査証跡を参照する場合）

監査証跡を参照するには，ADB_AUDITREAD 関数を指定したSELECT 文を実行します。

重要

監査参照権限を持っているHADBユーザだけが，ADB_AUDITREAD関数を指定したSELECT文を実行して，監査証跡を参照できます。

監査証跡を参照するSELECT文の例を次に示します。

例題1

監査証跡の保存先ディレクトリ（/mnt/audittrail/savearea）下の全監査証跡ファイル中の監査証跡を参照します。

SELECT文の実行例

```
SELECT * FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/savearea/*.aud'])) "DT"
```

[説明]

下線部分に，参照する監査証跡が格納されている監査証跡ファイルのパス名を指定します。この例の場合，監査証跡ファイル名に特殊文字の*を指定（*.aud）しているため，/mnt/audittrail/savearea ディレクトリ下の全監査証跡ファイル（ファイル拡張子がaud のファイル）に格納されている監査証跡を参照しています。

例題2

2017年4月1日～2017年4月30日の間に，HADBサーバにアクセスしたHADBユーザの一覧を参照します。監査証跡ファイルは，監査証跡の保存先ディレクトリ（/mnt/audittrail/savearea）下に保存されています。

SELECT文の実行例

```
SELECT DISTINCT "USER_NAME"
    FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/savearea/*.aud'])) "DT"
       WHERE "EXEC_TIME" BETWEEN TIMESTAMP'2017/04/01 00:00:00.000000'
                             AND TIMESTAMP'2017/04/30 23:59:59.999999'
```

[説明]

USER_NAME列には，HADBユーザの認可識別子が格納されています。EXEC_TIME列には，HADBユーザがHADBサーバに対する操作を実行した日時（イベントが完了した日時）が格納されています。USER_NAME列およびEXEC_TIME列は，ADB_AUDITREAD関数によって作成される表関数導出表の列名です。表関数導出表の列名については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

監査証跡を参照する際のSELECT文の例については，「12.7.5　定期監査の実施」も参照してください。

ADB_AUDITREAD関数の指定規則については，マニュアル『HADB　SQLリファレンス』の『ADB_AUDITREAD関数』を参照してください。

■留意事項

- ADB_AUDITREAD 関数の入力情報にできる監査証跡ファイルは，HADB サーバが出力した監査証跡ファイルか，その監査証跡ファイルを OS のgzip コマンドで圧縮したファイルです。また，複数の HADB サーバを運用している場合に，ほかの HADB サーバが出力した監査証跡ファイルもADB_AUDITREAD 関数の入力情報にできます。
- ADB_AUDITREAD 関数には，監査証跡ファイルをgzip コマンドで圧縮したファイルを指定できます。tar コマンドを使用してアーカイブされたgzip ファイルは指定できません。
- ADB_AUDITREAD 関数中に指定する監査証跡ファイルのパスに含まれる各ディレクトリに対して，HADB 管理者がアクセスできるように読み取り権限と実行権限を設定しておいてください。例えば，監査証跡ファイルが/mnt/auditrail/savearea ディレクトリ下に格納されている場合，/，/mnt，/mnt/auditrail，および/mnt/auditrail/savearea の各ディレクトリに対して，HADB 管理者がアクセスできるように読み取り権限と実行権限を設定しておく必要があります。また，/mnt/auditrail/savearea ディレクトリ下の監査証跡ファイルに対して，HADB 管理者が読み込めるように読み取り権限を設定しておいてください。正しく権限を設定していない場合，ADB_AUDITREAD 関数を指定したSELECT 文がエラーになります。
- 監査証跡の出力先ディレクトリ下の監査証跡ファイル中の監査証跡は参照しない運用を推奨します。監査証跡の参照中に，参照中の監査証跡が格納されている監査証跡ファイルが削除された場合，SQL 文の実行結果が想定したものにならないことがあります。監査証跡ファイルは，次の契機に削除されます。
  - 監査証跡の出力先ディレクトリ下の監査証跡ファイルを，監査証跡の保存先ディレクトリまたは監査証跡の長期保存先ディレクトリ下に移動させたとき
  - 次の条件をすべて満たす場合に，現用の監査証跡ファイルの切り替えが発生したとき
    - 監査証跡ファイルの世代数の上限を指定している場合
    - 監査証跡の出力先ディレクトリ下の監査証跡ファイルの数が世代数の上限に達している場合

  2 つ目の契機については，HADB サーバが監査証跡ファイルを自動的に削除するため，運用で回避できません。そのため，監査証跡の保存先ディレクトリまたは監査証跡の長期保存先ディレクトリ下の監査証跡ファイル中の監査証跡を参照することを推奨します。

  監査証跡の出力先ディレクトリ下の監査証跡ファイル中の監査証跡を参照したい場合は，その監査証跡ファイルを監査証跡の保存先ディレクトリに移動したあとに参照するようにしてください。
- 現用の監査証跡ファイル中の監査証跡は参照できません。現用の監査証跡ファイル中の監査証跡を参照したい場合は，現用の監査証跡ファイルを切り替えたあとに，その監査証跡ファイルを参照してください。現用の監査証跡ファイルの切り替え手順については，「12.4.2　現用の監査証跡ファイルの切り替え」を参照してください。

12.3.4　監査証跡の参照（CSV 形式に変換した監査証跡を参照する場合）

Microsoft Excel などの表計算ソフトで監査証跡を参照することができます。adbexport コマンドを実行して監査証跡を CSV 形式のデータに変換し，その CSV 形式のデータを表計算ソフトの入力情報にしてください。監査証跡を CSV 形式のデータに変換する手順を次に示します。

例題

監査証跡の保存先ディレクトリ（/mnt/audittrail/savearea）下の全監査証跡ファイル中の監査証跡を，CSV 形式のデータに変換します。

手順

1. adbexport コマンドの出力データパスファイルを作成する

   出力データパスファイルには，出力データファイルのパス名（監査証跡を CSV 形式で出力するファイルのパス名）を記述します。出力データパスファイルの記述例を次に示します。

```
/home/auditviewer/audit/audit1.csv      ←このファイルに監査証跡がCSV形式で出力されます。
```

   出力データパスファイルを，/home/auditviewer/export/expf01.txt に保存します。

2. adbexport コマンドの SQL 記述ファイルを作成する

   SQL 記述ファイルには，ADB_AUDITREAD 関数を指定したSELECT 文を記述します。SQL 記述ファイルの記述例を次に示します。

```
SELECT * FROM TABLE
    (ADB_AUDITREAD(MULTISET['/mnt/audittrail/savearea/*.aud'])) "T1"
```

   SQL 記述ファイルを，/home/auditviewer/export/sel01.txt に保存します。

   ヒント

   この例の場合，SELECT 文に探索条件を指定していないため，ADB_AUDITREAD 関数の読み込み対象となる監査証跡ファイル中の全監査証跡が検索対象（CSV 形式に変換される監査証跡）になります。監査証跡のうち，必要な情報だけを CSV 形式で出力したい場合は，次のどちらかの方法を実行してください。

   - SQL 記述ファイルに記述するSELECT 文に，探索条件を指定する
   - SQL 記述ファイルに，ビュー表を検索するSELECT 文を記述する

     この場合，ADB_AUDITREAD 関数を指定した表関数導出表を表参照に指定した問合せ指定に，探索条件を指定したビュー表を定義しておく必要があります。

3. adbexport コマンドを実行する

```
adbexport -u ADBAUDITOR -p '#HelloHADB_Aud01'
          -q /home/auditviewer/export/sel01.txt
          /home/auditviewer/export/expf01.txt
```

   -u と-p オプションには，監査参照権限を持っている HADB ユーザの認可識別子とパスワードを指定します。

   -q オプションには，手順 2.で作成した SQL 記述ファイルを指定します。

   /home/auditviewer/export/expf01.txt は，手順 1.で作成した出力データパスファイルのパス名です。

重要

監査証跡を CSV 形式のデータに変換する作業は，監査参照権限を持っている HADB ユーザだけが実行できます。

出力データパスファイル，および SQL 記述ファイルの指定規則の詳細については，マニュアル『HADB コマンドリファレンス』の『adbexport コマンドの指定形式』の『指定形式およびオプションの説明』を参照してください。

重要

- 監査証跡を CSV 形式のデータに変換してファイルに保存した場合，そのファイルの取り扱いに注意してください。そのファイルを参照できるのは，監査の実行者だけとなるようにしてください。また，ファイルが更新されないように対策をする必要があります。
- 監査証跡は，監査の実行者（監査参照権限を持っている HADB ユーザ）だけが参照することを推奨しています。そのため，基本的には「12.3.3　監査証跡の参照（SELECT 文で監査証跡を参照する場合）」で説明している方法で監査証跡を参照するようにしてください。

12.4　監査証跡機能の非定期運用

ここでは，監査証跡機能で非定期に行う運用項目について説明します。

- 「12.4.1　監査人の追加，削除，変更（監査権限の付与または取り消し）」
- 「12.4.2　現用の監査証跡ファイルの切り替え」
- 「12.4.3　監査対象定義の確認」
- 「12.4.4　監査対象定義の変更」
- 「12.4.5　監査証跡機能の動作状況の確認」

12.4.1　監査人の追加，削除，変更（監査権限の付与または取り消し）

監査人の追加，削除，変更方法（監査権限の付与または取り消し方法）について，例を使って説明します。

(1)　監査人の追加（監査権限の付与）

監査人の追加は，DBA 権限を持っている HADB ユーザが行います。監査人の追加手順を次に示します。

例題

監査管理権限を持つ監査人（認可識別子がADBAUDITADMIN01）と，監査参照権限を持つ監査人（認可識別子がADBAUDITOR01）を追加します。HADB ユーザを新規に作成し，その HADB ユーザに監査権限を付与します。

手順

1. HADB ユーザを追加する

```
CREATE USER "ADBAUDITADMIN01" IDENTIFIED BY '#HelloHADB_AUD01'
CREATE USER "ADBAUDITOR01" IDENTIFIED BY '#HelloHADB_AUD02'
```

認可識別子がADBAUDITADMIN01 と，認可識別子がADBAUDITOR01 の HADB ユーザを追加します。

2. 追加した HADB ユーザに，CONNECT 権限と監査権限を付与する

```
GRANT CONNECT,AUDIT ADMIN TO "ADBAUDITADMIN01"
GRANT CONNECT,AUDIT VIEWER TO "ADBAUDITOR01"
```

ADBAUDITADMIN01 にはCONNECT 権限と監査管理権限を，ADBAUDITOR01 にはCONNECT 権限と監査参照権限を付与します。

重要

追加された監査人は，すぐに初期パスワード（#HelloHADB_AUD01，#HelloHADB_AUD02）を変更してください。パスワードの変更方法については，「11.6.2　HADB ユーザのパスワードを変更する方法」を参照してください。

既存の HADB ユーザを監査人にする場合は，上記の手順 1.の操作は必要ありません。また，既存の HADB ユーザがCONNECT 権限を持っている場合は，上記の手順 2.で監査権限だけを付与します。

なお，DBA 権限を持っている HADB ユーザに監査管理権限を付与することはできません。

(2) 監査人の削除（監査権限の取り消し）

監査人の削除手順を次に示します。

例題 1

監査管理権限を持っている監査人（認可識別子はADBAUDITADMIN）を削除します。

手順

1. 監査人（ADBAUDITADMIN）が持っている監査管理権限を取り消す

```
REVOKE AUDIT ADMIN FROM "ADBAUDITADMIN"
```

この操作は，監査管理権限を持っている HADB ユーザが実行します。

2. 監査管理権限を取り消した HADB ユーザ（ADBAUDITADMIN）を削除する

```
DROP USER "ADBAUDITADMIN"
```

この操作は，DBA 権限を持っている HADB ユーザが実行します。

例題 2

監査参照権限を持っている監査人（認可識別子はADBAUDITOR）を削除します。

手順

1. 監査人（ADBAUDITOR）が持っている監査参照権限を取り消す

```
REVOKE AUDIT VIEWER FROM "ADBAUDITOR"
```

この操作は，監査管理権限を持っている HADB ユーザが実行します。

2. 監査参照権限を取り消した HADB ユーザ（ADBAUDITOR）を削除する

```
DROP USER "ADBAUDITOR"
```

この操作は，DBA 権限を持っている HADB ユーザが実行します。

(3) 監査人の変更

監査人の変更手順を次に示します。

例題

監査管理権限を持っている監査人を変更します。認可識別子がADBAUDITADMIN01 の HADB ユーザから，認可識別子がADBAUDITADMIN02 の HADB ユーザに監査人を変更します。

認可識別子がADBAUDITADMIN02 の HADB ユーザを新規に作成し，その HADB ユーザを新たな監査人にします。

手順

1. HADB ユーザ（ADBAUDITADMIN02）を追加する

```
CREATE USER "ADBAUDITADMIN02" IDENTIFIED BY '#HelloHADB_AUD02'
```

認可識別子がADBAUDITADMIN02 の HADB ユーザを追加します。

この操作は，DBA 権限を持っている HADB ユーザが実行します。

2. 追加した HADB ユーザ（ADBAUDITADMIN02）に，CONNECT 権限と監査管理権限を付与する

```
GRANT CONNECT,AUDIT ADMIN TO "ADBAUDITADMIN02"
```

この操作は，DBA 権限を持っている HADB ユーザが実行します。

3. 監査人（ADBAUDITADMIN01）が持っている監査管理権限を取り消す

```
REVOKE AUDIT ADMIN FROM "ADBAUDITADMIN01"
```

この操作は，監査管理権限を持っている HADB ユーザが実行します。

4. 監査管理権限を取り消した HADB ユーザ（ADBAUDITADMIN01）を削除する

```
DROP USER "ADBAUDITADMIN01"
```

この操作は，DBA 権限を持っている HADB ユーザが実行します。

この操作は，この HADB ユーザが不要な場合に限り実施してください。

重要

新たな監査人（ADBAUDITADMIN02）は，すぐに初期パスワード（#HelloHADB_AUD02）を変更してください。パスワードの変更方法については，「11.6.2　HADB ユーザのパスワードを変更する方法」を参照してください。

既存の HADB ユーザを新たな監査人にする場合は，上記の手順 1.の操作は必要ありません。また，既存の HADB ユーザがCONNECT 権限を持っている場合は，上記の手順 2.で監査管理権限だけを付与します。

なお，DBA 権限を持っている HADB ユーザに監査管理権限を付与することはできません。

(4)　留意事項

- 監査管理権限を持っている HADB ユーザは，自分が持っている監査権限，およびほかの HADB ユーザが持っている監査権限を取り消すことができます。
- 監査証跡機能が有効な場合，監査管理権限とCONNECT 権限の両方を持っている HADB ユーザが自分だけのときは，自分が持っている監査管理権限を取り消すことはできません。
- 監査証跡機能が無効な場合，監査権限を取り消すことはできません。ただし，次に示す条件をすべて満たす場合は，自分が持っている監査管理権限を取り消すことができます。
  - 監査管理権限を持っている HADB ユーザが自分だけである
  - 監査参照権限を持っている HADB ユーザがいない

12.4.2　現用の監査証跡ファイルの切り替え

現用の監査証跡ファイルの切り替えについて説明します。

現用の監査証跡ファイル中の監査証跡は，参照できません。そのため，現用の監査証跡ファイル中の監査証跡を参照したい場合は，監査証跡ファイルの切り替えを実施する必要があります。監査証跡ファイルの切り替えは，adbaudittrail コマンドで実施できます。

監査管理権限を持っている HADB ユーザが，--swap オプションを指定したadbaudittrail コマンドを実行してください。

> **メモ**
>
> adbaudittrail コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbaudittrail（監査証跡機能の管理)』を参照してください。

--swap オプションを指定してadbaudittrail コマンドを実行すると，監査証跡ファイルの切り替えが行われて，現用の監査証跡ファイルがリネームされます。リネームされた監査証跡ファイルに出力された監査証跡は，次に示す方法で参照してください。

- 「12.3.3　監査証跡の参照（SELECT 文で監査証跡を参照する場合)」
- 「12.3.4　監査証跡の参照（CSV 形式に変換した監査証跡を参照する場合)」

12.4.3　監査対象定義の確認

監査対象定義を確認する場合は，ディクショナリ表SQL_AUDITS を検索します。

> **重要**
>
> 監査管理権限を持っている HADB ユーザだけが，ディクショナリ表SQL_AUDITS を検索できます。

ディクショナリ表SQL_AUDITS の検索例を次に示します。

SQL 文の実行例

```
SELECT * FROM "MASTER"."SQL_AUDITS"
```

検索結果の例

```
OPERATION_TYPE   EVENT_TYPE  ...  AUDIT_TYPE  IS_EXCLUSION_AUDIT
--------------   ----------       ----------  ------------------
ANY              **********  ...  EVENT                       N
```

検索結果の行がある場合は，監査対象定義が存在します。この例の場合，監査対象定義が存在するため，選択監査イベントでも監査証跡が出力されます。

検索結果の行がない場合は，監査対象定義は存在しません。この場合，選択監査イベントでは監査証跡が出力されません。

なお，検索結果の例は，adbsql コマンドで SQL 文を実行したときの例です。**********は，ナル値を意味しています。

12.4.4　監査対象定義の変更

監査対象定義の変更方法（監査対象を定義する方法，または監査対象定義を削除する方法）について説明します。

重要

- 監査管理権限を持っている HADB ユーザだけが，監査対象定義を変更できます。
- 監査証跡機能が有効なときだけ，監査対象定義を変更できます。

■監査対象の定義

監査対象を定義する場合は，CREATE AUDIT 文を実行します。監査対象を定義する例を次に示します。

例題

選択監査イベントでも監査証跡を出力するために，監査対象を定義します。

```
CREATE AUDIT AUDITTYPE EVENT FOR ANY OPERATION
```

なお，定義済みの監査対象定義と同一の監査対象を定義することはできません。

■監査対象定義の削除

監査対象定義を削除する場合は，DROP AUDIT 文を実行します。監査対象定義を削除する例を次に示します。

例題

選択監査イベントでの監査証跡の出力をやめるために，監査対象定義を削除します。

```
DROP AUDIT AUDITTYPE EVENT FOR ANY OPERATION
```

12.4.5　監査証跡機能の動作状況の確認

監査証跡機能の動作状況を確認する方法について説明します。

監査証跡機能の動作状況を確認したい場合，adbaudittrail コマンドを実行する必要があります。

次のどちらかに該当する HADB ユーザが，-d オプションを指定したadbaudittrail コマンドを実行してください。

- 監査管理権限を持っている HADB ユーザ

- 監査参照権限を持っているHADBユーザ

> **メモ**
>
> adbauditトrailコマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbauditトrail（監査証跡機能の管理)』を参照してください。

-dオプションを指定してadbauditトrailコマンドを実行すると，監査証跡機能の動作状況が表示されます。

■監査証跡機能が有効かどうかを確認したい場合

監査証跡機能が有効かどうかを確認したい場合は，auditの出力内容を確認してください。

auditの出力内容がACTIVEまたはACTIVE(NO OUTPUT)の場合，監査証跡機能は有効です。なお，ACTIVE(NO OUTPUT)の場合は，監査証跡ファイルへの監査証跡の出力が抑止されている状態です。

auditの出力内容がINACTIVEの場合，監査証跡機能は無効です。

■監査証跡ファイルに監査証跡を書き込めなくなったときの処理方式を確認したい場合

ディスクの満杯やディスク障害などの原因によって，監査証跡ファイルに監査証跡を書き込めなくなったときの処理方式を確認したい場合は，write-errorの出力内容を確認してください。

write-errorの出力内容がDOWNの場合，監査証跡ファイルへの書き込みに失敗すると，HADBサーバを異常終了する設定になっています。

write-errorの出力内容がFAILSOFTの場合，監査証跡ファイルへの書き込みに失敗しても，HADBサーバは動作を継続する設定になっています。

■監査証跡機能に関するサーバ定義の指定値を確認したい場合

監査証跡機能に関するサーバ定義の指定値を確認したい場合は，次の項目に出力された内容を確認してください。

- audit-directory-pathの出力内容には，サーバ定義のadb_audit_log_pathオペランドに指定したディレクトリ名が表示されます。監査証跡の出力先ディレクトリを確認できます。
- audit-file-max-sizeの出力内容には，サーバ定義のadb_audit_log_max_sizeオペランドに指定した値が表示されます。監査証跡ファイルの最大容量を確認できます。
- audit-file-numberの出力内容には，サーバ定義のadb_audit_log_max_numオペランドに指定した値が表示されます。監査証跡ファイルの最大世代数を確認できます。

12.5　監査証跡機能のトラブルシュート

ここでは，監査証跡機能に関するトラブルがあった場合の対処方法について説明します。

監査証跡の出力時にエラーが発生した場合，KFAA51404-E メッセージが出力されます。KFAA51404-E メッセージの変数 *aa....aa* の内容によって，対処方法が異なります。

- 変数 *aa....aa* がThe audit trail file disk is full の場合
  「12.5.1　監査証跡の出力先ディレクトリを作成したディスクが満杯になった場合の対処方法」を参照してください。
- 変数 *aa....aa* がerrno *cc....cc* の場合
  「12.5.2　監査証跡の出力先ディレクトリを作成したディスクで障害が発生した場合の対処方法」を参照してください。
- 変数 *aa....aa* がThe audit trail file could not be opened, because file descriptors were insufficient の場合
  「12.5.3　ファイルディスクリプタが不足した場合の対処方法」を参照してください。

KFAA51404-E メッセージについては，マニュアル『HADB　メッセージ』を参照してください。

12.5.1　監査証跡の出力先ディレクトリを作成したディスクが満杯になった場合の対処方法

監査証跡の出力先ディレクトリを作成したディスクが満杯になった場合の対処方法について説明します。

監査証跡の出力先ディレクトリを作成したディスクが満杯になった場合，KFAA51404-E メッセージの変数 *aa....aa* にThe audit trail file disk is full が表示されます。HADB サーバが稼働しているかどうかによって，対処方法が異なります。

■HADB サーバが異常終了している場合

次に示す 3 つの対処方法から 1 つを選択して，ディスクの空き容量を確保してください。そのあとで，adbstart コマンドで HADB サーバを開始してください。

■HADB サーバが稼働している場合

次に示す 3 つの対処方法のうち，1.または 2.を選択して，ディスクの空き容量を確保してください。HADB サーバが稼働している状態で対処できます。

なお，3.を実施したい場合は，adbstop コマンドで HADB サーバを終了する必要があります。HADB サーバの終了後に，3.を実施してディスクの空き容量を確保してください。そのあとで，adbstart コマンドで HADB サーバを開始してください。

対処方法

1. 監査証跡の出力先ディレクトリに存在する現用以外の監査証跡ファイルを，監査証跡の保存先ディレクトリに移動させる
2. 監査証跡の出力先ディレクトリを作成したディスクに存在する，不要なファイルを削除する
3. 監査証跡の出力先ディレクトリを作成したディスクの容量を大きくして，ディスクの空き容量を増やす

12.5.2 監査証跡の出力先ディレクトリを作成したディスクで障害が発生した場合の対処方法

監査証跡の出力先ディレクトリを作成したディスクで障害が発生した場合の対処方法について説明します。

監査証跡の出力先ディレクトリを作成したディスクで障害が発生した場合，KFAA51404-E メッセージの変数 *aa....aa* にerrno *cc....cc* が表示されます。対処方法を次に示します。

対処方法

1. HADB サーバを終了する

   HADB サーバが稼働している場合は，adbstop コマンドで HADB サーバを終了してください。
2. エラーの原因を取り除く

   errno *cc....cc* に表示されたエラー番号を OS のマニュアルで調べて，エラーの原因を取り除いてください。

   エラーの原因を取り除くことができない場合は，監査証跡の出力先ディレクトリを新規に作成してください。そして，新規に作成した監査証跡の出力先ディレクトリのパス名を，サーバ定義の adb_audit_log_path オペランドに指定してください。
3. HADB サーバを開始する

   adbstart コマンドで HADB サーバを開始してください。

12.5.3 ファイルディスクリプタが不足した場合の対処方法

ファイルディスクリプタが不足した場合の対処方法について説明します。

ファイルディスクリプタが不足した場合，KFAA51404-E メッセージの変数*aa....aa* にThe audit trail file could not be opened, because file descriptors were insufficient が表示されます。対処方法を次に示します。

対処方法

1. HADB サーバプロセス以外のプロセスでクローズできるファイルがある場合は，そのファイルをクローズする

   ファイルをクローズしても解決しない場合は，次の手順に進んでください。

2. HADB サーバを終了する

   adbstop コマンドで，HADB サーバを終了してください。

3. カーネルパラメタの指定値を変更して，ファイルディスクリプタ数の上限を増やす

   カーネルパラメタのnofile およびfile-max の指定値を大きくしてください。カーネルパラメタについては，「6.2　カーネルパラメタの見積もり」を参照してください。

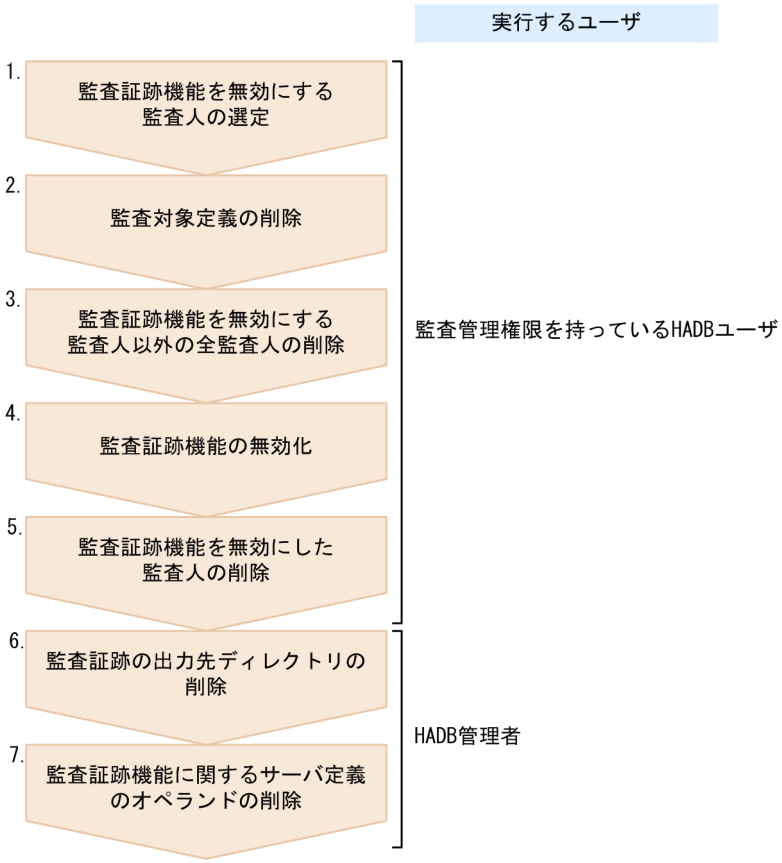
4. HADB サーバを開始する

   adbstart コマンドで，HADB サーバを開始してください。

12.6　監査証跡機能の使用をやめる場合

監査証跡機能の使用をやめるときの操作手順を次の図に示します。

図 12-4　監査証跡機能の使用をやめるときの操作手順



手順

1. 監査証跡機能を無効にする監査人を選定する

   監査証跡機能を無効にする監査人（監査管理権限を持っている HADB ユーザ）を 1 人決めてください。選定した監査人が，手順 2.～5.を実施します。

2. 監査対象定義を削除する

   手順 1.で選定した監査人が，定義されている監査対象定義を，DROP AUDIT 文ですべて削除してください。監査対象定義を削除する方法は，「12.4.4　監査対象定義の変更」を参照してください。

3. 監査証跡機能を無効にする監査人以外の監査人をすべて削除する

   監査証跡機能を無効にする前に，手順 1.で選定した監査人以外の監査人をすべて削除する必要があります。次の手順で削除してください。

   - まず，手順 1.で選定した監査人が，すべての監査人の認可識別子を調べます。監査人の認可識別子を調べる場合は，「付録 B.22　ディクショナリ表の検索」の「(36)　監査人の認可識別子と監査人が持っている監査権限を調べる方法」を参照してください。

- 次に，手順 1.で選定した監査人が，調べた監査人の認可識別子を基に，自分を含むすべての監査人が持っている監査参照権限をREVOKE 文で取り消します。監査参照権限を持っている HADB ユーザが存在しないようにしてください。監査参照権限を取り消す際の SQL 文の例については，「12.4.1 監査人の追加，削除，変更（監査権限の付与または取り消し）」の「(2)　監査人の削除（監査権限の取り消し）」を参照してください。
- 最後に，手順 1.で選定した監査人が，調べた監査人の認可識別子を基に，手順 1.で選定した監査人以外の監査人が持っている監査管理権限をREVOKE 文で取り消します。監査管理権限を持っている HADB ユーザが，手順 1.で選定した監査人だけになるようにしてください。監査管理権限を取り消す際の SQL 文の例については，「12.4.1　監査人の追加，削除，変更（監査権限の付与または取り消し）」の「(2)　監査人の削除（監査権限の取り消し）」を参照してください。

4. 監査証跡機能を無効にする

   手順 1.で選定した監査人が，adbauditrail コマンドで監査証跡機能を無効にします。--stop オプションを指定してadbauditrail コマンドを実行してください。監査証跡機能を無効にすると，監査証跡が出力されなくなります。

   コマンドの実行例

```
adbauditrail -u ADBAUDITADMIN
             -p '#HelloHADB_ADMIN'
             --stop
```

> **メモ**
>
> adbauditrail コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbauditrail（監査証跡機能の管理）』を参照してください。

5. 監査証跡機能を無効にした監査人を削除する

   手順 1.で選定した監査人は，REVOKE 文で監査人自身の監査管理権限を取り消してください。監査管理権限の取り消しについては，「12.4.1　監査人の追加，削除，変更（監査権限の付与または取り消し）」の「(2)　監査人の削除（監査権限の取り消し）」を参照してください。

6. 監査証跡の出力先ディレクトリを削除する

   監査証跡機能を無効にすると，監査証跡は出力されなくなります。監査証跡の出力先ディレクトリは不要になるため，必要に応じて削除してください。

   - まず，HADB 管理者が，監査証跡の出力先ディレクトリに存在する監査証跡ファイルを，監査証跡の保存先ディレクトリに移動します。監査証跡ファイルを移動する方法については，「12.3.1　監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）」を参照してください。
   - 次に，HADB 管理者が，監査証跡の出力先ディレクトリを削除します。

7. 監査証跡機能に関するサーバ定義のオペランドを削除する

   監査証跡機能の使用をやめる場合，サーバ定義に指定した次のオペランドは不要になります。HADB 管理者は，必要に応じて次のオペランドを削除してください。

   - adb_audit_log_path オペランド
   - adb_audit_log_max_size オペランド

- adb_audit_log_max_num オペランド

サーバ定義を変更する方法については，「8.5.2　サーバ定義の変更方法」を参照してください。

12.7　監査証跡機能の運用例

実際の業務を想定した監査証跡機能の運用例について説明します。

12.7.1　A 社の情報分析システムの構成

EC サイトを運営する A 社を例にして，監査証跡機能の運用例を説明します。

監査証跡機能の使用を決定した背景

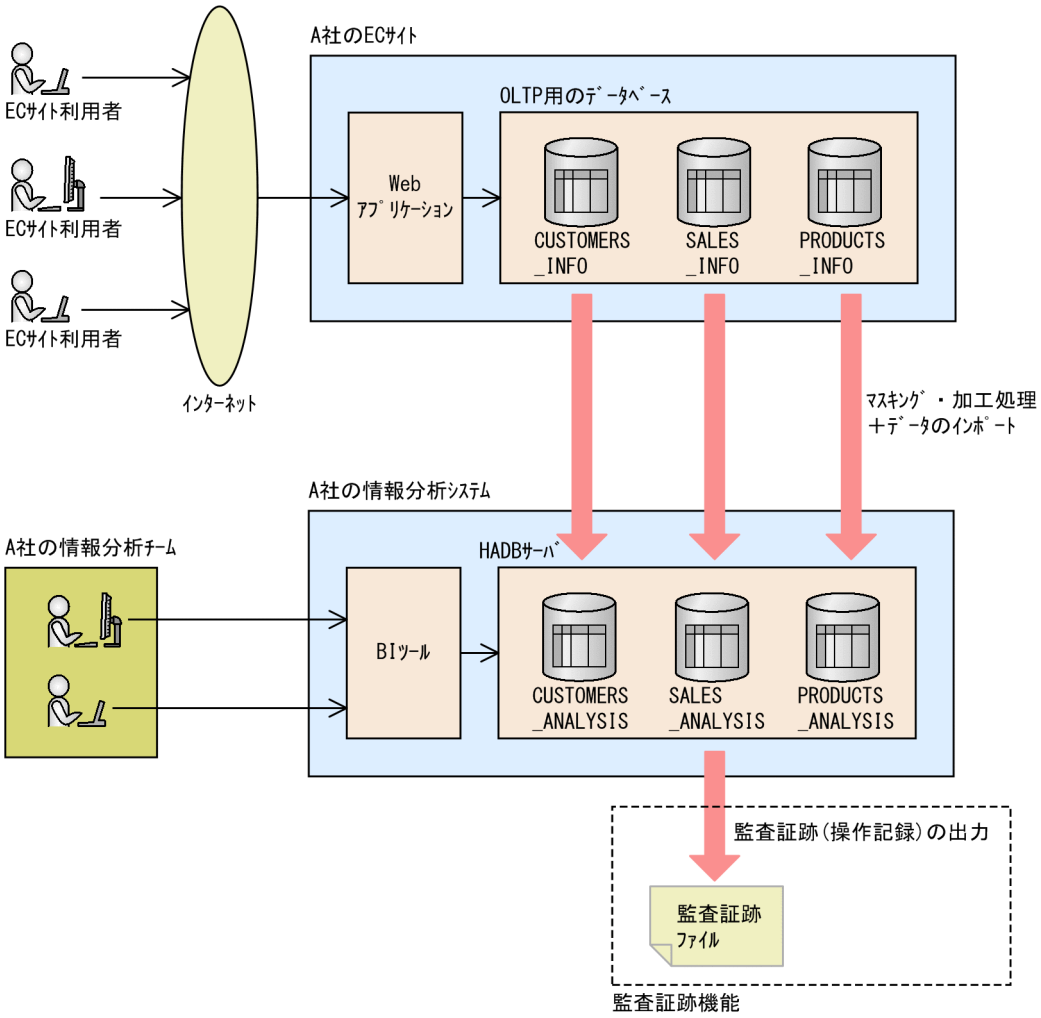
A 社は，インターネット上で商品やサービスを販売する EC サイトを運営しています。A 社では，顧客の個人情報，販売履歴，在庫状況などの情報を HADB のデータベースで管理しています。これらの情報（データ）は，人気商品や顧客の好みをリアルタイムに分析するために，30 分ごとにデータベースにインポートされています。データベースにインポートされるデータには，個人情報が含まれているため，A 社は，HADB の監査証跡機能を使用して，データベースの監査をできるようにしました。

A 社の情報分析システムの稼働状況

- EC サイトにログインした顧客の情報や，EC サイトの販売情報および在庫情報を，30 分ごとにデータベースにインポートする
- BI ツールを使用して情報分析をする
- HADB サーバで処理するクエリは，1 日に 2,000～2,400 回程度発生する
- 1SQL 文の長さは 16 キロバイト未満である

A 社の EC サイトと情報分析システムの構成を次の図に示します。

図 12-5　A 社の EC サイトと情報分析システムの構成



上記の条件とシステムの構成を前提として，監査証跡機能の運用例を以降で説明します。

12.7.2　設計

監査証跡機能を適切に使用するために，事前に設計しておくことを説明します。

(1)　監査の方針を決める

最初に，監査の対象や，監査の実施間隔などの監査方針を決めます。

A 社の情報分析システムでは，個人情報を扱っています。そのため，社員が個人情報に不正アクセスしていないかを監査で確認する必要があります。また，販売履歴や売り上げ情報なども社外秘の情報のため，社員がこれらの情報に不正アクセスしていないかを監査で確認する必要があります。

売り上げ情報や，訪問客数などは月に 1 回計上しているため，このタイミングで監査を実施することにします。

以上のことから，監査の対象とする操作と，監査の実施間隔を次のように決めます。

監査の対象とする操作

- 個人情報を格納しているデータベースへのアクセス操作
- 販売履歴や売り上げ情報などの社外秘の情報を格納しているデータベースへのアクセス操作

監査の実施間隔

- 月に 1 回実施

(2)　監査証跡を出力するイベントを決める

「(1)　監査の方針を決める」で決めた監査方針に従って，監査証跡を出力するイベントを決めます。

監査対象のデータを格納しているデータベースへのアクセス記録が，監査の際に必要になります。また，データベースにアクセスしたユーザが，どのような経路で HADB サーバに接続してきたかを特定するために，HADB サーバへの接続または切断処理が発生したときの記録も必要になります。

データベースへのアクセス記録を出力するには，**操作系 SQL イベント**を監査証跡の出力対象イベントとする必要があります。また，HADB サーバへの接続または切断処理に関する記録を出力するには，**セッションイベント**を監査証跡の出力対象イベントとする必要があります。操作系 SQL イベントおよびセッションイベントの発生時に，監査証跡を出力する場合は，`CREATE AUDIT` 文で**選択監査イベント**を監査対象として定義する必要があります。

> **メモ**
>
> イベントの種類，イベントの種類と操作の関係については，「12.9.1　監査対象イベントの一覧と出力項目」を参照してください。

(3)　監査証跡機能で使用するディレクトリを設計する

A 社の情報分析システムで扱うデータには，個人情報が含まれているため，監査証跡ファイルの保管規則を次のように決めました。

- 過去 3 か月分の監査証跡ファイルは，監査証跡の出力先ディレクトリ下に保管する
- 出力後 3 か月が経過した監査証跡ファイルは，監査証跡の保存先ディレクトリに移動する
- 出力後 1 年が経過するまで，監査証跡の保存先ディレクトリ下に監査証跡ファイルを保管する
- 監査証跡の保存先ディレクトリ下に保管している，出力後 1 年が経過した監査証跡ファイルは削除する

A 社の情報分析システムでは，データベースへのデータインポートが 30 分ごとに実行されます。また，BI ツールからの検索クエリが，1 日に約 2,400 回発生します。これらを考慮し，監査証跡機能で使用するディレクトリを設計します。

■監査証跡の出力先ディレクトリの設計

監査証跡の出力先ディレクトリには，3 か月分の監査証跡ファイルを格納できる容量が必要になります。監査証跡の出力先ディレクトリの容量は次のように見積もります。

計算式

```
｛2,400クエリ×｛2キロバイト＋16キロバイト(SQL文長)｝＋24×2×(1キロバイト＋1キロバイト＋1キロバイト)｝×31日×3か月
＝｛2,400×(2＋16)＋24×2×(1＋1＋1)｝×31×3＝4,030,992キロバイト
≒4ギガバイト
```

上記の計算結果から，監査証跡の出力先ディレクトリを作成するディスク領域は，余裕も含めて 10 ギガバイトにします。

このディスク領域を/mnt/audittrail/outputarea にマウントし，監査証跡の出力先ディレクトリのパスを次のようにします。

- /mnt/audittrail/outputarea/audit

■監査証跡の保存先ディレクトリの設計

監査証跡の保存先ディレクトリには，9 か月分の監査証跡ファイルを格納できる容量が必要になります。監査証跡の保存先ディレクトリの容量は次のように見積もります。

計算式

```
｛2,400クエリ×｛2キロバイト＋16キロバイト(SQL文長)｝＋24×2×（1キロバイト＋1キロバイト＋1キロバイト）｝×31日×9か月
＝｛2,400×(2＋16)＋24×2×(1＋1＋1)｝×31×9＝12,092,976キロバイト
≒12ギガバイト
```

上記の計算結果から，監査証跡の保存先ディレクトリを作成するディスク領域は，余裕も含めて 20 ギガバイトにします。

このディスク領域を/mnt/audittrail/savearea にマウントし，監査証跡の保存先ディレクトリのパスを次のようにします。

- /mnt/audittrail/savearea/audit_bak

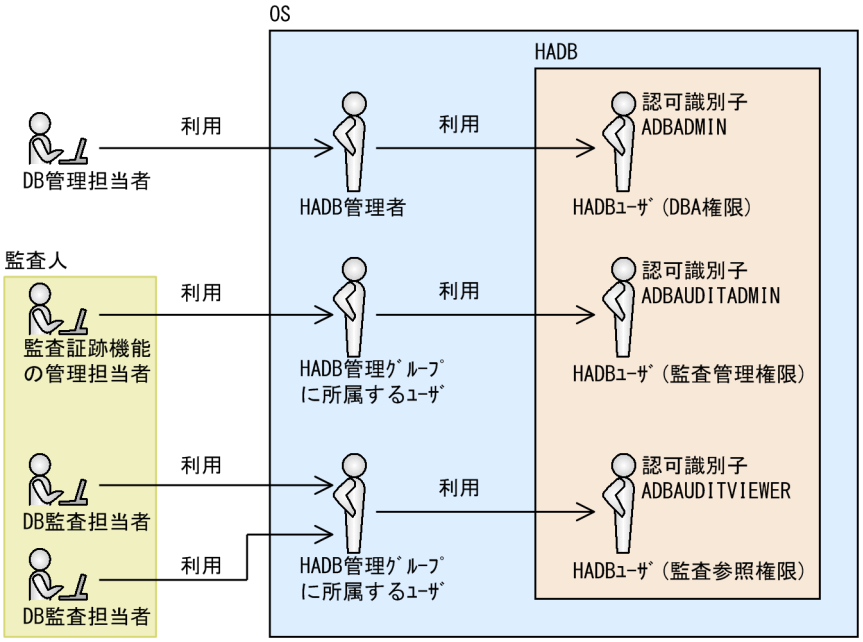
> **メモ**
>
> 出力後 3 か月が経過した監査証跡ファイルを監査証跡の保存先ディレクトリに移動したあとに，出力後 1 年が経過した監査証跡ファイルを削除します。そのため，監査証跡の保存先ディレクトリには，9 か月分の監査証跡ファイルを格納する容量が必要になります。

(4) 監査人を選定する

監査証跡機能を使用する場合，監査人を選定する必要があります。監査人とは，監査証跡機能の管理，運用を担当したり，監査を実施したりする人のことです。

データベースの運用担当者と，データベースの運用担当者の操作を確認する監査人は，別々の人物が担当する必要があります。A 社では，監査証跡機能の使用に伴い，情報分析システムの担当者の役割分担を次の図のように決めました。

図 12-6　A 社の情報分析システムの担当者の役割分担



- DB 管理担当者

  HADB のデータベースを管理，運用する人です。主にデータベースのメンテナンス作業などを担当します。OS にログインする際は，HADB 管理者の OS アカウントを使用します。また，HADB サーバに接続する際は，DBA 権限を持っている HADB ユーザの認可識別子ADBADMIN を使用します。

- 監査証跡機能の管理担当者

  監査証跡機能の管理，運用をする人です。監査証跡機能の有効または無効を管理したり，監査対象の定義をしたりします。OS にログインする際は，HADB 管理グループに所属する OS ユーザのアカウントを使用します。また，HADB サーバに接続する際は，監査管理権限を持っている HADB ユーザの認可識別子ADBAUDITADMIN を使用します。

- DB 監査担当者

  監査を担当する人です。監査証跡を参照して，データベースの操作記録などを確認します。OS にログインする際は，HADB 管理グループに所属する OS ユーザのアカウントを使用します。また，HADB サーバに接続する際は，監査参照権限を持っている HADB ユーザの認可識別子ADBAUDITVIEWER を使用します。

  DB 監査担当者は 2 名選定し，OS ユーザのアカウントや，HADB ユーザの認可識別子は，両者が使用します。

(5)　監査証跡ファイルへの書き込みに失敗した場合の処理方式を選択する

ディスクの満杯，またはディスク障害などによって，HADB サーバが監査証跡ファイルへの書き込みに失敗した場合の処理方式を選択する必要があります。次のどちらの処理方式にするかを選択します。

1. 監査証跡ファイルへの書き込みに失敗した場合，HADB サーバを停止（異常終了）します。

2. 監査証跡ファイルへの書き込みに失敗した場合，書き込みに失敗した監査証跡を破棄します。HADB サーバの動作は継続します。

上記の 2.の方法を選択した場合，監査証跡の欠損が発生するおそれがあります。この場合，セキュリティインシデントが発生したときの調査が不十分になるおそれがあります。また，安全管理措置を怠ったと見なされると，社会的信用の失墜につながり，大きな損失となるおそれもあります。

情報分析システムは，24 時間連続稼働が求められる EC サイトとは完全に切り離されています。そのため，情報分析システムが停止しても，EC サイトの運営に影響はありません。

以上のことから，情報分析システムでは，監査証跡ファイルの書き込みに失敗した場合，HADB サーバを停止する 1.の処理方式を選択することにしました。

さらに，監査証跡ファイルの書き込みに失敗したときに出力されるKFAA51404-E メッセージを監視し，書き込み失敗を検知した場合は，HADB サーバの管理担当者と監査証跡機能の管理担当者に対して，メールが送信される運用を決めました。

12.7.3　環境設定

「12.7.2　設計」で決定した方針や運用方法に基づいて，監査証跡機能の環境設定を行います。

(1)　監査証跡機能で使用するディレクトリを作成する

監査証跡機能で使用する次のディレクトリを作成します。

- 監査証跡の出力先ディレクトリ（/mnt/auditttrail/outputarea/audit）
- 監査証跡の保存先ディレクトリ（/mnt/auditttrail/savearea/audit_bak）

各ディレクトリに対して必要となるパーミッションについては，「12.2.2　監査証跡機能で使用するディレクトリの準備」を参照してください。

> **重要**
>
> DB 管理担当者が，HADB 管理者のアカウントで OS にログインして，ディレクトリを作成します。

(2)　サーバ定義に監査証跡機能に関するオペランドを追加する

サーバ定義にadb_audit_log_path オペランドを追加します。adb_audit_log_path オペランドには，「(1) 監査証跡機能で使用するディレクトリを作成する」で作成した監査証跡の出力先ディレクトリを指定します。

サーバ定義に追加するオペランドの指定例

```
set adb_audit_log_path = /mnt/auditttrail/outputarea/audit
```

なお，次に示す監査証跡機能に関するオペランドについては，デフォルト値を適用するため，指定を省略します。

- adb_audit_log_max_size
- adb_audit_log_max_num

重要

- DB 管理担当者が，HADB 管理者のアカウントで OS にログインしてこの操作を行います。
- HADB サーバが稼働中の場合は，HADB サーバを正常終了してからサーバ定義を修正してください。

(3) HADB サーバをオフラインモードで開始する

adbstart コマンドを実行して，HADB サーバをオフラインモードで開始します。HADB サーバをオフラインモードで開始することで，監査証跡機能の環境設定を実施している間に，不正なアクセスが行われることを防ぐことができます。

コマンドの実行例

```
adbstart --offline
```

重要

DB 管理担当者が，HADB 管理者のアカウントで OS にログインしてこの操作を行います。

(4) 監査人を作成する

監査証跡機能の管理担当者や，DB 監査担当者が使用する監査人（監査権限を持っている HADB ユーザ）を作成します。

監査人の作成

監査証跡機能の管理担当者が使用する HADB ユーザ（ADBAUDITADMIN）を新規に作成し，その HADB ユーザに監査管理権限を付与します。

- 監査管理権限を持つ HADB ユーザの作成例

```
CREATE USER "ADBAUDITADMIN" IDENTIFIED BY '@udi+@dmin'
GRANT AUDIT ADMIN,CONNECT TO "ADBAUDITADMIN"
```

また，DB 監査担当者が使用する HADB ユーザ（ADBAUDITVIEWER）を新規に作成し，その HADB ユーザに監査参照権限を付与します。

- 監査参照権限を持つ HADB ユーザの作成例

```
CREATE USER ”ADBAUDITVIEWER” IDENTIFIED BY ’@udi+viewer’
GRANT AUDIT VIEWER,CONNECT TO ”ADBAUDITVIEWER”
```

重要

DB 管理担当者が，DBA 権限を持っている HADB ユーザの認可識別子（ADBADMIN）で HADB サーバに接続してこの操作を行います。

監査人のパスワード変更

監査人の作成後，DB 管理担当者は，監査証跡機能の管理担当者に，監査管理権限を付与した HADB ユーザの認可識別子と初期パスワードを連絡します。また，DB 監査担当者には，監査参照権限を付与した HADB ユーザの認可識別子と初期パスワードを連絡します。各担当者は，その HADB ユーザの認可識別子で HADB サーバに接続し，初期パスワードを変更してください。

- 監査管理権限を持っている HADB ユーザのパスワードの変更例

```
ALTER USER ”ADBAUDITADMIN” IDENTIFIED BY ’@uditDDD’
```

認可識別子がADBAUDITADMIN の HADB ユーザで HADB サーバに接続し，上記の SQL 文を実行します。

- 監査参照権限を持っている HADB ユーザのパスワードの変更例

```
ALTER USER ”ADBAUDITVIEWER” IDENTIFIED BY ’@uditVVV’
```

認可識別子がADBAUDITVIEWER の HADB ユーザで HADB サーバに接続し，上記の SQL 文を実行します。

メモ

監査人（監査権限を持っている HADB ユーザ）のパスワードの変更は，監査人本人だけが実行できます。

(5) 監査証跡機能を有効にする

adbauditrail --start コマンドを実行して，監査証跡機能を有効にします。

コマンドの実行例

```
adbauditrail -u ”ADBAUDITADMIN” -p ’@uditDDD’ --start --write-error DOWN
```

[説明]

- -u と-p オプションには，監査管理権限を持っている HADB ユーザの認可識別子とパスワードを指定します。
- --start オプションは，監査証跡機能を有効にする際に指定するオプションです。

- --write-error オプションには，「12.7.2　設計」の「(5)　監査証跡ファイルへの書き込みに失敗した場合の処理方式を選択する」で決定した処理方式を指定します。この例では，DOWN（監査証跡ファイルへの書き込みに失敗した場合，HADB サーバを異常終了する処理方式）を指定します。

> **重要**
>
> 監査証跡機能の管理担当者が，HADB 管理グループに所属するユーザのアカウントで OS にログインし，この操作を行います。

adbauditttrail --start コマンドの実行後，adbauditttrail -d コマンドを実行して，監査証跡機能が有効になっていることを確認します。

コマンドの実行例

```
adbauditttrail -u ”ADBAUDITADMIN” -p ’@uditDDD’ -d
```

コマンドの実行結果

```
audit  write-error audit-directory-path             audit-file-max-size audit-file-number
ACTIVE DOWN        /mnt/auditttrail/outputarea/audit 256                 0
```

[説明]

- 監査証跡機能が有効になっている場合は，audit 列にACTIVE が表示されます。
- write-error 列には，--write-error オプションの指定値（監査証跡ファイルへの書き込みに失敗した場合の処理方式）が表示されます。

(6)　監査対象を定義する

「12.7.2　設計」の「(2)　監査証跡を出力するイベントを決める」で決めた設計方針に基づいて，選択監査イベントでも監査証跡を出力するようにします。CREATE AUDIT 文を実行して，監査対象を定義します。

CREATE AUDIT 文の実行例

```
CREATE AUDIT AUDITTYPE EVENT FOR ANY OPERATION
```

> **重要**
>
> 監査証跡機能の管理担当者が，監査管理権限を持っている HADB ユーザの認可識別子（ADBAUDITADMIN）で HADB サーバに接続してこの操作を行います。

(7)　HADB サーバの稼働モードを変更する

adbchgsrvmode コマンドを実行して，HADB サーバの稼働モードを通常モードに変更します。稼働モードを通常モードに変更すると，クライアントからの HADB サーバへの接続要求を受け付けます。

コマンドの実行例

```
adbchgsrvmode --normal
```

重要

DB 管理担当者が，HADB 管理者のアカウントで OS にログインしてこの操作を行います。

12.7.4　定期運用

次に示す操作を定期的に実施します。

- 監査証跡の出力先ディレクトリ下の監査証跡ファイルを，監査証跡の保存先ディレクトリに移動する
- 監査証跡の保存先ディレクトリ下の，保管期間を超えた監査証跡ファイルを削除する

それぞれの操作について説明します。

(1)　監査証跡ファイルを移動する

監査証跡の出力先ディレクトリ下の監査証跡ファイルを，定期的に監査証跡の保存先ディレクトリ下に移動します。「12.7.2　設計」の「(3)　監査証跡機能で使用するディレクトリを設計する」で決めた監査証跡ファイルの保管規則に基づいて，出力後 3 か月が経過した監査証跡ファイルを監査証跡の保存先ディレクトリに移動します。

コマンドの実行例

```
find /mnt/audittrail/outputarea/audit -name ”adbaud-*.aud” -and -mtime 93 -exec mv {} /mnt/a
udittrail/savearea/audit_bak ¥;
```

重要

DB 管理担当者が，HADB 管理者のアカウントで OS にログインしてこの操作を行います。

(2)　監査証跡ファイルを削除する

監査証跡の保存先ディレクトリ下の監査証跡ファイルを定期的に削除します。「12.7.2　設計」の「(3)　監査証跡機能で使用するディレクトリを設計する」で決めた監査証跡ファイルの保管規則に基づいて，出力後 1 年が経過した監査証跡ファイルを削除します。

コマンドの実行例

```
find /mnt/audittrail/savearea/audit_bak -name ”adbaud-*.aud” -and -mtime 365 -exec rm -f {}
¥;
```

重要

DB 管理担当者が，HADB 管理者のアカウントで OS にログインしてこの操作を行います。

監査証跡ファイルの移動と削除を自動化したい場合は，次のコマンドを cron に登録します。

```
crontab -e
# 以下，crontab の設定記述。
# 毎日0時5分に，出力後3か月が経過した監査証跡ファイルを移動する。
5 0 * * * /bin/find /mnt/audittrail/outputarea/audit -name ”adbaud-*.aud” -and -mtime 93 -ex
ec mv {} /mnt/audittrail/savearea/audit_bak ¥;
# 毎日0時5分に，出力後1年が経過した監査証跡ファイルを削除する。
5 0 * * * /bin/find /mnt/audittrail/savearea/audit_bak -name ”adbaud-*.aud” -and -mtime 365
-exec rm -f {} ¥;
```

12.7.5　定期監査の実施

「12.7.2　設計」の「(1)　監査の方針を決める」で決めた監査方針に基づいて，次のことを定期監査で確認します。

- HADB サーバへの不審な接続がないことを確認する

  A 社の運用では，基本的に BI ツールから HADB サーバに接続しています。データベースのメンテナンス作業などで，DB 管理担当者が HADB サーバに接続する場合は，HADB サーバへの接続申請を事前に行っています。そのため，これら以外の方法で HADB サーバへの接続が行われていないかを確認します。また，HADB サーバへの接続は，申請どおりに適切に行われているかを確認します。

- データベースの不正操作がないことを確認する

  データベースの操作は，BI ツールが自動的に作成した SQL 文によって行われます。そのため，実行される SQL 文は定型的なものが多く，基本的には数パターンの SQL 文しか実行されません。そのパターンから外れた，複雑な SQL 文が実行されていないかを確認します。複雑な SQL 文が実行されている場合，不正操作の可能性があります。

- 監査証跡機能の設定が不正に変更されていないことを確認する

  監査証跡機能の設定変更は，社内で決められた手順に従って申請を行い，変更の許可が下りたあとに実施されます。そのため，許可されていない設定変更が実施されていないかを確認します。

上記の各監査項目の確認方法の例を以降で説明します。

(1) HADB サーバへの不審な接続がないことを確認する

A 社の情報分析システムでは，BI ツールから HADB サーバに接続する際に使用される，認可識別子とアクセス元のマシンの IP アドレスが，次のように決まっています。

- 認可識別子：BI_ANALYSIS
- アクセス元のマシンの IP アドレス：11.168.xx.32

HADB サーバへの接続時に，上記以外の認可識別子や IP アドレスが使用された場合，その HADB サーバへの接続は，不正操作の可能性があります。

また，A 社では，DB 管理担当者が，データベースのメンテナンス作業などを行う場合は，事前に申請をする規則になっています。申請書には，メンテナンスの理由，作業内容，使用するマシン，作業の実施時刻，作業時に使用する HADB ユーザの認可識別子などの情報を記入します。申請以外の作業を実施した場合は，実施した作業とその理由を申請先に報告します。このような規則で運用されているため，申請以外の操作や，報告以外の操作で HADB サーバへの接続が行われた場合，その HADB サーバへの接続は，不正操作の可能性があります。

以上のことから，定期監査の際は次のことを確認します。

- HADB サーバへの接続に失敗した HADB ユーザがいるか
- 申請内容（認可識別子，マシンの IP アドレス，作業時間帯）にはない，HADB サーバへの接続が行われていないか

それぞれの調査方法を以降で説明します。

■HADB サーバへの接続に失敗した HADB ユーザがいるかどうかを調査する方法

BI ツールからの HADB サーバへの接続は，基本的には失敗しません。そのため，HADB サーバへの接続に失敗した HADB ユーザがいるかどうかを重点的に調査します。次の SQL 文を実行すると，HADB サーバへの接続に失敗した HADB ユーザを確認できます。

SQL 文の実行例

```
SELECT "USER_NAME","CLIENT_IP_ADDRESS","EXEC_TIME"                                  ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/outputarea/audit/*.aud'])) "DT" ...2
    WHERE "EVENT_SUBTYPE"='CONNECT' AND "EVENT_RESULT"='FAILURE'                    ...3
```

[説明]

1. HADB サーバへの接続に失敗した HADB ユーザの認可識別子（USER_NAME），接続依頼元のマシンの IP アドレス（CLIENT_IP_ADDRESS），接続操作を実行した日時（EXEC_TIME）の一覧を出力するために指定します。
2. 定期監査では，過去 3 か月の監査証跡を調査対象とするため，監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）下の全監査証跡ファイルを入力情報にします。
3. HADB サーバへの接続が失敗した操作記録を抽出するための条件を指定します。

上記の SQL 文の実行結果を確認します。認可識別子（USER_NAME），接続依頼元のマシンの IP アドレス（CLIENT_IP_ADDRESS），接続操作を実行した日時（EXEC_TIME）が，申請の内容や，事後報告の内容と合っているかを確認します。合っていない操作記録がある場合は，不正操作の可能性があるため，そのことを監査結果として報告します。

■申請内容（認可識別子，マシンの IP アドレス，作業時間帯）にはない，HADB サーバへの接続が行われていないかを調査する方法

申請された作業以外で，HADB サーバへの接続が行われていないかを調査します。次の SQL 文を実行すると，申請された作業以外の，HADB サーバへの接続記録を確認できます。

SQL 文の実行例

```
SELECT "USER_NAME","CLIENT_IP_ADDRESS","EXEC_TIME"                                    ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/outputarea/audit/*.aud'])) "DT" ...2
    WHERE "EVENT_SUBTYPE"='CONNECT'                                                   ...3
     AND ("CLIENT_IP_ADDRESS"!='10.196.xx.122'                                        ...4
       OR "USER_NAME"!='MAINTENANCE_USER25239'                                        ...4
       OR "EXEC_TIME" NOT BETWEEN TIMESTAMP'2017/06/08 00:00:00.000000'               ...4
                              AND TIMESTAMP'2017/06/08 00:30:00.000000')              ...4
```

[説明]

1. HADB サーバへの接続を試みた HADB ユーザの認可識別子（USER_NAME），接続依頼元のマシンの IP アドレス（CLIENT_IP_ADDRESS），接続操作を実行した日時（EXEC_TIME）の一覧を出力するために指定します。
2. 定期監査では，過去 3 か月の監査証跡を調査対象とするため，監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）下の全監査証跡ファイルを入力情報にします。
3. HADB サーバへの接続記録を抽出するための条件を指定します。
4. 申請内容に当てはまらない条件を指定します。申請内容に記載されている IP アドレス，認可識別子，作業時間を否定条件で指定します。

上記の SQL 文の例は，申請された作業が 1 件の場合の例です。申請された作業が複数ある場合は，その申請内容の条件を探索条件に追加してください。

上記の SQL 文の実行結果には，申請された作業以外で HADB サーバへの接続を試みた HADB ユーザの認可識別子などが出力されます。これらの HADB サーバへの接続は，不正操作の可能性があるため，そのことを監査結果として報告します。

(2) データベースの不正操作がないことを確認する

A 社の情報分析システムでは，次に示すデータベース操作が行われます。

- 表へのデータインポート（30 分ごとに実行）
- BI ツールを使用したデータベースの検索
- 申請された作業によるデータベース操作

上記以外のデータベース操作は，不正操作の可能性があるため，定期監査の際は次のことを確認します。

- 不審なデータインポートが実行されていないか
- BI ツール以外の方法で，データベースの検索が実行されていないか
- 申請されていないデータベース操作が実行されていないか

それぞれの調査方法を以降で説明します。

■不審なデータインポートが実行されていないかを調査する方法

次の SQL 文を実行すると，表へのデータインポートの操作記録を確認できます。

SQL 文の実行例

```
SELECT "USER_NAME","EXEC_TIME"                                                  ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/outputarea/audit/*.aud'])) "DT" ...2
    WHERE "EVENT_SUBTYPE"='ADBIMPORT'                                           ...3
```

[説明]

1. データインポートを実行した HADB ユーザの認可識別子（USER_NAME），データインポートを実行した日時（EXEC_TIME）の一覧を出力するために指定します。
2. 定期監査では，過去 3 か月の監査証跡を調査対象とするため，監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）下の全監査証跡ファイルを入力情報にします。
3. adbimport コマンドを実行した操作記録を抽出するための条件を指定します。

上記の SQL 文の実行結果のEXEC_TIME 列（データインポートを実行した日時）を確認します。A 社では，毎時 00 分頃または 30 分頃にデータインポートが実行されます。明らかに違う時刻にデータインポートが実行されている場合は，定期的なデータインポートではない可能性があります。また，事前に申請されているデータインポートではない場合も不正操作の可能性があるため，そのことを監査結果として報告します。

■BI ツール以外の方法で，データベースの検索が実行されていないかを調査する方法

次の SQL 文を実行すると，BI ツールを使用した検索を除いた，データベースの検索記録を確認できます。

SQL 文の実行例

```
SELECT "USER_NAME","EXEC_TIME"                                                  ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/outputarea/audit/*.aud'])) "DT" ...2
    WHERE "EVENT_SUBTYPE"='SELECT'                                              ...3
     AND ("CLIENT_IP_ADDRESS"!='11.168.xx.32' OR "USER_NAME"!='BI_ANALYSIS')    ...3
```

[説明]

1. データベースの検索（BI ツールを使用した検索を除く）を実行した HADB ユーザの認可識別子（USER_NAME），検索を実行した日時（EXEC_TIME）の一覧を出力するために指定します。
2. 定期監査では，過去 3 か月の監査証跡を調査対象とするため，監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）下の全監査証跡ファイルを入力情報にします。
3. データベースの検索記録（BI ツールを使用した検索を除く）を抽出するための条件を指定します。

BI ツールを使用してデータベースを検索する場合，HADB サーバへの接続時に使用する認可識別子は，BI_ANALYSIS 固定となっています。また，HADB サーバへの接続元のクライアントの IP アドレスは，11.168.xx.32 固定となっています。したがって，この認可識別子または IP アドレス以外で HADB サーバに接続している場合は，BI ツールを使用しないでデータベースを検索しています。この検索は，不正操作の可能性があるため，そのことを監査結果として報告します。

なお，事前に申請されているデータベースの検索操作でないことも確認してください。

■申請されていないデータベース操作が実行されていないかを調査する方法

申請されていないデータベース操作が実行されていないかを調査するには，データベースの操作記録の一覧を出力し，申請内容と照らし合わせるなどの方法が考えられます。

(3) 監査証跡機能の設定が不正に変更されていないことを確認する

監査証跡機能の設定は，社内で決められた手順に従って変更の申請を行い，変更の許可が下りた場合に限り設定変更ができる規則になっています。したがって，申請されていない設定変更が行われた場合，その設定変更は不正操作の可能性があります。

また，申請が行われている場合であっても，申請内容と異なる設定変更が行われたときは，その設定変更も不正操作の可能性があります。これらの不正操作の可能性を監査の結果として報告します。

次の SQL 文を実行すると，監査証跡機能の設定変更の操作記録を確認できます。

SQL 文の実行例

```
SELECT "USER_NAME","CLIENT_IP_ADDRESS","EXEC_TIME"                                    ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/outputarea/audit/*.aud'])) "DT" ...2
    WHERE "EVENT_TYPE"='AUDIT'                                                        ...3
```

[説明]

1. 監査証跡機能の設定変更を行った HADB ユーザの認可識別子（USER_NAME），設定変更を依頼したマシンの IP アドレス（CLIENT_IP_ADDRESS），設定変更を行った日時（EXEC_TIME）の一覧を出力するために指定します。
2. 定期監査では，過去 3 か月の監査証跡を調査対象とするため，監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）下の全監査証跡ファイルを入力情報にします。
3. 監査証跡機能の設定変更を行った際の操作記録を抽出するための条件を指定します。

上記の SQL 文の実行結果を確認します。申請されていない設定変更が行われている場合，その設定変更は不正操作の可能性があるため，そのことを監査結果として報告します。

12.7.6 セキュリティインシデント発生時の調査

次に示すセキュリティインシデントが発生したときの調査例を説明します。

セキュリティインシデントの発生状況

A 社の EC サイトを利用しているユーザの個人情報が，インターネット上に流出している可能性があるとの連絡があり，確認したところ，A 社の情報分析システムで管理していた顧客情報であることが判明しました。そこで，DB 管理担当者と監査証跡機能の管理担当者が，顧客情報の漏えい経路の確認，漏えい時期，漏えいしたデータの規模についての調査を行うことに決定しました。調査方法の例を以降で説明します。

(1) 情報の漏えい元を特定する

インターネット上に流出した情報には，顧客の性別や誕生日などの情報が含まれていることが調査の結果わかっています。これらの情報は，情報分析システムのデータベースのCUSTOMERS_ANALYSIS 表に格納されている情報であることから，この表のデータが情報の漏えい元であると特定しました。

(2) 漏えい元の情報にアクセスした HADB ユーザを調査する

漏えい元の情報を格納しているCUSTOMERS_ANALYSIS 表にアクセスした HADB ユーザ，アクセス元 IP アドレス，アクセス日時，およびアクセス件数を調査します。

次の SQL 文を実行すると，CUSTOMERS_ANALYSIS 表にアクセスした HADB ユーザなどの情報を確認できます。

SQL 文の実行例

```
SELECT "USER_NAME","CLIENT_IP_ADDRESS","EXEC_TIME","ACCESS_COUNT"                    ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/savearea/audit_bak/*.aud',      ...2
                                    '/mnt/audittrail/outputarea/audit/*.aud'])) "DT"  ...2
    WHERE "OBJECT_SCHEMA_NAME"='ANALYSIS_SYSTEM'                                     ...3
      AND "OBJECT_NAME"='CUSTOMERS_ANALYSIS'                                         ...3
      AND ("EVENT_SUBTYPE"='SELECT' OR "EVENT_SUBTYPE"='ADBEXPORT')                  ...4
      AND "EVENT_RESULT"='SUCCESS'                                                   ...5
```

[説明]

1. CUSTOMERS_ANALYSIS 表にアクセスした HADB ユーザの認可識別子（USER_NAME），表へのアクセスを依頼したマシンの IP アドレス（CLIENT_IP_ADDRESS），アクセス日時（EXEC_TIME），およびアクセス件数（ACCESS_COUNT）の一覧を出力するために指定します。
2. 不正アクセスの実行時期が不明なため，すべての監査証跡を調査対象にします。そのため，次のディレクトリ下の全監査証跡ファイルを入力情報にします。
   - 監査証跡の保存先ディレクトリ（/mnt/audittrail/savearea/audit_bak）
   - 監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）
3. 漏えい元のCUSTOMERS_ANALYSIS 表のスキーマ名と表識別子を検索の条件に指定します。
4. 実行された操作（SELECT 文による表検索，adbexport コマンドによる表データのエクスポート）を検索の条件に指定します。
5. 4.の操作（イベント）が成功したという条件を，検索の条件に指定します。

重要

- DB 監査担当者が，監査参照権限を持っている HADB ユーザの認可識別子（ADBAUDITVIEWER）で HADB サーバに接続してこの操作を行います。
- 現用ファイル中の監査証跡も調査対象にする場合は，adbauditttrail --swap コマンドで現用の監査証跡ファイルを切り替えたあとに SQL 文を実行してください。

(3) 不正アクセスを実行した HADB ユーザが，ほかの表にアクセスした形跡がないかを調査する

「(2) 漏えい元の情報にアクセスした HADB ユーザを調査する」で，不正アクセスを実行した HADB ユーザを特定したあとに，その HADB ユーザがほかの表にアクセスした形跡がないかを調査します。不正アクセスを実行した HADB ユーザが，ほかの表にもアクセスしている場合，その表に格納されている情報も漏えいしているおそれがあります。

次の SQL 文を実行すると，不正アクセスを実行した HADB ユーザがアクセスした表を確認できます。

SQL 文の実行例

不正アクセスを実行した HADB ユーザの認可識別子をADBUSER05 とした場合の例です。

```
SELECT "EXEC_TIME","OBJECT_OWNER_NAME","OBJECT_SCHEMA_NAME",                    ...1
                  "OBJECT_NAME","ACCESS_COUNT"                                  ...1
  FROM TABLE(ADB_AUDITREAD(MULTISET['/mnt/audittrail/savearea/audit_bak/*.aud',  ...2
                                    '/mnt/audittrail/outputarea/audit/*.aud'])) "DT"  ...2
    WHERE "USER_NAME"='ADBUSER05'                                               ...3
     AND ("OBJECT_TYPE"='TABLE' OR "OBJECT_TYPE"='VIEW')                        ...4
     AND ("EVENT_SUBTYPE"='SELECT' OR "EVENT_SUBTYPE"='ADBEXPORT')              ...5
     AND "EVENT_RESULT"='SUCCESS'                                               ...6
     AND "OBJECT_NAME"!='CUSTOMERS_ANALYSIS'                                    ...7
```

[説明]

1. 次の項目の一覧を出力するために指定します。
   - 不正アクセスを実行した HADB ユーザが表（ビュー表も含む）にアクセスした日時（EXEC_TIME）
   - 不正アクセスを実行した HADB ユーザがアクセスした表の情報
     - 表の所有者の認可識別子（OBJECT_OWNER_NAME）
     - 表のスキーマ名（OBJECT_SCHEMA_NAME）
     - 表識別子（OBJECT_NAME）
   - アクセス件数（ACCESS_COUNT）
2. 不正アクセスの実行時期が不明なため，すべての監査証跡を調査対象にします。そのため，次のディレクトリ下の全監査証跡ファイルを入力情報にします。

- 監査証跡の保存先ディレクトリ（/mnt/audittrail/savearea/audit_bak）
- 監査証跡の出力先ディレクトリ（/mnt/audittrail/outputarea/audit）

3. 不正アクセスを実行した HADB ユーザの認可識別子（ADBUSER05）を検索の条件に指定します。
4. オブジェクトの種別（表とビュー表）を検索の条件に指定します。
5. 実行された操作（SELECT 文による表検索，adbexport コマンドによる表データのエクスポート）を検索の条件に指定します。
6. 5.の操作（イベント）が成功したという条件を，検索の条件に指定します。
7. CUSTOMERS_ANALYSIS 表を除外することを検索の条件に指定します。CUSTOMERS_ANALYSIS 表に対して不正アクセスを実行していることがわかっているため，この条件を指定します。

重要

- DB 監査担当者が，監査参照権限を持っている HADB ユーザの認可識別子（ADBAUDITVIEWER）で HADB サーバに接続してこの操作を行います。
- 現用ファイル中の監査証跡も調査対象にする場合は，adbaudittrail --swap コマンドで現用の監査証跡ファイルを切り替えたあとに SQL 文を実行してください。

(4) 調査結果を報告する

次のような調査結果をまとめて，報告をします。

- 情報を漏えいさせた HADB ユーザ

  「(2) 漏えい元の情報にアクセスした HADB ユーザを調査する」で特定した HADB ユーザが，情報を漏えいさせた HADB ユーザの可能性があります。
- 情報が漏えいした日時

  「(2) 漏えい元の情報にアクセスした HADB ユーザを調査する」および「(3) 不正アクセスを実行した HADB ユーザが，ほかの表にアクセスした形跡がないかを調査する」で特定した日時が，情報が漏えいした日時である可能性があります。
- 漏えいした情報

  「(1) 情報の漏えい元を特定する」および「(3) 不正アクセスを実行した HADB ユーザが，ほかの表にアクセスした形跡がないかを調査する」で特定した表に格納されている情報が漏えいしている可能性があります。
- 漏えいした情報の件数（被害規模）

  「(2) 漏えい元の情報にアクセスした HADB ユーザを調査する」および「(3) 不正アクセスを実行した HADB ユーザが，ほかの表にアクセスした形跡がないかを調査する」で特定したアクセス件数が，漏えいした可能性のある情報の件数になります。

12.8　監査証跡機能と JP1/Audit Management - Manager との連携

ここでは，監査証跡機能と JP1/Audit Management - Manager（JP1/Audit）を連携する方法について説明します。

監査証跡機能と JP1/Audit の連携に関する説明は，JP1/Audit および JP1/Base についての知識があることを前提としています。JP1/Audit の詳細については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。また，JP1/Base の詳細については，マニュアル『JP1/Base　運用ガイド』を参照してください。

監査証跡の変換に関する概要を知りたい場合は，「2.18.9　監査証跡の変換（JP1/Audit Management - Manager との連携）」を参照してください。

> **重要**
>
> JP1/Audit のマニュアルでは，監査証跡のことを**監査ログ**と表記しています。そのため，HADB のマニュアルでも，監査証跡を監査ログと表記する個所があります。

12.8.1　監査証跡機能と JP1/Audit を連携する際の概要

ここでは，監査証跡機能と JP1/Audit を連携する際の概要として，次の 3 点について説明します。

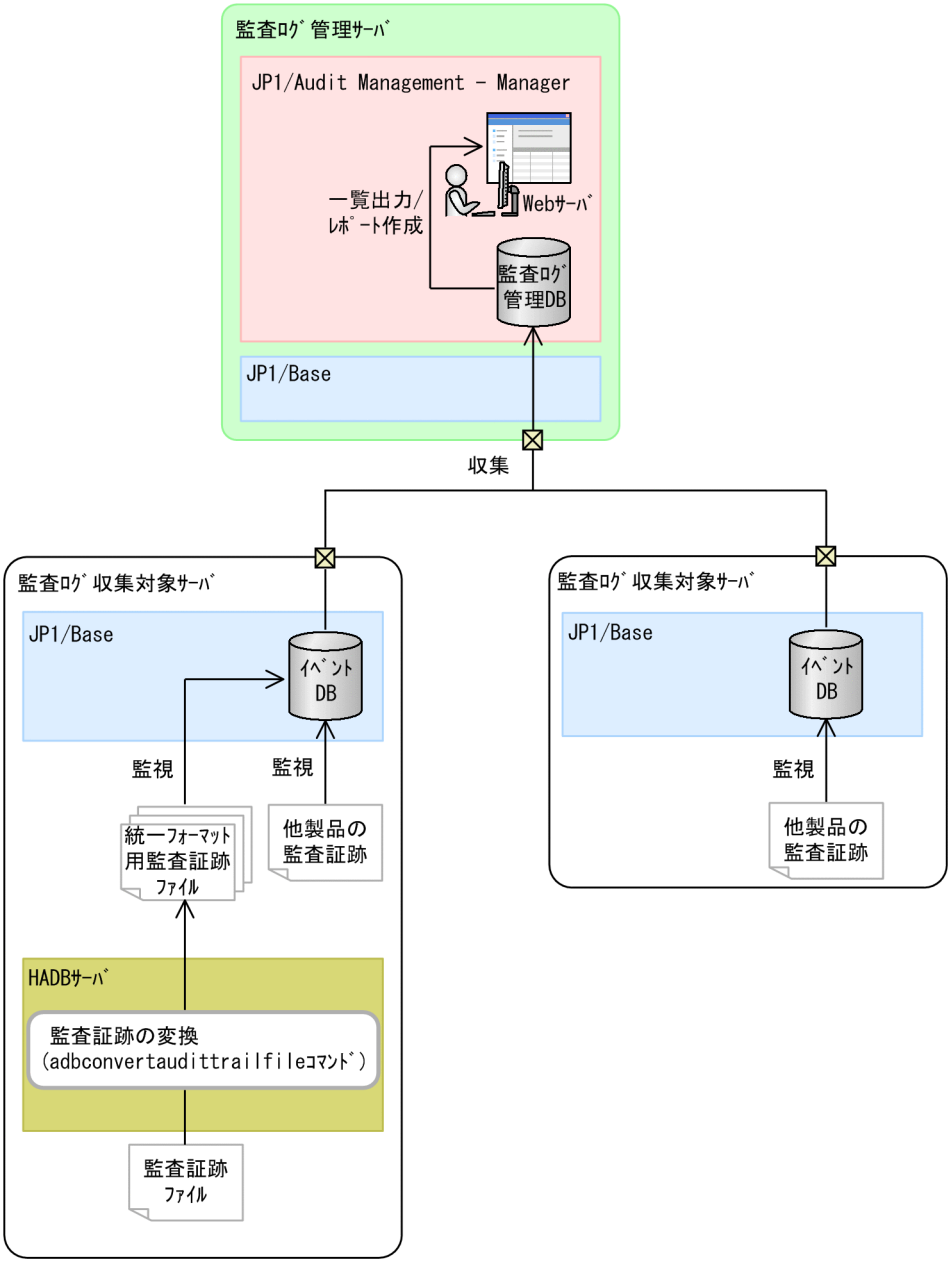
- システム構成例
- サーバ構成と前提ソフトウェア
- `adbconvertaudittrailfile` コマンドを実行する際の注意事項

(1)　システム構成例

ここでは，監査証跡機能と JP1/Audit を連携する場合のシステム構成について説明します。

監査証跡機能と JP1/Audit を連携する場合のシステム構成例を，次の図に示します。

図 12-7　監査証跡機能と JP1/Audit を連携する場合のシステム構成例



[説明]

図に示すシステム構成例では，2 種類のサーバ（**監査ログ管理サーバ**と**監査ログ収集対象サーバ**）で構成されています。

監査ログ管理サーバでは，監査ログ収集対象サーバ上の JP1/Base のイベントデータベースから監査証跡を収集して，一元管理します。

監査ログ収集対象サーバでは，監査ログ収集対象サーバ上の HADB サーバが監査証跡を出力して，`adbconvertaudittrailfile` コマンドで変換します。そして，変換後の監査証跡を収集して，監査ログ収集対象サーバ上の JP1/Base のイベントデータベースに蓄積します。

> **メモ**
>
> 監査を行うユーザは，Web ブラウザ上から監査ログ管理サーバにアクセスして，監査を実施できます。

(2) サーバ構成と前提ソフトウェア

ここでは，監査証跡機能と JP1/Audit を連携する場合のサーバ構成と，前提ソフトウェアについて説明します。

■サーバ構成

「(1) システム構成例」で説明しているとおり，監査証跡機能と JP1/Audit を連携する場合は，次の 2 種類のサーバが必要です。

- 監査ログ管理サーバ

  JP1/Audit および JP1/Base が稼働するサーバマシンです。監査ログ収集対象サーバから監査証跡を収集して，一元管理します。

- 監査ログ収集対象サーバ

  HADB サーバおよび JP1/Base が稼働するサーバマシンです。`adbconvertaudittrailfile` コマンドで変換された HADB サーバの監査証跡を収集して，JP1/Base のイベントデータベースに蓄積します。

> **重要**
>
> HADB サーバと JP1/Audit では，前提 OS が異なります。そのため，HADB サーバが稼働するサーバマシンに，JP1/Audit をインストールすることはできません。
>
> 監査ログ管理サーバの前提 OS については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。

■前提ソフトウェア

監査証跡機能と JP1/Audit を連携する場合の前提ソフトウェアを，次の表に示します。

表 12-4　監査証跡機能と JP1/Audit を連携する場合の前提ソフトウェア

導入対象のサーバ	前提ソフトウェア
監査ログ管理サーバ	• HADB サーバの OS が RHEL 6 または RHEL 7 の場合 JP1/Audit Management - Manager 11-00 以降 JP1/Base 11-10 以降 • HADB サーバの OS が RHEL 8 の場合 JP1/Audit Management - Manager 11-00-04 以降 JP1/Base 12-10-02 以降
監査ログ収集対象サーバ	• HADB サーバの OS が RHEL 6 または RHEL 7 の場合 JP1/Base 11-10 以降

導入対象のサーバ	前提ソフトウェア
	• HADB サーバの OS が RHEL 8 の場合 JP1/Base 12-10-02 以降

(3) adbconvertaudittrailfile コマンドを実行する際の注意事項

adbconvertaudittrailfile コマンドを実行する際の注意事項を次に示します。

- adbconvertaudittrailfile コマンドは，JP1/Audit で監査ログを監視しているときに実行してください。監査ログの監視を停止しているときに，adbconvertaudittrailfile コマンドを実行しても，変換された監査証跡は JP1/Audit に収集されません。監査ログの監視については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。
- adbconvertaudittrailfile コマンドは，HADB サーバが稼働中のときだけ実行できます。adbconvertaudittrailfile コマンドの実行中に，ほかのユーザが HADB サーバを利用することを避けたいときは，HADB サーバの稼働モードをメンテナンスモードに変更してください。そのあとで，adbconvertaudittrailfile コマンドを実行してください。メンテナンスモードについては，「10.2.3 HADB サーバの稼働モード」を参照してください。
- adbconvertaudittrailfile コマンドは，**現用の監査証跡ファイル**に対しては実行できません。現用の監査証跡ファイル内の監査証跡を変換したい場合は，事前にadbaudittrail --swap コマンドで，現用の監査証跡ファイルの切り替えを行ってください。そのあとで，リネームされた監査証跡ファイルに対して，adbconvertaudittrailfile コマンドを実行してください。現用の監査証跡ファイルの切り替えについては，「12.4.2　現用の監査証跡ファイルの切り替え」を参照してください。

メモ

- adbconvertaudittrailfile コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbconvertaudittrailfile（監査証跡ファイルの変換)』を参照してください。
- adbaudittrail コマンドについては，『HADB　コマンドリファレンス』の『adbaudittrail（監査証跡機能の管理)』を参照してください。

12.8.2　統一フォーマット用監査証跡ファイルの形式と出力項目

ここでは，adbconvertaudittrailfile コマンドで変換された監査証跡が出力される，統一フォーマット用監査証跡ファイルの形式について説明します。

adbconvertaudittrailfile コマンドを実行すると，統一フォーマット用監査証跡ファイルには，1 行ごとに 1 つの監査証跡が出力されます。また，統一フォーマット用監査証跡ファイルには，最初の監査証跡が出力される前に空白行が出力されます。

統一フォーマット用監査証跡ファイルに出力される，adbconvertaudittrailfile コマンドで変換された監査証跡の出力形式と出力項目を次に示します。

■adbconvertaudittrailfile コマンドで変換された監査証跡の出力形式

CALFHM 1.0,出力項目1=値1,出力項目2=値2,出力項目3=値3,…出力項目*n*=値*n*

■adbconvertaudittrailfile コマンドで変換された監査証跡の出力項目

adbconvertaudittrailfile コマンドで変換された監査証跡の出力項目を，次の表に示します。

adbconvertaudittrailfile コマンドで変換された監査証跡の出力項目の大部分は，ADB_AUDITREAD 関数で監査証跡ファイル中の監査証跡を表形式のデータ集合（表関数導出表）に変換したときの出力項目と同じです。ADB_AUDITREAD 関数による表関数導出表の列構成については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

なお，出力項目に出力されるデータが出力項目のバイト数の上限を超える場合は，データの先頭またはデータの末尾が省略されます。このとき，該当個所には，省略されたことを示す「...」が付与されます。次の表で，特に記述がない出力項目についてはデータの末尾が省略されます。

また，出力されるデータがNULL の場合，出力項目そのものが出力されません。

表 12-5　adbconvertaudittrailfile コマンドで変換された監査証跡の出力項目の一覧

項番	出力項目			出力される内容	ADB_AUDITREAD 関数で変換した表関数導出表に該当する列名
	区分	項目名	属性名		
1	ヘッダ情報	共通仕様識別子	－	CALFHM が出力されます。 統一フォーマット用監査証跡ファイル内の，すべての監査証跡の先頭に共通で出力されます。	なし
2		共通仕様リビジョン番号	－	1.0 が出力されます。 統一フォーマット用監査証跡ファイル内の，すべての監査証跡の先頭に共通で出力されます。	なし
3	共通情報	通番	seqnum	1～2,147,483,647 の値が，出力行ごとにカウントされて出力されます。 最大値に達した場合は，1 に戻ります。 また，adbconvertaudittrailfile コマンドの実行ごとに 1 に戻ります。	なし
4		メッセージ ID	msgid	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の EXIT_STATUS 列の内容が出力されます。	EXIT_STATUS
5		日付・時刻	date	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の EXEC_TIME 列の内容が，次の形式で出力されます。 *YYYY-MM-DD*T*hh*:*mm*:*ss*.*nnn*TZD *YYYY-MM-DD*：年月日 T：日付と時刻の区切り	EXEC_TIME

項番	出力項目			出力される内容	ADB_AUDITREAD 関数で変換した表関数導出表に該当する列名
	区分	項目名	属性名		
				hh:*mm*:*ss*.*nnn*：時分秒ミリ秒 *TZD*：監査証跡ファイルに監査証跡が出力されたときのタイムゾーン※1	
6		発生プログラム名	progid	HADB が出力されます。	なし
7		発生コンポーネント名	compid	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のADBDIR 列の内容が，64 バイト以内で出力されます。 64 バイトを超える場合は，データの先頭が省略されます。 出力される値は，エスケープされます。※2	ADBDIR
8		発生プロセス ID	pid	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の HADB_PROCESS_ID 列の内容が出力されます。	HADB_PROCESS_ID
9		発生場所	ocp:host	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の HADB_HOST_NAME 列の内容が，64 バイト以内で出力されます。	HADB_HOST_NAME
10		事象の種別	ctgry	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の EVENT_SUBTYPE 列の内容が，統一フォーマットで定められている種別に変換されて出力されます。 変換内容については，「表 12-6　イベント種別の値と監査事象の種別の値の対応」を参照してください。	EVENT_SUBTYPE
11		事象の結果	result	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の EVENT_RESULT 列の内容が出力されます。	EVENT_RESULT
12		サブジェクト識別情報	subj:uid	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の USER_NAME 列の内容が出力されます。 なお，USER_NAME 列の内容が，NULL のときは出力されません。	USER_NAME
13			subj:euid	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の	OS_USER_NAME

項番	出力項目			出力される内容	ADB_AUDITREAD 関数で変換した表関数導出表に該当する列名
	区分	項目名	属性名		
				OS_USER_NAME 列の内容が，100 バイト以内で出力されます。 なお，USER_NAME 列の内容がNULL のときだけ出力されます。	
14				「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の OS_USER_NAME 列とUSER_NAME 列の内容が，両方ともNULL の場合に，「*」が出力されます。	なし
15	固有情報	オブジェクト情報	obj	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の OBJECT_SCHEMA_NAME 列と OBJECT_NAME 列の内容が連結されて出力されます。 出力形式については，「表 12-7　オブジェクト情報の出力書式」を参照してください。 出力される値は，エスケープされます。※2	• OBJECT_SCHEMA_NAME • OBJECT_NAME
16		動作情報	op	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の EVENT_SUBTYPE 列の内容が，32 バイト以内で出力されます。 出力される値は，エスケープされます。※2	EVENT_SUBTYPE
17		リクエスト送信元ホスト	from:ipv4	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の CLIENT_IP_ADDRESS 列の内容が出力されます。	CLIENT_IP_ADDRESS
18		リクエスト送信元ポート番号	from:port	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」の CLIENT_PORT_NUMBER 列の内容が出力されます。	CLIENT_PORT_NUMBER
19		メッセージ	msg	変換元の監査証跡ファイルのファイル名が出力されます。ディレクトリ名は含まれません。 出力される値は，エスケープされます。※2	なし

（凡例）

－：該当しません。

注※1

UTC（協定世界時）からのオフセットを示したタイムゾーン指定です。次のどれかが出力されます。

- *+hh:mm*
  UTC（協定世界時）から*hh:mm* だけ進んでいることを示します。
- *-hh:mm*
  UTC（協定世界時）から*hh:mm* だけ遅れていることを示します。
- *Z*
  UTC（協定世界時）と同じであることを示します。

日本標準時は，「+09:00」となります。

注※2

エスケープによって，出力される値全体が二重引用符（”）で囲まれます。また，出力される値に二重引用符（”）が含まれる場合は，２つの二重引用符（””）に置き換えられます。

データの長さには，エスケープによる二重引用符（”）も含まれます。そのため，エスケープによる二重引用符（”）を含んだデータが，出力できるバイト数の上限よりも長い場合は，データの先頭またはデータの末尾が省略されます。

HADB の監査証跡のイベント種別の値と，統一フォーマットの監査事象の種別の値の対応を，次の表に示します。

表 12-6　イベント種別の値と監査事象の種別の値の対応

項番	イベントの分類	イベントの種類	監査証跡に出力されるイベント種別の値	統一フォーマットの監査事象の種別の値
1	必須監査イベント	システムイベント	ADBSTART	StartStop
2			ADBSTOP	
3			ADBCHGSRVMODE	ConfigurationAccess
4			ADBCHGNODETYPE	
5			ADBCHGSQLTRC	
6			ADBCLIENTDEFMANG	
7			ADBMODAREA	
8			ADBMODBUFF	
9			ADBCOLUMNIZE	
10		監査イベント	GRANT	AccessControl
11			REVOKE	
12			CREATE AUDIT	ConfigurationAccess
13			DROP AUDIT	

項番	イベントの分類	イベントの種類	監査証跡に出力されるイベント種別の値	統一フォーマットの監査事象の種別の値
14			ALTER USER	AccessControl
15			ADBAUDITTRAIL START	ConfigurationAccess
16			ADBAUDITTRAIL STOP	
17			ADBAUDITTRAIL SWAP	
18			ADBAUDITTRAIL DISPLAY	
19			SELECT	ContentAccess
20			ADBEXPORT	
21			ADBCONVERTAUDITTRAILFILE	
22	選択監査イベント	セッションイベント	CONNECT	Authentication
23			DISCONNECT	
24		権限管理イベント	GRANT	AccessControl
25			REVOKE	
26			CREATE USER	
27			DROP USER	
28			ALTER USER	
29		定義系 SQL イベント	CREATE INDEX	ContentAccess
30			CREATE SCHEMA	
31			CREATE TABLE	
32			CREATE VIEW	
33			DROP INDEX	
34			DROP SCHEMA	
35			DROP TABLE	
36			DROP VIEW	
37			ALTER TABLE	
38			ALTER VIEW	
39		操作系 SQL イベント	SELECT	
40			INSERT	
41			UPDATE	
42			DELETE	
43			TRUNCATE TABLE	

項番	イベントの分類	イベントの種類	監査証跡に出力されるイベント種別の値	統一フォーマットの監査事象の種別の値
44			PURGE CHUNK	
45			UNKNOWN	
46			GETDATA	
47			GETCOUNT	
48			TABLES	
49			COLUMNS	
50			INDEXES	
51			CHUNKS	
52			GETUSER	
53		コマンド操作イベント	ADBIMPORT	
54			ADBIDXREBUILD	
55			ADBGETCST	
56			ADBDBSTATUS	
57			ADBEXPORT	
58			ADBMERGECHUNK	
59			ADBCHGCHUNKCOMMENT	
60			ADBCHGCHUNKSTATUS	
61			ADBARCHIVECHUNK	
62			ADBUNARCHIVECHUNK	
63			ADBREORGSYSTEMDATA	
64			ADBSYNDICT	

オブジェクト情報の出力書式を，次の表に示します。オブジェクト情報は，OBJECT_SCHEMA_NAME 列および OBJECT_NAME 列の値によって，次に示すどれかの書式で出力されます。

表 12-7　オブジェクト情報の出力書式

項番	OBJECT_SCHEMA_NAME 列	OBJECT_NAME 列	書式	出力例
1	非ナル値	非ナル値	OBJECT_SCHEMA_NAME 列とOBJECT_NAME 列の値を，ピリオド（.）で連結します。	obj=”ADBUSER01.T1”
2	ナル値	非ナル値	OBJECT_NAME 列の値だけを出力します。	obj=”ADBUSER01”※
3	ナル値	ナル値	出力されません。	－

（凡例）

－：該当しません。

注※

CREATE SCHEMA 文の出力例です。この例の場合，OBJECT_NAME 列にはスキーマ名が出力されます。

12.8.3　監査証跡機能と JP1/Audit を連携する際の環境設定

ここでは，監査証跡機能と JP1/Audit を連携する際の環境設定について説明します。

監査証跡機能と JP1/Audit を連携する場合，HADB サーバと JP1/Audit の両方で環境設定が必要です。

(1)　HADB サーバの環境設定

HADB サーバの環境設定としては，**統一フォーマット用監査証跡の出力先ディレクトリ**を準備する必要があります。

統一フォーマット用監査証跡の出力先ディレクトリを作成する際の注意点を，次に示します。

- ４ギガバイト以上の空き容量があるディスクに，統一フォーマット用監査証跡の出力先ディレクトリを作成してください。統一フォーマット用監査証跡の出力先ディレクトリには，次のファイルが格納されます。
  - 統一フォーマット用監査証跡ファイル（adbcommonaudit*XX*.log）

    adbconvertaudittrailfile コマンドで変換された監査証跡が出力されるファイルです。最大で４つのファイルが格納されます。ファイル名の *XX* は通番であり，01～04 のどれかになります。

    ４つの統一フォーマット用監査証跡ファイルの合計サイズは，最大で２ギガバイトです。ただし，**統一フォーマット用監査証跡ファイルの切り替え**を行う際に，古いデータが一時的に残るおそれがあります。そのため，４ギガバイト以上の空き容量があるディスクを準備してください。
  - 統一フォーマット用監査証跡管理ファイル（.adbcommonaudit）

    統一フォーマット用監査証跡ファイル群を管理するファイルです。１つのファイルだけ格納されます。

  > **メモ**
  >
  > 統一フォーマット用監査証跡ファイルの切り替えについては，マニュアル『HADB　コマンドリファレンス』の『adbconvertaudittrailfile（監査証跡ファイルの変換)』を参照してください。

- 統一フォーマット用監査証跡の出力先ディレクトリのフルパスの長さは，末尾の「/」を含めて 239 バイト以内にしてください。フルパスの長さが 240 バイト以上ある場合，JP1/Audit が正しく設定できません。
- HADB 管理者がアクセスできるように，統一フォーマット用監査証跡の出力先ディレクトリには，読み取り権限，書き込み権限および実行権限を付与してください。

- 運用テストを行うなどで，1 つのサーバマシンで複数の HADB サーバを動作させる場合，それぞれの環境で，専用の統一フォーマット用監査証跡の出力先ディレクトリを準備してください。1 つの統一フォーマット用監査証跡の出力先ディレクトリを，複数の HADB サーバで共有しないでください。

(2)　JP1/Audit の環境設定

JP1/Audit では，HADB サーバは標準サポート外のプログラムとして扱われます。そのため，次の環境設定が必要になります。

- JP1/Base のイベントサービスの設定
  「(a)　JP1/Base のイベントサービスの設定」を参照してください。
- 正規化ルールファイルの定義
  「(b)　正規化ルールファイルの定義」を参照してください。
- 動作定義ファイルの作成
  「(c)　動作定義ファイルの作成」を参照してください。
- 製品定義ファイルの作成
  「(d)　製品定義ファイルの作成」を参照してください。
- 監査ログ標準レポート定義ファイルの編集
  「(e)　監査ログ標準レポート定義ファイルの編集」を参照してください。
- 監査ログの収集対象の設定
  「(f)　監査ログの収集対象の設定」を参照してください。

各環境設定の詳細手順については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。または，マニュアル『JP1/Base　運用ガイド』を参照してください。

(a)　JP1/Base のイベントサービスの設定

JP1/Base のイベントサービスの設定を行う必要があります。HADB サーバが出力する監査証跡の件数を踏まえて，監査ログ専用イベントデータベースのサイズを適切に指定してください。

なお，1 回の`adbconvertaudittrailfile` コマンドで変換する監査証跡の件数を見積もることができない場合は，監査ログ専用イベントデータベースには最大サイズを指定してください。

(b)　正規化ルールファイルの定義

正規化ルールファイルについては，新規に定義する必要はありません。

`adbconvertaudittrailfile` コマンドで統一フォーマット用監査証跡ファイルに出力する監査証跡は，日立オープンミドルウェア製品の統一フォーマット形式に従っています。そのため，JP1/Audit が標準で提供する統一フォーマット用の正規化ルールファイル（`admrglrule_CALFHM.conf`）を使用できます。

(c)　動作定義ファイルの作成

JP1/Audit の連携で使用する動作定義ファイルを作成する必要があります。作成手順を次に示します。

手順

1. 動作定義ファイルのサンプルファイルをコピーする

   サーバディレクトリ下の$ADBDIR/sample/jp1audit ディレクトリに格納されているサンプルファイル（admjevlog_HADB.conf）をコピーしてください。そして，*JP1/Audit Management - Managerのインストール先フォルダ*¥conf¥logdef に，サンプルファイル（admjevlog_HADB.conf）を格納してください。

2. 動作定義ファイルのサンプルファイルの設定内容を確認する

   サンプルファイルの設定内容を変更する必要はありません。次の設定内容かどうかを確認してください。

   設定内容

```
retry-times=60
retry-interval=10
FILETYPE=SEQ2
ACTDEF =<Information>1000 "^CALFHM"
```

なお，運用テストを行うなどで，1 つのサーバマシンで複数の HADB サーバを動作させる場合，動作定義ファイルはそれぞれの環境に作成する必要があります。その場合，動作定義ファイルのファイル名を「admjevlog_HADB_*任意の文字列*.conf」にリネームして，ほかのファイル名と重複させないでください。また，「*任意の文字列*」は，「(d)　製品定義ファイルの作成」で説明している製品定義ファイルと一致させてください。

(d)　製品定義ファイルの作成

JP1/Audit の連携で使用する製品定義ファイルを作成する必要があります。作成手順を次に示します。

手順

1. 製品定義ファイルのサンプルファイルをコピーする

   サーバディレクトリ下の$ADBDIR/sample/jp1audit ディレクトリに格納されているサンプルファイル（HADB.conf）をコピーしてください。そして，*JP1/Audit Management - Managerのインストール先フォルダ*¥conf¥product に，サンプルファイル（HADB.conf）を格納してください。

2. 製品定義ファイルのサンプルファイルの設定内容を確認する

   サンプルファイルの設定内容を変更する必要はありません。次の設定内容かどうかを確認してください。

   設定内容

```
AuditLogNum=4
AuditLogName=adbcommonaudit01.log
AuditLogName=adbcommonaudit02.log
AuditLogName=adbcommonaudit03.log
AuditLogName=adbcommonaudit04.log
RegularPattern=admrglrule_CALFHM.conf
ReadOnly=1
```

なお，運用テストを行うなどで，1 つのサーバマシンで複数の HADB サーバを動作させる場合，製品定義ファイルはそれぞれの環境に作成する必要があります。その場合，製品定義ファイルのファイル名を「HADB_*任意の文字列*.conf」にリネームし，ほかのファイル名と重複させないでください。また，「*任意の文字列*」は，「(c)　動作定義ファイルの作成」で説明している動作定義ファイルと一致させてください。

(e)　監査ログ標準レポート定義ファイルの編集

監査ログ標準レポート定義ファイルの編集を行う必要があります。編集手順を次に示します。

重要

通常，JP1/Audit が正規化ルールファイルを使用して収集した監査証跡については，監査ログレポート定義ファイルを使用します。しかし，HADB サーバは，日立オープンミドルウェア製品の統一フォーマット形式を採用しています。そのため，監査ログレポート定義ファイルではなく，監査ログ標準レポート定義ファイルを使用できます。

手順

JP1/Audit Management - Managerの仮想ディレクトリ¥conf 下に格納されている監査ログ標準レポート定義ファイル（admAnalysis.ini）に，次に示す内容を追記してください。

記述内容

```
[HADB]
TYPE=Common
```

なお，記述行の最後には，必ず改行を記述してください。

(f)　監査ログの収集対象の設定

監査ログの収集対象の設定を行う必要があります。JP1/Audit の監査ログ収集マネージャを使用して，HADB サーバを収集対象として設定してください。設定方法の詳細については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。

HADB サーバを収集対象に設定する場合に，［収集対象の設定］ダイアログに設定する値を次に示します。

表 12-8　監査ログの収集対象の設定値

項番	項目	設定値
1	サーバ	HADB サーバのホスト名を指定してください。
2	プログラム	プルダウンメニューから，製品定義ファイル名と一致するものを選択してください。 例えば，製品定義ファイルのファイル名が「HADB.conf」の場合は，［HADB］を選択してください。 なお，「_」は「/」に置き換えられて表示されます。
3	ログフォルダ	「(1)　HADB サーバの環境設定」で準備した，統一フォーマット用監査証跡の出力先ディレクトリをフルパスで指定してください。

項番	項目	設定値
4	コメント	任意のコメントを入力してください。
5	OS 起動時に監査ログの監視を開始する	設定は不要です。

重要

統一フォーマット用監査証跡ファイル内の監査証跡を JP1/Audit が収集するためには，JP1/Audit が監査ログの監視を開始している必要があります。JP1/Audit の環境設定が終わったら，必ず監査ログの監視を開始してください。また，OS 起動時に監査ログの監視を開始するように，OS を設定することを推奨します。

12.8.4　監査証跡機能と JP1/Audit を連携する際の運用方法

JP1/Audit と連携する場合の運用方法を次に示します。運用方法には，adbconvertaudittrailfile コマンドによる監査証跡の変換タイミングによって，次の 2 つがあります。

- 運用方法 1

  監査証跡ファイルが切り替わったタイミングでadbconvertaudittrailfile コマンドを実行して，監査証跡を変換する運用方法です。

  詳細については，「(1)　運用方法 1（監査証跡ファイルが切り替わったタイミングで変換する運用)」を参照してください。

- 運用方法 2

  定期的なタイミングでadbconvertaudittrailfile コマンドを実行して，監査証跡を変換する運用方法です。

  定期的に変換を行う場合，変換対象の監査証跡ファイルの数だけ，adbconvertaudittrailfile コマンドを実行する必要があります。

  詳細については，「(2)　運用方法 2（定期的なタイミングで変換する運用)」を参照してください。

なお，一度に複数の監査証跡ファイルを変換すると，**統一フォーマット用監査証跡ファイルの切り替え**によって，収集途中の監査証跡が消失するおそれがあります。この場合，JP1/Audit が一部の監査証跡を収集できないことがあります。そのため，運用方法 2 よりも，運用方法 1 を採用することを推奨します。

(1)　運用方法 1（監査証跡ファイルが切り替わったタイミングで変換する運用)

監査証跡ファイルが切り替わったタイミングで監査証跡を変換する方法を，次に示します。

この運用方法では，現用の監査証跡ファイルの切り替え時に出力されるメッセージを監視して，監査証跡ファイルが切り替わるたびにadbconvertaudittrailfile コマンドを実行する必要があります。対象メッセージの出力時に，adbconvertaudittrailfile コマンドを自動で実行するジョブを作成して，運用することを推奨します。

なお，HADB サーバを終了したときに切り替わった監査証跡ファイルは，次に HADB サーバを開始するまでは変換できません。そのため，adbconvertaudittrailfile コマンドを実行する前に，HADB サーバが稼働中であることを確認する必要があります。

手順

1. メッセージを監視する

   サーバメッセージログファイル（$ADBDIR/spool/adbmessage*XX*.log※）を確認して，監査証跡ファイルの切り替え時に出力されるKFAA81401-I メッセージ，およびKFAA81402-I メッセージを監視してください。このとき，切り替わった監査証跡ファイルのファイルパスを取得してください。

   注※

   XX は通番であり，01～04 のどれかになります。

2. 監査証跡ファイルを，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリに移動する

   手順 1.で取得した監査証跡ファイルのファイルパスを基に，対象の監査証跡ファイルを移動してください。

3. adbls -d srv コマンドを実行する

   HADB サーバが稼働中であることを確認します。出力項目STATUS に出力される HADB サーバの状態が，次のどれかであることを確認してください。

   - ACTIVE：稼働中（通常モード）
   - QUIESCE：稼働中（静止モード）
   - OFFLINE：稼働中（オフラインモード）
   - MAINTNCE：稼働中（メンテナンスモード）

   もし，HADB サーバが稼働中でない場合は，HADB サーバの開始を待ってください（HADB サーバの開始を待つように，自動で実行するジョブを作成してください）。HADB 管理者や別のジョブによって，HADB サーバが開始したあとで，次の手順に進んでください。

   > **重要**
   >
   > adbconvertaudittrailfile コマンドを自動で実行するジョブに，HADB サーバを開始する処理を組み込んでしまうと，HADB サーバが終了できなくなります。そのため，HADB サーバが稼働中ではない場合は，adbconvertaudittrailfile コマンドを自動で実行するジョブとは別の方法で，HADB サーバを開始するようにしてください。

4. adbconvertaudittrailfile コマンドを実行する

   手順 2.で移動した監査証跡ファイルを，adbconvertaudittrailfile コマンドで変換してください。

> **メモ**
>
> adbconvertaudittrailfile コマンドを自動で実行するジョブとは別に，定期的に現用の監査証跡ファイルを切り替えることを推奨します。現用の監査証跡ファイルの切り替えは，

adbauditrail --swap コマンドで行ってください。定期的にadbauditrail --swap コマンドを実行することで，ジョブによって監査証跡の変換が行われるようにしてください。

(2) 運用方法 2（定期的なタイミングで変換する運用）

定期的なタイミングで監査証跡を変換する運用方法を，次に示します。

この運用方法では，**監査証跡の出力先ディレクトリ**に格納されているすべての監査証跡ファイルが，adbconvertaudittrailfile コマンドの実行対象です。1 日 1 回，週に 1 回など，定期的に adbconvertaudittrailfile コマンドを実行するバッチプログラムを作成して，そのバッチプログラムを実行することを推奨します。

手順

1. adbls -d srv コマンドを実行する

   HADB サーバが稼働中であることを確認します。出力項目STATUS に出力される HADB サーバの状態が，次のどれかであることを確認してください。

   - ACTIVE：稼働中（通常モード）
   - QUIESCE：稼働中（静止モード）
   - OFFLINE：稼働中（オフラインモード）
   - MAINTNCE：稼働中（メンテナンスモード）

   もし，HADB サーバが稼働中でない場合は，HADB サーバを開始してください。そのあとで，次の手順に進んでください。

2. 移動が必要な監査証跡ファイルの一覧を作成する

   OS のfind コマンドで，監査証跡の出力先ディレクトリに存在する監査証跡ファイルを検索して，移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を作成します。なお，次のコマンドでは，現用の監査証跡ファイルは検索対象外となっています。

   コマンドの実行例

   ```
   find /mnt/audittrail/outputarea/audit ¥
       -name ”adbaud-????????-??????-???*.aud” ¥
       > /home/adbmanager/tmp/auditfilelist.txt
   ```

3. 移動が必要な監査証跡ファイルをコピーする

   移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）に存在する全ファイルを，OS のcp コマンドで，監査証跡の出力先ディレクトリから監査証跡の保存先ディレクトリにコピーします。

   ```
   while read filename ; do
     cp ${filename} /mnt/audittrail/shorttimesavearea/audit_bak
   done < /home/adbmanager/tmp/auditfilelist.txt
   ```

4. コピーが完了した監査証跡ファイルを削除する

監査証跡ファイルのコピーが完了したら，OS のrm コマンドで，コピー元の監査証跡ファイルを監査証跡の出力先ディレクトリから削除します。

```
while read filename ; do
  rm ${filename}
done < /home/adbmanager/tmp/auditfilelist.txt
```

5. adbconvertaudittrailfile コマンドを実行する

   手順 2.で作成した移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を基に，移動した監査証跡ファイルすべてをadbconvertaudittrailfile コマンドで変換してください。

   なお，一度に複数の監査証跡ファイルを変換すると，**統一フォーマット用監査証跡ファイルの切り替え**によって，収集途中の監査証跡が消失するおそれがあります。この場合，JP1/Audit が一部の監査証跡を収集できないことがあります。

6. 移動が必要な監査証跡ファイルの一覧を削除する

   OS のrm コマンドで，手順 2.で作成した移動が必要な監査証跡ファイルの一覧（/home/adbmanager/tmp/auditfilelist.txt）を削除します。

```
rm /home/adbmanager/tmp/auditfilelist.txt
```

12.8.5　監査の実施

JP1/Audit が収集した監査証跡は，Web ブラウザ上から JP1/Audit の監査ログ管理画面で検索・集計できます。詳細については，マニュアル『JP1/Audit　構築・運用ガイド』を参照してください。

なお，adbconvertaudittrailfile コマンドは，JP1/Audit で表示できる監査証跡の情報だけを統一フォーマット用監査証跡ファイルに出力しています。そのため，JP1/Audit で確認できる内容では，監査に必要な情報が不足することがあります。例えば，実行した SQL 文や表のアクセス件数などは，JP1/Audit では確認できません。

JP1/Audit で確認した監査証跡について，さらに詳しい情報を参照したい場合は，JP1/Audit 上の［固有情報］に記録された監査証跡ファイル名を対象に，ADB_AUDITREAD 関数を指定した SQL 文を実行して，監査証跡を参照してください。ADB_AUDITREAD 関数を指定した SQL 文については，「12.3.3　監査証跡の参照（SELECT 文で監査証跡を参照する場合）」を参照してください。

12.9　監査対象イベントと表関数導出表の列構成

ここでは，監査対象イベントと表関数導出表の列構成について説明します。

- 「12.9.1　監査対象イベントの一覧と出力項目」
- 「12.9.2　監査証跡を検索するときの表関数導出表の列構成」
- 「12.9.3　監査証跡の出力契機と出力項目」

12.9.1　監査対象イベントの一覧と出力項目

ここでは，監査対象イベントの一覧と出力項目について説明します。

監査対象イベントの一覧と出力項目を次の表に示します。

表 12-9　監査対象イベントの一覧と出力項目

項番	イベントの分類	イベントの種類	イベントの内容
1	必須監査イベント	システムイベント	システムの開始，終了および変更を，監査対象にします。 次に示すイベントが実行されたときに，監査証跡が出力されます。 • HADB サーバの開始※1（adbstart コマンド） • HADB サーバの終了※2（adbstop コマンド） • HADB サーバの稼働モードの変更（adbchgsrvmode コマンド） • ノード種別の変更（adbchgnodetype コマンド） • SQL トレース情報の出力開始・停止（adbchgsqltrc コマンド） • クライアント定義の集中管理（adbclientdefmang コマンド） • DB エリアの追加・変更（adbmodarea コマンド） • バッファの変更（adbmodbuff コマンド） • 更新行のカラム化機能の管理（adbcolumnize コマンド）
2		監査イベント	監査に関する操作を，監査対象にします。 次に示すイベントが実行されたときに，監査証跡が出力されます。 • 監査管理権限の付与（GRANT 文のAUDIT ADMIN） • 監査参照権限の付与（GRANT 文のAUDIT VIEWER） • 監査管理権限の取り消し（REVOKE 文のAUDIT ADMIN） • 監査参照権限の取り消し（REVOKE 文のAUDIT VIEWER） • 監査対象の定義（CREATE AUDIT 文） • 監査対象定義情報の削除（DROP AUDIT 文） • 監査対象定義情報の参照（ディクショナリ表のSQL_AUDITS 表の検索） • ディクショナリ表のSQL_AUDITS 表に依存するビュー表の検索 • 監査人のパスワード変更（ALTER USER 文） • 監査証跡機能の有効化（--start オプションを指定したadbaudittrail コマンド）

項番	イベントの分類	イベントの種類	イベントの内容
			• 監査証跡機能の無効化（--stop オプションを指定したadbaudittrail コマンド） • 監査証跡ファイルの切り替え（--swap オプションを指定したadbaudittrail コマンド） • 監査証跡機能に関連する情報の参照（-d オプションを指定したadbaudittrail コマンド） • 監査証跡用のシステム定義関数の使用（ADB_AUDITREAD 関数） • ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索 • 監査証跡ファイルの変換（adbconvertaudittrailfile コマンド）
3	選択監査イベント	セッションイベント	HADB ユーザの認可識別子とパスワードによるユーザ認証を，監査対象にします。 次に示すイベントが実行されたときに，監査証跡が出力されます。 • HADB サーバへの接続 • HADB サーバからの切り離し
4		権限管理イベント	権限の付与または取り消しを，監査対象にします。 次に示すイベントが実行されたときに，監査証跡が出力されます。 • 権限の付与（GRANT 文） • 権限の取り消し（REVOKE 文） • HADB ユーザの作成（CREATE USER 文） • HADB ユーザの削除（DROP USER 文） • HADB ユーザの情報変更（ALTER USER 文）

5		定義系 SQL イベント	オブジェクトの定義，削除または変更を，監査対象にします。
			次に示すイベントが実行されたときに，監査証跡が出力されます。
			• オブジェクトの定義
			  • CREATE INDEX 文
			  • CREATE SCHEMA 文
			  • CREATE TABLE 文
			  • CREATE VIEW 文
			• オブジェクトの削除
			  • DROP INDEX 文
			  • DROP SCHEMA 文
			  • DROP TABLE 文
			  • DROP VIEW 文
			• オブジェクトの変更
			  • ALTER TABLE 文
			  • ALTER VIEW 文

6		操作系 SQL イベント	オブジェクトに対するアクセスを，監査対象にします。 次に示すイベントが実行されたときに，監査証跡が出力されます。 • 表の検索（SELECT 文）※3 • 表への行挿入（INSERT 文） • 表の行更新（UPDATE 文） • 表の行削除（DELETE 文）

項番	イベントの分類	イベントの種類	イベントの内容
			• 表の全行削除（TRUNCATE TABLE 文） • チャンク内の全行削除（PURGE CHUNK 文） • SQL 構文の解析エラー • チャンクに格納されているデータの取得（adbsql コマンドのサブコマンド#GETDATA） • チャンクに格納されているデータの件数の取得（adbsql コマンドのサブコマンド#GETCOUNT） • 表情報の表示（adbsql コマンドのサブコマンド#TABLES） • 列情報の表示（adbsql コマンドのサブコマンド#COLUMNS） • インデクス情報の表示（adbsql コマンドのサブコマンド#INDEXES） • チャンク情報の表示（adbsql コマンドのサブコマンド#CHUNKS） • 認可識別子の表示（adbsql コマンドのサブコマンド#GETUSER）
7		コマンド操作イベント※4	HADB サーバに接続するコマンドの実行を，監査対象にします。 次に示すイベントが実行されたときに，監査証跡が出力されます。 • データのインポート（adbimport コマンド） • インデクスの再作成（adbidxrebuild コマンド） • コスト情報の収集（adbgetcst コマンド） • データベースの状態解析（adbdbstatus コマンド） • データのエクスポート（adbexport コマンド） • チャンクのマージ（adbmergechunk コマンド） • チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド） • チャンクの状態変更（adbchgchunkstatus コマンド） • チャンクのアーカイブ（adbarchivechunk コマンド） • チャンクのアンアーカイブ（adbunarchivechunk コマンド） • システム表（実表）の再編成（adbreorgsystemdata コマンド） • 同義語辞書の登録・削除（adbsyndict コマンド）

注※1

HADB サーバを起動できた場合に，監査対象となります。

注※2

HADB サーバの終了は，--force オプションを指定したadbstop コマンドの実行以外が，監査対象となります。

注※3

SQL パラレル実行機能が適用された検索系 SQL の場合，SQL メインノードの監査証跡ファイルにだけ監査証跡が出力されます。SQL サブノードの監査証跡ファイルには監査証跡は出力されません。

注※4

adbsql コマンドは，コマンド操作イベントの対象外です。adbsql コマンドを通じて実行した操作に応じて，ほかの種類のイベントとして記録されます。

HADB ユーザが明示的に実行しない操作は，監査の対象外です。具体的には，次の情報は監査の対象外となります。

- adbsql コマンドのサブコマンドが内部的に実行する SQL 文
- コマンドが内部的に実行する SQL 文※
- 次に示す SQL 文にCASCADE を指定した場合または省略した場合に，CASCADE 指定によって再帰的に行われるオブジェクトの削除や権限の取り消し
  - DROP USER 文
  - DROP SCHEMA 文
  - DROP TABLE 文
  - REVOKE 文

注※

adbexport コマンドの-q オプションに指定した SQL 文は，監査対象となります。

12.9.2　監査証跡を検索するときの表関数導出表の列構成

ここでは，監査証跡を検索するときの表関数導出表の列構成について説明します。

監査証跡を検索するときの表関数導出表の列構成を次の表に示します。

表 12-10　監査証跡を検索するときの表関数導出表の列構成

項番	列名称	列のデータ型	説明	非ナル値制約
1	HADB_VERSION	CHAR(8)	イベントを受け付けた HADB サーバのバージョンが出力されます。	なし
2	AUDIT_TRAIL_TYPE	VARCHAR(32)	監査証跡を取得した契機が出力されます。 • EVENT（イベントの終了）	なし
3	EXEC_TIME	TIMESTAMP(6)	イベントが完了した時刻が出力されます。	なし
4	USER_NAME	VARCHAR(100)	イベントを実行した HADB ユーザの認可識別子が出力されます。	なし
5	EVENT_TYPE	VARCHAR(32)	イベントの種類が出力されます。 • SYSTEM（システムイベント） • AUDIT（監査イベント） • SESSION（セッションイベント） • PRIVILEGE（権限管理イベント） • DEFINE（定義系 SQL イベント） • MANIPULATE（操作系 SQL イベント） • COMMAND（コマンド操作イベント）	なし
6	EVENT_SUBTYPE	VARCHAR(64)	イベント種別が出力されます。	なし

項番	列名称	列のデータ型	説明	非ナル値制約
			詳細は「表 12-11　監査証跡に出力されるイベント種別の値」を参照してください。	
7	EVENT_RESULT	CHAR(10)	イベントが成功したかどうかが出力されます。 • SUCCESS（成功） • FAILURE（失敗） • OCCURRENCE（発生）※ 注※ イベント種別DISCONNECT のイベント結果として出力される場合があります。詳細については，「12.10.2　コマンドの実行時に出力される監査証跡についての留意事項」の「(2)　adbcancel コマンドを実行した場合の留意事項」を参照してください。	なし
8	USED_PRIVILEGE	VARCHAR(32)	AUDIT_TRAIL_TYPE 列にEVENT が出力されている場合は，NULL が出力されます。	なし
9	OS_USER_NAME	VARCHAR(256)	イベント発行元クライアントの OS アカウント名が出力されます。 OS アカウント名が識別できない場合は，NULL が出力されます。	なし
10	AP_NAME	VARCHAR(30)	イベント発行者の AP 識別子が出力されます。 HADB サーバに接続するコマンドを実行した場合は，コマンド名が AP 識別子として出力されます。 イベント発行者の AP 識別子を識別できない場合は，NULL が出力されます。	なし
11	CLIENT_IP_ADDRESS※	VARCHAR(46)	イベント発行元クライアントの IP アドレスが出力されます。 イベント発行元クライアントの IP アドレスを識別できない場合は，NULL が出力されます。	なし
12	CLIENT_PORT_NUMBER※	SMALLINT	イベント発行元クライアントのポート番号が出力されます。 イベント発行元クライアントのポート番号を識別できない場合は，NULL が出力されます。	なし
13	CLIENT_PROCESS_NAME	VARCHAR(255)	イベント発行元クライアントのプロセス名が出力されます。 次に該当する場合は，NULL が出力されます。 • イベント発行元クライアントが JDBC ドライバの場合 • イベント発行元クライアントのプロセス名を識別できない場合	なし
14	CLIENT_PROCESS_ID	SMALLINT	イベント発行元クライアントのプロセス ID が出力されます。	なし

項番	列名称	列のデータ型	説明	非ナル値制約
			イベント発行元クライアントのプロセス ID を識別できない場合は，NULL が出力されます。	
15	HADB_HOST_NAME	VARCHAR(255)	イベントを受け付けた HADB サーバのホスト名が出力されます。 HADB サーバのホスト名を識別できない場合は，NULL が出力されます。	なし
16	HADB_IP_ADDRESS	VARCHAR(46)	イベントを受け付けた HADB サーバの IP アドレスが出力されます。 HADB サーバの IP アドレスを識別できない場合は，NULL が出力されます。	なし
17	HADB_PORT_NUMBER	SMALLINT	イベントを受け付けた HADB サーバのポート番号が出力されます。 HADB サーバのポート番号を識別できない場合は，NULL が出力されます。	なし
18	HADB_PROCESS_ID	SMALLINT	イベントを受け付けた HADB サーバのプロセス ID が出力されます。 HADB サーバのプロセス ID を識別できない場合は，NULL が出力されます。	なし
19	ADBDIR	VARCHAR(118)	環境変数ADBDIR に指定したサーバディレクトリのパス名が出力されます。 指定したパス名の長さが 118 バイトを超えている場合は，118 バイトまでの内容が出力されます。	なし
20	HADB_NODE_NUMBER	SMALLINT	マルチノード機能を使用している場合，イベントを受け付けた HADB サーバのノード番号（1～12）が出力されます。SQL パラレル実行機能が適用された検索系 SQL に関するイベントのときは，SQL メインノードのノード番号が出力されます。 マルチノード機能を使用していない場合は，NULL が出力されます。	なし
21	CONNECTION_ID	SMALLINT	イベントを実行したコネクションのコネクション ID（1～1,024）が出力されます。 コネクション ID を識別できない場合は，NULL が出力されます。	なし
22	CONNECTION_NUMBER	INTEGER	イベントを実行したコネクションのコネクション通番（1～4,294,967,295）が出力されます。 コネクション通番を識別できない場合は，NULL が出力されます。	なし
23	STATEMENT_HANDLE	SMALLINT	イベントを実行する際に使用した文ハンドル（1～4,095）が出力されます。 文ハンドルを識別できない場合は，NULL が出力されます。	なし

項番	列名称	列のデータ型	説明	非ナル値制約
24	SQL_SERIAL_NUMBER	INTEGER	イベントの SQL 文通番（1～9,223,372,036,854,775,807）が出力されます。 SQL 文通番を識別できない場合は，NULL が出力されます。	なし
25	MESSAGE_LOG_INFO	CHAR(20)	イベントのメッセージログ情報が出力されます。 メッセージログ情報を識別できない場合は，NULL が出力されます。	なし
26	EXIT_STATUS	SMALLINT	AUDIT_TRAIL_TYPE 列にEVENT が出力されている場合は，次のどちらかが出力されます。 • SQL 文を実行した場合は，SQLCODE が出力されます。 • コマンドを実行した場合は，リターンコードが出力されます。	なし
27	OBJECT_TYPE	VARCHAR(32)	イベントの対象として指定されたオブジェクトの種別が出力されます。 • INDEX（インデクス） • SCHEMA（スキーマ） • TABLE（実表） • TABLE FUNCTION（表関数導出表） • VIEW（ビュー表） オブジェクトの種別を識別できない場合は，NULL が出力されます。 各種別の出力契機は「表 12-19　オブジェクトの種別が監査証跡に出力される契機」を参照してください。	なし
28	OBJECT_OWNER_NAME	VARCHAR(100)	イベントの対象として指定されたオブジェクトの所有者の認可識別子が出力されます。 オブジェクトの所有者を識別できない場合は，NULL が出力されます。	なし
29	OBJECT_SCHEMA_NAME	VARCHAR(100)	イベントの対象として指定されたオブジェクトのスキーマ名が出力されます。 オブジェクトのスキーマ名を識別できない場合は，NULL が出力されます。	なし
30	OBJECT_NAME	VARCHAR(100)	イベントの対象として指定されたオブジェクトの識別子が出力されます。 オブジェクトの識別子を識別できない場合は，NULL が出力されます。	なし
31	ACCESS_COUNT	INTEGER	イベントによって，オブジェクト（実表，ビュー表）に対してアクセスした行数が出力されます。次の行数が出力対象です。 • 検索した行数（データエクスポートも含む） • 挿入した行数（データインポートも含む） • 更新した行数	なし

項番	列名称	列のデータ型	説明	非ナル値制約
			• 削除した行数 アクセスした行数の取得に失敗した場合は，NULL が出力されます。 各操作で出力される行数の詳細は「表 12-20 ACCESS_COUNT 列に出力される行数」を参照してください。	
32	PRIVILEGE_TYPE	VARCHAR(32)	イベントによって，付与または取り消しの対象として指定された権限名が出力されます。 各権限の名称は「表 12-12　監査証跡に出力される権限名」を参照してください。 権限名を識別できない場合は，NULL が出力されます。	なし
33	PRIVILEGE_USER_NAME	VARCHAR(100)	次のどちらかに該当する HADB ユーザの認可識別子が出力されます。 • 権限の付与または取り消しの対象として，指定された HADB ユーザの認可識別子 • HADB ユーザの作成，HADB ユーザの削除，または HADB ユーザの情報変更の対象として，指定された HADB ユーザの認可識別子 HADB ユーザの認可識別子を識別できない場合は，NULL が出力されます。	なし
34	SQL_SOURCE	VARCHAR(64000)	イベントで実行した SQL 文が出力されます。 1 つの SQL 文で複数のオブジェクトを指定した場合，それぞれの監査証跡に SQL 文が出力されます。 SQL 文は 64,000 バイトまで出力され，それ以降は切り捨てられます。 実行した SQL 文がない場合は，NULL が出力されます。	なし
35	PARAM	VARCHAR(64000)	イベントで実行した SQL 文の？パラメタが出力されます。 1 つの SQL 文で複数のオブジェクトを指定した場合，それぞれの監査証跡に？パラメタが出力されます。 ？パラメタは，コンマ（ , ）で区切って表示されます。例を次に示します。 ？パラメタ,？パラメタ,？パラメタ,… ？パラメタは 64,000 バイトまで出力され，それ以降は切り捨てられます。 ？パラメタが指定されていない場合は，NULL が出力されます。 ？パラメタにバインドされたデータの各データ型の表示形式は，「表 12-13　？パラメタの出力形式」を参照してください。	なし
36	BEFORE_SYSTEM_INFO	VARCHAR(2000)	システム変更前の情報が出力されます。 次に示すイベントが成功したときに出力されます。 • adbstart	なし

項番	列名称	列のデータ型	説明	非ナル値制約
			HADB サーバの稼働モードが出力されます。出力される情報の詳細は，「■イベント（adbstart と adbchgsrvmode）の成功時に出力される値」を参照してください。 • adbchgsrvmode HADB サーバの稼働モードが出力されます。出力される情報の詳細は，「■イベント（adbstart と adbchgsrvmode）の成功時に出力される値」を参照してください。 • adbchgnodetype ノード種別が出力されます。出力される情報の詳細は，「■イベント（adbchgnodetype）の成功時に出力される値」を参照してください。 • adbchgsqltrc 有効になっているトレース情報とトレースレベルが出力されます。出力される情報の詳細は，「■イベント（adbchgsqltrc）の成功時に出力される値」を参照してください。 • adbclientdefmang クライアント定義ファイルと HADB ユーザの関連づけが出力されます。出力される情報の詳細は，「■イベント（adbclientdefmang）の成功時に出力される値」を参照してください。 • adbmodarea 更新前の DB エリア情報が出力されます。出力される情報の詳細は，「■イベント（adbmodarea）の成功時に出力される値」を参照してください。 • adbmodbuff ローカル作業表用バッファのページ数が出力されます。出力される情報の詳細は，「■イベント（adbmodbuff）の成功時に出力される値」を参照してください。 • adbcolumnize 更新行のカラム化機能の状態に関する情報が出力されます。出力される情報の詳細は，「■イベント（adbcolumnize）の成功時に出力される値」を参照してください。 上記以外のイベントの場合は，NULL が出力されます。 システム変更前の情報は 2,000 バイトまで出力され，それ以降は切り捨てられます。 なお，システム変更がない場合は，「イベント実行時のシステム情報」が出力されます。そのため，BEFORE_SYSTEM_INFO 列とAFTER_SYSTEM_INFO 列で，同じ内容が出力されます。	
37	AFTER_SYSTEM_INFO	VARCHAR(2000)	システム変更後の情報が出力されます。 次に示すイベントが成功したときに出力されます。	なし

項番	列名称	列のデータ型	説明	非ナル値制約
			• adbstart HADB サーバの稼働モードが出力されます。出力される情報の詳細は，「■イベント（adbstart と adbchgsrvmode）の成功時に出力される値」を参照してください。 • adbchgsrvmode HADB サーバの稼働モードが出力されます。出力される情報の詳細は，「■イベント（adbstart と adbchgsrvmode）の成功時に出力される値」を参照してください。 • adbchgnodetype ノード種別が出力されます。出力される情報の詳細は，「■イベント（adbchgnodetype）の成功時に出力される値」を参照してください。 • adbchgsqltrc 有効になっているトレース情報とトレースレベルが出力されます。出力される情報の詳細は，「■イベント（adbchgsqltrc）の成功時に出力される値」を参照してください。 • adbclientdefmang クライアント定義ファイルと HADB ユーザの関連づけが出力されます。出力される情報の詳細は，「■イベント（adbclientdefmang）の成功時に出力される値」を参照してください。 • adbmodarea 更新後の DB エリア情報が出力されます。出力される情報の詳細は，「■イベント（adbmodarea）の成功時に出力される値」を参照してください。 • adbmodbuff ローカル作業表用バッファのページ数が出力されます。出力される情報の詳細は，「■イベント（adbmodbuff）の成功時に出力される値」を参照してください。 • adbcolumnize 更新行のカラム化機能の状態に関する情報が出力されます。出力される情報の詳細は，「■イベント（adbcolumnize）の成功時に出力される値」を参照してください。 上記以外のイベントの場合は，NULL が出力されます。 システムの変更情報は 2,000 バイトまで出力され，それ以降は切り捨てられます。 なお，システム変更がない場合は，「イベント実行時のシステム情報」が出力されます。そのため，BEFORE_SYSTEM_INFO 列とAFTER_SYSTEM_INFO 列で，同じ内容が出力されます。	

項番	列名称	列のデータ型	説明	非ナル値制約
38	SERVER_OPERAND	VARCHAR(4000)	HADB サーバの開始時に，サーバ定義に指定されているオペランドの値が出力されます。 各サーバ定義は，スラッシュ（/）で区切って出力されます。 サーバ定義の情報は 4,000 バイトまで出力され，それ以降は切り捨てられます。 HADB サーバの開始時以外（EVENT_SUBTYPE 列に ADBSTART と出力されたとき以外）は，NULL が出力されます。 SERVER_OPERAND 列に出力されるサーバ定義の形式は，「表 12-14　サーバ定義の出力形式」を参照してください。	なし
39	USER_INFO_1	VARCHAR(100)	HADB ユーザが任意に設定した付加情報が出力されます。 次に示す場合は，NULL が出力されます。 • HADB ユーザが設定した付加情報を識別できない場合 • HADB ユーザが付加情報を設定していない場合	なし
40	USER_INFO_2	VARCHAR(100)		なし
41	USER_INFO_3	VARCHAR(100)		なし

注※

HADB サーバと同一マシンの HADB クライアントで実行された SQL 文の監査証跡については，次の情報が出力されます。

- IP アドレス：127.0.0.1
- ポート番号：NULL

マルチノード機能を使用している場合，HADB サーバと同一マシンの HADB クライアントで実行された SQL 文の監査証跡でも，SQL 文が処理されたノードによって，出力される情報が変わることがあります。

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のEVENT_SUBTYPE 列に出力されるイベント種別の値を，次の表に示します。

表 12-11　監査証跡に出力されるイベント種別の値

項番	イベントの分類	イベントの種類	イベントの種別	監査証跡に出力されるイベント種別の値
1	必須監査イベント	システムイベント	HADB サーバの開始（adbstart コマンド）	ADBSTART
2			HADB サーバの終了（adbstop コマンド）	ADBSTOP
3			HADB サーバの稼働モードの変更（adbchgsrvmode コマンド）	ADBCHGSRVMODE
4			ノード種別の変更（adbchgnodetype コマンド）	ADBCHGNODETYPE

項番	イベントの分類	イベントの種類	イベントの種別	監査証跡に出力されるイベント種別の値
5			SQL トレース情報の出力開始・停止（adbchgsqltrc コマンド）	ADBCHGSQLTRC
6			クライアント定義の集中管理（adbclientdefmang コマンド）	ADBCLIENTDEFMANG
7			DB エリアの追加・変更（adbmodarea コマンド）	ADBMODAREA
8			バッファの変更（adbmodbuff コマンド）	ADBMODBUFF
9			更新行のカラム化機能の管理（adbcolumnize コマンド）	ADBCOLUMNIZE
10		監査イベント	監査管理権限の付与（GRANT 文のAUDIT ADMIN）	GRANT
11			監査参照権限の付与（GRANT 文のAUDIT VIEWER）	
12			監査管理権限の取り消し（REVOKE 文のAUDIT ADMIN）	REVOKE
13			監査参照権限の取り消し（REVOKE 文のAUDIT VIEWER）	
14			監査対象の定義（CREATE AUDIT 文）	CREATE AUDIT
15			監査対象定義の削除（DROP AUDIT 文）	DROP AUDIT
16			監査人のパスワード変更（ALTER USER 文）	ALTER USER
17			監査証跡機能の有効化（--start オプションを指定したadbaudittrail コマンド）	ADBAUDITTRAIL START
18			監査証跡機能の無効化（--stop オプションを指定したadbaudittrail コマンド）	ADBAUDITTRAIL STOP
19			監査証跡ファイルの切り替え（--swap オプションを指定したadbaudittrail コマンド）	ADBAUDITTRAIL SWAP
20			監査証跡機能に関連する情報の参照（-d オプションを指定したadbaudittrail コマンド）	ADBAUDITTRAIL DISPLAY
21			• 監査対象定義情報の参照（ディクショナリ表のSQL_AUDITS 表の検索） • ディクショナリ表のSQL_AUDITS 表に依存するビュー表の検索	SELECT
22				ADBEXPORT
23			• 監査証跡用のシステム定義関数の使用（ADB_AUDITREAD 関数） • ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索	SELECT
24				ADBEXPORT
25			監査証跡ファイルの変換（adbconvertaudittrailfile コマンド）	ADBCONVERTAUDITTRAILFILE
26	選択監査イベント	セッションイベント	HADB サーバへの接続	CONNECT
27			HADB サーバからの切り離し	DISCONNECT

項番	イベントの分類	イベントの種類	イベントの種別	監査証跡に出力されるイベント種別の値
28		権限管理イベント	DBA 権限の付与（GRANT 文のDBA）	GRANT
29			CONNECT 権限の付与（GRANT 文のCONNECT）	
30			スキーマ定義権限の付与（GRANT 文のSCHEMA）	
31			SELECT 権限の付与（GRANT 文のSELECT）	
32			INSERT 権限の付与（GRANT 文のINSERT）	
33			UPDATE 権限の付与（GRANT 文のUPDATE）	
34			DELETE 権限の付与（GRANT 文のDELETE）	
35			TRUNCATE 権限の付与（GRANT 文のTRUNCATE）	
36			REFERENCES 権限の付与（GRANT 文のREFERENCES）	
37			IMPORT TABLE 権限の付与（GRANT 文のIMPORT TABLE）	
38			REBUILD INDEX 権限の付与（GRANT 文のREBUILD INDEX）	
39			GET COSTINFO 権限の付与（GRANT 文のGET COSTINFO）	
40			EXPORT TABLE 権限の付与（GRANT 文のEXPORT TABLE）	
41			MERGE CHUNK 権限の付与（GRANT 文のMERGE CHUNK）	
42			CHANGE CHUNK COMMENT 権限の付与（GRANT 文のCHANGE CHUNK COMMENT）	
43			CHANGE CHUNK STATUS 権限の付与（GRANT 文のCHANGE CHUNK STATUS）	
44			ARCHIVE CHUNK 権限の付与（GRANT 文のARCHIVE CHUNK）	
45			UNARCHIVE CHUNK 権限の付与（GRANT 文のUNARCHIVE CHUNK）	
46			DBA 権限の取り消し（REVOKE 文のDBA）	REVOKE
47			CONNECT 権限の取り消し（REVOKE 文のCONNECT）	
48			スキーマ定義権限の取り消し（REVOKE 文のSCHEMA）	
49			SELECT 権限の取り消し（REVOKE 文のSELECT）	
50			INSERT 権限の取り消し（REVOKE 文のINSERT）	
51			UPDATE 権限の取り消し（REVOKE 文のUPDATE）	
52			DELETE 権限の取り消し（REVOKE 文のDELETE）	
53			TRUNCATE 権限の取り消し（REVOKE 文のTRUNCATE）	
54			REFERENCES 権限の取り消し（REVOKE 文のREFERENCES）	
55			IMPORT TABLE 権限の取り消し（REVOKE 文のIMPORT TABLE）	

項番	イベントの分類	イベントの種類	イベントの種別	監査証跡に出力されるイベント種別の値
56			REBUILD INDEX 権限の取り消し（REVOKE 文のREBUILD INDEX）	
57			GET COSTINFO 権限の取り消し（REVOKE 文のGET COSTINFO）	
58			EXPORT TABLE 権限の取り消し（REVOKE 文のEXPORT TABLE）	
59			MERGE CHUNK 権限の取り消し（REVOKE 文のMERGE CHUNK）	
60			CHANGE CHUNK COMMENT 権限の取り消し（REVOKE 文の CHANGE CHUNK COMMENT）	
61			CHANGE CHUNK STATUS 権限の取り消し（REVOKE 文の CHANGE CHUNK STATUS）	
62			ARCHIVE CHUNK 権限の取り消し（REVOKE 文のARCHIVE CHUNK）	
63			UNARCHIVE CHUNK 権限の取り消し（REVOKE 文のUNARCHIVE CHUNK）	
64			HADB ユーザの作成（CREATE USER 文）	CREATE USER
65			HADB ユーザの削除（DROP USER 文）	DROP USER
66			HADB ユーザの情報変更（ALTER USER 文）	ALTER USER
67		定義系 SQL イベント	インデクスの定義（CREATE INDEX 文）	CREATE INDEX
68			スキーマの定義（CREATE SCHEMA 文）	CREATE SCHEMA
69			実表の定義（CREATE TABLE 文）	CREATE TABLE
70			ビュー表の定義（CREATE VIEW 文）	CREATE VIEW
71			インデクスの削除（DROP INDEX 文）	DROP INDEX
72			スキーマの削除（DROP SCHEMA 文）	DROP SCHEMA
73			実表の削除（DROP TABLE 文）	DROP TABLE
74			ビュー表の削除（DROP VIEW 文）	DROP VIEW
75			表定義の変更（ALTER TABLE 文）	ALTER TABLE
76			ビュー定義の変更（ALTER VIEW 文）	ALTER VIEW
77		操作系 SQL イベント	表の検索（SELECT 文）	SELECT
78			表への行挿入（INSERT 文）	INSERT
79			表の行更新（UPDATE 文）	UPDATE
80			表の行削除（DELETE 文）	DELETE
81			表の全行削除（TRUNCATE TABLE 文）	TRUNCATE TABLE

項番	イベントの分類	イベントの種類	イベントの種別	監査証跡に出力されるイベント種別の値
82			チャンク内の全行削除（PURGE CHUNK 文）	PURGE CHUNK
83			SQL 構文の解析エラー	UNKNOWN
84			チャンクに格納されているデータの取得（adbsql コマンドのサブコマンド#GETDATA）	GETDATA
85			チャンクに格納されているデータの件数の取得（adbsql コマンドのサブコマンド#GETCOUNT）	GETCOUNT
86			表情報の表示（adbsql コマンドのサブコマンド#TABLES）	TABLES
87			列情報の表示（adbsql コマンドのサブコマンド#COLUMNS）	COLUMNS
88			インデクス情報の表示（adbsql コマンドのサブコマンド#INDEXES）	INDEXES
89			チャンク情報の表示（adbsql コマンドのサブコマンド#CHUNKS）	CHUNKS
90			認可識別子の表示（adbsql コマンドのサブコマンド#GETUSER）	GETUSER
91		コマンド操作イベント	データのインポート（adbimport コマンド）	ADBIMPORT
92			インデクスの再作成（adbidxrebuild コマンド）	ADBIDXREBUILD
93			コスト情報の収集（adbgetcst コマンド）	ADBGETCST
94			データベースの状態解析（adbdbstatus コマンド）	ADBDBSTATUS
95			データのエクスポート（adbexport コマンド）	ADBEXPORT
96			チャンクのマージ（adbmergechunk コマンド）	ADBMERGECHUNK
97			チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド）	ADBCHGCHUNKCOMMENT
98			チャンクの状態変更（adbchgchunkstatus コマンド）	ADBCHGCHUNKSTATUS
99			チャンクのアーカイブ（adbarchivechunk コマンド）	ADBARCHIVECHUNK
100			チャンクのアンアーカイブ（adbunarchivechunk コマンド）	ADBUNARCHIVECHUNK
101			システム表（実表）の再編成（adbreorgsystemdata コマンド）	ADBREORGSYSTEMDATA
102			同義語辞書の登録・削除（adbsyndict コマンド）	ADBSYNDICT

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のPRIVILEGE_TYPE 列に出力される**権限名**の値を，次の表に示します。

表 12-12　監査証跡に出力される権限名

項番	権限の種別	監査証跡に出力される権限名
1	DBA 権限	DBA
2	CONNECT 権限	CONNECT
3	スキーマ定義権限	SCHEMA
4	監査管理権限	AUDIT ADMIN
5	監査参照権限	AUDIT VIEWER
6	SELECT 権限	SELECT
7	INSERT 権限	INSERT
8	UPDATE 権限	UPDATE
9	DELETE 権限	DELETE
10	TRUNCATE 権限	TRUNCATE
11	REFERENCES 権限	REFERENCES
12	IMPORT TABLE 権限	IMPORT TABLE
13	REBUILD INDEX 権限	REBUILD INDEX
14	GET COSTINFO 権限	GET COSTINFO
15	EXPORT TABLE 権限	EXPORT TABLE
16	MERGE CHUNK 権限	MERGE CHUNK
17	CHANGE CHUNK COMMENT 権限	CHANGE CHUNK COMMENT
18	CHANGE CHUNK STATUS 権限	CHANGE CHUNK STATUS
19	ARCHIVE CHUNK 権限	ARCHIVE CHUNK
20	UNARCHIVE CHUNK 権限	UNARCHIVE CHUNK

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のPARAM 列に出力される？パラメタにバインドされたデータの各データ型の表示形式を，次の表に示します。

表 12-13　？パラメタの出力形式

項番	データ型	出力形式
1	INTEGER	10 進数形式
2	DECIMAL(*m*,*n*)	10 進数形式
3	NUMERIC(*m*,*n*)	
4	SMALLINT	10 進数形式
5	DOUBLE PRECISION	仮数（10 進数）E 指数（10 進数）

項番	データ型	出力形式
6	FLOAT	
7	CHARACTER(*n*)	'文字列形式'
8	VARCHAR(*n*)	'文字列形式'※ 実長 0 の場合は，''として出力する
9	DATE	日付データの既定の出力表現
10	TIME(*p*)	時刻データの既定の出力表現
11	TIMESTAMP(*p*)	時刻印データの既定の出力表現
12	BINARY(*n*)	X'16 進数値'
13	VARBINARY(*n*)	X'16 進数値' 実長 0 の場合は，X''として出力する
14	ROW	X'16 進数値'

（凡例）

m，*n*，*p*：正の整数

注

？パラメタにバインドされたデータがナル値の場合は，PARAM 列にNULL が出力されます。

注※

NULL 述語の値式に？パラメタを単独で指定した場合に，？パラメタにナル値以外の値を指定すると，指定した値に関係なく'*'が出力されます。

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のSERVER_OPERAND 列に出力されるサーバ定義の出力形式を，次の表に示します。

表 12-14　サーバ定義の出力形式

項番	形式	出力形式
1	set 形式	*オペランド名 ＝指定値*※1
2	コマンド形式	*コマンド名 オプション名 指定値*[, *オプション名 指定値* …]※2

注※1

「＝*指定値*」部分の最大長は 1,024 バイトです。それ以上長い場合は 1,024 バイトで打ち切られます。

注※2

「*オプション名 指定値*[, *オプション名 指定値* …]」部分の最大長は 1,024 バイトです。それ以上長い場合は 1,024 バイトで打ち切られます。

また，サーバ定義の指定例とSERVER_OPERAND 列の出力例を示します。指定が有効になっているサーバ定義のオペランド名と指定値の組み合わせごとに，スラッシュ（/）で区切られて情報が出力されます。

■サーバ定義の指定例

```
set adb_db_path = XXXXX
set adb_rpc_port = YYYYY
set adb_sys_max_users = 10
set adb_sys_rthd_num = 40
set adb_sys_uthd_num = 128
set adb_sql_exe_max_rthd_num = 4
set adb_sys_memory_limit = 64000
adbbuff -g TBLBUF01 -n ADBUTBL01 -p 1000000 -v 1024
adbbuff -g IDXBUF01 ¥
        -n ADBUIDX01 ¥
        -p 2500000
adbcltgrp -g group -m 10 -u 0 -r 40 -e 0 -w 0
```

■SERVER_OPERAND 列の出力例

```
adb_db_path = XXXXX / adb_rpc_port = YYYYY / adb_sys_max_users = 10 / adb_sys_rthd_num =
40 / adb_sys_uthd_num = 128 / adb_sql_exe_max_rthd_num = 4 /
adb_sys_memory_limit = 64000 / adbbuff -g TBLBUF01 -n ADBUTBL01 -p 1000000 -v 1024 / adbb
uff -g IDXBUF01 -n ADBUIDX01 -p 2500000 / adbcltgrp -g group -m 10 -u 0 -r 40 -e 0 -w 0
```

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のBEFORE_SYSTEM_INFO 列，および AFTER_SYSTEM_INFO 列に出力される値を，イベントごとに次に示します。

■イベント（adbstart とadbchgsrvmode）の成功時に出力される値

イベント（adbstart とadbchgsrvmode）の成功時には，次の表に示す各稼働モードの値が出力されます。

表 12-15　稼働モードの値

項番	稼働モード	値
1	通常モード	NORMAL
2	静止モード	QUIESCENCE
3	オフラインモード	OFFLINE
4	メンテナンスモード	MAINTENANCE

- マルチノード機能を使用している場合

  プライマリノードでadbstart コマンドを実行した場合，BEFORE_SYSTEM_INFO 列には，前回起動時にプライマリノードだったノードの HADB サーバの稼働モードが出力されます。

  セカンダリノードまたはワーカーノードでadbstart コマンドを実行した場合，BEFORE_SYSTEM_INFO 列には，常にNULL が出力されます。

■イベント（adbchgnodetype）の成功時に出力される値

イベント（adbchgnodetype）の成功時には，次の表に示す各ノード種別の値が出力されます。

表 12-16　ノード種別の値

項番	ノード種別	値
1	プライマリノード	PRIMARY

項番	ノード種別	値
2	セカンダリノード	SECONDARY

■イベント（adbchgsqltrc）の成功時に出力される値

イベント（adbchgsqltrc）の成功時には，次の 2 つの表に示すトレース情報とトレースレベルの値が出力されます。

表 12-17　トレース情報の値

項番	トレース情報	値
1	アクセスパス情報	accesspath
2	？パラメタ情報	param

表 12-18　トレースレベルの値

項番	トレースレベル	値
1	SQL 文単位で出力	sql
2	コール単位で出力	call

イベント（adbchgsqltrc）の成功時には，有効になっているトレース情報とトレースレベルが，トレース情報，トレースレベルの順にスラッシュ（/）で区切られて出力されます。なお，SQL トレース機能が無効の場合は，空文字が出力されます。

アクセスパスが有効で，トレースレベルが SQL 文単位となっている場合の出力例を次に示します。

```
accesspath / sql
```

■イベント（adbclientdefmang）の成功時に出力される値

イベント（adbclientdefmang）の成功時には，「表 12-14　サーバ定義の出力形式」のコマンド形式に従って，各クライアント管理定義がスラッシュ（/）で区切られて出力されます。クライアント定義の集中管理機能が無効な場合は，空文字が出力されます。

クライアント管理定義の指定例と出力例を，次に示します。

- クライアント管理定義の指定例

```
adbclientmang -f client01.def -i USER01, USER02
adbclientmang -f client02.def -i USER03
```

- 出力例

```
adbclientmang -f client01.def -i USER01, USER02 / adbclientmang -f client02.def -i USE
R03
```

■イベント（adbmodarea）の成功時に出力される値

イベント（adbmodarea）の成功時には，次の形式で値が出力されます。

```
AREA NAME = DBエリア名, FILE NUMBER = DBエリアファイル数
```

説明

DB エリア名

DB エリア名が出力されます。指定値，または正規化済み文字列のどちらかが出力されます。

DB エリアファイル数

DB エリアファイル数が出力されます。ファイル数が不明の場合は，?が出力されます。DB エリアが存在しない場合は，0 が出力されます。

DB エリア追加・変更オプションファイルの指定例と出力例を，次に示します。

- DB エリア追加・変更オプションファイルの指定例

```
adbaddarea -n ADBUTBL02
```

- 出力例（BEFORE_SYSTEM_INFO 列）

```
AREA NAME = ADBUTBL02, FILE NUMBER = 0
```

- 出力例（AFTER_SYSTEM_INFO 列）

```
AREA NAME = ADBUTBL02, FILE NUMBER = 1
```

■イベント（adbmodbuff）の成功時に出力される値

イベント（adbmodbuff）の成功時には，adbmodbuff コマンドで変更したローカル作業表用バッファのページ数（5～100,000,000）が出力されます。

adbmodbuff コマンドのバッファ変更オプションファイルの指定例と出力例を，次に示します。

- バッファ変更オプションファイルの指定例

```
set adb_dbbuff_wrktbl_clt_blk_num = 256
```

- 出力例

```
256
```

■イベント（adbcolumnize）の成功時に出力される値

イベント（adbcolumnize）の成功時には，更新行のカラム化機能の状態に関する情報が出力されます。

- 更新行のカラム化機能が有効の場合：ACTIVE
- 更新行のカラム化機能が無効の場合：INACTIVE

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のOBJECT_TYPE 列に出力されるオブジェクトの種別が出力される契機を，次の表に示します。

表 12-19　オブジェクトの種別が監査証跡に出力される契機

項番	オブジェクトの種別	出力契機
1	INDEX※1	• インデクスの定義・削除 • インデクスの再作成（adbidxrebuild コマンド）
2	SCHEMA	スキーマの定義・削除
3	TABLE	• 実表の定義

項番	オブジェクトの種別	出力契機
		• 実表の定義の変更 • 実表の参照（データエクスポートも含む） • 実表の更新（データインポートも含む） • 実表の削除 • アクセス権限の付与・取り消し • コスト情報の収集（adbgetcst コマンド） • チャンクのマージ（adbmergechunk コマンド） • チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド） • チャンクの状態変更（adbchgchunkstatus コマンド） • チャンクのアーカイブ（adbarchivechunk コマンド） • チャンクのアンアーカイブ（adbunarchivechunk コマンド） • システム表（実表）の再構成（adbreorgsystemdata コマンド）
4	TABLE FUNCTION※2	表関数導出表の参照（データエクスポートも含む）
5	VIEW	• ビュー表の定義 • ビュー表の再作成 • ビュー表の参照（データエクスポートも含む） • ビュー表の更新 • ビュー表の削除 • アクセス権限の付与・取り消し

注※1

表の検索で利用したインデクスの情報は出力されません。

注※2

オブジェクトの識別子は，表関数の関数名となります。表関数の引数は出力されません。

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のACCESS_COUNT 列に出力される**各操作で出力される行数**を，次の表に示します。

表 12-20　ACCESS_COUNT 列に出力される行数

項番	イベント	操作によって出力される監査証跡の EVENT_SUBTYPE	出力される行数
1	SELECT 文	SELECT	検索結果の行数※1，※2，※3
2	INSERT 文	INSERT	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のOBJECT_NAME 列に出力された表に挿入した行数※3
3		SELECT	NULL※2
4	UPDATE 文	UPDATE	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のOBJECT_NAME 列に出力された表を更新した行数※3
5		SELECT	NULL※2

項番	イベント	操作によって出力される監査証跡のEVENT_SUBTYPE	出力される行数
6	DELETE 文	DELETE	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のOBJECT_NAME 列に出力された表から削除した行数※3
7		SELECT	NULL※2
8	TRUNCATE TABLE 文	TRUNCATE TABLE	0
9	PURGE CHUNK 文	PURGE CHUNK	0
10		SELECT	NULL※2
11	定義系 SQL	それぞれのイベント種別	0※4
12	adbimport コマンド	ADBIMPORT	「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のOBJECT_NAME 列に出力された表にデータインポートした行数※3
13	adbexport コマンド	ADBEXPORT	データエクスポートした行数※2, ※3
14	上記以外	それぞれのイベント種別	NULL

注※1

カーソルをクローズした時点で，HADB サーバ側でFETCH 処理が完了している行数が出力されます。そのため，検索結果の一括転送によって，HADB クライアント側でFETCH 処理を行っている行数ではなく，次の計算式で求めた行数が出力されることがあります。その場合，HADB クライアント側でFETCH 処理を行っている行数よりも大きな値になります。

一括転送行数 × 2 − 1

注※2

1 つの SQL 文または 1 つのコマンドを実行した場合に，「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のEVENT_SUBTYPE 列の内容が同一である監査証跡が複数出力されたときは，それらの監査証跡のACCESS_COUNT 列には同じ値が出力されます。

注※3

「表 12-10　監査証跡を検索するときの表関数導出表の列構成」のEVENT_RESULT 列がFAILURE の場合，0 が出力されます。

注※4

1 つの定義系 SQL を実行した場合に，複数の監査証跡が出力されたときは，それらの監査証跡のACCESS_COUNT 列には同じ値（0）が出力されます。

各イベントの監査証跡の出力契機と出力項目については，「12.9.3　監査証跡の出力契機と出力項目」を参照してください。

12.9.3　監査証跡の出力契機と出力項目

ここでは，監査証跡の出力契機と出力項目について説明します。

監査証跡の出力契機と出力項目を，次の表に示します。表の「監査証跡用の表関数導出表の列構成」列が横に長いため，ここでは５つの表に分割して記載しています。

- 「表 12-21　監査証跡の出力契機と出力項目（その 1）」
- 「表 12-22　監査証跡の出力契機と出力項目（その 2）」
- 「表 12-23　監査証跡の出力契機と出力項目（その 3）」
- 「表 12-24　監査証跡の出力契機と出力項目（その 4）」
- 「表 12-25　監査証跡の出力契機と出力項目（その 5）」

上記の５つの表にある「監査イベント」列の詳細については，「12.9.1　監査対象イベントの一覧と出力項目」の「表 12-9　監査対象イベントの一覧と出力項目」を参照してください。

上記の５つの表にある「監査証跡用の表関数導出表の列構成」列の詳細については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」の「表 12-10　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

また，５つの表の凡例と注釈は，「表 12-25　監査証跡の出力契機と出力項目（その 5）」の末尾にまとめて記載しています。

表 12-21　監査証跡の出力契機と出力項目（その 1）

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン（HADB_VERSION）	監査証跡の種別（AUDIT_TRAIL_TYPE）	イベント実行日時（EXEC_TIME）	イベント実行ユーザ（USER_NAME）	イベントの種類（EVENT_TYPE）	イベント種別（EVENT_SUBTYPE）
1	必須監査イベント	システムイベント	HADB サーバの開始（adbstart コマンド）		○	EVENT	○		SYSTEM	ADBSTART
2			HADB サーバの終了（adbstop コマンド）	正常終了	○	EVENT	○		SYSTEM	ADBSTOP
3			HADB サーバの稼働モードの変更（adbchgsrvmode コマンド）	正常終了	○	EVENT	○		SYSTEM	ADBCHGSRVMODE
4			ノード種別の変更	正常終了	○	EVENT	○		SYSTEM	ADBCHGNODETYPE

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
			(adbchgnodetype コマンド)							
5			SQL トレース情報の出力開始・停止 (adbchgsqltrc コマンド)	正常終了	○	EVENT	○		SYSTEM	ADBCHGSQLTRC
6			クライアント定義の集中管理 (--update オプションを指定した adbclientdefmang コマンド)	正常終了 ※2	○	EVENT	○	○	SYSTEM	ADBCLIENTDEFMANG
7			クライアント定義の集中管理 (-i オプションを指定した adbclientdefmang コマンド)	正常終了 ※2	○	EVENT	○	○	SYSTEM	ADBCLIENTDEFMANG
8			DB エリアの追加・変更 (adbmodarea コマンド)	正常終了 ※2	○	EVENT	○		SYSTEM	ADBMODAREA
9			バッファの変更 (adbmodbuff コマンド)	正常終了 ※2	○	EVENT	○		SYSTEM	ADBMODBUFF
10			更新行のカラム化機能の管理 (adbcolumnize コマンド)	正常終了	○	EVENT	○		SYSTEM	ADBCOLUMNIZE
11		監査イベント	監査管理権限の付与 (GRANT 文の AUDIT ADMIN)	正常終了 ※6, ※7	○	EVENT	○	○	AUDIT	GRANT
12			監査参照権限の付与 (GRANT 文の AUDIT VIEWER)	正常終了 ※6, ※7	○	EVENT	○	○	AUDIT	GRANT

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
13			監査管理権限の取り消し（REVOKE 文の AUDIT ADMIN）	正常終了 ※6，※7	○	EVENT	○	○	AUDIT	REVOKE
14			監査参照権限の取り消し（REVOKE 文の AUDIT VIEWER）	正常終了 ※6，※7	○	EVENT	○	○	AUDIT	REVOKE
15			監査対象の定義（CREATE AUDIT 文）	正常終了	○	EVENT	○	○	AUDIT	CREATE AUDIT
16			監査対象定義情報の削除（DROP AUDIT 文）	正常終了	○	EVENT	○	○	AUDIT	DROP AUDIT
17			監査人のパスワード変更（ALTER USER 文）	正常終了	○	EVENT	○	○	AUDIT	ALTER USER
18			監査証跡機能の有効化（--start オプションを指定した adbauditttrail コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBAUDITTRAIL START
19			監査証跡機能の無効化（--stop オプションを指定した adbauditttrail コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBAUDITTRAIL STOP
20			監査証跡ファイルの切り替え（--swap オプションを指定した adbauditttrail コマンド）	正常終了 ※2，※9	○	EVENT	○	○	AUDIT	ADBAUDITTRAIL SWAP

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
21			監査証跡機能に関連する情報の参照（-d オプションを指定した adbaudittrail コマンド）	正常終了 ※2, ※9	○	EVENT	○	○	AUDIT	ADBAUDITTRAILDISPLAY
22			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（操作系 SQL）	正常終了	○	EVENT	○	○	AUDIT	SELECT
23			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（操作系 SQL）	正常終了	○	EVENT	○	○	AUDIT	SELECT
24			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（-q オプションを指定したadbexport コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBEXPORT
25			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBEXPORT
26			ディクショナリ表のSQL_AUDITS 表の検索（操作系 SQL）	正常終了	○	EVENT	○	○	AUDIT	SELECT

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
27			ディクショナリ表のSQL_AUDITS表に依存するビュー表の検索（操作系 SQL）	正常終了	○	EVENT	○	○	AUDIT	SELECT
28			ディクショナリ表のSQL_AUDITS表の検索（-q オプションを指定していない adbexport コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBEXPORT
29			ディクショナリ表のSQL_AUDITS表に依存するビュー表の検索（-q オプションを指定していないadbexport コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBEXPORT
30			ディクショナリ表のSQL_AUDITS表の検索（-q オプションを指定したadbexport コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBEXPORT
31			ディクショナリ表のSQL_AUDITS表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）	正常終了 ※2	○	EVENT	○	○	AUDIT	ADBEXPORT
32			監査証跡ファイルの変換（adbconvertaudittrailfile コマンド）	正常終了 ※2, ※9	○	EVENT	○	○	AUDIT	ADBCONVERTAUDITTRAILFILE

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
33	選択監査イベント	セッションイベント	HADB サーバへの接続	正常終了	○	EVENT	○	○	SESSION	CONNECT
34			HADB サーバからの切り離し	正常終了※1	○	EVENT	○	○	SESSION	DISCONNECT
35		権限管理イベント	DBA 権限の付与（GRANT 文のDBA）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
36			CONNECT 権限の付与（GRANT 文のCONNECT）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
37			SCHEMA 権限の付与（GRANT 文のSCHEMA）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
38			SELECT 権限の付与（GRANT 文のSELECT）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
39			INSERT 権限の付与（GRANT 文のINSERT）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
40			UPDATE 権限の付与（GRANT 文のUPDATE）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
41			DELETE 権限の付与（GRANT 文のDELETE）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
42			TRUNCATE 権限の付与（GRANT 文のTRUNCATE）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
43			REFERENCES 権限の付与（GRANT 文のREFERENCES）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT
44			IMPORT TABLE 権限の付与（GRANT 文のIMPORT TABLE）	正常終了※6	○	EVENT	○	○	PRIVILEGE	GRANT

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
45			REBUILD INDEX 権限の付与（GRANT 文の REBUILD INDEX）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
46			GET COSTINFO 権限の付与（GRANT 文のGET COSTINFO）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
47			EXPORT TABLE 権限の付与（GRANT 文のEXPORT TABLE）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
48			MERGE CHUNK 権限の付与（GRANT 文のMERGE CHUNK）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
49			CHANGE CHUNK COMMENT 権限の付与（GRANT 文のCHANGE CHUNK COMMENT）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
50			CHANGE CHUNK STATUS 権限の付与（GRANT 文の CHANGE CHUNK STATUS）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
51			ARCHIVE CHUNK 権限の付与（GRANT 文の ARCHIVE CHUNK）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
52			UNARCHIVE CHUNK 権限の付与（GRANT 文の UNARCHIVE CHUNK）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	GRANT
53			DBA 権限の取り消し（REVOKE 文のDBA）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
54			CONNECT 権限の取り消し（REVOKE 文の CONNECT）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
55			SCHEMA 権限の取り消し（REVOKE 文のSCHEMA）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
56			SELECT 権限の取り消し（REVOKE 文のSELECT）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
57			INSERT 権限の取り消し（REVOKE 文のINSERT）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
58			UPDATE 権限の取り消し（REVOKE 文のUPDATE）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
59			DELETE 権限の取り消し（REVOKE 文のDELETE）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
60			TRUNCATE 権限の取り消し（REVOKE 文の TRUNCATE）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
61			REFERENCES 権限の取り消し（REVOKE 文の REFERENCES）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
62			IMPORT TABLE 権限の取り消し（REVOKE 文の IMPORT TABLE）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
63			REBUILD INDEX 権限の取り消し（REVOKE 文の REBUILD INDEX）	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
64			GET COSTINFO 権限の取り消し	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
			(REVOKE 文のGET COSTINFO)							
65			EXPORT TABLE 権限の取り消し (REVOKE 文の EXPORT TABLE)	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
66			MERGE CHUNK 権限の取り消し (REVOKE 文の MERGE CHUNK)	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
67			CHANGE CHUNK COMMENT 権限の取り消し (REVOKE 文の CHANGE CHUNK COMMENT)	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
68			CHANGE CHUNK STATUS 権限の取り消し (REVOKE 文のCHANGE CHUNK STATUS)	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
69			ARCHIVE CHUNK 権限の取り消し (REVOKE 文の ARCHIVE CHUNK)	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
70			UNARCHIVE CHUNK 権限の取り消し (REVOKE 文のUNARCHIVE CHUNK)	正常終了 ※6	○	EVENT	○	○	PRIVILEGE	REVOKE
71			HADB ユーザの作成 (CREATE USER 文)	正常終了	○	EVENT	○	○	PRIVILEGE	CREATE USER
72			HADB ユーザの削除 (DROP USER 文)	正常終了	○	EVENT	○	○	PRIVILEGE	DROP USER

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
73			HADB ユーザの情報変更（ALTER USER 文）	正常終了	○	EVENT	○	○	PRIVILEGE	ALTER USER
74		定義系 SQL イベント	インデクスの定義（CREATE INDEX 文）	正常終了	○	EVENT	○	○	DEFINE	CREATE INDEX
75			スキーマの定義（CREATE SCHEMA 文）	正常終了	○	EVENT	○	○	DEFINE	CREATE SCHEMA
76			実表の定義（CREATE TABLE 文）	正常終了	○	EVENT	○	○	DEFINE	CREATE TABLE
77			ビュー表の定義（CREATE VIEW 文）	正常終了	○	EVENT	○	○	DEFINE	CREATE VIEW
78			インデクスの削除（DROP INDEX 文）	正常終了	○	EVENT	○	○	DEFINE	DROP INDEX
79			スキーマの削除（DROP SCHEMA 文）	正常終了	○	EVENT	○	○	DEFINE	DROP SCHEMA
80			実表の削除（DROP TABLE 文）	正常終了	○	EVENT	○	○	DEFINE	DROP TABLE
81			ビュー表の削除（DROP VIEW 文）	正常終了	○	EVENT	○	○	DEFINE	DROP VIEW
82			実表の定義変更（ALTER TABLE 文）	正常終了	○	EVENT	○	○	DEFINE	ALTER TABLE
83			ビュー表の定義変更（ALTER VIEW 文）	正常終了	○	EVENT	○	○	DEFINE	ALTER VIEW
84		操作系 SQL イベント	表の検索（SELECT 文）	正常終了	○	EVENT	○	○	MANIPULATE	SELECT
85			表への行挿入（INSERT 文※3）	正常終了	○	EVENT	○	○	MANIPULATE	INSERT

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
86			表の行更新 (UPDATE 文※3)	正常終了	○	EVENT	○	○	MANIPULATE	UPDATE
87			表の行削除 (DELETE 文※3)	正常終了	○	EVENT	○	○	MANIPULATE	DELETE
88			表の全行削除 (TRUNCATE TABLE 文)	正常終了	○	EVENT	○	○	MANIPULATE	TRUNCATE TABLE
89			チャンク内の全行削除（PURGE CHUNK 文※3)	正常終了	○	EVENT	○	○	MANIPULATE	PURGE CHUNK
90			SQL 構文の解析エラー		○	EVENT	○	○	MANIPULATE	UNKNOWN
91			チャンクに格納されているデータの取得 (adbsql コマンドのサブコマンド#GETDATA)	正常終了	○	EVENT	○	○	MANIPULATE	GETDATA
92			チャンクに格納されているデータの件数の取得 (adbsql コマンドのサブコマンド#GETCOUNT)	正常終了	○	EVENT	○	○	MANIPULATE	GETCOUNT
93			表情報の表示 (adbsql コマンドのサブコマンド#TABLES)	正常終了	○	EVENT	○	○	MANIPULATE	TABLES
94			列情報の表示 (adbsql コマンドのサブコマンド#COLUMNS)	正常終了	○	EVENT	○	○	MANIPULATE	COLUMNS
95			インデクス情報の表示（adbsql コマンドのサブコマンド #INDEXES)	正常終了	○	EVENT	○	○	MANIPULATE	INDEXES

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
96			チャンク情報の表示（adbsql コマンドのサブコマンド#CHUNKS）	正常終了	○	EVENT	○	○	MANIPULATE	CHUNKS
97			認可識別子の表示（adbsql コマンドのサブコマンド#GETUSER）	正常終了	○	EVENT	○	○	MANIPULATE	GETUSER
98		コマンド操作イベント	データのインポート（adbimport コマンド）	正常終了※2	○	EVENT	○	○	COMMAND	ADBIMPORT
99			データのエクスポート（-q オプションを指定していない adbexport コマンド）	正常終了※2	○	EVENT	○	○	COMMAND	ADBEXPORT
100			データのエクスポート（-q オプションを指定したadbexport コマンド）	正常終了※2	○	EVENT	○	○	COMMAND	ADBEXPORT
101			インデクスの再作成（adbidxrebuild コマンド）	正常終了※2	○	EVENT	○	○	COMMAND	ADBIDXREBUILD
102			コスト情報の収集（adbgetcst コマンド）	正常終了※2	○	EVENT	○	○	COMMAND	ADBGETCST
103			データベースの状態解析（adbdbstatus コマンド）	正常終了※2	○	EVENT	○		COMMAND	ADBDBSTATUS
104			チャンクのマージ（adbmergechunk コマンド）	正常終了※2	○	EVENT	○	○	COMMAND	ADBMERGECHUNK

監査イベント					監査証跡用の表関数導出表の列構成（その 1）					
項番	イベントの分類	イベントの種類	イベントの内容		HADB のバージョン (HADB_VERSION)	監査証跡の種別 (AUDIT_TRAIL_TYPE)	イベント実行日時 (EXEC_TIME)	イベント実行ユーザ (USER_NAME)	イベントの種類 (EVENT_TYPE)	イベント種別 (EVENT_SUBTYPE)
105			チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド）	正常終了 ※2	○	EVENT	○	○	COMMAND	ADBCHGCHUNKCOMMENT
106			チャンクの状態変更（adbchgchunkstatus コマンド）	正常終了 ※2	○	EVENT	○	○	COMMAND	ADBCHGCHUNKSTATUS
107			チャンクのアーカイブ（adbarchivechunk コマンド）	正常終了 ※2	○	EVENT	○	○	COMMAND	ADBARCHIVECHUNK
108			チャンクのアンアーカイブ（adbunarchivechunk コマンド）	正常終了 ※2	○	EVENT	○	○	COMMAND	ADBUNARCHIVECHUNK
109			システム表（実表）の再編成（adbreorgsystemdata コマンド）	正常終了 ※2	○	EVENT	○		COMMAND	ADBREORGSYSTEMDATA
110			同義語辞書の登録・削除（adbsyndict コマンド）	正常終了 ※2	○	EVENT	○		COMMAND	ADBSYNDICT

表 12-22　監査証跡の出力契機と出力項目（その 2）

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
1	必須監査イベント	システムイベント	HADB サーバの開始（adbstart コマンド）		SUCCESS		○				○	○
2			HADB サーバの終了（adbstop コマンド）	正常終了	SUCCESS		○		▼	▼	○	○
3			HADB サーバの稼働モードの変更（adbchgsrvmode コマンド）	正常終了	SUCCESS		○				○	○
4			ノード種別の変更（adbchgnodetype コマンド）	正常終了	SUCCESS		○				○	○
5			SQL トレース情報の出力開始・停止（adbchgsqltrc コマンド）	正常終了	SUCCESS		○				○	○
6			クライアント定義の集中管理（--update オプションを指定	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			した adbclientdefmang コマンド）									
7			クライアント定義の集中管理（-i オプションを指定した adbclientdefmang コマンド）	正常終了※2	SUCCESS		○	○			○	○
8			DB エリアの追加・変更（adbmodarea コマンド）	正常終了※2	SUCCESS		○	○			○	○
9			バッファの変更（adbmodbuff コマンド）	正常終了※2	SUCCESS		○	○			○	○
10			更新行のカラム化機能の管理（adbcolumnize コマンド）	正常終了	SUCCESS		○				○	○
11		監査イベント	監査管理権限の付与（GRANT 文の AUDIT ADMIN）	正常終了※6，※7	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
12			監査参照権限の付与（GRANT 文の AUDIT VIEWER）	正常終了※6，※7	SUCCESS		○	○	○	■	◇	○
13			監査管理権限の取り消し（REVOKE 文の AUDIT ADMIN）	正常終了※6，※7	SUCCESS		○	○	○	■	◇	○
14			監査参照権限の取り消し（REVOKE 文の AUDIT VIEWER）	正常終了※6，※7	SUCCESS		○	○	○	■	◇	○
15			監査対象の定義（CREATE AUDIT 文）	正常終了	SUCCESS		○	○	○	■	◇	○
16			監査対象定義情報の削除（DROP AUDIT 文）	正常終了	SUCCESS		○	○	○	■	◇	○
17			監査人のパスワード変更（ALTER USER 文）	正常終了	SUCCESS		○	○	○	■	◇	○
18			監査証跡機能の有効化（--start オプション	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			を指定した adbaudittrail コマンド）									
19			監査証跡機能の無効化（--stop オプションを指定した adbaudittrail コマンド）	正常終了※2	SUCCESS		○	○			○	○
20			監査証跡ファイルの切り替え（--swap オプションを指定した adbaudittrail コマンド）	正常終了※2，※9	SUCCESS		○	○	▼	▼	○	○
21			監査証跡機能に関連する情報の参照（-d オプションを指定した adbaudittrail コマンド）	正常終了※2，※9	SUCCESS		○	○	▼	▼	○	○
22			監査証跡用のシステム定義	正常終了	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			関数の実行（ADB_AUDITREAD 関数）（操作系 SQL）									
23			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（操作系 SQL）	正常終了	SUCCESS		○	○	○	■	◇	○
24			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（-q オプションを指定した adbexport コマンド）	正常終了※2	SUCCESS		○	○			○	○
25			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（-	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			q オプションを指定した adbexport コマンド）									
26			ディクショナリ表の SQL_AUDITS 表の検索（操作系 SQL）	正常終了	SUCCESS		○	○	○	■	◇	○
27			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（操作系 SQL）	正常終了	SUCCESS		○	○	○	■	◇	○
28			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定していない adbexport コマンド）	正常終了※2	SUCCESS		○	○			○	○
29			ディクショナリ	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定していない adbexport コマンド）									
30			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定した adbexport コマンド）	正常終了※2	SUCCESS		○	○			○	○
31			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
32			監査証跡ファイルの変換（adbconvertaudittrailfile コマンド）	正常終了※2，※9	SUCCESS		○	○			○	○
33	選択監査イベント	セッションイベント	HADB サーバへの接続	正常終了	SUCCESS		○	○	▽	△	◇	○
34			HADB サーバからの切り離し	正常終了※1	SUCCESS		○	○	▽	△	◇	○
35		権限管理イベント	DBA 権限の付与（GRANT 文の DBA）	正常終了※6	SUCCESS		○	○	○	■	◇	○
36			CONNECT 権限の付与（GRANT 文の CONNECT）	正常終了※6	SUCCESS		○	○	○	■	◇	○
37			SCHEMA 権限の付与（GRANT 文の SCHEMA）	正常終了※6	SUCCESS		○	○	○	■	◇	○
38			SELECT 権限の付与（GRANT 文の SELECT）	正常終了※6	SUCCESS		○	○	○	■	◇	○
39			INSERT 権限の付与（GRANT 文の INSERT）	正常終了※6	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
40			UPDATE 権限の付与（GRANT 文の UPDATE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
41			DELETE 権限の付与（GRANT 文の DELETE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
42			TRUNCATE 権限の付与（GRANT 文の TRUNCATE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
43			REFERENCES 権限の付与（GRANT 文の REFERENCES）	正常終了※6	SUCCESS		○	○	○	■	◇	○
44			IMPORT TABLE 権限の付与（GRANT 文の IMPORT TABLE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
45			REBUILD INDEX 権限の付与（GRANT 文の REBUILD INDEX）	正常終了※6	SUCCESS		○	○	○	■	◇	○
46			GET COSTINFO 権限の付与（GRANT	正常終了※6	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			文のGET COSTINFO）									
47			EXPORT TABLE 権限の付与（GRANT 文のEXPORT TABLE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
48			MERGE CHUNK 権限の付与（GRANT 文のMERGE CHUNK）	正常終了※6	SUCCESS		○	○	○	■	◇	○
49			CHANGE CHUNK COMMENT 権限の付与（GRANT 文の CHANGE CHUNK COMMENT）	正常終了※6	SUCCESS		○	○	○	■	◇	○
50			CHANGE CHUNK STATUS 権限の付与（GRANT 文のCHANGE CHUNK STATUS）	正常終了※6	SUCCESS		○	○	○	■	◇	○
51			ARCHIVE CHUNK 権限の付与（GRANT 文のARCHIVE CHUNK）	正常終了※6	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
52			UNARCHIVE CHUNK 権限の付与（GRANT 文の UNARCHIVE CHUNK）	正常終了※6	SUCCESS		○	○	○	■	◇	○
53			DBA 権限の取り消し（REVOKE 文のDBA）	正常終了※6	SUCCESS		○	○	○	■	◇	○
54			CONNECT 権限の取り消し（REVOKE 文の CONNECT）	正常終了※6	SUCCESS		○	○	○	■	◇	○
55			SCHEMA 権限の取り消し（REVOKE 文の SCHEMA）	正常終了※6	SUCCESS		○	○	○	■	◇	○
56			SELECT 権限の取り消し（REVOKE 文の SELECT）	正常終了※6	SUCCESS		○	○	○	■	◇	○
57			INSERT 権限の取り消し（REVOKE 文の INSERT）	正常終了※6	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
58			UPDATE 権限の取り消し（REVOKE 文の UPDATE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
59			DELETE 権限の取り消し（REVOKE 文の DELETE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
60			TRUNCATE 権限の取り消し（REVOKE 文の TRUNCATE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
61			REFERENCES 権限の取り消し（REVOKE 文の REFERENCES）	正常終了※6	SUCCESS		○	○	○	■	◇	○
62			IMPORT TABLE 権限の取り消し（REVOKE 文の IMPORT TABLE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
63			REBUILD INDEX 権限の取り消し	正常終了※6	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			（REVOKE 文の REBUILD INDEX）									
64			GET COSTINFO 権限の取り消し（REVOKE 文の GET COSTINFO）	正常終了※6	SUCCESS		○	○	○	■	◇	○
65			EXPORT TABLE 権限の取り消し（REVOKE 文の EXPORT TABLE）	正常終了※6	SUCCESS		○	○	○	■	◇	○
66			MERGE CHUNK 権限の取り消し（REVOKE 文の MERGE CHUNK）	正常終了※6	SUCCESS		○	○	○	■	◇	○
67			CHANGE CHUNK COMMENT 権限の取り消し（REVOKE 文の CHANGE CHUNK COMMENT）	正常終了※6	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
68			CHANGE CHUNK STATUS 権限の取り消し（REVOKE 文の CHANGE CHUNK STATUS）	正常終了※6	SUCCESS		○	○	○	■	◇	○
69			ARCHIVE CHUNK 権限の取り消し（REVOKE 文の ARCHIVE CHUNK）	正常終了※6	SUCCESS		○	○	○	■	◇	○
70			UNARCHIVE CHUNK 権限の取り消し（REVOKE 文の UNARCHIVE CHUNK）	正常終了※6	SUCCESS		○	○	○	■	◇	○
71			HADB ユーザの作成（CREATE USER 文）	正常終了	SUCCESS		○	○	○	■	◇	○
72			HADB ユーザの削除（DROP USER 文）	正常終了	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
73			HADB ユーザの情報変更（ALTER USER 文）	正常終了	SUCCESS		○	○	○	■	◇	○
74		定義系 SQL イベント	インデクスの定義（CREATE INDEX 文）	正常終了	SUCCESS		○	○	○	■	◇	○
75			スキーマの定義（CREATE SCHEMA 文）	正常終了	SUCCESS		○	○	○	■	◇	○
76			実表の定義（CREATE TABLE 文）	正常終了	SUCCESS		○	○	○	■	◇	○
77			ビュー表の定義（CREATE VIEW 文）	正常終了	SUCCESS		○	○	○	■	◇	○
78			インデクスの削除（DROP INDEX 文）	正常終了	SUCCESS		○	○	○	■	◇	○
79			スキーマの削除（DROP SCHEMA 文）	正常終了	SUCCESS		○	○	○	■	◇	○
80			実表の削除（DROP TABLE 文）	正常終了	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
81			ビュー表の削除（DROP VIEW 文）	正常終了	SUCCESS		○	○	○	■	◇	○
82			実表の定義変更（ALTER TABLE 文）	正常終了	SUCCESS		○	○	○	■	◇	○
83			ビュー表の定義変更（ALTER VIEW 文）	正常終了	SUCCESS		○	○	○	■	◇	○
84		操作系 SQL イベント	表の検索（SELECT 文）	正常終了	SUCCESS		○	○	○	■	◇	○
85			表への行挿入（INSERT 文※3）	正常終了	SUCCESS		○	○	○	■	◇	○
86			表の行更新（UPDATE 文※3）	正常終了	SUCCESS		○	○	○	■	◇	○
87			表の行削除（DELETE 文※3）	正常終了	SUCCESS		○	○	○	■	◇	○
88			表の全行削除（TRUNCATE TABLE 文）	正常終了	SUCCESS		○	○	○	■	◇	○
89			チャンク内の全行削除（PURGE	正常終了	SUCCESS		○	○	○	■	◇	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			CHUNK 文 ※3）									
90			SQL 構文の解析エラー		FAILURE		○	○	○	■	◇	○
91			チャンクに格納されているデータの取得（adbsql コマンドのサブコマンド #GETDATA）	正常終了	SUCCESS		○	○	○	■	○	○
92			チャンクに格納されているデータの件数の取得（adbsql コマンドのサブコマンド #GETCOUNT）	正常終了	SUCCESS		○	○	○	■	○	○
93			表情報の表示（adbsql コマンドのサブコマンド #TABLES）	正常終了	SUCCESS		○	○	○	■	○	○
94			列情報の表示（adbsql コマンド	正常終了	SUCCESS		○	○	○	■	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			のサブコマンド #COLUMNS）									
95			インデクス情報の表示（adbsql コマンドのサブコマンド #INDEXES）	正常終了	SUCCESS		○	○	○	■	○	○
96			チャンク情報の表示（adbsql コマンドのサブコマンド #CHUNKS）	正常終了	SUCCESS		○	○	○	■	○	○
97			認可識別子の表示（adbsql コマンドのサブコマンド #GETUSER）	正常終了	SUCCESS		○	○	○	■	○	○
98		コマンド操作イベント	データのインポート（adbimport コマンド）	正常終了※2	SUCCESS		○	○			○	○
99			データのエクスポート（-q オプションを	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
			指定していない adbexport コマンド）									
100			データのエクスポート（-q オプションを指定した adbexport コマンド）	正常終了※2	SUCCESS		○	○			○	○
101			インデクスの再作成（adbidxrebuild コマンド）	正常終了※2	SUCCESS		○	○			○	○
102			コスト情報の収集（adbgetcst コマンド）	正常終了※2	SUCCESS		○	○			○	○
103			データベースの状態解析（adbdbstatus コマンド）	正常終了※2	SUCCESS		○	○			○	○
104			チャンクのマージ（adbmergechunk コマンド）	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
105			チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド）	正常終了※2	SUCCESS		○	○			○	○
106			チャンクの状態変更（adbchgchunkstatus コマンド）	正常終了※2	SUCCESS		○	○			○	○
107			チャンクのアーカイブ（adbarchivechunk コマンド）	正常終了※2	SUCCESS		○	○			○	○
108			チャンクのアンアーカイブ（adbunarchivechunk コマンド）	正常終了※2	SUCCESS		○	○			○	○
109			システム表（実表）の再編成（adbreorgsystemdata コマンド）	正常終了※2	SUCCESS		○	○			○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 2）							
項番	イベントの分類	イベントの種類	イベントの内容		イベント成否（EVENT_RESULT）	確認した権限（USED_PRIVILEGE）	クライアント側の OS アカウント名（OS_USER_NAME）	AP 識別子（AP_NAME）	クライアントの IP アドレス（CLIENT_IP_ADDRESS）	クライアントのポート番号（CLIENT_PORT_NUMBER）	クライアントのプロセス名（CLIENT_PROCESS_NAME）	クライアントのプロセス ID（CLIENT_PROCESS_ID）
110			同義語辞書の登録・削除（adbsyndict コマンド）	正常終了※2	SUCCESS		○	○			○	○

表 12-23　監査証跡の出力契機と出力項目（その 3）

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名（HADB_HOST_NAME）	HADB の IP アドレス（HADB_IP_ADDRESS）	HADB のポート番号（HADB_PORT_NUMBER）	HADB のプロセス ID（HADB_PROCESS_ID）	HADB のサーバディレクトリ（ADBDIR）	HADB のノード番号※8（HADB_NODE_NUMBER）	コネクション ID（CONNECTION_ID）	コネクション通番（CONNECTION_NUMBER）	文ハンドル（STATEMENT_HANDLE）	SQL 文通番（SQL_SERIAL_NUMBER）
1	必須監査イベント	システムイベント	HADB サーバの開始（adbstart コマンド）		○	○	○	○	○	○				
2			HADB サーバの終了（adbstop コマンド）	正常終了	○	○	○	○	○	○				
3			HADB サーバの稼働モードの変更	正常終了	○	○	○	○	○	○				

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			(adbchgsrvmode コマンド)											
4			ノード種別の変更 (adbchgnodetype コマンド)	正常終了	○	○	○	○	○	○				
5			SQL トレース情報の出力開始・停止 (adbchgsqltrc コマンド)	正常終了	○	○	○	○	○	○				
6			クライアント定義の集中管理（--update オプションを指定した adbclientdefmang コ	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			マンド）											
7			クライアント定義の集中管理（-i オプションを指定した adbclientdefmang コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
8			DB エリアの追加・変更（adbmodarea コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
9			バッファの変更（adbmodbuff コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
10			更新行のカラム化機	正常終了	○	○	○	○	○	○				

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			能の管理（adbcolumnize コマンド）											
11		監査イベント	監査管理権限の付与（GRANT 文の AUDIT ADMIN）	正常終了※6, ※7	○	○	○	○	○	○	○	○	○	○
12			監査参照権限の付与（GRANT 文の AUDIT VIEWER）	正常終了※6, ※7	○	○	○	○	○	○	○	○	○	○
13			監査管理権限の取り消し（REVOKE 文の AUDIT ADMIN）	正常終了※6, ※7	○	○	○	○	○	○	○	○	○	○
14			監査参照権限の取り消し（REVOK	正常終了※6, ※7	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			E 文の AUDIT VIEWER）											
15			監査対象の定義（CREATE AUDIT 文）	正常終了	○	○	○	○	○	○	○	○	○	○
16			監査対象定義情報の削除（DROP AUDIT 文）	正常終了	○	○	○	○	○	○	○	○	○	○
17			監査人のパスワード変更（ALTER USER 文）	正常終了	○	○	○	○	○	○	○	○	○	○
18			監査証跡機能の有効化（--start オプションを指定した	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			adbaudittrail コマンド）											
19			監査証跡機能の無効化（--stop オプションを指定した adbaudittrail コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
20			監査証跡ファイルの切り替え（--swap オプションを指定した adbaudittrail コマンド）	正常終了※2, ※9	○	○	○	○	○	○	○	○		
21			監査証跡機能に関連する情報の参照（-d	正常終了※2, ※9	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			オプションを指定した adbaudittrail コマンド）											
22			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（操作系 SQL）	正常終了	○	○	○	○	○	○	○	○	○	○
23			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（操作系 SQL）	正常終了	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
24			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
25			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（-q オプションを指定した adbexport コ	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			マンド）											
26			ディクショナリ表の SQL_AUDITS 表の検索（操作系 SQL）	正常終了	○	○	○	○	○	○	○	○	○	○
27			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（操作系 SQL）	正常終了	○	○	○	○	○	○	○	○	○	○
28			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定していない adbexp	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			ort コマンド）											
29			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定していない adbexport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
30			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
31			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
32			監査証跡ファイルの変換（adbconvertauditrailfile コマンド）	正常終了※2, ※9	○	○	○	○	○	○	○	○		
33	選択監査イベント	セッションイベ	HADB サーバへの接続	正常終了	○	○	○	○	○	○	○	○		
34			HADB サーバからの	正常終了※1	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
		ント	切り離し											
35		権限管理イベント	DBA 権限の付与 (GRANT 文の DBA)	正常終了※6	○	○	○	○	○	○	○	○	○	○
36			CONNECT 権限の付与 (GRANT 文の CONNECT)	正常終了※6	○	○	○	○	○	○	○	○	○	○
37			SCHEMA 権限の付与 (GRANT 文の SCHEMA)	正常終了※6	○	○	○	○	○	○	○	○	○	○
38			SELECT 権限の付与 (GRANT 文の SELECT)	正常終了※6	○	○	○	○	○	○	○	○	○	○
39			INSERT 権限の付与 (GRANT 文の	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			INSERT）											
40			UPDATE 権限の付与（GRANT 文の UPDATE）	正常終了※6	○	○	○	○	○	○	○	○	○	○
41			DELETE 権限の付与（GRANT 文の DELETE）	正常終了※6	○	○	○	○	○	○	○	○	○	○
42			TRUNCATE 権限の付与（GRANT 文の TRUNCATE）	正常終了※6	○	○	○	○	○	○	○	○	○	○
43			REFERENCES 権限の付与（GRANT 文の REFERENCES）	正常終了※6	○	○	○	○	○	○	○	○	○	○
44			IMPORT TABLE 権限の	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			付与（GRANT 文の IMPORT TABLE）											
45			REBUILD INDEX 権限の付与（GRANT 文の REBUILD INDEX）	正常終了※6	○	○	○	○	○	○	○	○	○	○
46			GETCOSTINFO 権限の付与（GRANT 文の GETCOSTINFO）	正常終了※6	○	○	○	○	○	○	○	○	○	○
47			EXPORT TABLE 権限の付与（GRANT 文の EXPORT TABLE）	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
48			MERGE CHUNK 権限の付与 (GRANT 文の MERGE CHUNK)	正常終了※6	○	○	○	○	○	○	○	○	○	○
49			CHANGE CHUNK COMMENT 権限の付与 (GRANT 文の CHANGE CHUNK COMMENT)	正常終了※6	○	○	○	○	○	○	○	○	○	○
50			CHANGE CHUNK STATUS 権限の付与 (GRANT 文の CHANGE CHUNK STATUS)	正常終了※6	○	○	○	○	○	○	○	○	○	○
51			ARCHIVE CHUNK 権限の付与	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			(GRANT 文の ARCHIVE CHUNK)											
52			UNARCHIVE CHUNK 権限の付与 (GRANT 文の UNARCHIVE CHUNK)	正常終了※6	○	○	○	○	○	○	○	○	○	○
53			DBA 権限の取り消し (REVOKE 文の DBA)	正常終了※6	○	○	○	○	○	○	○	○	○	○
54			CONNECT 権限の取り消し (REVOKE 文の CONNECT)	正常終了※6	○	○	○	○	○	○	○	○	○	○
55			SCHEMA 権限の取り消し (REVOK	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			E 文の SCHEMA ）											
56			SELECT 権限の取り消し（REVOKE 文の SELECT ）	正常終了※6	○	○	○	○	○	○	○	○	○	○
57			INSERT 権限の取り消し（REVOKE 文の INSERT ）	正常終了※6	○	○	○	○	○	○	○	○	○	○
58			UPDATE 権限の取り消し（REVOKE 文の UPDATE ）	正常終了※6	○	○	○	○	○	○	○	○	○	○
59			DELETE 権限の取り消し（REVOKE 文の	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			DELETE）											
60			TRUNCATE 権限の取り消し（REVOKE 文の TRUNCATE）	正常終了※6	○	○	○	○	○	○	○	○	○	○
61			REFERENCES 権限の取り消し（REVOKE 文の REFERENCES）	正常終了※6	○	○	○	○	○	○	○	○	○	○
62			IMPORT TABLE 権限の取り消し（REVOKE 文の IMPORT TABLE）	正常終了※6	○	○	○	○	○	○	○	○	○	○
63			REBUILD INDEX 権限の取り消し（REVOK	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			E 文の REBUILD INDEX）											
64			GET COSTINFO 権限の取り消し（REVOKE 文の GET COSTINFO）	正常終了※6	○	○	○	○	○	○	○	○	○	○
65			EXPORT TABLE 権限の取り消し（REVOKE 文の EXPORT TABLE）	正常終了※6	○	○	○	○	○	○	○	○	○	○
66			MERGE CHUNK 権限の取り消し（REVOKE 文の MERGE CHUNK）	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
67			CHANGE CHUNK COMMENT 権限の取り消し (REVOKE 文の CHANGE CHUNK COMMENT)	正常終了※6	○	○	○	○	○	○	○	○	○	○
68			CHANGE CHUNK STATUS 権限の取り消し (REVOKE 文の CHANGE CHUNK STATUS)	正常終了※6	○	○	○	○	○	○	○	○	○	○
69			ARCHIVE CHUNK 権限の取り消し (REVOKE 文の ARCHIVE CHUNK)	正常終了※6	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
70			UNARCHIVE CHUNK 権限の取り消し (REVOKE 文の UNARCHIVE CHUNK)	正常終了※6	○	○	○	○	○	○	○	○	○	○
71			HADB ユーザの作成 (CREATE USER 文)	正常終了	○	○	○	○	○	○	○	○	○	○
72			HADB ユーザの削除 (DROP USER 文)	正常終了	○	○	○	○	○	○	○	○	○	○
73			HADB ユーザの情報変更 (ALTER USER 文)	正常終了	○	○	○	○	○	○	○	○	○	○
74		定義系 SQ	インデクスの定義 (CREAT	正常終了	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
		L イベント	E INDEX 文）											
75			スキーマの定義（CREATE SCHEMA 文）	正常終了	○	○	○	○	○	○	○	○	○	○
76			実表の定義（CREATE TABLE 文）	正常終了	○	○	○	○	○	○	○	○	○	○
77			ビュー表の定義（CREATE VIEW 文）	正常終了	○	○	○	○	○	○	○	○	○	○
78			インデクスの削除（DROP INDEX 文）	正常終了	○	○	○	○	○	○	○	○	○	○
79			スキーマの削除（DROP SCHEMA 文）	正常終了	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
80			実表の削除（DROP TABLE 文）	正常終了	○	○	○	○	○	○	○	○	○	○
81			ビュー表の削除（DROP VIEW 文）	正常終了	○	○	○	○	○	○	○	○	○	○
82			実表の定義変更（ALTER TABLE 文）	正常終了	○	○	○	○	○	○	○	○	○	○
83			ビュー表の定義変更（ALTER VIEW 文）	正常終了	○	○	○	○	○	○	○	○	○	○
84		操作系 SQL イベント	表の検索（SELECT 文）	正常終了	○	○	○	○	○	○	○	○	○	○
85			表への行挿入（INSERT 文※3）	正常終了	○	○	○	○	○	○	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
86			表の行更新 (UPDATE 文※3)	正常終了	○	○	○	○	○	○	○	○	○	○
87			表の行削除 (DELETE 文※3)	正常終了	○	○	○	○	○	○	○	○	○	○
88			表の全行削除 (TRUNCATE TABLE 文)	正常終了	○	○	○	○	○	○	○	○	○	○
89			チャンク内の全行削除 (PURGE CHUNK 文※3)	正常終了	○	○	○	○	○	○	○	○	○	○
90			SQL 構文の解析エラー		○	○	○	○	○	○	○	○	○	○
91			チャンクに格納されているデータの取得 (adbsql コマンドの	正常終了	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			サブコマンド #GETDATA)											
92			チャンクに格納されているデータの件数の取得 (adbsql コマンドのサブコマンド #GETCOUNT)	正常終了	○	○	○	○	○	○	○	○		
93			表情報の表示 (adbsql コマンドのサブコマンド #TABLES)	正常終了	○	○	○	○	○	○	○	○		
94			列情報の表示 (adbsql コマンドのサブコマンド	正常終了	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			#COLUMNS)											
95			インデクス情報の表示 (adbsql コマンドのサブコマンド #INDEXES)	正常終了	○	○	○	○	○	○	○	○		
96			チャンク情報の表示 (adbsql コマンドのサブコマンド #CHUNKS)	正常終了	○	○	○	○	○	○	○	○		
97			認可識別子の表示 (adbsql コマンドのサブコマンド #GETUSER)	正常終了	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
98		コマンド操作イベント	データのインポート（adbimport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
99			データのエクスポート（-q オプションを指定していない adbexport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
100			データのエクスポート（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	○	○	○	○	○	○		
101			インデクスの再作成	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			(adbidxrebuild コマンド)											
102			コスト情報の収集 (adbgetcst コマンド)	正常終了※2	○	○	○	○	○	○	○	○		
103			データベースの状態解析 (adbdbstatus コマンド)	正常終了※2	○	○	○	○	○	○	○	○		
104			チャンクのマージ (adbmergechunk コマンド)	正常終了※2	○	○	○	○	○	○	○	○		
105			チャンクのコメントの設定・変更・削除 (adbch	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			gchunk comment コマンド)											
106			チャンクの状態変更 (adbchgchunkstatus コマンド)	正常終了※2	○	○	○	○	○	○	○	○		
107			チャンクのアーカイブ (adbarchivechunk コマンド)	正常終了※2	○	○	○	○	○	○	○	○		
108			チャンクのアンアーカイブ (adbunarchivechunk コマンド)	正常終了※2	○	○	○	○	○	○	○	○		
109			システム表 (実表) の再編	正常終了※2	○	○	○	○	○	○	○	○		

監査イベント					監査証跡用の表関数導出表の列構成（その 3）									
項番	イベントの分類	イベントの種類	イベントの内容		HADB のホスト名 (HADB_HOST_NAME)	HADB の IP アドレス (HADB_IP_ADDRESS)	HADB のポート番号 (HADB_PORT_NUMBER)	HADB のプロセス ID (HADB_PROCESS_ID)	HADB のサーバディレクトリ (ADBDIR)	HADB のノード番号※8 (HADB_NODE_NUMBER)	コネクション ID (CONNECTION_ID)	コネクション通番 (CONNECTION_NUMBER)	文ハンドル (STATEMENT_HANDLE)	SQL 文通番 (SQL_SERIAL_NUMBER)
			成（adbreorgsystemdata コマンド）											
110			同義語辞書の登録・削除（adbsyndict コマンド）	正常終了※2	○	○	○	○	○	○	○	○		

表 12-24　監査証跡の出力契機と出力項目（その 4）

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報 (MESSAGE_LOG_INFO)	終了コード (EXIT_STATUS)	オブジェクトの種別※4 (OBJECT_TYPE)	オブジェクトの所有者※4 (OBJECT_OWNER_NAME)	オブジェクトのスキーマ名※4 (OBJECT_SCHEMA_NAME)	オブジェクト名※4 (OBJECT_NAME)	アクセス件数※5 (ACCESS_COUNT)
1	必須監査イベント	システムイベント	HADB サーバの開始（adbstart コマンド）		○	○					
2			HADB サーバの終了（adbstop コマンド）	正常終了	○	○					

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
3			HADB サーバの稼働モードの変更（adbchgsrvmode コマンド）	正常終了	○	○					
4			ノード種別の変更（adbchgnodetype コマンド）	正常終了	○	○					
5			SQL トレース情報の出力開始・停止（adbchgsqltrc コマンド）	正常終了	○	○					
6			クライアント定義の集中管理（--update オプションを指定した adbclientdefmang コマンド）	正常終了※2	○	○					
7			クライアント定義の集中管理（-i オプションを指定した adbclientdefmang コマンド）	正常終了※2	○	○					
8			DB エリアの追加・変	正常終了※2	○	○					

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			更（adbmodarea コマンド）								
9			バッファの変更（adbmodbuff コマンド）	正常終了※2	○	○					
10			更新行のカラム化機能の管理（adbcolumnize コマンド）	正常終了	○	○					
11		監査イベント	監査管理権限の付与（GRANT 文のAUDIT ADMIN）	正常終了※6，※7	○	○					○
12			監査参照権限の付与（GRANT 文のAUDIT VIEWER）	正常終了※6，※7	○	○					○
13			監査管理権限の取り消し（REVOKE 文のAUDIT ADMIN）	正常終了※6，※7	○	○					○
14			監査参照権限の取り消し（REVOKE 文のAUDIT VIEWER）	正常終了※6，※7	○	○					○
15			監査対象の定義	正常終了	○	○					○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			（CREATE AUDIT 文）								
16			監査対象定義情報の削除（DROP AUDIT 文）	正常終了	○	○					○
17			監査人のパスワード変更（ALTER USER 文）	正常終了	○	○					○
18			監査証跡機能の有効化（--start オプションを指定した adbaudittrail コマンド）	正常終了※2	○	○					
19			監査証跡機能の無効化（--stop オプションを指定した adbaudittrail コマンド）	正常終了※2	○	○					
20			監査証跡ファイルの切り替え（--swap オプションを指定した adbaudittrail コマンド）	正常終了※2，※9	○	○					

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
21			監査証跡機能に関連する情報の参照（-d オプションを指定した adbaudittrail コマンド）	正常終了※2，※9	○	○					
22			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（操作系 SQL）	正常終了	○	○	TABLE FUNCTION	○	○	○	○
23			ADB_AUDITREAD 関数を指定した導出表に依存するビュー表の検索（操作系 SQL）	正常終了	○	○	VIEW	○	○	○	○
24			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	TABLE FUNCTION	○	○	○	○
25			ADB_AUDITREAD 関数を	正常終了※2	○	○	VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			指定した導出表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）								
26			ディクショナリ表の SQL_AUDITS 表の検索（操作系 SQL）	正常終了	○	○	VIEW	○	○	○	○
27			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（操作系 SQL）	正常終了	○	○	VIEW	○	○	○	○
28			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定していない adbexport コマンド）	正常終了※2	○	○	VIEW	○	○	○	○
29			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q	正常終了※2	○	○	VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			オプションを指定していない adbexport コマンド）								
30			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	VIEW	○	○	○	○
31			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	VIEW	○	○	○	○
32			監査証跡ファイルの変換（adbconvertaudittrailfile コマンド）	正常終了※2，※9	○	○					
33	選択監査イベント	セッションイベント	HADB サーバへの接続	正常終了	○	○					
34			HADB サーバからの切り離し	正常終了※1	○	○					

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
35		権限管理イベント	DBA 権限の付与（GRANT 文のDBA）	正常終了※6	○	○					○
36			CONNECT 権限の付与（GRANT 文のCONNECT）	正常終了※6	○	○					○
37			SCHEMA 権限の付与（GRANT 文のSCHEMA）	正常終了※6	○	○					○
38			SELECT 権限の付与（GRANT 文のSELECT）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
39			INSERT 権限の付与（GRANT 文のINSERT）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
40			UPDATE 権限の付与（GRANT 文のUPDATE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
41			DELETE 権限の付与（GRANT 文のDELETE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
42			TRUNCATE 権限の付与（GRANT 文の TRUNCATE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
43			REFERENCES 権限の付与（GRANT 文	正常終了※6	○	○	TABLE または VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			の REFERENCES）								
44			IMPORT TABLE 権限の付与（GRANT 文のIMPORT TABLE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
45			REBUILD INDEX 権限の付与（GRANT 文のREBUILD INDEX）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
46			GET COSTINFO 権限の付与（GRANT 文のGET COSTINFO）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
47			EXPORT TABLE 権限の付与（GRANT 文のEXPORT TABLE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
48			MERGE CHUNK 権限の付与（GRANT 文のMERGE CHUNK）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
49			CHANGE CHUNK COMMENT 権限の付与	正常終了※6	○	○	TABLE または VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			（GRANT 文のCHANGE CHUNK COMMENT）								
50			CHANGE CHUNK STATUS 権限の付与（GRANT 文のCHANGE CHUNK STATUS）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
51			ARCHIVE CHUNK 権限の付与（GRANT 文のARCHIVE CHUNK）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
52			UNARCHIVE CHUNK 権限の付与（GRANT 文の UNARCHIVE CHUNK）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
53			DBA 権限の取り消し（REVOKE 文のDBA）	正常終了※6	○	○					○
54			CONNECT 権限の取り消し（REVOKE 文の CONNECT）	正常終了※6	○	○					○
55			SCHEMA 権限の取り消し（REVOKE	正常終了※6	○	○					○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			文の SCHEMA）								
56			SELECT 権限の取り消し（REVOKE 文の SELECT）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
57			INSERT 権限の取り消し（REVOKE 文の INSERT）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
58			UPDATE 権限の取り消し（REVOKE 文の UPDATE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
59			DELETE 権限の取り消し（REVOKE 文の DELETE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
60			TRUNCATE 権限の取り消し（REVOKE 文の TRUNCATE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
61			REFERENCES 権限の取り消し（REVOKE 文の REFERENCES）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
62			IMPORT TABLE 権限の取り消し（REVOKE 文のIMPORT TABLE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
63			REBUILD INDEX 権限の取り消し（REVOKE 文のREBUILD INDEX）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
64			GET COSTINFO 権限の取り消し（REVOKE 文のGET COSTINFO）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
65			EXPORT TABLE 権限の取り消し（REVOKE 文のEXPORT TABLE）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
66			MERGE CHUNK 権限の取り消し（REVOKE 文のMERGE CHUNK）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
67			CHANGE CHUNK COMMENT 権限の取り消し（REVOKE 文のCHANGE	正常終了※6	○	○	TABLE または VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			CHUNK COMMENT）								
68			CHANGE CHUNK STATUS 権限の取り消し（REVOKE 文のCHANGE CHUNK STATUS）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
69			ARCHIVE CHUNK 権限の取り消し（REVOKE 文のARCHIVE CHUNK）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
70			UNARCHIVE CHUNK 権限の取り消し（REVOKE 文の UNARCHIVE CHUNK）	正常終了※6	○	○	TABLE または VIEW	○	○	○	○
71			HADB ユーザの作成（CREATE USER 文）	正常終了	○	○					○
72			HADB ユーザの削除（DROP USER 文）	正常終了	○	○					○
73			HADB ユーザの情報変更（ALTER USER 文）	正常終了	○	○					○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
74		定義系 SQL イベント	インデクスの定義（CREATE INDEX 文）	正常終了	○	○	INDEX	○	○	○	○
75			スキーマの定義（CREATE SCHEMA 文）	正常終了	○	○	SCHEMA	○		○	○
76			実表の定義（CREATE TABLE 文）	正常終了	○	○	TABLE	○	○	○	○
77			ビュー表の定義（CREATE VIEW 文）	正常終了	○	○	VIEW	○	○	○	○
78			インデクスの削除（DROP INDEX 文）	正常終了	○	○	INDEX	○	○	○	○
79			スキーマの削除（DROP SCHEMA 文）	正常終了	○	○	SCHEMA	○		○	○
80			実表の削除（DROP TABLE 文）	正常終了	○	○	TABLE	○	○	○	○
81			ビュー表の削除（DROP VIEW 文）	正常終了	○	○	VIEW	○	○	○	○
82			実表の定義変更（ALTER TABLE 文）	正常終了	○	○	TABLE	○	○	○	○
83			ビュー表の定義変更	正常終了	○	○	VIEW	○	○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			（ALTER VIEW 文）								
84		操作系 SQL イベント	表の検索（SELECT 文）	正常終了	○	○	TABLE，VIEW または TABLE FUNCTION	○	○	○	○
85			表への行挿入（INSERT 文※3）	正常終了	○	○	TABLE または VIEW	○	○	○	○
86			表の行更新（UPDATE 文※3）	正常終了	○	○	TABLE または VIEW	○	○	○	○
87			表の行削除（DELETE 文※3）	正常終了	○	○	TABLE または VIEW	○	○	○	○
88			表の全行削除（TRUNCATE TABLE 文）	正常終了	○	○	TABLE	○	○	○	○
89			チャンク内の全行削除（PURGE CHUNK 文※3）	正常終了	○	○	TABLE	○	○	○	○
90			SQL 構文の解析エラー		○	○					○
91			チャンクに格納されているデータの取得（adbsql コマンドのサブコマンド #GETDATA）	正常終了	○	○					○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
92			チャンクに格納されているデータの件数の取得（adbsql コマンドのサブコマンド #GETCOUNT）	正常終了	○	○					○
93			表情報の表示（adbsql コマンドのサブコマンド #TABLES）	正常終了	○	○					○
94			列情報の表示（adbsql コマンドのサブコマンド #COLUMNS）	正常終了	○	○					○
95			インデクス情報の表示（adbsql コマンドのサブコマンド #INDEXES）	正常終了	○	○					○
96			チャンク情報の表示（adbsql コマンドのサブコマンド #CHUNKS）	正常終了	○	○					○
97			認可識別子の表示（adbsql コマンドのサ	正常終了	○	○					○

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			ブコマンド #GETUSER）								
98		コマンド操作イベント	データのインポート（adbimport コマンド）	正常終了※2	○	○	TABLE	○	○	○	○
99			データのエクスポート（-q オプションを指定していない adbexport コマンド）	正常終了※2	○	○	TABLE または VIEW	○	○	○	○
100			データのエクスポート（-q オプションを指定した adbexport コマンド）	正常終了※2	○	○	TABLE, VIEW または TABLE FUNCTION	○	○	○	○
101			インデクスの再作成（adbidxrebuild コマンド）	正常終了※2	○	○	INDEX	○	○	○	
102			コスト情報の収集（adbgetcst コマンド）	正常終了※2	○	○	TABLE	○	○	○	
103			データベースの状態解析（adbdbstatus コマンド）	正常終了※2	○	○					

監査イベント					監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容		メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
104			チャンクのマージ（adbmergechunk コマンド）	正常終了※2	○	○	TABLE	○	○	○	
105			チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド）	正常終了※2	○	○	TABLE	○	○	○	
106			チャンクの状態変更（adbchgchunkstatus コマンド）	正常終了※2	○	○	TABLE	○	○	○	
107			チャンクのアーカイブ（adbarchivechunk コマンド）	正常終了※2	○	○	TABLE	○	○	○	
108			チャンクのアンアーカイブ（adbunarchivechunk コマンド）	正常終了※2	○	○	TABLE	○	○	○	
109			システム表（実表）の再編成（adbreorgsystemdata コマンド）	正常終了※2	○	○	TABLE	○	○	○	
110			同義語辞書の登録・削除	正常終了※2	○	○					

監査イベント				監査証跡用の表関数導出表の列構成（その 4）						
項番	イベントの分類	イベントの種類	イベントの内容	メッセージログ情報（MESSAGE_LOG_INFO）	終了コード（EXIT_STATUS）	オブジェクトの種別※4（OBJECT_TYPE）	オブジェクトの所有者※4（OBJECT_OWNER_NAME）	オブジェクトのスキーマ名※4（OBJECT_SCHEMA_NAME）	オブジェクト名※4（OBJECT_NAME）	アクセス件数※5（ACCESS_COUNT）
			（adbsyndict コマンド）							

表 12-25　監査証跡の出力契機と出力項目（その 5）

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
1	必須監査イベント	システムイベント	HADB サーバの開始（adbstart コマンド）						●	○	○			
2			HADB サーバの終了（adbstop コマンド）	正常終了										
3			HADB サーバの稼働モードの変更（adbchgsrvmo	正常終了					○	○				

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			de コマンド）											
4			ノード種別の変更（adbchgnodetype コマンド）	正常終了					○	○				
5			SQL トレース情報の出力開始・停止（adbchgsqltrc コマンド）	正常終了					○	○				
6			クライアント定義の集中管理（--update オプションを指定した adbclientdefmang コ	正常終了 ※2					○	○				

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			マンド）											
7			クライアント定義の集中管理（-i オプションを指定した adbclientdefmang コマンド）	正常終了 ※2										
8			DB エリアの追加・変更（adbmodarea コマンド）	正常終了 ※2					○	○				
9			バッファの変更（adbmodbuff コマンド）	正常終了 ※2					○	○				

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
10			更新行のカラム化機能の管理（adbcolumnize コマンド）	正常終了					○	○				
11		監査イベント	監査管理権限の付与（GRANT 文の AUDIT ADMIN）	正常終了 ※6, ※7	○	○	○					○	○	○
12			監査参照権限の付与（GRANT 文の AUDIT VIEWER）	正常終了 ※6, ※7	○	○	○					○	○	○
13			監査管理権限の取り消し（REVOKE 文の AUDIT ADMIN）	正常終了 ※6, ※7	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
14			監査参照権限の取り消し（REVOKE 文の AUDIT VIEWER）	正常終了 ※6, ※7	○	○	○					○	○	○
15			監査対象の定義（CREATE AUDIT 文）	正常終了			○					○	○	○
16			監査対象定義情報の削除（DROP AUDIT 文）	正常終了			○					○	○	○
17			監査人のパスワード変更（ALTER USER 文）	正常終了		○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
18			監査証跡機能の有効化（--start オプションを指定した adbaudittrail コマンド）	正常終了 ※2										
19			監査証跡機能の無効化（--stop オプションを指定した adbaudittrail コマンド）	正常終了 ※2										
20			監査証跡ファイルの切り替え（--swap オプションを指定した	正常終了 ※2, ※9										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			adbaudittrail コマンド）											
21			監査証跡機能に関連する情報の参照（-d オプションを指定した adbaudittrail コマンド）	正常終了 ※2, ※9										
22			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（操作系 SQL）	正常終了			○	○				○	○	○
23			ADB_AUDITREA	正常終了			○	○				○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			D 関数を指定した導出表に依存するビュー表の検索（操作系 SQL）											
24			監査証跡用のシステム定義関数の実行（ADB_AUDITREAD 関数）（-q オプションを指定した adbexport コマンド）	正常終了 ※2			○							
25			ADB_AUDITREAD 関数を指定した導	正常終了 ※2			○							

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			出表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）											
26			ディクショナリ表の SQL_AUDITS 表の検索（操作系 SQL）	正常終了			○	○				○	○	○
27			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（操	正常終了			○	○				○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			作系 SQL）											
28			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定していない adbexport コマンド）	正常終了 ※2										
29			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定していない adbexport コ	正常終了 ※2										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			マンド）											
30			ディクショナリ表の SQL_AUDITS 表の検索（-q オプションを指定した adbexport コマンド）	正常終了 ※2			○							
31			ディクショナリ表の SQL_AUDITS 表に依存するビュー表の検索（-q オプションを指定した adbexport コマンド）	正常終了 ※2			○							

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
32			監査証跡ファイルの変換（adbconvertaudittrailfile コマンド）	正常終了 ※2, ※9										
33	選択監査イベント	セッションイベント	HADB サーバへの接続	正常終了										
34			HADB サーバからの切り離し	正常終了 ※1								○	○	○
35		権限管理イベント	DBA 権限の付与（GRANT 文の DBA）	正常終了 ※6	○	○	○					○	○	○
36			CONNECT 権限の付与（GRANT 文の CONNECT）	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
37			SCHEMA 権限の付与（GRANT 文の SCHEMA）	正常終了 ※6	○	○	○					○	○	○
38			SELECT 権限の付与（GRANT 文の SELECT）	正常終了 ※6	○	○	○					○	○	○
39			INSERT 権限の付与（GRANT 文の INSERT）	正常終了 ※6	○	○	○					○	○	○
40			UPDATE 権限の付与（GRANT 文の UPDATE）	正常終了 ※6	○	○	○					○	○	○
41			DELETE 権限の付与（GRANT	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			文の DELETE）											
42			TRUNCATE 権限の付与（GRANT 文の TRUNCATE）	正常終了 ※6	○	○	○					○	○	○
43			REFERENCES 権限の付与（GRANT 文の REFERENCES）	正常終了 ※6	○	○	○					○	○	○
44			IMPORT TABLE 権限の付与（GRANT 文の IMPORT TABLE）	正常終了 ※6	○	○	○					○	○	○
45			REBUILD INDEX 権限の付与（GRANT	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			文の REBUILD INDEX）											
46			GET COSTINFO 権限の付与（GRANT 文の GET COSTINFO）	正常終了 ※6	○	○	○					○	○	○
47			EXPORT TABLE 権限の付与（GRANT 文の EXPORT TABLE）	正常終了 ※6	○	○	○					○	○	○
48			MERGE CHUNK 権限の付与（GRANT 文の MERGE CHUNK）	正常終了 ※6	○	○	○					○	○	○
49			CHANGE CHUNK COMMEN	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			T 権限の付与（GRANT 文の CHANGE CHUNK COMMENT）											
50			CHANGE CHUNK STATUS 権限の付与（GRANT 文の CHANGE CHUNK STATUS）	正常終了 ※6	○	○	○					○	○	○
51			ARCHIVE CHUNK 権限の付与（GRANT 文の ARCHIVE CHUNK）	正常終了 ※6	○	○	○					○	○	○
52			UNARCHIVE CHUNK 権限の	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			付与（GRANT 文の UNARCHIVE CHUNK）											
53			DBA 権限の取り消し（REVOKE 文の DBA）	正常終了 ※6	○	○	○					○	○	○
54			CONNECT 権限の取り消し（REVOKE 文の CONNECT）	正常終了 ※6	○	○	○					○	○	○
55			SCHEMA 権限の取り消し（REVOKE 文の SCHEMA）	正常終了 ※6	○	○	○					○	○	○
56			SELECT 権限の取り消し	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			（REVOKE 文の SELECT）											
57			INSERT 権限の取り消し（REVOKE 文の INSERT）	正常終了※6	○	○	○					○	○	○
58			UPDATE 権限の取り消し（REVOKE 文の UPDATE）	正常終了※6	○	○	○					○	○	○
59			DELETE 権限の取り消し（REVOKE 文の DELETE）	正常終了※6	○	○	○					○	○	○
60			TRUNCATE 権限の取り消し	正常終了※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			（REVOKE 文の TRUNCATE）											
61			REFERENCES 権限の取り消し（REVOKE 文の REFERENCES）	正常終了※6	○	○	○					○	○	○
62			IMPORT TABLE 権限の取り消し（REVOKE 文の IMPORT TABLE）	正常終了※6	○	○	○					○	○	○
63			REBUILD INDEX 権限の取り消し（REVOKE 文の REBUILD INDEX）	正常終了※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
64			GET COSTINFO 権限の取り消し（REVOKE 文の GET COSTINFO）	正常終了※6	○	○	○					○	○	○
65			EXPORT TABLE 権限の取り消し（REVOKE 文の EXPORT TABLE）	正常終了※6	○	○	○					○	○	○
66			MERGE CHUNK 権限の取り消し（REVOKE 文の MERGE CHUNK）	正常終了※6	○	○	○					○	○	○
67			CHANGE CHUNK COMMENT 権限の取り	正常終了※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			消し（REVOKE 文の CHANGE CHUNK COMMENT）											
68			CHANGE CHUNK STATUS 権限の取り消し（REVOKE 文の CHANGE CHUNK STATUS）	正常終了 ※6	○	○	○					○	○	○
69			ARCHIVE CHUNK 権限の取り消し（REVOKE 文の ARCHIVE CHUNK）	正常終了 ※6	○	○	○					○	○	○
70			UNARCHIVE CHUNK	正常終了 ※6	○	○	○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			権限の取り消し（REVOKE 文の UNARCHIVE CHUNK）											
71			HADB ユーザの作成（CREATE USER 文）	正常終了		○	○					○	○	○
72			HADB ユーザの削除（DROP USER 文）	正常終了		○	○					○	○	○
73			HADB ユーザの情報変更（ALTER USER 文）	正常終了		○	○					○	○	○
74		定義系 SQL	インデクスの定義（CREATE	正常終了			○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
		イベント	INDEX文）											
75			スキーマの定義（CREATE SCHEMA文）	正常終了			○					○	○	○
76			実表の定義（CREATE TABLE文）	正常終了			○					○	○	○
77			ビュー表の定義（CREATE VIEW文）	正常終了			○					○	○	○
78			インデクスの削除（DROP INDEX文）	正常終了			○					○	○	○
79			スキーマの削除（DROP	正常終了			○					○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			SCHEMA 文）											
80			実表の削除（DROP TABLE 文）	正常終了			○					○	○	○
81			ビュー表の削除（DROP VIEW 文）	正常終了			○					○	○	○
82			実表の定義変更（ALTER TABLE 文）	正常終了			○					○	○	○
83			ビュー表の定義変更（ALTER VIEW 文）	正常終了			○					○	○	○
84		操作系 SQL イベ	表の検索（SELECT 文）	正常終了			○	○				○	○	○
85			表への行挿入	正常終了			○	○				○	○	○

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
		ント	（INSERT 文※3）											
86			表の行更新（UPDATE 文※3）	正常終了			○	○				○	○	○
87			表の行削除（DELETE 文※3）	正常終了			○	○				○	○	○
88			表の全行削除（TRUNCATE TABLE 文）	正常終了			○					○	○	○
89			チャンク内の全行削除（PURGE CHUNK 文※3）	正常終了			○	○				○	○	○
90			SQL 構文の解析エラー				○					○	○	○
91			チャンクに格納され	正常終了										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			ているデータの取得（adbsql コマンドのサブコマンド #GETDATA）											
92			チャンクに格納されているデータの件数の取得（adbsql コマンドのサブコマンド #GETCOUNT）	正常終了										
93			表情報の表示（adbsql コマンドのサブコマンド #TABLES）	正常終了										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
94			列情報の表示（adbsql コマンドのサブコマンド #COLUMNS）	正常終了										
95			インデクス情報の表示（adbsql コマンドのサブコマンド #INDEXES）	正常終了										
96			チャンク情報の表示（adbsql コマンドのサブコマンド #CHUNKS）	正常終了										
97			認可識別子の表示	正常終了										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			（adbsql コマンドのサブコマンド #GETUSER）											
98		コマンド操作イベント	データのインポート（adbimport コマンド）	正常終了 ※2										
99			データのエクスポート（-q オプションを指定していない adbexport コマンド）	正常終了 ※2										
100			データのエクスポート（-q オプション	正常終了 ※2			○							

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			を指定した adbexport コマンド）											
101			インデクスの再作成（adbidxrebuild コマンド）	正常終了 ※2										
102			コスト情報の収集（adbgetcst コマンド）	正常終了 ※2										
103			データベースの状態解析（adbdbstatus コマンド）	正常終了 ※2										
104			チャンクのマージ（adbme	正常終了 ※2										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
			rgechunk コマンド）											
105			チャンクのコメントの設定・変更・削除（adbchgchunkcomment コマンド）	正常終了 ※2										
106			チャンクの状態変更（adbchgchunkstatus コマンド）	正常終了 ※2										
107			チャンクのアーカイブ（adbarchivechunk コマンド）	正常終了 ※2										

監査イベント					監査証跡用の表関数導出表の列構成（その 5）									
項番	イベントの分類	イベントの種類	イベントの内容		付与，または取り消した権限（PRIVILEGE_TYPE）	イベント対象になった認可識別子（PRIVILEGE_USER_NAME）	SQL 文（SQL_SOURCE）	？パラメタ（PARAM）	システムの変更情報（変更前）（BEFORE_SYSTEM_INFO）	システムの変更情報（変更後）（AFTER_SYSTEM_INFO）	データベースの設定情報（SERVER_OPERAND）	ユーザ付加情報 1（USER_INFO_1）	ユーザ付加情報 2（USER_INFO_2）	ユーザ付加情報 3（USER_INFO_3）
108			チャンクのアンアーカイブ（adbunarchivechunk コマンド）	正常終了 ※2										
109			システム表（実表）の再編成（adbreorgsystemdata コマンド）	正常終了 ※2										
110			同義語辞書の登録・削除（adbsyndict コマンド）	正常終了 ※2										

（凡例）

○

出力情報がある場合は，該当する情報が出力されます。出力情報がない場合は，NULL が出力されます。

空欄

NULL が出力されます。

■

UNIX ドメイン通信による接続の場合は，NULL が出力されます。

◇

クライアントが JDBC ドライバの場合は，NULL が出力されます。

●

マルチノード機能を使用している場合，ノードによって出力される情報が異なります。

- プライマリノードの場合は，前回起動時にプライマリノードだったノードの HADB サーバの稼働モードが出力されます。
- セカンダリノードまたはワーカーノードの場合は，常にNULL が出力されます。

▽

実行したコマンドの処理の延長である場合は，NULL が出力されます。

△

次のどちらかに該当する場合は，NULL が出力されます。

- UNIX ドメイン通信による接続の場合
- 実行したコマンドの処理の延長である場合

▼

マルチノード機能を使用している場合は，次のどれかのケースに該当する監査証跡に，サーバ間通信で使用するプライマリノードの IP アドレスとポート番号が出力されます。次のケースに該当しない監査証跡には，NULL が出力されます。

- プライマリノードで--node オプションを指定しないでadbstop コマンドを実行した際に，「イベント種別（EVENT_SUBTYPE)」列にADBSTOP が出力されている監査証跡

  セカンダリノードまたはワーカーノードに出力されている監査証跡が対象です。
- -n オプションにセカンダリノードまたはワーカーノードのノード番号を指定してadbaudittrail --swap コマンドを実行した際に，「イベント種別（EVENT_SUBTYPE)」列にADBAUDITTRAIL SWAP が出力されている監査証跡

  セカンダリノードで処理が実行されて，セカンダリノードに出力されている監査証跡が対象です。または，ワーカーノードで処理が実行されて，ワーカーノードに出力されている監査証跡が対象です。
- -n オプションにセカンダリノードまたはワーカーノードのノード番号を指定してadbaudittrail -d コマンドを実行した際に，「イベント種別（EVENT_SUBTYPE)」列にADBAUDITTRAIL DISPLAY が出力されている監査証跡

  セカンダリノードで処理が実行されて，セカンダリノードに出力されている監査証跡が対象です。または，ワーカーノードで処理が実行されて，ワーカーノードに出力されている監査証跡が対象です。

注

上記の凡例以外の出力情報（EVENT，SYSTEM など）は，表内に記載されているとおりの内容が出力されます。詳細については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」の「表 12-10 監査証跡を検索するときの表関数導出表の列構成」を参照してください。

なお，「オブジェクトの種別（OBJECT_TYPE)」列については，出力情報がない場合，NULL が出力されます。

注※1

コネクションプーリングを使用している場合は，Connection オブジェクトのclose は含みません（SQL トレース情報に出力されるコール種別DISP が該当します)。

注※2

対象イベントの前後には，「イベント種別（EVENT_SUBTYPE)」列にCONNECT とDISCONNECT が出力された監査証跡が，それぞれ出力されます。それぞれの監査証跡について，次に示す列には実行したコマンド名が出力されます。

- 「AP 識別子（AP_NAME)」列
- 「クライアントのプロセス名（CLIENT_PROCESS_NAME)」列

注※3

SELECT 文以外の操作系 SQL で問合せ式本体を含む SQL 文を実行した場合，問合せ式本体に記述したオブジェクトに関する監査証跡は，「表の検索（SELECT 文)」行と同じ情報が出力されます。

注※4

次のどちらかに該当する場合は，表内に出力情報が記載されていても，NULL が出力されます。

- イベント対象のオブジェクトがすべて表値構成子の場合
- コマンド実行時に，どのオブジェクトに対しても処理が実行されなかった場合

注※5

SQL 文の実行まで完了した場合と区別するために，SQL 文の前処理だけが完了した状態で文ハンドルをクローズした場合はNULL が出力されます。なお，コマンドは該当しません。

注※6

SQL 文に指定されたすべての認可識別子に対して，権限の付与または権限の取り消しが不要だった場合は，次に示す列にNULL が出力されます。

- 「オブジェクトの種別（OBJECT_TYPE)」列
- 「オブジェクトの所有者（OBJECT_OWNER_NAME)」列
- 「オブジェクトのスキーマ名（OBJECT_SCHEMA_NAME)」列
- 「オブジェクト名（OBJECT_NAME)」列
- 「付与，または取り消した権限（PRIVILEGE_TYPE)」列
- 「イベント対象になった認可識別子（PRIVILEGE_USER_NAME)」列

注※7

SQL 文に指定されたすべての認可識別子に対して，権限の付与または権限の取り消しが不要だった場合は，「イベントの種類（EVENT_TYPE)」列にPRIVILEGE が出力されます。

注※8

マルチノード機能を使用した場合に，情報が出力されます。マルチノード機能を使用していない場合は，NULL が出力されます。

注※9

マルチノード機能を使用している場合，対象イベントの「イベント種別（EVENT_SUBTYPE)」列にCONNECTとDISCONNECT が出力される監査証跡は，常にプライマリノードで出力されます。

12.10　監査証跡機能使用時の留意事項

ここでは，監査証跡機能使用時の留意事項について説明します。

- 「12.10.1　1 つのイベントで複数の監査証跡が出力されるケース」
- 「12.10.2　コマンドの実行時に出力される監査証跡についての留意事項」

12.10.1　1 つのイベントで複数の監査証跡が出力されるケース

イベントの対象となるオブジェクトが複数ある場合，対象となるオブジェクトの数だけ監査証跡が出力されます。例えば，次の SQL 文を実行した場合，表T1 についての監査証跡と，表T2 についての監査証跡がそれぞれ出力されます。

```
SELECT * FROM "T1","T2"
```

同様に，イベントの対象となる権限や認可識別子が複数ある場合も，権限や認可識別子の組み合わせの数だけ監査証跡が出力されます。1 つのイベントで複数の監査証跡が出力されるケースを次の表に示します。

表 12-26　1 つのイベントで複数の監査証跡が出力されるケース

項番	実行したイベント	出力される監査証跡の数
1	監査管理権限の付与	権限を付与した認可識別子の数
2	監査参照権限の付与	
3	監査管理権限の取り消し	権限を取り消した認可識別子の数
4	監査参照権限の取り消し	
5	権限の付与（アクセス権限以外の場合）	MAX（1，付与した権限の数 × 権限を付与した認可識別子の数）
6	権限の付与（アクセス権限の場合）	MAX（1，付与したアクセス権限の数※1 × 対象オブジェクトの数※2 × 権限を付与した認可識別子の数）※6，※7
7	権限の取り消し（アクセス権限以外の場合）	MAX（1，取り消した権限の数 × 権限を取り消した認可識別子の数）
8	権限の取り消し（アクセス権限の場合）	MAX（1，取り消したアクセス権限の数※1 × 対象オブジェクトの数※2 × 権限を取り消した認可識別子の数）※6，※7
9	表の検索（SELECT 文の場合）	MAX（1，問合せ式本体に指定した対象オブジェクトの数※4）
10	表の検索（SELECT 文以外の操作系 SQL の場合※3）	問合せ式本体に指定した対象オブジェクトの数※4
11	インデクスの再作成	MAX（1，再作成したインデクスの数※5）
12	コスト情報の収集	MAX（1，コスト情報を収集した表の数※5）

項番	実行したイベント	出力される監査証跡の数
13	データエクスポート（-q オプションを指定したadbexport コマンドの場合）	MAX（1，SQL 記述ファイル中に記述した SQL 文の問合せ式本体に指定した対象オブジェクトの数※4）

注※1

ALL PRIVILEGES を指定してGRANT 文またはREVOKE 文を実行した場合，すべてのアクセス権限を指定したときと同じ動作になります。

注※2

ALL TABLES を指定してGRANT 文またはREVOKE 文を実行した場合，対象オブジェクトの数は，GRANT 文またはREVOKE 文を実行した HADB ユーザのスキーマ内の全実表数になります。ただし，SQL 文がエラーになったときは，全実表分の監査証跡が出力されないことがあります。

注※3

問合せ式本体を含むINSERT 文などの操作系 SQL を実行した場合，問合せ式本体中に指定したオブジェクトについての監査証跡は，表の検索イベントとして出力されます。

注※4

同じオブジェクトを複数指定した場合は，重複排除されます。また，表値構成子はオブジェクト数としてカウントされません。

注※5

オブジェクトごとにイベントの成功，失敗の情報が監査証跡に出力されます。また，失敗した時点でコマンドの処理がエラーになるため，失敗したオブジェクトの後ろに指定されたオブジェクトからは監査証跡が出力されません。

注※6

指定した権限を 1 つも付与しないでGRANT 文が正常終了した場合，GRANT 文についての監査証跡が 1 つだけ出力されます。その際，オブジェクト，権限種別，HADB ユーザの認可識別子には，NULL が出力されます。REVOKE 文の場合も同様です。

注※7

アクセス権限の付与または取り消しの際に SQL 文がエラーになった場合，付与または取り消されたアクセス権限がないため，指定したすべてのアクセス権限が監査証跡の出力対象になります。エラーによって対象オブジェクトの種別が判別できない場合は，対象オブジェクトの種別にNULL が出力されます。

12.10.2　コマンドの実行時に出力される監査証跡についての留意事項

(1)　HADB サーバに接続するコマンドを実行したときに出力される監査証跡

HADB サーバに接続するコマンドを実行した場合，コマンドについての監査証跡以外に，次のイベントの監査証跡が出力されます。

- HADB サーバへの接続
- HADB サーバからの切り離し

HADB サーバに接続するコマンドについては，マニュアル『HADB　コマンドリファレンス』の『コマンドの一覧』を参照してください。

(2)　adbcancel コマンドを実行した場合の留意事項

HADB サーバに接続するコマンドに対してadbcancel コマンドを実行したタイミングや，HADB サーバに接続するコマンドの実行中に HADB サーバが通信エラーを検知したタイミングによっては，監査証跡に出力される情報とコマンドの戻り値が次の表に示すようになります。

表 12-27　監査証跡に出力される情報とコマンドの戻り値

adbcancel コマンドを実行したタイミング，または HADB サーバが通信エラーを検知したタイミング	HADB サーバへの接続についての監査証跡	コマンドについての監査証跡	HADB サーバからの切り離しについての監査証跡	コマンドの戻り値
HADB サーバへの接続とコマンドの処理の間	「成功」で出力される	出力されない	「発生」で出力される※	エラー
コマンドの処理と HADB サーバからの切り離しの間	「成功」で出力される	コマンドの処理が成功している場合は，「成功」で出力される	「発生」で出力される※	エラー

注※

イベント結果に「発生」が出力された場合，HADB サーバからの切り離し自体は完了しています。「発生」が出力された原因については，メッセージログを参照してください。

(3)　adbaudittrail コマンドを実行した場合の留意事項

- 次のタイミングで HADB サーバが現用の監査証跡ファイルのリネームに失敗した場合，監査証跡に出力される終了コードとコマンドのリターンコードが不一致になります。
  - HADB サーバを終了したとき
  - 監査証跡機能を無効化したとき
- マルチノード機能の使用時にadbaudittrail コマンドを実行した場合，監査証跡は次の表に示すノードに出力されます。

表 12-28　adbaudittrail コマンドの実行時に監査証跡が出力されるノード

項番	adbaudittrail コマンドに指定したオプション	監査証跡の出力先ノード		
		HADB サーバへの接続についての監査証跡	adbaudittrail コマンドについての監査証跡	HADB サーバからの切り離しについての監査証跡
1	--start	プライマリノード	プライマリノード	プライマリノード

項番	adbaudittrail コマンドに指定したオプション	監査証跡の出力先ノード		
		HADB サーバへの接続についての監査証跡	adbaudittrail コマンドについての監査証跡	HADB サーバからの切り離しについての監査証跡
2	--stop			
3	--swap	プライマリノード※1	コマンドを実行したノード	プライマリノード※1
4	--swap -n *ノード番号*	プライマリノード	-n オプションに指定したノード番号のノード※2, ※3	プライマリノード
5	-d	プライマリノード※1	コマンドを実行したノード	プライマリノード※1
6	-d -n *ノード番号*	プライマリノード	-n オプションに指定したノード番号のノード※2, ※3	プライマリノード

注※1

プライマリノードで実行した場合，CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER にはNULL が出力されます。セカンダリノードまたはワーカーノードで実行した場合，CLIENT_IP_ADDRESS と CLIENT_PORT_NUMBER には，adbaudittrail コマンドがプライマリノードの HADB サーバに接続するときに使用した IP アドレスとポート番号が出力されます。CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER は，監査証跡を検索するときの表関数導出表の列名です。詳細については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

注※2

adbaudittrail コマンドの-n オプションにセカンダリノードまたはワーカーノードのノード番号を指定した場合，CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER には，サーバ間通信で使用するプライマリノードの IP アドレスとポート番号が出力されます。プライマリノードのノード番号を指定した場合は，CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER にはNULL が出力されます。CLIENT_IP_ADDRESS と CLIENT_PORT_NUMBER は，監査証跡を検索するときの表関数導出表の列名です。詳細については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

注※3

次に示すエラーが発生した場合，adbaudittrail コマンドについての監査証跡は，プライマリノードに出力されます。

- 監査管理権限（-d オプションの場合は監査権限）を持っていない HADB ユーザがadbaudittrail コマンドを実行した場合
- 監査証跡機能が無効なときにadbaudittrail コマンドを実行した場合

(4)　adbconvertaudittrailfile コマンドを実行した場合の留意事項

マルチノード機能の使用時にadbconvertaudittrailfile コマンドを実行した場合，監査証跡は次の表に示すノードに出力されます。

表 12-29　adbconvertaudittrailfile コマンドの実行時に監査証跡が出力されるノード

監査証跡の出力先ノード		
HADB サーバへの接続についての監査証跡	**adbconvertaudittrailfile コマンドについての監査証跡**	**HADB サーバからの切り離しについての監査証跡**
プライマリノード※	コマンドを実行したノード	プライマリノード※

注※

プライマリノードで実行した場合，CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER にはNULL が出力されます。セカンダリノードまたはワーカーノードで実行した場合，CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER には，adbconvertaudittrailfile コマンドがプライマリノードの HADB サーバに接続するときに使用した IP アドレスとポート番号が出力されます。CLIENT_IP_ADDRESS とCLIENT_PORT_NUMBER は，監査証跡を検索するときの表関数導出表の列名です。詳細については，「12.9.2　監査証跡を検索するときの表関数導出表の列構成」を参照してください。

13 チューニング

この章では，HADB のチューニングについて説明します。

13.1　処理性能の向上に関するチューニング

ここでは，HADB サーバの処理性能を向上させるためのチューニング方法について説明します。

各コマンドの処理性能を向上させるためのチューニング方法については，「13.3　コマンドの実行時間の短縮に関するチューニング」を参照してください。

13.1.1　レンジインデクスの一括先読み

HADB サーバの開始後，初回の表検索時のテーブルスキャンが遅い場合，レンジインデクスの一括先読みの適用を検討してください。レンジインデクスの一括先読みについては，「5.5.3　レンジインデクスの一括先読み」を参照してください。

13.1.2　定義したレンジインデクスの見直し

レンジインデクスを使用した検索処理の状況を確認することで，定義したレンジインデクスを見直す方法について説明します。

通常は，SQL 文の統計情報を確認してください。レンジインデクスを使用した検索処理の状況を確認して，必要があれば，定義したレンジインデクスを見直します。

ただし，レンジインデクスが定義された複数の表に対する検索処理の状況を確認する場合は，アクセスパスの統計情報を確認してください。SQL 文の統計情報では，どのレンジインデクスを見直せばよいかわかりません。

定義したレンジインデクスを見直す方法を次に示します。

- SQL 文の統計情報を確認する場合
  「(1)　SQL 文の統計情報を確認してレンジインデクスを見直す」を参照してください。
- アクセスパスの統計情報を確認する場合
  「(2)　アクセスパスの統計情報を確認してレンジインデクスを見直す」を参照してください。

(1)　SQL 文の統計情報を確認してレンジインデクスを見直す

SQL 文の統計情報を確認して，レンジインデクスを使用した検索処理の状況を確認します。

SQL 文の統計情報は，次に示すどちらかの方法で確認してください。

- `adbstat` コマンドを実行する
  マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析）』を参照してください。

- SQL トレースファイルを確認する
  「10.11.2　SQL トレース情報に出力される情報」の「(10)　SQL 文の統計情報」を参照してください。

SQL 文の統計情報を確認して，レンジインデクスを使用した検索処理の状況を確認する手順を，次に示します。

手順

1. 「チャンクのスキップ」によるレンジインデクスの使用状況を確認する
   レンジインデクスを使用した検索処理で，チャンクのスキップがどれだけ実行されているかについて確認します。
   SQL 文の統計情報に出力された，次に示す項目を確認してください。
   - Ridx_chunk_skip_cnt（レンジインデクスを使用した検索で，表のチャンクをスキップした回数）
   - Ridx_chunk_judge_cnt（レンジインデクスを使用した検索で，表のチャンクが探索条件を満たす値域のデータを格納しているかどうかを判定した回数）

   確認した項目を基に，次に示す計算式を求めてください。
   計算式

   ```
   Ridx_chunk_skip_cnt÷Ridx_chunk_judge_cnt
   ```

2. 「セグメントのスキップ」によるレンジインデクスの使用状況を確認する
   レンジインデクスを使用した検索処理で，セグメントのスキップがどれだけ実行されているかについて確認します。
   SQL 文の統計情報に出力された，次に示す項目を確認してください。
   - Ridx_sgmt_skip_cnt（レンジインデクスを使用した検索で，表のセグメントをスキップした回数）
   - Ridx_sgmt_judge_cnt（レンジインデクスを使用した検索で，表のセグメントが探索条件を満たす値域のデータを格納しているかどうかを判定した回数）

   確認した項目を基に，次に示す計算式を求めてください。
   計算式

   ```
   Ridx_sgmt_skip_cnt÷Ridx_sgmt_judge_cnt
   ```

3. 定義したレンジインデクスを見直す
   手順 1.および手順 2.で計算した結果が両方とも 0 の場合，または両方とも 0 に近い場合は，レンジインデクスの効果がありません（チャンクのスキップ，およびセグメントのスキップが実行されていません）。そのため，定義したレンジインデクスを削除することを検討してください。
   効果がないレンジインデクスを使用すると，レンジインデクスを使用しない検索処理よりも，検索処理に時間が掛かるおそれがあります。
   なお，手順 1.および手順 2.で計算した結果が両方とも 1 に近い場合，または片方だけ 1 に近い場合は，レンジインデクスの効果があります（チャンクのスキップ，またはセグメントのスキップが実行されています）。そのため，定義したレンジインデクスの見直しは不要です。

重要

シングルチャンク表の場合でも，手順 1.を実施してください。レンジインデクスは，シングルチャンク表を 1 つのチャンクとして，チャンクのスキップを実行します。そのため，レンジインデクスの効果の有無を確認する必要があります。

(2) アクセスパスの統計情報を確認してレンジインデクスを見直す

アクセスパスの統計情報を確認して，レンジインデクスを使用した検索処理の状況を確認します。

アクセスパスの統計情報は，SQL トレースファイルを確認してください。アクセスパスの統計情報については，「10.11.3 アクセスパスの統計情報の出力例と出力項目」を参照してください。

アクセスパスの統計情報を確認して，レンジインデクスを使用した検索処理の状況を確認する手順を，次に示します。

手順

1. 「チャンクのスキップ」によるレンジインデクスの使用状況を確認する

   レンジインデクスを使用した検索処理で，チャンクのスキップがどれだけ実行されているかについて確認します。

   アクセスパスの統計情報に出力された，次に示す項目を確認してください。

   - Data_ridx_chunk_skip_cnt（レンジインデクスを使用した検索で，表のチャンクをスキップした回数）
   - Data_ridx_chunk_judge_cnt（レンジインデクスを使用した検索で，表のチャンクを判定した回数）

   確認した項目を基に，次に示す計算式を求めてください。

   計算式

   ```
   Data_ridx_chunk_skip_cnt÷Data_ridx_chunk_judge_cnt
   ```

2. 「セグメントのスキップ」によるレンジインデクスの使用状況を確認する

   レンジインデクスを使用した検索処理で，セグメントのスキップがどれだけ実行されているかについて確認します。

   アクセスパスの統計情報に出力された，次に示す項目を確認してください。

   - Data_ridx_sgmt_skip_cnt（レンジインデクスを使用した検索で，表のセグメントをスキップした回数）
   - Data_ridx_sgmt_judge_cnt（レンジインデクスを使用した検索で，表のセグメントを判定した回数）

   確認した項目を基に，次に示す計算式を求めてください。

   計算式

   ```
   Data_ridx_sgmt_skip_cnt÷Data_ridx_sgmt_judge_cnt
   ```

3. 定義したレンジインデクスを見直す

手順 1.および手順 2.で計算した結果が両方とも 0 の場合，または両方とも 0 に近い場合は，レンジインデクスの効果がありません（チャンクのスキップ，およびセグメントのスキップが実行されていません）。そのため，定義したレンジインデクスを削除することを検討してください。

効果がないレンジインデクスを使用すると，レンジインデクスを使用しない検索処理よりも，検索処理に時間が掛かるおそれがあります。

なお，手順 1.および手順 2.で計算した結果が両方とも 1 に近い場合，または片方だけ 1 に近い場合は，レンジインデクスの効果があります（チャンクのスキップ，またはセグメントのスキップが実行されています）。そのため，定義したレンジインデクスの見直しは不要です。

重要

シングルチャンク表の場合でも，手順 1.を実施してください。レンジインデクスは，シングルチャンク表を 1 つのチャンクとして，チャンクのスキップを実行します。そのため，レンジインデクスの効果の有無を確認する必要があります。

13.1.3 定義したテキストインデクスの見直し

テキストインデクスが効果的に使用されているかどうかを確認する方法について説明します。

確認方法

SQL トレースファイルに出力されているアクセスパスの統計情報の次の出力項目を確認してください。

- Data_tidx_all_search_cnt

上記の項目には，テキストインデクスを使用してチャンク内のデータを検索した際，チャンク内のデータを全件検索した回数が出力されます。上記の項目に 1 以上の値が出力されている場合は，対処を検討してください。

メモ

アクセスパスの統計情報については，「10.11.3 アクセスパスの統計情報の出力例と出力項目」を参照してください。

対処方法

次に示すどちらかの対処を行うと，性能が改善する可能性があります。

- UNION を使用してそれぞれの検索結果を統合するように SQL 文を修正する
- テキストインデクスを使用しないで検索するように SQL 文を修正する（インデクス指定を指定する）

SQL 文の修正例については，「11.16.11 同義語検索のチューニングを行う」の「(2) テキストインデクスが効果的に使用されているかを確認する」を参照してください。

次の条件をすべて満たす SQL 文を指定している場合は，性能低下の原因となっている可能性があります。

- LIKE 述語，LIKE_REGEX 述語，またはスカラ関数CONTAINS を検索条件に指定している

- 上記の述語およびスカラ関数を指定した検索条件を，論理演算子OR を使って複数指定している
- 上記の述語およびスカラ関数を指定した検索条件中にテキストインデクスの構成列を指定している

上記の条件を満たす場合，すべての検索条件に指定されている検索文字列を使用して検索が行われます。検索条件の数が非常に多い場合，テキストインデクスによる絞り込みは行われません。このとき，テキストインデクスを使用した全件検索が行われるため，検索時間が非常に長くなるおそれがあります。

また，同義語検索が行われた場合，内部処理では，各同義語を指定した検索条件が論理演算子OR を使って複数個指定された形に変更されます。そのため，同義語グループに登録されている同義語が非常に多い場合，テキストインデクスによる絞り込みではなく，テキストインデクスを使用した全件検索が行われます。

13.1.4　HADB サーバ開始時間の短縮（HugePages の適用）

ここでは，HADB サーバの開始時間を短縮する方法について説明します。

HADB のグローバルバッファは OS の共有メモリを使用しています。そのため，HADB のサーバ定義で大量のグローバルバッファを確保する定義を指定すると，OS のメモリを管理する TLB 内に要求されたアドレスがない状態（**TLB ミス**）が多発します。その結果，共有メモリのアクセス性能が低下し，HADB サーバの開始時間が長くなります。

このような場合，HADB の共有メモリにHugePages を適用してください。HugePages は TLB のページテーブルを拡張する機能です。TLB のページテーブルを拡張することで TLB ミスが少なくなり，共有メモリへのアクセス性能が改善されます。

HADB の共有メモリにHugePages を適用する手順を次に示します。

設定方法

1. HADB サーバを終了する
   adbstop コマンドを実行して，HADB サーバを終了してください。
2. HugePages を有効にするために，次のカーネルパラメタを指定する
   - vm.nr_hugepages パラメタ
   - vm.hugetlb_shm_group パラメタ
   - soft memlock パラメタ
   - hard memlock パラメタ

   各カーネルパラメタの詳細については，「6.2　カーネルパラメタの見積もり」を参照してください。
3. OS を再起動する
   カーネルパラメタの指定を変更した場合，OS を再起動してください。
4. サーバ定義を変更する
   サーバ定義にadb_sys_shm_huge_page_size オペランドを指定してください。

また，手順 2.でvm.nr_hugepages パラメタの指定値を変更した場合，サーバ定義のadb_sys_memory_limit オペランドの指定値も変更してください。

5. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_sys_shm_huge_page_size オペランド，adb_sys_memory_limit オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_shm_huge_page_size オペランド，adb_sys_memory_limit オペランドを参照してください。

13.1.5　SQL 文の実行待ち状態の発生防止

ここでは，サーバ定義のadb_sys_rthd_num オペランドに指定した処理リアルスレッド数の見直しによる処理性能の向上について説明します。

指定した処理リアルスレッド数が少ない場合，SQL 文の実行待ち状態が発生することがあります。SQL 文の実行待ち状態が発生すると，SQL 文の処理に時間が掛かるおそれがあるため，指定した処理リアルスレッド数が適切かどうかを確認してください。

手順

1. SQL 文の実行待ち状態を確認する

adbstat コマンドを実行して出力される「HADB サーバの統計情報」のうち，次に示す情報を確認してください。

- SQL_execute_wait_total_time（SQL 文の実行待ち状態の合計時間）
- SQL_execute_wait_cnt（SQL 文の実行待ち状態の発生回数）

SQL_execute_wait_total_time およびSQL_execute_wait_cnt に出力された値が，どちらも 0 でない場合，SQL 文の実行待ち状態が発生しています。この場合，adb_sys_rthd_num オペランドの指定値を大きくすることで，SQL 文の実行待ち状態が発生しなくなり，処理性能が向上することがあります。

2. HADB サーバを終了する

adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

サーバ定義のadb_sys_rthd_num オペランドの指定値を大きくして，HADB サーバが使用する処理リアルスレッド数を多くしてください。

重要

adb_sys_rthd_num オペランドの指定値を変更した場合，必ず HADB サーバのメモリ所要量を再見積もりしてください。HADB サーバのメモリ所要量の見積もりについては，「6.3 HADB サーバのメモリ所要量の見積もり」を参照してください。

4. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

メモ

サーバ定義のadb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_num オペランドを参照してください。

13.1.6　SQL 文の処理時間の短縮

ここでは，サーバ定義およびクライアント定義のadb_sql_exe_max_rthd_num オペランドに指定した最大 SQL 処理リアルスレッド数の見直しによる処理性能の向上について説明します。

指定した最大 SQL 処理リアルスレッド数が少ない場合，SQL 文の実行時に使用される処理リアルスレッドの数が少なくなるため，SQL 文の処理に時間が掛かるおそれがあります。指定した最大 SQL 処理リアルスレッド数が適切かどうかを確認してください。

(1)　SELECT 文の処理時間を短縮する方法

SELECT 文の処理時間を短縮する手順を次に示します。

手順

1. SELECT 文の実行時間を確認する

adbstat コマンドを実行して出力される「コネクションの稼働情報」のうち，次に示す情報を確認してください。

- SELECT_total_time（SELECT 文の実行時間）

2. adb_sql_exe_max_rthd_num オペランドの指定値を見直す

手順 1.で確認したSELECT_total_time に出力された値が大きい場合，SQL 文の処理に時間が掛かっているおそれがあります。この場合，サーバ定義およびクライアント定義のadb_sql_exe_max_rthd_num オペランドの指定値を大きくすることで，処理性能が向上することがあります。

重要

- クライアント定義のadb_sql_exe_max_rthd_num オペランドと，サーバ定義の adb_sql_exe_max_rthd_num オペランドの両方の指定値を確認してください。クライアン

ト定義の指定値だけを大きくした結果，クライアント定義の指定値がサーバ定義の指定値を超えた場合，サーバ定義の指定値が適用されます（変更後のクライアント定義の指定値が適用されません）。該当する場合は，サーバ定義とクライアント定義の両方の adb_sql_exe_max_rthd_num オペランドの指定値を大きくしてください。

- adb_sql_exe_max_rthd_num オペランドの指定値を大きくした場合は，サーバ定義の adb_sys_rthd_num オペランドの指定値も大きくしてください。
  adb_sql_exe_max_rthd_num オペランドの指定値だけを大きくすると，SQL 文の実行待ち状態が発生するおそれがあります。
- adb_sql_exe_max_rthd_num オペランドおよびadb_sys_rthd_num オペランドの指定値を変更した場合，必ず HADB サーバのメモリ所要量を再見積もりしてください。HADB サーバのメモリ所要量の見積もりについては，「6.3　HADB サーバのメモリ所要量の見積もり」を参照してください。

3. HADB サーバを終了する
   adbstop コマンドを実行して，HADB サーバを終了してください。
4. サーバ定義およびクライアント定義を変更する
   次のオペランドの指定値を大きくしてください。
   - サーバ定義およびクライアント定義のadb_sql_exe_max_rthd_num オペランド
   - サーバ定義のadb_sys_rthd_num オペランド
5. HADB サーバを開始する
   adbstart コマンドを実行して，HADB サーバを開始してください。

メモ

- サーバ定義のadb_sql_exe_max_rthd_num オペランド，adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_max_rthd_num オペランド，adb_sys_rthd_num オペランドを参照してください。
- クライアント定義のadb_sql_exe_max_rthd_num オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

(2)　定義系 SQL と操作系 SQL の処理時間を短縮する方法

定義系 SQL，および操作系 SQL の処理時間を短縮する方法について説明します。

対象の定義系 SQL，および操作系 SQL を次に示します。

定義系 SQL

- ALTER TABLE 文（アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する場合）
- DROP TABLE 文

- DROP INDEX 文
- DROP USER 文
- DROP SCHEMA 文
- REVOKE 文（スキーマ操作権限の取り消し）

操作系 SQL

- PURGE CHUNK 文
- TRUNCATE TABLE 文

定義系 SQL，および操作系 SQL の処理時間を短縮する手順を次に示します。

手順

1. DB エリアファイル数を確認する

   adbdbstatus コマンドを実行して出力される「DB エリアのサマリ情報」のうち，次に示す情報を確認してください。

   - Number_of_files（DB エリアファイル数）

   なお，確認する DB エリアは，SQL 文の対象となる表およびインデクスが格納されている DB エリアになります。その DB エリアを構成する DB エリアファイル数のうち，最も大きい数を確認してください。

2. adb_sql_exe_max_rthd_num オペランドの指定値を見直す

   手順 1.で確認したNumber_of_files に出力された各 DB エリアの DB エリアファイル数のうち，最も大きい DB エリアファイル数と，adb_sql_exe_max_rthd_num オペランドの指定値を比較してください。DB エリアファイル数の方が大きい場合は，SQL 文の処理に時間が掛かっているおそれがあります。この場合，adb_sql_exe_max_rthd_num オペランドの指定値を大きくして，DB エリアファイル数と同じ値にしてください。処理性能が向上することがあります。

   重要

   - クライアント定義のadb_sql_exe_max_rthd_num オペランドと，サーバ定義の adb_sql_exe_max_rthd_num オペランドの両方の指定値を確認してください。クライアント定義の指定値だけを大きくした結果，クライアント定義の指定値がサーバ定義の指定値を超えた場合，サーバ定義の指定値が適用されます（変更後のクライアント定義の指定値が適用されません）。該当する場合は，サーバ定義とクライアント定義の両方の adb_sql_exe_max_rthd_num オペランドの指定値を大きくしてください。
   - adb_sql_exe_max_rthd_num オペランドの指定値を大きくした場合，SQL 文の実行待ち状態が発生するおそれがあるため，あわせてサーバ定義のadb_sys_rthd_num オペランドの指定値も大きくしてください。
   - adb_sql_exe_max_rthd_num オペランドおよびadb_sys_rthd_num オペランドの指定値を変更した場合，必ず HADB サーバのメモリ所要量を再見積もりしてください。HADB サー

バのメモリ所要量の見積もりについては，「6.3　HADB サーバのメモリ所要量の見積もり」を参照してください。

3. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義およびクライアント定義を変更する

   次のオペランドの指定値を大きくしてください。

   - サーバ定義およびクライアント定義のadb_sql_exe_max_rthd_num オペランド
   - サーバ定義のadb_sys_rthd_num オペランド

5. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

メモ

- サーバ定義のadb_sql_exe_max_rthd_num オペランド，adb_sys_rthd_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_max_rthd_num オペランド，adb_sys_rthd_num オペランドを参照してください。
- クライアント定義のadb_sql_exe_max_rthd_num オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

13.1.7　ユーザログバッファ面数の拡張

ここでは，サーバ定義のadb_log_usrbuf_num オペランドの見直しによる処理性能の向上について説明します。

ユーザログバッファ面数が満杯となった場合，ユーザログバッファからユーザログファイルに，ユーザログが掃き出されます。ユーザログバッファの掃き出し時に，ユーザログファイルへの I/O が発生するため，更新性能が低下するおそれがあります。そのため，確保するユーザログバッファ面数が適切かどうかを確認してください。

手順

1. ユーザログバッファの満杯による掃き出し回数を確認する

   adbstat コマンドを実行して出力される情報のうち，次に示す情報を確認してください。

   - Log_usrbuf_out_cnt（ユーザログバッファの満杯による掃き出し回数）

> **メモ**
>
> adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析)』を参照してください。

2. adb_log_usrbuf_num オペランドの指定値を見直す

   手順 1.で確認したLog_usrbuf_out_cnt に出力された値が1 以上の場合，ユーザログバッファの掃き出しが発生しています。この場合，サーバ定義のadb_log_usrbuf_num オペランドの指定値を大きくして，ユーザログバッファ面数を拡張してください。ユーザログバッファ面数を拡張することで，更新処理時に，ユーザログバッファの満杯による掃き出しが発生しなくなります。

   > **重要**
   >
   > ユーザログバッファ面数を拡張した場合，HADB サーバ開始時のプロセス共通メモリの所要量が増加します。「6.3.3　HADB サーバ開始時のメモリ所要量の求め方」の「(3)　プロセス共通メモリの所要量の求め方（HADB サーバ開始時)」の「(g)　変数 RECCTL の求め方」を参照して，再見積もりをしてください。

3. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義を変更する

   サーバ定義のadb_log_usrbuf_num オペランドの指定値を大きくしてください。

5. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_log_usrbuf_num オペランドについては，「7.2.3　システムログに関するオペランド（set 形式)」の adb_log_usrbuf_num オペランドを参照してください。

13.1.8　ユーザログファイルの初期化容量の拡張

ここでは，サーバ定義のadb_log_usrfile_size オペランドの見直しによる処理性能の向上について説明します。

ユーザログの増加によってユーザログファイルの容量が初期化した容量を超えた場合，ユーザログファイルは自動的に拡張されます。ユーザログファイルの拡張処理は，更新性能への影響が大きいため，ユーザログファイルの初期化容量が適切かどうかを確認してください。

手順

1. ユーザログファイルの最大使用量を確認する

adbstat コマンドを実行して出力される情報のうち，次に示す情報を確認してください。

- Log_usrfile_max_size（ユーザログファイルの最大使用量）

> **メモ**
>
> adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析）』を参照してください。

2. adb_log_usrfile_size オペランドの指定値を見直す

手順 1.で確認したLog_usrfile_max_size に出力された値が，サーバ定義のadb_log_usrfile_size オペランドの*ユーザログファイルの初期化容量*の指定値よりも大きい場合，ユーザログファイルの拡張が発生しています。この場合，adb_log_usrfile_size オペランドの*ユーザログファイルの初期化容量*を，Log_usrfile_max_size に出力された値を目安に大きくしてください。*ユーザログファイルの初期化容量*を大きくすることで，更新処理時にユーザログファイルの拡張が発生しなくなります。

> **重要**
>
> *ユーザログファイルの初期化容量*を大きくする場合，「6.12　システムログファイルの容量見積もり」を参照して，再見積もりをしてください。

3. HADB サーバを終了する

adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義を変更する

サーバ定義のadb_log_usrfile_size オペランドの*ユーザログファイルの初期化容量*の指定値を大きくしてください。

5. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_log_usrfile_size オペランドについては，「7.2.3　システムログに関するオペランド（set 形式）」の adb_log_usrfile_size オペランドを参照してください。

13.1.9　ユーザログファイルの縮小化容量の見直し

ここでは，サーバ定義のadb_log_usrfile_size オペランドに指定したユーザログファイルの容量の縮小契機の見直しによる処理性能の向上について説明します。

更新トランザクションの決着時，更新トランザクションが使用していたユーザログファイルの容量が，ユーザログファイルの容量の縮小契機に指定した容量を超えている場合，HADB サーバはユーザログファイルを初期化容量に縮小します。ユーザログファイルを初期化容量に縮小する回数が増加すると，更新性能に

影響を及ぼすおそれがあるため，次に示す手順でadb_log_usrfile_size オペランドの指定値を見直してください。

手順

1. ユーザログファイルが初期化容量に縮小された回数を確認する

   adbstat コマンドを実行して出力される情報のうち，次に示す情報を確認してください。

   - Log_usrfile_reduced_cnt（ユーザログファイルが初期化容量に縮小された回数）

   > **メモ**
   >
   > adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat (HADB サーバの統計解析)』を参照してください。

2. adb_log_usrfile_size オペランドの指定値を見直す

   手順 1.で確認したLog_usrfile_reduced_cnt に出力された値が大きい場合，ユーザログファイルの初期化容量への縮小回数が多いことを意味しています。この場合，サーバ定義のadb_log_usrfile_size オペランドの次の指定値を大きくすることを推奨します。

   - *ユーザログファイルの初期化容量*
   - *ユーザログファイルの容量の縮小契機*

   上記の指定値を大きくすると，ユーザログファイルの初期化容量への縮小回数が減ります。

   > **ヒント**
   >
   > adbstat コマンドの実行結果のLog_usrfile_max_size に出力された値が，定常運用中のユーザログファイルの最大使用量になります。この値に，安全係数として 1.5～2 倍した値を adb_log_usrfile_size オペランドのユーザログファイルの容量の縮小契機に指定することを推奨します。これによって，非定期運用などでデータベースの大量更新を行ったときに限り，ユーザログファイルの縮小処理が動作するようにできます。

3. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義を変更する

   サーバ定義のadb_log_usrfile_size オペランドの次の指定値を大きくしてください。

   - *ユーザログファイルの初期化容量*
   - *ユーザログファイルの容量の縮小契機*

5. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_log_usrfile_size オペランドについては，「7.2.3　システムログに関するオペランド（set 形式）」の adb_log_usrfile_size オペランドを参照してください。

13.1.10　DB エリアの自動増分の発生防止

ここでは，DB エリアの自動増分の発生による更新性能の低下を防ぐ方法について説明します。

データ用 DB エリアファイル，ディクショナリ用 DB エリアファイル，およびシステム表用 DB エリアファイルがレギュラーファイルで構成されている場合に，DB エリアの自動増分が発生すると，DB エリアファイル拡張時の更新処理の性能が低下するおそれがあります。そのため，対象の DB エリアで自動増分が発生しているかどうかを確認してください。

adbstat コマンドを実行して出力される情報のうち，次に示す情報を確認してください。

- DBarea_extension_cnt（DB エリアが自動増分した回数）

> **メモ**
>
> adbstat コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbstat（HADB サーバの統計解析）』を参照してください。

DBarea_extension_cnt に出力された値が1 以上の場合，DB エリアの自動増分が発生しています。自動増分による DB エリアファイル拡張時の更新性能の低下を防ぎたい場合は，データ用 DB エリアファイル，ディクショナリ用 DB エリアファイル，およびシステム表用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更して，DB エリアの自動増分が発生しないようにしてください。データ用 DB エリアファイル，ディクショナリ用 DB エリアファイル，およびシステム表用 DB エリアファイルをブロックスペシャルファイルに変更する場合，次のどちらかの方法で対象の DB エリアの容量を求めて，必要なサイズを割り当てたブロックスペシャルファイルを準備してください。

- **対象の DB エリアの容量見積もりを実施する**

  データ用 DB エリアの容量は，「5.8　データ用 DB エリアの容量見積もり」を参照して求めてください。

  ディクショナリ用 DB エリアの容量は，「5.11　ディクショナリ用 DB エリアの容量見積もり」を参照して求めてください。

  システム表用 DB エリアの容量は，「5.12　システム表用 DB エリアの容量見積もり」を参照して求めてください。

- **adbdbstatus コマンドを実行する**

  マニュアル『HADB　コマンドリファレンス』の『adbdbstatus（データベースの状態解析）』の『使用例』の『データベース全体の使用量を調べる（DB エリアのサマリ情報の出力）』を参照して，対象の DB エリアの使用量を求めてください。

必要なサイズを割り当てたブロックスペシャルファイルを準備したあとで，対象の DB エリアファイルをブロックスペシャルファイルに変更します。変更する方法については，「11.10.5　データ用 DB エリアファイルの格納場所を変更する方法」の「(1)　データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する」を参照してください。

13.1.11　作業表用 DB エリアのファイル構成の変更

ここでは，作業表用 DB エリアの自動増分の発生による，SQL 文の処理性能の低下を防ぐ方法について説明します。

作業表用 DB エリアファイルがレギュラーファイルで構成されている場合，まず，adbinit コマンドの実行時に，最も小さい容量で作業表用 DB エリアが作成されます。そのあとで，作業表を作成する SQL 文を実行したときに，DB エリアの自動増分によって，作業表用 DB エリアに必要な容量が確保されます。

作業表用 DB エリアの自動増分が発生すると，作業表を作成する SQL 文の処理性能に影響します。自動増分の発生によって，SQL 文の処理性能が低下することを防ぎたい場合は，作業表用 DB エリアファイルをブロックスペシャルファイルで構成することを検討してください。

adbstat コマンドを実行して出力される「HADB サーバの統計情報」のうち，次に示す情報を確認してください。

- Wrktbl_page_use_max（最大割り当て作業表ページ数）

Wrktbl_page_use_max に出力された値を，「5.9　作業表用 DB エリアの容量見積もり」の計算式中の変数 *WRK_PAGE_NUM* に代入して，作業表用 DB エリアの容量を見積もってください。見積もり結果を基に，必要なサイズを割り当てたブロックスペシャルファイルを準備してください。

必要なサイズを割り当てたブロックスペシャルファイルを準備したあとで，作業表用 DB エリアファイルをブロックスペシャルファイルに変更します。変更する方法については，「11.11.1　作業表用 DB エリアファイルの格納場所を変更する方法」を参照してください。

13.2　バッファの見直しによるSQL文の実行時間の短縮に関するチューニング

ここでは，次に示すバッファに関するオペランドの指定値の見直しによる処理性能の向上について説明します。

- サーバ定義のadbbuff オペランド
- サーバ定義のadb_dbbuff_wrktbl_glb_blk_num オペランド
- クライアント定義のadb_dbbuff_wrktbl_clt_blk_num オペランド

なお，各オペランドの指定値を見直す場合，必ずメモリ所要量を再見積もりしてください。

13.2.1　チューニングを行う前に確認すること

ここでは，SQL文やコマンドの実行時間を短縮するチューニングを行う前に確認する内容について説明します。次の2点を見直してください。

実表やインデクスを格納するデータ用DBエリアの見直し

次の状態であるかどうかを確認してください。

- 1つのデータ用DBエリアには，1つの実表だけが格納されている

  対象のデータ用DBエリアに格納されている実表を調べる場合は，「付録 B.22　ディクショナリ表の検索」の「(14)　DBエリアに格納されている表の表名を調べる場合」を参照してください。

- 1つのデータ用DBエリアには，1つのインデクスだけが格納されている

  対象のデータ用DBエリアに格納されているインデクスを調べる場合は，「付録 B.22　ディクショナリ表の検索」の「(15)　DBエリアに格納されているインデクスのインデクス名を調べる場合」を参照してください。

「5.6.1　データ用DBエリアを設計する際の考慮点」のとおり，1つのデータ用DBエリアに，複数の実表や複数のインデクスが格納されていると，I/Oが集中して性能が低下するおそれがあります。特に，1つのデータ用DBエリアに，ローストア表とカラムストア表を混在して格納しないようにしてください。そのため，1つのデータ用DBエリアには，1つの実表だけ，または1つのインデクスだけを格納するようにしてください。

実表やインデクスが格納されているデータ用DBエリアを見直す場合は，「11.1.16　実表が格納されているデータ用DBエリアを変更する方法」および「11.3.8　インデクスが格納されているデータ用DBエリアを変更する方法」を参照してください。

データ用DBエリアに割り当てるグローバルバッファの見直し

1つのデータ用DBエリアに，専用のグローバルバッファを割り当てているかどうかを確認してください。対象のデータ用DBエリアに割り当てているグローバルバッファを確認する場合は，adbls -d gbuf コマンドを使用してください。adbls -d gbuf コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d gbuf（グローバルバッファの情報表示)』を参照してください。

1 つのグローバルバッファを複数のデータ用 DB エリアに割り当てている場合は，性能が低下するおそれがあります。そのため，1 つのグローバルバッファを 1 つのデータ用 DB エリアだけに割り当てるようにしてください。グローバルバッファを見直す場合は，サーバ定義adbbuff オペランドの指定値を変更してください。サーバ定義adbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」を参照してください。サーバ定義を変更する方法については，「8.5.2　サーバ定義の変更方法」を参照してください。

13.2.2　SQL 文の実行時間を短縮する方法

adbbuff オペランドの指定値を見直すことで，SQL 文の実行時間を短縮できる場合があります。

adbstat コマンドで出力される情報によって，チューニング方法が異なります。

- グローバルバッファの統計情報
- SQL 文の統計情報

まずは，グローバルバッファの統計情報を確認して，チューニングを行ってください。そのあとで，特定の SQL 文の実行時間を短縮したい場合は，SQL 文の統計情報を確認して，チューニングを行ってください。

(1)　グローバルバッファの統計情報による SQL 文の実行時間の短縮

手順

1. グローバルバッファのヒット率を確認する

   adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す情報を確認してください。

   - DBbuff_page_hit_rate（グローバルバッファのページヒット率）

2. グローバルバッファの割り当て方を見直す

   手順 1.で確認したDBbuff_page_hit_rate に出力された値が50 以下の場合，次の 3 点を見直してください。

   - **グローバルバッファのページ数を大きくする**

     DB エリアに割り当てるグローバルバッファのページ数が少な過ぎることが考えられます。adbbuff オペランドの-p オプションに指定した値を大きくしてください。

   - **専用のグローバルバッファを割り当てる**

     複数の DB エリアに対して，1 つのグローバルバッファを割り当てている場合，adbbuff オペランドで，各 DB エリアに専用のグローバルバッファを割り当ててください。

   - **テーブルスキャンバッファを確保する**

     テーブルスキャンを実行する SQL 文がある場合は，adbbuff オペランドの-v オプションを指定して，テーブルスキャンバッファを確保してください。

3. HADB サーバを終了する

adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義を変更する

   サーバ定義のadbbuff オペランドの指定を，手順 2.で決めた方針で変更してください。

5. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドを参照してください。

(2)　SQL 文の統計情報による SQL 文の実行時間の短縮

手順

1. グローバルバッファのヒット率を確認する

   adbstat コマンドを実行して出力される「SQL 文の統計情報」のうち，次に示す情報を確認してください。

   - DBbuff_page_hit_rate（グローバルバッファのページヒット率）

2. グローバルバッファの割り当て方を見直す

   手順 1.で確認したDBbuff_page_hit_rate に出力された値が50 以下の場合，次の 3 点を見直してください。

   - **グローバルバッファのページ数を大きくする**

     DB エリアに割り当てるグローバルバッファのページ数が少な過ぎることが考えられます。adbbuff オペランドの-p オプションに指定した値を大きくしてください。

   - **専用のグローバルバッファを割り当てる**

     複数の DB エリアに対して，1 つのグローバルバッファを割り当てている場合，「SQL 文の統計情報」のDBbuff_page_request_cnt（グローバルバッファのページ要求回数）を確認してください。そして，ページ要求回数が多い DB エリアに，adbbuff オペランドで専用のグローバルバッファを割り当ててください。

   - **テーブルスキャンバッファを確保する**

     テーブルスキャンを実行する SQL 文がある場合は，adbbuff オペランドの-v オプションを指定して，テーブルスキャンバッファを確保してください。

3. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義を変更する

   サーバ定義のadbbuff オペランドの指定を，手順 2.で決めた方針で変更してください。

5. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドを参照してください。

13.2.3　グローバル作業表を作成する SQL 文の実行時間を短縮する方法

グローバル作業表を作成する SQL 文を実行する場合，グローバル作業表用のグローバルバッファのページ数を増やすことで，SQL 文の実行時間を短縮できることがあります。

グローバル作業表を作成する SQL 文については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『SQL を実行した場合に作成される作業表について』を参照してください。

adbstat コマンドで出力される情報によって，チューニング方法が異なります。

- グローバルバッファの統計情報
- SQL 文の統計情報

まずは，グローバルバッファの統計情報を確認して，チューニングを行ってください。そのあとで，特定の SQL 文の実行時間を短縮したい場合は，SQL 文の統計情報を確認して，チューニングを行ってください。

(1)　グローバルバッファの統計情報によるグローバル作業表を作成する SQL 文の実行時間の短縮

手順

1. グローバルバッファからのファイル書き込み回数を確認する

   adbstat コマンドを実行して出力されるグローバルバッファの統計情報のうち，まず，次に示す情報を確認してください。

   - DBbuff_name（グローバルバッファ名）

   DBbuff_name がADBWRK となっているグローバルバッファについて，次に示す情報を確認してください。

   - DBbuff_page_write_cnt（グローバルバッファからのファイル書き込み回数）

   DBbuff_page_write_cnt に出力された値が 1 以上の場合，グローバルバッファからファイルへの書き込みが発生しています。該当する場合は，手順 2.に進んでください。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

サーバ定義のadb_dbbuff_wrktbl_glb_blk_num オペランドの指定値を大きくしてください。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. グローバル作業表を作成する SQL 文を再実行する

   グローバル作業表を作成する SQL 文を再実行したあとに，手順 1.を実施して，DBbuff_page_write_cnt に出力された値が減っているかどうかを確認してください。

6. 手順 1.～手順 5.を繰り返す

   手順 1.～手順 5.を繰り返して，DBbuff_page_write_cnt に出力された値が0 になるまで調整すると，グローバル作業表を作成する SQL 文の実行時間を短縮できることがあります。

> **メモ**
>
> サーバ定義のadb_dbbuff_wrktbl_glb_blk_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_dbbuff_wrktbl_glb_blk_num オペランドを参照してください。

(2)　SQL 文の統計情報によるグローバル作業表を作成する SQL 文の実行時間の短縮

手順

1. グローバルバッファからのファイル書き込み回数を確認する

   adbstat コマンドを実行して出力される「SQL 文の統計情報」のうち，まず，次に示す情報を確認してください。

   - DBbuff_name（グローバルバッファ名）

   DBbuff_name がADBWRK となっているグローバルバッファについて，次に示す情報を確認してください。

   - DBbuff_page_write_cnt（グローバルバッファからのファイル書き込み回数）
   - DBbuff_page_put_cnt（グローバルバッファへの書き込み回数）

2. グローバルバッファからファイルへの書き込みが多発していないか確認する

   手順 1.で確認したDBbuff_page_write_cnt とDBbuff_page_put_cnt の関係が，次に示す計算式を満たしている場合，グローバルバッファからファイルへの書き込みが多発しているおそれがあります。

   計算式

   DBbuff_page_write_cntに出力された値 ＞ *DBbuff_page_put_cntに出力された値*÷2

   計算式を満たしている場合，手順 3.に進んでください。

3. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

4. サーバ定義を変更する

   サーバ定義のadb_dbbuff_wrktbl_glb_blk_num オペランドの指定値を大きくしてください。

5. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

6. グローバル作業表を作成する SQL 文を再実行する

   グローバル作業表を作成する SQL 文を再実行したあとに，手順 1.を実施して，DBbuff_page_write_cnt に出力された値が，計算式を満たさないことを確認してください。

7. 手順 1.～手順 6.を繰り返す

   手順 1.～手順 6.を繰り返して，DBbuff_page_write_cnt に出力された値が計算式を満たさなくなるまで調整すると，グローバル作業表を作成する SQL 文の実行時間を短縮できることがあります。

> **メモ**
>
> サーバ定義のadb_dbbuff_wrktbl_glb_blk_num オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_dbbuff_wrktbl_glb_blk_num オペランドを参照してください。

13.2.4　ローカル作業表を作成する SQL 文の実行時間を短縮する方法

ローカル作業表を作成する SQL 文を実行する場合，ローカル作業表用バッファのページ数を増やすことで，SQL 文の実行時間を短縮できることがあります。

ローカル作業表を作成する SQL 文については，マニュアル『HADB　AP 開発ガイド』の『AP の性能向上に関する設計』の『作業表が作成される SQL を実行する際の考慮点』の『SQL を実行した場合に作成される作業表について』を参照してください。

adbstat コマンドで出力される情報によって，チューニング方法が異なります。

- コネクションの稼働情報
- SQL 文の統計情報

まずは，コネクションの稼働情報を確認して，チューニングを行ってください。そのあとで，特定の SQL 文の実行時間を短縮したい場合は，SQL 文の統計情報を確認して，チューニングを行ってください。

> **メモ**
>
> クライアント定義adb_dbbuff_wrktbl_clt_blk_num オペランドを指定していないコネクションの場合は，HADB サーバの稼働中にadbmodbuff コマンドで，サーバ定義 adb_dbbuff_wrktbl_clt_blk_num オペランドで指定したローカル作業表用バッファのページ数を変更できます。「11.12.1　ローカル作業表用バッファを変更する方法」を参照してください。

(1) コネクションの稼働情報によるローカル作業表を作成するSQL文の実行時間の短縮

手順

1. ローカル作業表用バッファからのファイル書き込み回数を確認する

   adbstatコマンドを実行して出力される「コネクションの稼働情報」のうち，次に示す情報を確認してください。

   - DBbuff_wrktbl_clt_write_cnt（ローカル作業表用バッファからのファイル書き込み回数）

2. クライアント定義を変更する

   手順1.で確認したDBbuff_wrktbl_clt_write_cntに出力された値が1以上の場合，ローカル作業表用バッファからファイルへの書き込みが発生しています。クライアント定義のadb_dbbuff_wrktbl_clt_blk_numオペランドの指定値を大きくしてください。

   クライアント定義を変更する場合は，マニュアル『HADB　AP開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

3. ローカル作業表を作成するSQL文を再実行する

   ローカル作業表を作成するSQL文を再実行したあとに，手順1.を実施して，DBbuff_wrktbl_clt_write_cntに出力された値が減っているかどうかを確認してください。

4. 手順1.～手順3.を繰り返す

   手順1.～手順3.を繰り返して，DBbuff_wrktbl_clt_write_cntに出力された値が0になるまで調整すると，ローカル作業表を作成するSQL文の実行時間を短縮できることがあります。

> **メモ**
>
> クライアント定義のadb_dbbuff_wrktbl_clt_blk_numオペランドについては，マニュアル『HADB　AP開発ガイド』の『性能に関するオペランド』を参照してください。

(2) SQL文の統計情報によるローカル作業表を作成するSQL文の実行時間の短縮

手順

1. ローカル作業表用バッファからのファイル書き込み回数を確認する

   adbstatコマンドを実行して出力される「SQL文の統計情報」のうち，次に示す情報を確認してください。

   - DBbuff_wrktbl_clt_write_cnt（ローカル作業表用バッファからのファイル書き込み回数）
   - DBbuff_wrktbl_clt_put_cnt（ローカル作業表用バッファへの書き込み回数）

2. クライアント定義を変更する

手順 1.で確認したDBbuff_wrktbl_clt_write_cnt とDBbuff_wrktbl_clt_put_cnt の情報の関係が，次に示す計算式を満たしている場合，ローカル作業表用バッファからファイルへの書き込みが多発しているおそれがあります。

計算式

DBbuff_wrktbl_clt_write_cntに出力された値 ＞ DBbuff_wrktbl_clt_put_cntに出力された値÷2

計算式を満たしている場合，クライアント定義のadb_dbbuff_wrktbl_clt_blk_num オペランドの指定値を大きくしてください。

クライアント定義を変更する場合は，マニュアル『HADB　AP 開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

3. ローカル作業表を作成する SQL 文を再実行する

   ローカル作業表を作成する SQL 文を再実行したあとに，手順 1.を実施して，DBbuff_wrktbl_clt_write_cnt に出力された値が，計算式を満たさないことを確認してください。

4. 手順 1.～手順 3.を繰り返す

   手順 1.～手順 3.を繰り返して，DBbuff_wrktbl_clt_write_cnt に出力された値が計算式を満たさなくなるまで調整すると，ローカル作業表を作成する SQL 文の実行時間を短縮できることがあります。

メモ

クライアント定義のadb_dbbuff_wrktbl_clt_blk_num オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法

adbbuff オペランドの-v オプションを指定した場合に，-v オプションの指定値を見直すことで，テーブルスキャンを実行する SQL 文の実行時間を短縮できることがあります。

adbstat コマンドで出力される情報によって，チューニング方法が異なります。

- グローバルバッファの統計情報
- SQL 文の統計情報

最初に，グローバルバッファの統計情報を確認して，チューニングを行ってください。そのあとで，特定の SQL 文の実行時間を短縮したい場合は，SQL 文の統計情報を確認して，チューニングを行ってください。

(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮

手順

1. テーブルスキャンバッファでバッファ不足が発生した回数，およびテーブルスキャンバッファのページヒット率を確認する

   adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す情報を確認してください。

   - DBbuff_tblscan_failed_cnt（テーブルスキャンバッファでバッファ不足が発生した回数）
   - DBbuff_tblscan_hit_rate（テーブルスキャンバッファのページヒット率）

   DBbuff_tblscan_failed_cnt に出力された値が1 以上の場合で，かつDBbuff_tblscan_hit_rate に出力された値が低い場合，テーブルスキャンバッファの面数が不足しています。そのため，テーブルスキャン実行時にグローバルバッファでの読み込みが発生しています。該当する場合は，手順 2.に進んでください。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドの-v オプションの指定値に，サーバ定義のadb_sys_rthd_num オペランドの指定値分を加算してください。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. テーブルスキャンを実行する SQL 文を再実行する

   テーブルスキャンを実行する SQL 文を再実行したあとに，手順 1.を実施して，DBbuff_tblscan_failed_cnt に出力された値が減っているかどうかを確認してください。また，DBbuff_tblscan_hit_rate に出力された値が増えているかどうかを確認してください。

6. 手順 1.～手順 5.を繰り返す

   手順 1.～手順 5.を繰り返して，DBbuff_tblscan_failed_cnt に出力された値が0 に近い値になるまで調整すると，DBbuff_tblscan_hit_rate に出力された値が増えてテーブルスキャンバッファのページヒット率が改善されます。そのため，テーブルスキャンを実行する SQL 文の実行時間を短縮できる場合があります。

メモ

サーバ定義のadbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドを参照してください。

(2)　SQL 文の統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮

手順

1. テーブルスキャンバッファでバッファ不足が発生した回数を確認する

adbstat コマンドを実行して出力される「SQL 文の統計情報」のうち，次に示す情報を確認してください。

- DBbuff_tblscan_failed_cnt（テーブルスキャンバッファでバッファ不足が発生した回数）
- DBbuff_tblscan_insufficient_buff_num（テーブルスキャンバッファの不足バッファ面数）

DBbuff_tblscan_failed_cnt に出力された値が1 以上の場合，テーブルスキャンバッファの面数が不足しています。そのため，テーブルスキャン実行時にグローバルバッファでの読み込みが発生しています。該当する場合は，手順 2.に進んでください。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドの-v オプションの指定値に，DBbuff_tblscan_insufficient_buff_num に出力された値分を加算してください。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. テーブルスキャンを実行する SQL 文を再実行する

   テーブルスキャンを実行する SQL 文の実行時間を短縮できる場合があります。

メモ

サーバ定義のadbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドを参照してください。

13.3　コマンドの実行時間の短縮に関するチューニング

ここでは，コマンドの実行時間を短縮する方法について説明します。

> **メモ**
>
> コマンドの実行時間を短縮するチューニングを行う前に，データ用 DB エリアやグローバルバッファの見直しを行ってください。詳細については，「13.2.1　チューニングを行う前に確認すること」を参照してください。

13.3.1　adbimport コマンドの実行時間の短縮

adbimport コマンドのインポートオプションの性能に関するオプションに適切な値を指定することで，インポートの性能を向上させ，adbimport コマンドの実行時間を短縮できることがあります。

adbimport コマンドのインポートオプションの指定形式については，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート）』の『adbimport コマンドの指定形式』の『インポートオプションの形式』を参照してください。

13.3.2　adbidxrebuild コマンドの実行時間の短縮

ここでは，adbidxrebuild コマンドの実行時間を短縮する方法について説明します。

adbidxrebuild コマンドを実行する場合，サーバ定義adbbuff オペランドの-v オプションを指定することで，adbidxrebuild コマンドの実行時間を短縮できることがあります。-v オプションを指定しているときは，指定値を見直すことで，adbidxrebuild コマンドの実行時間を短縮できることがあります。

手順

1. -v オプションの指定値を見積もる

   「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」にあるサーバ定義の adbbuff オペランドの-v オプションに指定する値を見積もってください。その際，変数 *sql_rthd_num* には，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値ではなく，次の値を代入してください。

   ↓(*インデクス再作成オプションadb_idxrebuild_rthd_numの指定値* −1)÷2↓

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

サーバ定義のadbbuff オペランドに-v オプションを追加するか，または-v オプションの指定値を変更してください。手順 1.で見積もった値を-v オプションに指定します。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. adbidxrebuild コマンドを実行する

6. adbstat コマンドを実行して，グローバルバッファの統計情報を出力する

   「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」を参照して，adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のDBbuff_tblscan_failed_cnt に出力される値が 0 に近い値になるまで調整してください。調整することで，adbidxrebuild コマンドの実行時間を短縮できる場合があります。

   なお，参照先の「**テーブルスキャンを実行する SQL 文**」は，「**adbidxrebuild コマンド**」に読み替えてください。

13.3.3　adbgetcst コマンドの実行時間の短縮

ここでは，adbgetcst コマンドの実行時間を短縮する方法について説明します。

adbgetcst コマンドを実行する場合，サーバ定義adbbuff オペランドの-v オプションを指定することで，adbgetcst コマンドの実行時間を短縮できることがあります。-v オプションを指定しているときは，指定値を見直すことで，adbgetcst コマンドの実行時間を短縮できることがあります。

手順

1. -v オプションの指定値を見積もる

   「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」にあるサーバ定義の adbbuff オペランドの-v オプションに指定する値を見積もってください。その際，変数 *sql_rthd_num* には，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値ではなく，次の値を代入してください。

コスト情報収集オプションadb_getcst_rthd_numの指定値 －1

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドに-v オプションを追加するか，または-v オプションの指定値を変更してください。手順 1.で見積もった値を-v オプションに指定します。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. adbgetcst コマンドを実行する

6. adbstat コマンドを実行して，グローバルバッファの統計情報を出力する

   「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」を参照して，adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のDBbuff_tblscan_failed_cnt に出力される値が 0 に近い値になるまで調整してください。調整することで，adbgetcst コマンドの実行時間を短縮できる場合があります。

   なお，参照先の「**テーブルスキャンを実行する SQL 文**」は，「**adbgetcst コマンド**」に読み替えてください。

13.3.4　adbexport コマンドの実行時間の短縮

ここでは，adbexport コマンドの実行時間を短縮する方法について説明します。

adbexport コマンドで SQL 文の検索結果をエクスポートする場合，次に示す項目を確認することで，adbexport コマンドの実行時間を短縮できることがあります。

- 「13.2.2　SQL 文の実行時間を短縮する方法」
- 「13.2.3　グローバル作業表を作成する SQL 文の実行時間を短縮する方法」
- 「13.2.4　ローカル作業表を作成する SQL 文の実行時間を短縮する方法」

  参照先の「**クライアント定義 adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値**」は，「**エクスポートオプション adb_export_wrktbl_blk_num の指定値**」に読み替えてください。また，「**ローカル作業表を作成する SQL 文**」は，「**adbexport コマンド**」に読み替えてください。
- 「13.5　ハッシュテーブル領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング」
- 「13.8　ハッシュフィルタ領域サイズの見直しによる SQL 文の実行時間の短縮に関するチューニング」

上記の項目を参照して，チューニングしてください。

また，サーバ定義adbbuff オペランドの-v オプションを指定することで，adbexport コマンドの実行時間を短縮できることがあります。-v オプションを指定しているときは，指定値を見直すことで，adbexport コマンドの実行時間を短縮できることがあります。

手順

1. -v オプションの指定値を見積もる

   「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」にあるサーバ定義の adbbuff オペランドの-v オプションに指定する値を見積もってください。その際，変数 *sql_rthd_num* には，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値ではなく，次の値を代入してください。

   ↓(*エクスポートオプションadb_export_rthd_numの指定値* −1)÷2↓

2. HADB サーバを終了する

adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドに-v オプションを追加するか，または-v オプションの指定値を変更してください。手順 1.で見積もった値を-v オプションに指定します。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. adbexport コマンドを実行する

6. adbstat コマンドを実行して，グローバルバッファの統計情報を出力する

   「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」を参照して，adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のDBbuff_tblscan_failed_cnt に出力される値が 0 に近い値になるまで調整してください。調整することで，adbexport コマンドの実行時間を短縮できる場合があります。

   なお，参照先の「**テーブルスキャンを実行する SQL 文**」は，「**adbexport コマンド**」に読み替えてください。

13.3.5　adbmergechunk コマンドの実行時間の短縮

ここでは，adbmergechunk コマンドの実行時間を短縮する方法について説明します。

adbmergechunk コマンドを実行する場合，サーバ定義adbbuff オペランドの-v オプションを指定することで，adbmergechunk コマンドの実行時間を短縮できることがあります。-v オプションを指定しているときは，指定値を見直すことで，adbmergechunk コマンドの実行時間を短縮できることがあります。

手順

1. -v オプションの指定値を見積もる

   「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」にあるサーバ定義の adbbuff オペランドの-v オプションに指定する値を見積もってください。その際，変数 *sql_rthd_num* には，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値ではなく，次の値を代入してください。

   ```
   ↓(マージチャンクオプションadb_mergechunk_rthd_numの指定値 −1)÷2↓
   ```

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドに-v オプションを追加するか，または-v オプションの指定値を変更してください。手順 1.で見積もった値を-v オプションに指定します。

4. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

5. adbmergechunk コマンドを実行する

6. adbstat コマンドを実行して，グローバルバッファの統計情報を出力する

   「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」を参照して，adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のDBbuff_tblscan_failed_cnt に出力される値が 0 に近い値になるまで調整してください。調整することで，adbmergechunk コマンドの実行時間を短縮できる場合があります。

   なお，参照先の「**テーブルスキャンを実行する SQL 文**」は，「**adbmergechunk コマンド**」に読み替えてください。

13.3.6　adbarchivechunk コマンドの実行時間の短縮

ここでは，adbarchivechunk コマンドの実行時間を短縮する方法について説明します。

adbarchivechunk コマンドを実行する場合，サーバ定義adbbuff オペランドの-v オプションを指定することで，adbarchivechunk コマンドの実行時間を短縮できることがあります。-v オプションを指定しているときは，指定値を見直すことで，adbarchivechunk コマンドの実行時間を短縮できることがあります。

手順

1. -v オプションの指定値を見積もる

   「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」にあるサーバ定義の adbbuff オペランドの-v オプションに指定する値を見積もってください。その際，変数 *sql_rthd_num* には，サーバ定義adb_sql_exe_max_rthd_num オペランドの指定値ではなく，次の値を代入してください。

   ↓(*アーカイブチャンクオプションadb_arcv_rthd_numの指定値* −1)÷2↓

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドに-v オプションを追加するか，または-v オプションの指定値を変更してください。手順 1.で見積もった値を-v オプションに指定します。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. adbarchivechunk コマンドを実行する

6. adbstat コマンドを実行して，グローバルバッファの統計情報を出力する

   「13.2.5　テーブルスキャンを実行する SQL 文の実行時間を短縮する方法」の「(1)　グローバルバッファの統計情報によるテーブルスキャンを実行する SQL 文の実行時間の短縮」を参照して，adbstat

コマンドを実行して出力される「グローバルバッファの統計情報」のDBbuff_tblscan_failed_cnt に出力される値が0 に近い値になるまで調整してください。調整することで，adbarchivechunk コマンドの実行時間を短縮できる場合があります。

なお，参照先の「**テーブルスキャンを実行する SQL 文**」は，「adbarchivechunk コマンド」に読み替えてください。

13.4　メモリ使用量の削減に関するチューニング

ここでは，adbstat コマンドで出力される統計情報や，SQL トレースファイルに出力される SQL トレース情報を確認して，HADB サーバが使用するメモリ量を削減する方法について説明します。

13.4.1　HugePages を適用した共有メモリの使用量の削減

HADB サーバの共有メモリにHugePages を適用した場合，必要以上にメモリを確保することがあります。その結果，Linux のカーネルなどの性能が低下するおそれがあります。

HADB サーバの共有メモリにHugePages を適用した場合は，確保しているメモリ量が適切かどうかを確認してください。チューニング方法を次に示します。

手順

1. 全メモリの最大使用量を確認する

   adbstat コマンドを実行して出力される「HADB サーバの統計情報」のうち，次に示す情報を確認してください。

   - Total_memory_max_size（全メモリの最大使用量）

2. カーネルパラメタを確認する

   HADB サーバをインストールしたマシンの OS に設定したカーネルパラメタvm.nr_hugepages の指定値を確認してください。

3. 確保しているメモリ量が適切かどうかを確認する

   手順 1.で確認したTotal_memory_max_size に出力された値よりも，手順 2.で確認したvm.nr_hugepages の指定値が大きい場合，必要以上にメモリを確保しています。vm.nr_hugepages の指定値を，Total_memory_max_size に出力された値にあわせて小さくしてください。

   > **重要**
   >
   > adbstop コマンドを実行して HADB サーバを終了したあとに，vm.nr_hugepages の指定値を変更してください。

4. OS を再起動する

5. サーバ定義を変更する

   手順 3.でvm.nr_hugepages の指定値を変更した場合，サーバ定義のadb_sys_memory_limit オペランドの指定値を変更してください。

6. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

メモ

サーバ定義のadb_sys_memory_limit オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_memory_limit オペランドを参照してください。

13.4.2　グローバルバッファの見直しによるメモリ使用量の削減

使用していないグローバルバッファの割り当てを解除したり，グローバルバッファに指定したページ数を見直したりすることで，HADB サーバが使用するメモリ量を削減できます。

(1)　使用していないグローバルバッファの割り当て解除

ここでは，レンジインデクス専用のグローバルバッファ，およびテーブルスキャンバッファを割り当てている場合のチューニング方法を次に示します。

■adbbuff オペランドに-a オプションを指定している場合

1. レンジインデクスバッファのページ要求回数を確認する

   adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す情報に出力された値が0 かどうかを確認してください。

   - DBbuff_page_rng_request_cnt（レンジインデクスバッファのページ要求回数）

   DBbuff_page_rng_request_cnt に出力される値が0 の場合，レンジインデクスが定義されていない，またはレンジインデクスを使用しない DB エリアにレンジインデクス専用のグローバルバッファを割り当てているおそれがあります。

   この場合は，adbbuff オペランドの-a オプションを削除してください。HADB サーバが使用するメモリ量を削減できます。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドの-a オプションを削除してください。

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

■adbbuff オペランドに-v オプションを指定している場合

1. テーブルスキャンバッファのページ要求回数を確認する

   adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す情報に出力された値が0 かどうかを確認してください。

   - DBbuff_tblscan_request_cnt（テーブルスキャンバッファのページ要求回数）

DBbuff_tblscan_request_cnt に出力される値が0 の場合，テーブルスキャンを実行する SQL 文を実行しない DB エリアにテーブルスキャンバッファを割り当てているおそれがあります。

この場合は，adbbuff オペランドの-v オプションを削除してください。HADB サーバが使用するメモリ量を削減できます。

2. HADB サーバを終了する

adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

サーバ定義のadbbuff オペランドの-v オプションを削除してください。

4. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドを参照してください。

(2)　グローバルバッファに指定したページ数の見直し

ここでは，グローバルバッファに指定したページ数が多過ぎる場合のチューニング方法を次に示します。

■adbbuff オペランドに-p オプションを指定している場合

1. グローバルバッファ面数を確認する

adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す 2 つの情報に出力された値を確認してください。

- DBbuff_page_use_cnt（割り当て済みのグローバルバッファ面数）
- DBbuff_page_num（総グローバルバッファ面数）

DBbuff_page_use_cnt に出力された値が，DBbuff_page_num に出力された値よりも小さい場合，adbbuff オペランドの-p オプションに指定した値が大き過ぎるおそれがあります。該当する場合は，手順 2.に進んでください。

2. HADB サーバを終了する

adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

サーバ定義のadbbuff オペランドの-p オプションの指定値を，次の計算式で求めた値に変更してください。HADB サーバが使用するメモリ量を削減できます。

```
DBbuff_page_use_cntに出力された値×1.2
```

4. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

■adbbuff オペランドに-a オプションを指定している場合

1. レンジインデクスバッファ面数を確認する

   adbstat コマンドを実行して出力される「グローバルバッファの統計情報」のうち，次に示す 2 つの情報に出力された値を確認してください。

   - DBbuff_page_rng_use_cnt（割り当て済みのレンジインデクスバッファ面数）
   - DBbuff_page_rng_num（総レンジインデクスバッファ面数）

   DBbuff_page_rng_use_cnt に出力された値が，DBbuff_page_rng_num に出力された値よりも小さい場合，adbbuff オペランドの-a オプションに指定した値が大き過ぎるおそれがあります。該当する場合は，手順 2.に進んでください。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadbbuff オペランドの-a オプションの指定値を，次の計算式で求めた値に変更してください。HADB サーバが使用するメモリ量を削減できます。

   ```
   DBbuff_page_rng_use_cntに出力された値×1.2
   ```

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」の adbbuff オペランドを参照してください。

13.4.3　ローカル作業表用バッファの見直しによるメモリ使用量の削減

SQL トレース情報に出力された「アクセスパスの統計情報」を確認することで，ローカル作業表用バッファのページ数を見直す方法について説明します。

ローカル作業表用バッファのページ数を見直すことで，メモリ使用量を削減できることがあります。

手順

1. アクセスパスの統計情報を確認する

   SQL トレース情報に出力された「アクセスパスの統計情報」を確認してください。アクセスパスの統計情報に出力された「データアクセス情報」のうち，次に示す出力項目を確認してください。

   - Data_dbbuff_wrktbl_clt_write_cnt（ローカル作業表用バッファからファイルへの書き込み回数）
   - Data_wrktbl_page_use_max（割り当てた作業表のページ数のうち，最も大きいページ数）

出力項目の値を確認したら，手順 2.に進んでください。

2. オペランドの指定値を確認する

次に示すオペランドの指定値を確認してください。

- adb_sql_exe_max_rthd_num
- adb_dbbuff_wrktbl_clt_blk_num

各オペランドの指定値を確認したら，手順 3.に進んでください。

3. ローカル作業表用バッファの見直しが必要かどうかを確認する

確認した値を基に，次に示す計算式を求めてください。

計算式（ローカル作業表用バッファの見直しが必要かどうかの確認）

```
A＝0，かつ B÷C>D

  変数の説明
    A：アクセスパスの統計情報のData_dbbuff_wrktbl_clt_write_cntの値
    B：アクセスパスの統計情報のData_wrktbl_page_use_maxの値
    C：adb_sql_exe_max_rthd_numオペランドの値
    D：adb_dbbuff_wrktbl_clt_blk_numオペランドの値
```

上記の計算式を満たす場合，確保しているローカル作業表用バッファのページ数が，実際に必要なページ数よりも多いおそれがあります。この場合，手順 4.に進んでください。

4. クライアント定義を変更する

該当するコネクションのクライアント定義のadb_dbbuff_wrktbl_clt_blk_num オペランドの指定値を，次に示す計算式で求めた値に変更してください。

計算式（adb_dbbuff_wrktbl_clt_blk_num オペランドの値）

```
A÷B×1.5

  変数の説明
    A：アクセスパスの統計情報のData_wrktbl_page_use_maxの値
    B：adb_sql_exe_max_rthd_numオペランドの値
```

上記の計算式で求めた値に，該当するコネクションのクライアント定義の adb_dbbuff_wrktbl_clt_blk_num オペランドの指定値を変更すると，処理性能に影響を与えることなく，メモリ使用量を削減することができます。

クライアント定義を変更する場合は，マニュアル『HADB　AP 開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

メモ

クライアント定義のadb_dbbuff_wrktbl_clt_blk_num オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

13.5 ハッシュテーブル領域サイズの見直しによるSQL文の実行時間の短縮に関するチューニング

ここでは，次に示すオペランドに指定したハッシュテーブル領域サイズの見直しによる，SQL文の実行時間を短縮する方法について説明します。

- サーバ定義のadb_sql_exe_hashtbl_area_sizeオペランド
- クライアント定義のadb_sql_exe_hashtbl_area_sizeオペランド

なお，adbexportコマンドにSQL文を指定して実行した場合でも，同様にSQL文の実行時間を短縮できます。その場合は，adb_sql_exe_hashtbl_area_sizeオペランドをエクスポートオプションadb_export_hashtbl_area_sizeに読み替えてください。

ハッシュテーブル領域を使用する次の処理が適用されるSQL文を実行する場合に，ハッシュテーブル領域サイズを大きくすると，SQL文の実行時間を短縮できることがあります。

- 表の結合方式のハッシュジョイン
- グループ化の処理方式のグローバルハッシュグループ化
- 副問合せの処理方式のハッシュ実行
- SELECT DISTINCTの処理方式のハッシュ実行
- 集合演算の処理方式のハッシュ実行

手順

1. ハッシュテーブル領域の領域不足が発生した処理の有無を確認する

   SQLトレース情報に出力された「アクセスパスの統計情報」を確認してください。アクセスパスの統計情報に出力された「ハッシュテーブル領域に関する情報」のうち，次に示す出力項目がYとなっている処理があるかどうかを確認してください。

   - Hashtbl_area_shortage（ハッシュテーブル領域の領域不足の発生有無）

   Yとなっている処理があった場合は，手順2.に進んでください。

2. KFAA51130-Wメッセージの出力有無を確認する

   手順1.でYとなっている処理がある場合，次の2つを突き合わせてください。

   - 該当する処理の「SQL文の実行情報」に出力されるmessage_log_info
   - サーバメッセージログファイル

   突き合わせの結果，サーバメッセージログファイルに，KFAA51130-Wメッセージが出力されているかどうかを確認してください。KFAA51130-Wメッセージが出力されている場合は，手順3.に進んでください。

3. adb_sql_exe_hashtbl_area_sizeオペランドの指定値を見直す

手順 2.でKFAA51130-W メッセージが出力されている場合は，ハッシュテーブル領域内で処理できなかったデータを，複数回にわたって作業表に格納して処理している可能性があります。そのため，SQL 文の実行時間が長くなっているおそれがあります。

この場合，サーバ定義およびクライアント定義のadb_sql_exe_hashtbl_area_size オペランドの指定値を大きくしてください。ハッシュテーブル領域を使用する処理が適用される SQL 文の実行時間を短縮できることがあります。

重要

- クライアント定義のadb_sql_exe_hashtbl_area_size オペランドと，サーバ定義の adb_sql_exe_hashtbl_area_size オペランドの両方の指定値を確認してください。クライアント定義の指定値だけを大きくした結果，クライアント定義の指定値がサーバ定義の指定値を超えた場合，サーバ定義の指定値が適用されます（変更後のクライアント定義の指定値が適用されません）。該当する場合は，サーバ定義とクライアント定義の両方のadb_sql_exe_hashtbl_area_size オペランドの指定値を大きくしてください。
- adb_sql_exe_hashtbl_area_size オペランドの指定値を変更した場合，必ず HADB サーバのメモリ所要量を再見積もりしてください。HADB サーバのメモリ所要量の再見積もりについては，「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」の変数 *HASHTBL_PROC_SIZE* を参照してください。

4. サーバ定義およびクライアント定義を変更する

   サーバ定義およびクライアント定義のadb_sql_exe_hashtbl_area_size オペランドの指定値を大きくしてください。

   - サーバ定義を変更する場合

     adbstop コマンドで HADB サーバをいったん終了してください。そのあとに，サーバ定義を変更し，adbstart コマンドで HADB サーバを開始してください。

   - クライアント定義を変更する場合

     マニュアル『HADB　AP 開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

5. SQL 文を再実行する

   SQL 文を再実行して，SQL 文の実行時間が短縮されるかどうかを確認してください。

メモ

- アクセスパスの統計情報に出力された「ハッシュテーブル領域に関する情報」については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」の「(2)　アクセスパスの統計情報に出力される項目」の「(e)　ハッシュテーブル領域に関する情報」を参照してください。
- サーバ定義のadb_sql_exe_hashtbl_area_size オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashtbl_area_size オペランドを参照してください。

- クライアント定義のadb_sql_exe_hashtbl_area_size オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

13.6 ハッシュグループ化領域サイズの見直しによるSQL文の実行時間の短縮に関するチューニング

ここでは，次に示すオペランドに指定したハッシュグループ化領域サイズの見直しによる，SQL文の実行時間を短縮する方法について説明します。

- サーバ定義のadb_sql_exe_hashgrp_area_sizeオペランド
- クライアント定義のadb_sql_exe_hashgrp_area_sizeオペランド

ローカルハッシュグループ化が適用されるSQL文を実行する場合に，ハッシュグループ化領域サイズを大きくすると，SQL文の実行時間を短縮できることがあります。

手順

1. ハッシュグループ化領域の領域不足が発生した処理の有無を確認する

   SQLトレース情報に出力された「アクセスパスの統計情報」を確認してください。アクセスパスの統計情報に出力された「ハッシュグループ化領域に関する情報」のうち，次に示す出力項目にYがあるかどうかを確認してください。

   - Hashgrp_area_shortage（ハッシュグループ化領域の領域不足の発生有無）

   もし，Yがあった場合で，かつローカルハッシュグループ化の処理がある場合は，手順2.に進んでください。

2. adb_sql_exe_hashgrp_area_sizeオペランドの指定値を見直す

   手順1.で，ハッシュグループ化領域の領域不足が発生したローカルハッシュグループ化の処理がある場合は，ハッシュグループ化領域内で処理できなかったデータを，作業表に格納して処理している可能性があります。そのため，SQL文の実行時間が長くなっているおそれがあります。

   この場合，手順1.で確認したHashgrp_area_shortageがYとなっているすべてのローカルハッシュグループ化の処理について，「アクセスパスの統計情報」に出力された「ハッシュグループ化領域に関する情報」のうち，次に示す出力項目を確認してください。

   - Hashgrp_area_sufficient_size（ハッシュグループ化領域で領域不足が発生しない十分な領域サイズ）

   そして，出力された値のうち，最大値以上の値を，サーバ定義またはクライアント定義のadb_sql_exe_hashgrp_area_sizeオペランドに指定してください。ローカルハッシュグループ化が適用されるSQL文の実行時間を短縮できることがあります。

   > **重要**
   >
   > adb_sql_exe_hashgrp_area_sizeオペランドの指定値を変更した場合，必ずHADBサーバのメモリ所要量を再見積もりしてください。HADBサーバのメモリ所要量の見積もりについては，「6.3 HADBサーバのメモリ所要量の見積もり」を参照してください。

3. サーバ定義またはクライアント定義を変更する

サーバ定義またはクライアント定義のadb_sql_exe_hashgrp_area_size オペランドの指定値を大きくしてください。

- サーバ定義を変更する場合

  adbstop コマンドで HADB サーバをいったん終了してください。そのあとに，サーバ定義を変更し，adbstart コマンドで HADB サーバを開始してください。

- クライアント定義を変更する場合

  マニュアル『HADB　AP 開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

4. SQL 文を再実行する

   SQL 文を再実行して，SQL 文の実行時間が短縮されるかどうかを確認してください。

メモ

- アクセスパスの統計情報に出力された「ハッシュグループ化領域に関する情報」については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」の「(2)　アクセスパスの統計情報に出力される項目」の「(d)　ハッシュグループ化領域に関する情報」を参照してください。
- サーバ定義のadb_sql_exe_hashgrp_area_size オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashgrp_area_size オペランドを参照してください。
- クライアント定義adb_sql_exe_hashgrp_area_size オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

13.7 ハッシュジョインが適用されるINNER JOINの結合順序の見直しによるSQL文の実行時間の短縮に関するチューニング

ここでは，ハッシュジョインが適用されるINNER JOINの結合順序の見直しによる，SQL文の実行時間を短縮する方法について説明します。

ハッシュジョインが適用されるINNER JOINを含むSQL文を実行する場合，INNER JOINの外表と内表を入れ替えた結合方式指定を行うと，SQL文の実行時間を短縮できることがあります。

手順

1. ハッシュジョインが適用されるINNER JOINの有無を確認する

   SQLトレース情報に出力された「実行したSQL文とアクセスパス情報」を確認してください。アクセスパス情報に，ハッシュジョインが適用されたINNER JOINがあるかどうかを確認してください。

   ハッシュジョインが適用されたINNER JOINがある場合は，手順2.に進んでください。

2. ハッシュジョインでハッシュテーブル領域の領域不足の発生有無を確認する

   ハッシュジョインが適用されたINNER JOINがある場合，そのハッシュジョインについて，SQLトレース情報に出力された「アクセスパスの統計情報」を確認してください。アクセスパスの統計情報に出力された「ハッシュテーブル領域に関する情報」のうち，次に示す出力項目がYとなっているかどうかを確認してください。

   - Hashtbl_area_shortage（ハッシュテーブル領域の領域不足の発生有無）

   Yとなっている場合は，手順3.に進んでください。

3. ハッシュジョインの外表と内表の処理行数を確認する

   ハッシュテーブル領域の領域不足が発生している場合，SQLトレース情報に出力された「アクセスパスの統計情報」を確認してください。アクセスパスの統計情報に出力された「検索処理に関する情報」のうち，次に示す出力項目を確認してください。

   - Scan_row_cnt（検索処理した行数）

   ハッシュジョインの外表の処理行数が，内表の処理行数と比べて，十分に多いかどうかを確認してください。ハッシュジョインの外表の処理行数が，内表の処理行数よりも十分に多い場合は，手順4.に進んでください。

4. ハッシュジョインの結合順序を見直す

   ハッシュジョインが適用されるINNER JOINに，処理行数の少なかった内表を外表としたハッシュジョインとなるように，結合方式指定を指定してください。

   ハッシュジョインの外表の処理行数が，内表の処理行数よりも十分に多い場合は，ハッシュテーブル領域で処理できなかった外表のデータをいったん作業表に格納しています。そのため，SQL文の実行時間が長くなっているおそれがあります。

   結合方式指定を指定することで，ハッシュジョインが適用されるINNER JOINを含むSQL文の実行時間を短縮できることがあります。

メモ

- アクセスパス情報については，「10.11.2　SQL トレース情報に出力される情報」の「(5) 実行した SQL 文とアクセスパス情報」を参照してください。
- アクセスパスの統計情報に出力された「ハッシュテーブル領域に関する情報」については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」の「(2)　アクセスパスの統計情報に出力される項目」の「(e)　ハッシュテーブル領域に関する情報」を参照してください。
- アクセスパスの統計情報に出力された「検索処理に関する情報」については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」の「(2)　アクセスパスの統計情報に出力される項目」の「(b)　検索処理に関する情報」を参照してください。
- 結合方式指定については，マニュアル『HADB　SQL リファレンス』の『構成要素』の『結合方式指定』を参照してください。

13.8 ハッシュフィルタ領域サイズの見直しによるSQL文の実行時間の短縮に関するチューニング

ここでは，次に示すオペランドに指定したハッシュフィルタ領域サイズの見直しによる，SQL文の実行時間を短縮する方法について説明します。

- サーバ定義のadb_sql_exe_hashflt_area_sizeオペランド
- クライアント定義のadb_sql_exe_hashflt_area_sizeオペランド

なお，adbexportコマンドにSQL文を指定して実行した場合でも，同様にSQL文の実行時間を短縮できます。その場合は，adb_sql_exe_hashflt_area_sizeオペランドをエクスポートオプションadb_export_hashflt_area_sizeに読み替えてください。

ハッシュフィルタを適用する次の処理が実行される場合に，ハッシュフィルタ領域サイズを大きくすると，SQL文の実行時間を短縮できることがあります。

- 表の結合方式のハッシュジョイン
- 副問合せの処理方式のハッシュ実行

以降で説明する『■ハッシュフィルタが適用されているかどうかを確認する』と『■ハッシュフィルタが有効に働いているかどうかを確認する』の両方を確認してください。

■ハッシュフィルタが適用されているかどうかを確認する

手順

1. ハッシュフィルタが適用されていない処理があるかどうかを確認する

   SQLトレース情報の出力結果で，次に示す条件を満たす処理があるかどうかを確認してください。

   - 「クライアント定義情報」のadb_sql_exe_hashflt_area_sizeの値が1以上である
   - 「SQL文の統計情報」のHashflt_disabled（ハッシュフィルタが無効化されたかどうか）の値がYである

   上記のすべての条件を満たす処理は，ハッシュフィルタが適用されていません。手順2.に進んでください。上記のすべての条件を満たす処理がない場合は，『■ハッシュフィルタが有効に働いているかどうかを確認する』の内容を確認してください。

2. 無効化されたハッシュフィルタの合計数を確認する

   手順1.の条件に該当する処理がある場合，SQLトレース情報の出力結果で次の値を確認してください。

   - 「アクセスパスの統計情報」に出力された「ハッシュテーブル領域に関する情報」のHashtbl_filter_disabled_num（無効化されたハッシュフィルタの合計数）

3. adb_sql_exe_hashflt_area_sizeオペランドの指定値を見直す

手順 2.で確認したHashtbl_filter_disabled_num の値が0 の場合は，ハッシュフィルタ領域が小さ過ぎるためにハッシュフィルタが適用されていない可能性があります。そのため，SQL 文の実行時間が長くなっているおそれがあります。

この場合，サーバ定義およびクライアント定義のadb_sql_exe_hashflt_area_size オペランドの指定値を大きくしてください。そうすると，ハッシュフィルタが適用されて SQL 文の実行時間を短縮できることがあります。adb_sql_exe_hashflt_area_size オペランドの指定値の目安については，マニュアル『HADB　AP 開発ガイド』の次の個所を参照してください。

- 『ハッシュジョインとは』の『ハッシュフィルタが適用される条件』
- 『外への参照列を含まない副問合せの処理方式とは』の『ハッシュ実行』
- 『外への参照列を含む副問合せの処理方式とは』の『ハッシュ実行』

重要

- クライアント定義のadb_sql_exe_hashflt_area_size オペランドと，サーバ定義の adb_sql_exe_hashflt_area_size オペランドの両方の指定値を確認してください。クライアント定義の指定値だけを大きくした結果，クライアント定義の指定値がサーバ定義の指定値を超えた場合，サーバ定義の指定値が適用されます（変更後のクライアント定義の指定値が適用されません）。該当する場合は，サーバ定義とクライアント定義の両方のadb_sql_exe_hashflt_area_size オペランドの指定値を大きくしてください。
- adb_sql_exe_hashflt_area_size オペランドの指定値を変更した場合，必ず HADB サーバのメモリ所要量を再見積もりしてください。HADB サーバのメモリ所要量の再見積もりについては，「6.3.4　通常運用時のメモリ所要量の求め方」の「(2)　リアルスレッド固有メモリの所要量の求め方（通常運用時）」の「(c)　変数 RTHD_EXESQLSZ の求め方」の変数 *HASHFLT_PROC_SIZE* を参照してください。

また，手順 2.で確認したHashtbl_filter_disabled_num の値が1 以上の場合でも，サーバ定義またはクライアント定義のadb_sql_exe_hashflt_area_size オペランドの指定値を大きくすると，SQL 文の実行時間を短縮できることがあります。

4. サーバ定義およびクライアント定義を変更する

   サーバ定義およびクライアント定義のadb_sql_exe_hashflt_area_size オペランドの指定値を大きくしてください。

   - サーバ定義を変更する場合

     adbstop コマンドで HADB サーバをいったん終了してください。そのあとに，サーバ定義を変更し，adbstart コマンドで HADB サーバを開始してください。

   - クライアント定義を変更する場合

     マニュアル『HADB　AP 開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

5. SQL 文を再実行する

   SQL 文を再実行して，SQL 文の実行時間が短縮されるかどうかを確認してください。ハッシュフィルタの適用によって，SQL 文の実行時間が短縮されることがあります。

■ハッシュフィルタが有効に働いているかどうかを確認する

手順

1. ハッシュフィルタが有効に働いているかどうかを確認する

   SQL トレース情報の「アクセスパスの統計情報」に出力された「ハッシュテーブル領域に関する情報」のうち，次の 2 つの値を確認してください。

   - Hashtbl_sum_filter_check_cnt（ハッシュフィルタでハッシュ値をチェックした回数の合計値）
   - Hashtbl_sum_filtering_cnt（ハッシュ値がハッシュフィルタで除外された回数の合計値）

   Hashtbl_sum_filtering_cnt の値が，Hashtbl_sum_filter_check_cnt に比べて非常に小さい場合，ハッシュフィルタの適用効果がほとんどありません。そのため，ハッシュフィルタの適用が原因で，SQL 文の実行時間が長くなっているおそれがあります。

2. adb_sql_exe_hashflt_area_size オペランドの指定値を見直す

   手順 1.の条件に該当する場合は，クライアント定義のadb_sql_exe_hashflt_area_size オペランドの指定値に0 を指定して，該当する SQL 文の処理でハッシュフィルタを適用しないことを検討してください。

   クライアント定義を変更する場合は，マニュアル『HADB　AP 開発ガイド』の『クライアント定義変更時の注意事項』で説明している手順に従ってクライアント定義を変更してください。

メモ

- クライアント定義情報については，「10.11.2　SQL トレース情報に出力される情報」の「(3)　クライアント定義情報」を参照してください。
- SQL 文の統計情報については，「10.11.2　SQL トレース情報に出力される情報」の「(10)　SQL 文の統計情報」を参照してください。
- アクセスパスの統計情報に出力された「ハッシュテーブル領域に関する情報」については，「10.11.3　アクセスパスの統計情報の出力例と出力項目」の「(2)　アクセスパスの統計情報に出力される項目」の「(e)　ハッシュテーブル領域に関する情報」を参照してください。
- サーバ定義のadb_sql_exe_hashflt_area_size オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_exe_hashflt_area_size オペランドを参照してください。
- クライアント定義のadb_sql_exe_hashflt_area_size オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

13.9 表定義プールサイズの見直しによるSQL文の実行時間の短縮に関するチューニング

ここでは，次に示すオペランドに指定した表定義プールサイズの見直しによる，SQL文の実行時間を短縮する方法について説明します。

- サーバ定義のadb_sql_tbldef_cache_sizeオペランド

表定義プールサイズを大きくすると，SQL文の実行時間を短縮できることがあります。表定義プールサイズの大きさが適切かどうかを確認して変更する手順を次に示します。

手順

1. adbstatコマンドを実行して，HADBサーバの統計情報を取得する

   HADBサーバの統計情報に，表定義プールに関する情報が出力されます。次の情報を確認してください。

   - Tbldef_req_cnt（表定義情報の取得要求回数）

     表定義情報の取得要求回数が出力されます。
   - Tbldef_access_cnt（ディクショナリ表からの表定義情報の取得回数）

     表定義プール内で該当の表定義情報が取得できないため，ディクショナリ表から表定義情報を取得した回数が出力されます。
   - Tbldef_cache_access_cnt（表定義プールからの表定義情報の取得回数）

     表定義プール内で該当の表定義情報を取得した回数が出力されます。
   - Tbldef_cache_register_cnt（表定義プールへの表定義情報の登録回数）

     表定義プールに表定義情報を登録した回数が出力されます。
   - Tbldef_cache_sweep_cnt（表定義プールからの表定義情報の掃き出し回数）

     表定義プールから表定義情報を掃き出した回数が出力されます。
2. 表定義情報のヒット率を確認する

   次に示す計算式で，表定義情報のヒット率を求めてください。計算の結果，ヒット率が80%以下の場合は，サーバ定義のadb_sql_tbldef_cache_sizeオペランドの指定値を大きくしてください。

   表定義情報のヒット率（%） = （*A* ÷ *B*） × 100

   A：Tbldef_cache_access_cnt（表定義プールからの表定義情報の取得回数）

   B：Tbldef_req_cnt（表定義情報の取得要求回数）
3. Tbldef_cache_sweep_cnt（表定義プールからの表定義情報の掃き出し回数）を確認する

   表定義プールからの表定義情報の掃き出し回数が多い場合は，サーバ定義のadb_sql_tbldef_cache_sizeオペランドの指定値を大きくしてください。
4. HADBサーバを終了する

   adbstopコマンドを実行して，HADBサーバを終了してください。

5. サーバ定義を変更する

   サーバ定義のadb_sql_tbldef_cache_size オペランドの指定値を大きくしてください。

6. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> 表定義プール，およびサーバ定義のadb_sql_tbldef_cache_size オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sql_tbldef_cache_size オペランドを参照してください。

14 障害運用

この章では，HADB で障害が発生した場合に，HADB 管理者が実施することについて説明します。

14.1　障害発生時の対処の流れ

HADB サーバで障害が発生した場合，原因を確認して障害を取り除く必要があります。発生した障害によって，HADB サーバが異常終了したかどうかで対処が異なります。

まず，HADB サーバが異常終了したかどうかを確認します。次に示すコマンドを実行してください。

実行するコマンド

```
adbls -d srv
```

HADB サーバの状態が表示されます。実行結果のSTATUS の項目を確認してください。

- STATUS がABORT の場合

  HADB サーバが異常終了しています。「14.1.1　HADB サーバが異常終了した場合の対処」を参照してください。

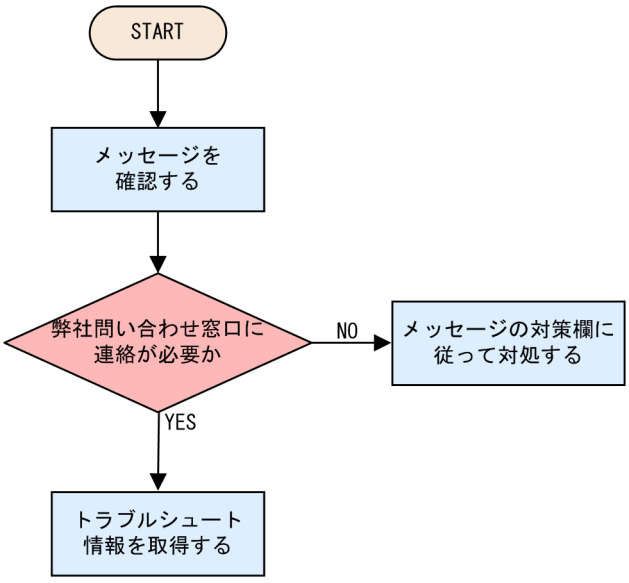
- STATUS が上記以外の場合

  HADB サーバは異常終了していません。「14.1.2　HADB サーバが異常終了していない場合の対処」を参照してください。

14.1.1　HADB サーバが異常終了した場合の対処

HADB サーバが異常終了した場合の対処を次に示します。

図 14-1　HADB サーバが異常終了した場合の対処の流れ



手順

1. メッセージを確認する

HADB サーバが異常終了した場合，サーバメッセージログファイルをテキストエディタで開き，出力されているエラーメッセージを確認してください。

サーバメッセージログファイルは，サーバディレクトリ下のspool/adbmessage*XX*.log になります。ファイル名の*XX* は通番であり，01～04 のどれかになります。最終更新日時が最も新しいファイルを開いてください。

出力されているエラーメッセージの ID を基に，マニュアル『HADB　メッセージ』を参照して，障害の原因および対策欄を確認してください。

出力されたメッセージテキストに「Save troubleshooting information by the "adbinfoget" command」が含まれる場合，または対策欄に「弊社問い合わせ窓口に連絡してください。」と記載されている場合は，手順 2.に進んでください。

それ以外の場合は，手順 3.に進んでください。

2. 弊社問い合わせ窓口に連絡が必要な場合

「14.2　トラブルシュート情報の取得（adbinfoget コマンド）」を参照して，トラブルシュート情報を取得してください。取得したトラブルシュート情報は，弊社問い合わせ窓口に連絡する際，合わせて送付してください。

3. 弊社問い合わせ窓口に連絡が不要な場合

出力されたメッセージの対策に従って障害を取り除いてください。

障害を取り除いてから，HADB サーバを再開始してください。障害発生時点で処理中だったトランザクションをロールバックしたあとに，HADB サーバが起動します。

14.1.2　HADB サーバが異常終了していない場合の対処

HADB サーバが異常終了していない場合の対処を次に示します。

手順

1. メッセージを確認する

HADB サーバが異常終了していない場合，サーバメッセージログファイルをテキストエディタで開き，出力されているエラーメッセージを確認してください。

サーバメッセージログファイルは，サーバディレクトリ下のspool/adbmessage*XX*.log になります。ファイル名の*XX* は通番であり，01～04 のどれかになります。最終更新日時が最も新しいファイルを開いてください。

出力されているエラーメッセージの ID を基に，マニュアル『HADB　メッセージ』を参照して，障害の原因および対策欄を確認してください。

2. 対策欄に従って対処する

対策欄の対処方法に従って障害を取り除いてください。障害を取り除いてから，HADB サーバを再開始してください。障害発生時点で処理中だったトランザクションをロールバックしたあとに，HADB サーバが起動します。

重要

APまたはコマンドが終了しない，HADBサーバが無応答状態になったなど，障害が発生してもメッセージが出力されない場合があります。その場合は，「14.2　トラブルシュート情報の取得（adbinfoget コマンド）」を参照して，トラブルシュート情報を取得後，弊社問い合わせ窓口に連絡してください。

14.2　トラブルシュート情報の取得（adbinfoget コマンド）

弊社問い合わせ窓口に対して送付するトラブルシュート情報を取得する場合，adbinfoget コマンドを実行してください。

adbinfoget コマンドを実行すると，次に示すトラブルシュート情報ファイルを取得できます。

取得できるトラブルシュート情報ファイル

- **detail 版トラブルシュート情報ファイル**
  トラブルシュートに必要なすべての情報が格納されたファイルです。
  ファイル名は，adbinfo*YYYYMMDDhhmmss*_detail.tar.gz になります。
- **light 版トラブルシュート情報ファイル**
  detail 版トラブルシュート情報ファイルからコアファイル，共有メモリ，ダンプファイルなどを除いた情報が格納されたファイルです。
  ファイル名は，adbinfo*YYYYMMDDhhmmss*_light.tar.gz になります。
- **mail 版トラブルシュート情報ファイル**
  light 版トラブルシュート情報ファイルから，さらに幾つかのファイルを除いて，トラブルシュートに必要な最低限の情報だけが格納されたファイルです。
  ファイル名は，adbinfo*YYYYMMDDhhmmss*_mail.tar.gz になります。
- **root 版トラブルシュート情報ファイル**
  root 権限があると取得できる情報が格納されたファイルです。
  ファイル名は，adbinfo*YYYYMMDDhhmmss*_root.tar.gz になります。

注

ファイル名の"*YYYYMMDDhhmmss*"は，adbinfoget コマンドを実行した日時になります。

adbinfoget コマンドの実行手順を次に示します。adbinfoget コマンドのオプションについては，マニュアル『HADB　コマンドリファレンス』の『adbinfoget（トラブルシュート情報の取得)』を参照してください。

手順

1. トラブルシュート情報を出力する
   AP またはコマンドが終了しない，HADB サーバが無応答状態になったなど，障害が発生してもメッセージが出力されない場合は，トラブルシュート情報が出力されません。そのため，まずは，adbinfoget コマンドを実行して，トラブルシュート情報を出力します。次に示すコマンドを実行してください。
   なお，メッセージが出力されている場合（HADB サーバが異常終了した場合）は，トラブルシュート情報の出力は不要です。手順 2.に進んでください。
   実行するコマンド

```
adbinfoget -g
```

2. 取得するトラブルシュート情報ファイルの容量を確認する

adbinfoget コマンドを実行して，取得するトラブルシュート情報ファイルの容量を確認します。次に示すコマンドを実行してください。-r オプションを指定しているため，要求に応じてスーパユーザのパスワードを入力してください。

実行するコマンド

```
adbinfoget -m -r
```

コマンドを実行した結果とディスクの空き容量を比較して，すべてのトラブルシュート情報ファイルを取得できるかどうかを確認してください。もし，detail 版トラブルシュート情報ファイルの容量が大き過ぎて取得できない場合は，detail 版以外のトラブルシュート情報ファイルを取得します。

3. トラブルシュート情報ファイルを格納するディレクトリを作成する

取得するトラブルシュート情報ファイルを格納するディレクトリを作成してください。

4. トラブルシュート情報ファイルを取得する

adbinfoget コマンドを実行して，トラブルシュート情報ファイルを取得します。ディスクの空き容量によって，次のどちらかのコマンドを実行してください。-r オプションを指定しているため，要求に応じてスーパユーザのパスワードを入力してください。

■すべてのトラブルシュート情報ファイルを取得する場合

```
adbinfoget -o トラブルシュート情報の出力先ディレクトリ -r
```

■detail 版以外のトラブルシュート情報ファイルを取得する場合

```
adbinfoget -l -o トラブルシュート情報の出力先ディレクトリ -r
```

注

-o オプションには，手順 3.で作成したディレクトリを指定してください。

5. 取得したトラブルシュート情報ファイルを送付する

トラブルシュート情報ファイルを取得したら，detail 版トラブルシュート情報ファイルおよび root 版トラブルシュート情報ファイルを，弊社問い合わせ窓口に送付してください。

ディスクの空き容量が少なくて detail 版トラブルシュート情報ファイルを取得できない場合，または取得した detail 版トラブルシュート情報ファイルの容量が大き過ぎて送付できない場合は，light 版トラブルシュート情報ファイルおよび root 版トラブルシュート情報ファイルを，弊社問い合わせ窓口に送付してください。

light 版トラブルシュート情報ファイルの容量も大き過ぎて送付できない場合は，mail 版トラブルシュート情報ファイルおよび root 版トラブルシュート情報ファイルを，弊社問い合わせ窓口に送付してください。

重要

- 取得したトラブルシュート情報ファイルが不要になった場合は，ディスクの空き容量を確保するために，トラブルシュート情報ファイルを削除することを検討してください。

- adbinfoget コマンドを実行したら，「14.3　トラブルシュート情報の削除（adbinfosweep コマンド）」を参照して，adbinfosweep コマンドを実行することを検討してください。

14.3　トラブルシュート情報の削除（adbinfosweep コマンド）

HADB サーバが出力したトラブルシュート情報は，HADB サーバが停止中のときにadbinfosweep コマンドを実行すると削除できます。

次に示すときに，HADB サーバが出力したトラブルシュート情報は，自動的に削除されません。

- サーバプロセスの強制終了時に出力されたトラブルシュート情報
- サーバプロセスでの内部矛盾の検知時に出力されたトラブルシュート情報

そのため，トラブルシュート情報の容量が増加し続けると，ディスクの空き容量が不足する原因となります。定期的にadbinfosweep コマンドを実行して，ディスクの空き容量を確保してください。

adbinfosweep コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbinfosweep (トラブルシュート情報の削除)』を参照してください。

adbinfosweep コマンドを実行するタイミングを次に示します。

adbinfosweep コマンドを実行するタイミング

HADB サーバが異常終了した場合に，adbinfoget コマンドでトラブルシュート情報を取得したあとで，adbinfosweep コマンドを実行してください。

adbinfosweep コマンドで，削除できるトラブルシュート情報を次に示します。

削除できるトラブルシュート情報の一覧

- 障害情報（コアファイル）
  - $DBDIR/SPOOL/core.*サーバプロセスのプロセス ID*
  - *サーバ定義 adb_core_path オペランドで指定したパス*/core.*サーバプロセスのプロセス ID*
- サーバ定義格納ファイル
  - $ADBDIR/spool/.defrslt
- HADB ダンプファイル
  - $ADBDIR/spool/adbdump*YYYYMMDDhhmmss*.*サーバプロセスのプロセス ID*

    YYYYMMDDhhmmss は，ファイルを生成した時刻になります。
  - $ADBDIR/spool/adbdumperror*YYYYMMDDhhmmssSSSSSS_TTTTTTTTTTTTTTTTTTTT*.*サーバプロセスのプロセス ID*

    YYYYMMDDhhmmssSSSSSS は，ファイルを生成した時刻になります。

    また，*TTTTTTTTTTTTTTTTTTTT* は，HADB サーバの内部矛盾エラーを検知したリアルスレッドのスレッド ID になります。

重要

HADB 管理者以外の OS ユーザによって出力されたトラブルシュート情報は，adbinfosweep コマンドを実行しても削除されないことがあります。この場合，スーパユーザが OS の rm コマンドなどを実行して，不要なトラブルシュート情報を削除してください。

15 トラブルシュート

この章では，HADB サーバでトラブルが発生した場合の対処方法について説明します。

15.1　AP に関するトラブル

ここでは，AP に関するトラブルがあった場合の対処方法について説明します。

15.1.1　AP が終了しない場合や AP が異常終了した場合の対処方法

AP が終了しない場合，または AP が異常終了した場合は，「10.8　トランザクションの処理状況の確認」に従って，AP の処理状況を確認してください。

必要に応じてadbcancel コマンドを実行して AP が実行しているトランザクションを終了させてください。

また，adbcancel コマンドを実行してもトランザクションが終了しない場合は，次の手順で HADB サーバを終了させてください。

手順

1. adbstop --force コマンドで HADB サーバを強制終了させてください。
2. 1.を実行しても HADB サーバが終了しない場合は，HADB サーバプロセス（adbsrvd）を OS のkill コマンドで終了させてください。

上記の手順でも HADB サーバが終了しない場合は，OS を再起動してください。

15.1.2　AP が実行できない場合の対処方法（KFAA40005-E メッセージが出力された場合）

AP の実行時に，ほかの AP およびコマンドを同時に実行していなくても，KFAA40005-E メッセージが出力される場合は，排他資源を確保したまま，終了していないトランザクションがある可能性があります。

adbls -d cnct コマンドを実行して，トランザクションの処理状況を確認してください。出力項目「STATUS」を確認して，"STARTED"となっている AP およびコマンドがないかどうかを確認してください。

"STARTED"となっている AP およびコマンドがある場合は，対象のトランザクションの終了を待つか，またはadbcancel コマンドで対象のトランザクションを終了させてください。

15.1.3　AP の HADB サーバへの接続に時間が掛かる場合の対処方法

AP の動作環境が Linux の場合，DNS サーバへのホスト名などの問い合わせに時間が掛かり，それが原因で AP の HADB サーバへの接続に時間が掛かることがあります。

/etc/nsswitch.conf ファイルの設定が次のようになっていることを確認してください。

```
hosts: files myhostname dns
```

15.1.4　クライアントマシンで使用中の通信ポート数を確認する方法

APからキャンセル要求を頻繁に実行した場合，APを実行しているクライアントマシンの通信ポートが不足することがあります。SQL文のキャンセル処理の遅延や，APがHADBサーバに接続できないなどの現象が発生した場合は，通信ポートが不足しているおそれがあります。使用中の通信ポートのおおよその数は，ssコマンドやnetstatコマンドなどのOSのコマンドで確認できます。

メモ

ここでは，RHEL 7でのssコマンド，Windows 10でのnetstatコマンドを例に説明しています。RHEL 7またはWindows 10以外のOSを使用している場合は，コマンドの仕様が下記の例とは異なることがあります。そのため，下記の例のとおりにコマンドを実行しても，誤動作してしまいます。例えば，下記で説明しているssコマンドをRHEL 6で実行した場合，出力フォーマットが異なるため，下記のコマンド例の実行結果は期待した結果になりません。したがって，RHEL 7またはWindows 10以外のOSを使用している場合は，そのOSで実行できるコマンドや引数をOSのマニュアルで確認したあとにコマンドを実行してください。

- RHEL 7の場合のコマンド例

```
ss -na | grep ”tcp” | awk ’{print $5}’ | sed -e ”s/[0-9a-f*:.]*://g” | sort -u | wc -l
```

[説明]

ss

ホスト内のソケット情報の一覧を，次のフォーマットで出力します。

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port

grep

ssコマンドの出力結果から，Netidがtcpの行を絞り込みます。

awk

grepコマンドの出力結果から，Local Address:Portの列を抽出します。

sed

awkコマンドの出力結果から，ローカルアドレスを削除し，ポート番号だけを残します。

sort

sedコマンドの出力結果から，ポート番号の重複を排除します。

wc

sortコマンドの出力行数をカウントします。

上記のコマンドを実行すると，使用中の通信ポートのおおよその数が表示されます。

- Windows 10 の場合のコマンド例

```
netstat -nq | find ”TCP” /c
```

[説明]

netstat

ホスト内のアクティブな接続状態の一覧を，次のフォーマットで出力します。

プロトコル	ローカル アドレス	外部アドレス	状態

find

netstat コマンドの出力結果のうち，プロトコルがTCP の行数をカウントします。

上記のコマンドを実行すると，使用中の通信ポートのおおよその数が表示されます。

メモ

HADB クライアントでは，HADB サーバのadbinfoget コマンドのようにトラブルシュート情報を自動的に取得する機能をサポートしていません。そのため，上記の方法で使用中の通信ポート数を確認してください。

15.2　コマンドに関するトラブル

ここでは，コマンドに関するトラブルがあった場合の対処方法について説明します。

15.2.1　コマンドが応答待ちタイムアウトになる場合の対処方法

コマンドを実行しても応答待ちタイムアウトになる場合，OS の負荷が高くコマンドの応答時間内に処理が完了しないことが考えられます。その場合は，再度コマンドを実行してください。

コマンドを再度実行しても，応答待ちタイムアウトになる場合は，必要のないプロセスを停止して OS の負荷を下げてください。そのあとで，コマンドを実行してください。

15.2.2　コマンドが実行できない場合の対処方法（KFAA40005-E メッセージが出力された場合）

コマンドの実行時に，ほかのコマンドおよび AP を同時に実行していなくても，KFAA40005-E メッセージが出力される場合は，排他資源を確保したまま，終了していないトランザクションがある可能性があります。

adbls -d cnct コマンドを実行して，トランザクションの処理状況を確認してください。出力項目「STATUS」を確認して，"STARTED"となっているコマンドおよび AP がないかどうかを確認してください。

"STARTED"となっているコマンドおよび AP がある場合は，対象のトランザクションの終了を待つか，またはadbcancel コマンドで対象のトランザクションを終了させてください。

15.2.3　コマンドが終了しない場合の対処方法

コマンドが終了しない場合，コマンドの実行状況を調べてください。コマンドの実行状況を調べる方法については，「10.8　トランザクションの処理状況の確認」を参照してください。コマンドを終了させたい場合は，adbcancel コマンドでコマンドを終了させてください。adbcancel コマンドでコマンドを終了させる方法については，マニュアル『HADB　コマンドリファレンス』の『実行中のコマンドを中断（強制終了）する場合』を参照してください。

また，adbcancel コマンドを実行してもコマンドが終了しない場合は，次の手順で HADB サーバを終了させてください。

手順

1. adbstop --force コマンドで HADB サーバを強制終了させてください。
2. 1.を実行しても HADB サーバが終了しない場合は，HADB サーバプロセス（adbsrvd）を OS のkill コマンドで終了させてください。

上記の手順でも HADB サーバが終了しない場合は，OS を再起動してください。

15.2.4 adbimport コマンドまたは adbidxrebuild コマンドが再実行できない場合の対処方法

中断したadbimport コマンドまたはadbidxrebuild コマンドが再実行できない原因として，次のどちらかのファイルにアクセスできないことが考えられます。その場合，対応するメッセージがメッセージログファイルに出力されます。

- コマンドステータスファイル
  KFAA50244-E メッセージが出力されます。
- 作業用一時ファイル
  KFAA50247-E メッセージが出力されます。

また，KFAA50244-E メッセージおよびKFAA50247-E メッセージ以外のメッセージが出力されている場合は，ほかの原因が考えられます。原因は，出力されたメッセージから確認できます。

対処方法は，出力されたメッセージによって異なります。それぞれの対処方法を次に示します。

(1) KFAA50244-E メッセージが出力されたとき（コマンドステータスファイルにアクセスできないとき）

--force オプションを指定して，adbidxrebuild コマンドを実行してください。

なお，中断したコマンドがバックグラウンドインポートを適用した（-b オプションを指定した）adbimport コマンドの場合は，adbidxrebuild コマンドに--force オプションを指定して実行すると，adbimport コマンドで処理中だった表データの格納情報が削除されます。この場合，中断したadbimport コマンドは実行されていない状態に戻ります。そのため，adbidxrebuild コマンドを実行したあとに，-b オプションを指定して，中断したadbimport コマンドを再実行してください。

対処手順の詳細については，次に示す項目を参照してください。

- マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』の『adbimport コマンドが異常終了したときの対処方法』の『adbimport コマンドの再実行時にエラーが発生した場合』
- マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドが異常終了したときの対処方法』の『adbidxrebuild コマンドの再実行時にエラーが発生した場合』

(2) KFAA50247-E メッセージが出力されたとき（作業用一時ファイルにアクセスできないとき）

--create-temp-file オプションを指定して，adbidxrebuild コマンドを実行してください。

なお，中断したコマンドがバックグラウンドインポートを適用した（-b オプションを指定した）adbimport コマンドの場合は，adbidxrebuild コマンドに--create-temp-file オプションを指定して実行することで，中断したadbimport コマンドの実行が完了します。

対処手順の詳細については，次に示す項目を参照してください。

- マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』の『adbimport コマンドが異常終了したときの対処方法』の『adbimport コマンドの再実行時にエラーが発生した場合』
- マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドが異常終了したときの対処方法』の『adbidxrebuild コマンドの再実行時にエラーが発生した場合』

(3)　KFAA50244-E メッセージおよび KFAA50247-E メッセージ以外のメッセージが出力されたとき

出力されたメッセージの対策に従って，中断したコマンドを再実行してください。

15.2.5　adbunarchivechunk コマンドが再実行できない場合の対処方法

中断したadbunarchivechunk コマンドが再実行できない原因として，次のどちらかのファイルにアクセスできないことが考えられます。

- コマンドステータスファイル
- 作業用一時ファイル

どちらのファイルが原因であるかは，出力されたメッセージで確認できます。

(1)　コマンドステータスファイルにアクセスできないときの対処方法

--force オプションを指定して，adbunarchivechunk コマンドを再実行してください。

(2)　作業用一時ファイルにアクセスできないときの対処方法

出力されたメッセージの対策に従って対処してください。そのあと，--force オプションを指定して，adbunarchivechunk コマンドを再実行してください。

15.2.6　コマンド実行時の作業用一時ファイルを格納するディスクの空き容量が不足した場合の対処方法

次に示すコマンドの実行中，-w オプションに指定したディレクトリに作業用一時ファイルが作成されます。

- adbimport コマンド
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド
- adbreorgsystemdata コマンド

コマンドの実行中に，次に示すメッセージのどれかが出力された場合，作業用一時ファイルを格納するディスクの空き容量が不足したおそれがあります。

- KFAA30959-E
- KFAA40204-E
- KFAA40205-E
- KFAA40207-E
- KFAA40208-E
- KFAA40214-E
- KFAA41205-E
- KFAA41206-I

作業用一時ファイルを格納するディスクの空き容量が不足したかどうかは，メッセージに出力された内容から判断できます。詳細については，「(1)　作業用一時ファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法」を参照してください。

作業用一時ファイルを格納するディスクの空き容量が不足した場合，次に示す原因が考えられます。該当する原因に従って，対処してください。

- ディスクに不要な作業用一時ファイルが残っている。
  「(2)　ディスクに不要な作業用一時ファイルが残っている場合」を参照してください。
- 作業用一時ファイルを格納するディスク容量が，作成される作業用一時ファイルよりも小さい。
  - 中断したコマンドがadbimport コマンドまたはadbidxrebuild コマンドの場合
    「(3)　作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき）」を参照してください。
  - 中断したコマンドがadbmergechunk コマンド，adbunarchivechunk コマンドまたは adbreorgsystemdata コマンドの場合
    「(4)　作業用一時ファイルを格納するディスク容量が小さい場合（adbmergechunk コマンド，adbunarchivechunk コマンドまたは adbreorgsystemdata コマンドが中断したとき）」を参照してください。

(1) 作業用一時ファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法

作業用一時ファイルを格納するディスクの空き容量が不足しているかどうかを確認するには，出力されたメッセージの内容を確認してください。出力されたメッセージの内容が，次に示す表の内容に一致している場合，作業用一時ファイルを格納するディスクの空き容量が不足しています。

表 15-1　作業用一時ファイルを格納するディスクの空き容量が不足した場合に出力されるメッセージと，メッセージに出力される内容

項番	メッセージ ID	メッセージに出力される内容		
		ファイル名	エラーコード	エラー番号
1	KFAA30959-E	ファイル名※1	－	28(ENOSPC)
2	KFAA40204-E	ファイルのパス名※2	－	－
3	KFAA40205-E	－	－	28(ENOSPC)
4	KFAA40207-E	－	-210 または-230	－
5	KFAA40208-E	－	-210 または-230	－
6	KFAA40214-E	－	－	28(ENOSPC)
7	KFAA41205-E	ファイルのパス名※2	－	28(ENOSPC)
8	KFAA41206-I	ファイルのパス名※2	－	28(ENOSPC)

（凡例）

－：該当するメッセージに，この内容は出力されません。

注※1

メッセージに出力されたファイル名のファイルが格納されているディレクトリを確認する必要があります。ファイルが格納されているディレクトリを確認する手順を次に示します。

1. メッセージに出力されたファイル名のファイルを，OS のfind コマンドで検索してください。

   OS のfind コマンドの実行例を次に示します。

   ■OS のfind コマンドの実行例

   ```
   find ~/ -name "xxxxx"
   ```

   [説明]

   メッセージに出力されたファイル名（*xxxxx*）を指定して，find コマンドを実行します。

2. コマンドの実行結果で，ファイルが格納されているディレクトリが，各コマンドの-w オプションに指定したディレクトリに一致しているかどうかを確認してください。

   一致している場合は，作業用一時ファイルを格納するディスクの空き容量が不足しています。

注※2

メッセージに出力されたパス名から，ファイルが格納されているディレクトリを確認する必要があります。

パス名に含まれるディレクトリが，各コマンドの-w オプションに指定したディレクトリに一致しているかどうかを確認してください。

一致している場合は，作業用一時ファイルを格納するディスクの空き容量が不足しています。

メモ

コマンドの-w オプションに指定したディレクトリとは，次のどちらかを指します。

- コマンドの-w オプションに指定したディレクトリ
- コマンドの-w オプションに指定したディレクトリパスファイルに指定したディレクトリ

(2) ディスクに不要な作業用一時ファイルが残っている場合

不要な作業用一時ファイルを削除する必要があります。削除の手順を次に示します。

手順

1. 再実行が必要なコマンドの有無を確認する

   -c オプションにtable を指定して，adbdbstatus コマンドを実行してください。コマンドの実行結果で，表のサマリ情報のRerun_command 列に，再実行が必要なコマンド名が出力されます。なお，再実行が必要なコマンドがない場合は，Rerun_command 列に何も出力されません。

   再実行が必要なコマンドがある場合は，手順 2.に進んでください。再実行が必要なコマンドがない場合は，手順 3.に進んでください。

   adbdbstatus コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbdbstatus (データベースの状態解析)』を参照してください。

2. コマンドを再実行する

   手順 1.で確認した，再実行が必要なコマンドを再実行してください。コマンドの再実行が正常終了すると，作業用一時ファイルが削除されます。このとき，手順 3.の操作は不要です。

3. 残っている作業用一時ファイルを削除する

   残っている作業用一時ファイルは，すべて不要なファイルです。OS のrm コマンドなどを実行して，不要なファイルを削除してください。

(3) 作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき）

作業用一時ファイルを格納するディスク容量が小さい場合，作業用一時ファイルの格納先を，容量が大きい別のディスクに変更する必要があります。ここでは，中断したコマンドが，adbimport コマンドまたはadbidxrebuild コマンドのときの，作業用一時ファイルの格納先を変更する方法を説明します。

中断したコマンドがadbmergechunk コマンド，adbunarchivechunk コマンドまたはadbreorgsystemdata コマンドのときの，作業用一時ファイルの格納先を変更する方法については，「(4)　作業用一時ファイルを格納するディスク容量が小さい場合（adbmergechunk コマンド，adbunarchivechunk コマンドまたはadbreorgsystemdata コマンドが中断したとき)」を参照してください。

作業用一時ファイルの格納先を変更する方法の手順は，コマンドの-w オプションに指定したディレクトリを変更するかどうかによって異なります。コマンドの-w オプションに指定したディレクトリとは，次のうちどちらかを指します。

- コマンドの-w オプションに指定したディレクトリ
- コマンドの-w オプションに指定したディレクトリパスファイルに指定したディレクトリ

コマンドの-w オプションに指定したディレクトリを変更するとき，および変更しないときの手順をそれぞれ次に示します。

(a)　コマンドの-w オプションに指定したディレクトリを変更するとき

手順

1. コマンドの-w オプションに指定したディレクトリを変更する

   コマンドの-w オプションに指定したディレクトリを，容量が大きい別のディスクのディレクトリに変更してください。

   ディレクトリパスファイルに指定したディレクトリを変更する場合について，変更前および変更後のディレクトリパスファイルの指定例を次に示します。

   ■ディレクトリパスファイルの指定例（変更前）

   ```
   /mnt/disk1/xxxxx
   /mnt/disk2/yyyyy
   /mnt/disk3/zzzzz
   ```

   [説明]

   変更前のディレクトリパスファイルでは，容量不足のディスク（'/mnt/disk1'～'/mnt/disk3'）のディレクトリを，作業用一時ファイルの格納先として指定しています。

   ■ディレクトリパスファイルの指定例（変更後）

   ```
   /mnt/largedisk1/xxxxx
   /mnt/largedisk2/yyyyy
   /mnt/largedisk3/zzzzz
   ```

   [説明]

   変更前に'/mnt/disk1'～'/mnt/disk3'を指定していた個所に，容量が大きい別のディスク（'/mnt/largedisk1'～'/mnt/largedisk3'）をそれぞれ指定して，作業用一時ファイルの格納先を変更します。

2. adbidxrebuild コマンドを実行する

-w オプションおよび--create-temp-file オプションの両方を指定して，adbidxrebuild コマンドを実行してください。adbidxrebuild コマンドの指定形式については，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』を参照してください。

(b)　コマンドの-w オプションに指定したディレクトリを変更しないとき

手順

1. 作業用一時ファイルを別のディレクトリにコピーする

   OS のcp コマンドなどで，コマンドの-w オプションに指定したディレクトリの直下に格納されているすべての作業用一時ファイルを，容量が大きい別のディスクのディレクトリにコピーしてください。

   ディレクトリパスファイルに指定したディレクトリの直下に格納されている作業用一時ファイルをコピーする場合について，ディレクトリパスファイルの指定例および OS のcp コマンドの実行例を次に示します。

   ■ディレクトリパスファイルの指定例

   ```
   /mnt/disk1/xxxxx
   /mnt/disk2/yyyyy
   /mnt/disk3/zzzzz
   ```

   [説明]

   容量不足のディスク（'/mnt/disk1'～'/mnt/disk3'）のディレクトリを，作業用一時ファイルの格納先として指定しています。

   ■OS のcp コマンドの実行例

   ```
   cp -R /mnt/disk1/xxxxx  /mnt/largedisk1/xxxxx
   cp -R /mnt/disk2/yyyyy  /mnt/largedisk2/yyyyy
   cp -R /mnt/disk3/zzzzz  /mnt/largedisk3/zzzzz
   ```

   [説明]

   容量不足のディスク（'/mnt/disk1'～'/mnt/disk3'）のディレクトリに格納されている作業用一時ファイルを，容量が大きい別のディスク（'/mnt/largedisk1'～'/mnt/largedisk3'）のディレクトリにコピーします。

2. コマンドの-w オプションに指定したディレクトリを削除する

   コマンドの-w オプションに指定したディレクトリを OS のrm コマンドなどで削除してください。ディレクトリを削除する際の，OS のrm コマンドの実行例を次に示します。

   ■OS のrm コマンドの実行例

   ```
   rm  -r  /mnt/disk1/xxxxx
   rm  -r  /mnt/disk2/yyyyy
   rm  -r  /mnt/disk3/zzzzz
   ```

   [説明]

   容量不足のディスク（'/mnt/disk1'～'/mnt/disk3'）のディレクトリを削除します。

3. 容量が大きい別のディスクのディレクトリをリンク先とするシンボリックリンクを作成する

OS のln コマンドを実行して，容量が大きい別のディスクのディレクトリをリンク先に設定したシンボリックリンクを作成してください。このとき，シンボリックリンクのパス名として，コマンドの-w オプションに指定したディレクトリを指定します。シンボリックリンクを作成する際の，OS のln コマンドの実行例を次に示します。

■OS のln コマンドの実行例

```
ln  -s /mnt/largedisk1/xxxxx  /mnt/disk1/xxxxx
ln  -s /mnt/largedisk2/yyyyy  /mnt/disk2/yyyyy
ln  -s /mnt/largedisk3/zzzzz  /mnt/disk3/zzzzz
```

[説明]

容量が大きい別のディスクのディレクトリ（'/mnt/largedisk1/xxxxx'～'/mnt/largedisk3/zzzzz'）をリンク先に設定して，コマンドの-w オプションに指定したディレクトリ（'/mnt/disk1/xxxxx'～'/mnt/disk3/zzzzz'）と同名のシンボリックリンク（'/mnt/disk1/xxxxx'～'/mnt/disk3/zzzzz'）を作成します。

4. 中断したコマンドを再実行する

中断したadbimport コマンドまたはadbidxrebuild コマンドを再実行してください。ソート処理またはB-tree インデクスとテキストインデクスの作成処理から，コマンドの処理が再開されます。

なお，コマンドの再実行後，KFAA50247-E メッセージが出力された場合は，手順 1.でコピーした作業用一時ファイルをすべて削除してください。そのあと，--create-temp-file オプションを指定してadbidxrebuild コマンドを実行してください。

(4) 作業用一時ファイルを格納するディスク容量が小さい場合（adbmergechunk コマンド，adbunarchivechunk コマンドまたはadbreorgsystemdata コマンドが中断したとき）

作業用一時ファイルを格納するディスク容量が小さい場合，作業用一時ファイルの格納先を，容量が大きい別のディスクに変更する必要があります。ここでは，adbmergechunk コマンド，adbunarchivechunk コマンドまたはadbreorgsystemdata コマンドが中断したときの，作業用一時ファイルの格納先を変更する方法を説明します。

中断したコマンドがadbimport コマンドまたはadbidxrebuild コマンドのときの，作業用一時ファイルの格納先を変更する方法については，「(3) 作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき）」を参照してください。

作業用一時ファイルの格納先を変更する方法の手順は，コマンドの-w オプションに指定したディレクトリを変更するかどうかによって異なります。コマンドの-w オプションに指定したディレクトリとは，次のうちどちらかを指します。

- コマンドの-w オプションに指定したディレクトリ
- コマンドの-w オプションに指定したディレクトリパスファイルに，指定したディレクトリ

コマンドの-w オプションに指定したディレクトリを変更するとき，および変更しないときの手順をそれぞれ次に示します。

(a)　コマンドの-w オプションに指定したディレクトリを変更するとき

手順

1. コマンドの-w オプションに指定したディレクトリを変更する

   コマンドの-w オプションに指定したディレクトリを，容量が大きい別のディスクのディレクトリに変更してください。

   ディレクトリパスファイルに指定したディレクトリを変更する場合，ディレクトリパスファイルの指定例については，「(3)　作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき)」を参照してください。

2. 中断したコマンドを再実行する

   中断したコマンド（adbmergechunk コマンド，adbunarchivechunk コマンドまたはadbreorgsystemdata コマンド）を再実行してください。

(b)　コマンドの-w オプションに指定したディレクトリを変更しないとき

手順

1. 容量が大きい別のディスクのディレクトリをリンク先とするシンボリックリンクを作成する

   OS の ln コマンドを実行して，容量が大きい別のディスクのディレクトリをリンク先に設定したシンボリックリンクを作成してください。このとき，シンボリックリンクのパス名として，コマンドの-w オプションに指定したディレクトリを指定します。

   OS の ln コマンドの実行例については，「(3)　作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき)」を参照してください。

2. 中断したコマンドを再実行する

   中断したコマンド（adbmergechunk コマンド，adbunarchivechunk コマンドまたはadbreorgsystemdata コマンド）を再実行してください。

15.3　ディスクの空き容量に関するトラブル

ここでは，DB ディレクトリ下にある DB エリアファイル，または DB エリアファイル以外のファイルの容量が増加し，データを格納するディスクの空き容量が少なくなった場合の対処方法について説明します。

15.3.1　DB エリアファイルの容量増加が原因の場合

DB エリアファイルの容量が増加し，DB エリアファイルを格納しているディスクの空き容量が少なくなった場合，次に示すどれかの方法で対処してください。

- DB エリアファイルの容量を拡張する
- DB エリアファイルの格納場所を変更する
- 作業表用 DB エリアファイルのサイズを小さくする

上記の方法で対処できない場合は，「11.10.6　データ用 DB エリアの空き容量を確保する方法」を参照して，DB エリアの空き容量を確保してください。

(1)　DB エリアファイルの容量を拡張する

DB エリアファイルの容量を拡張する場合，レギュラーファイルとブロックスペシャルファイルで，対処方法が異なります。それぞれの対処方法を，次に示します。

- **レギュラーファイルの場合**

  不要なファイルを削除して，ディスクの空き容量を増やしてください。

  ディスクの空き容量を増やすことで，DB エリアファイルの上限値までは HADB サーバが自動的に増分します。DB エリアファイルの上限値については，「2.4.5　DB エリアの自動増分」を参照してください。

- **ブロックスペシャルファイルの場合**

  OS でサポートしている LVM を使用している場合，LV（Logical Volume）を拡張してください。LV の拡張方法については，OS のマニュアルを参照してください。

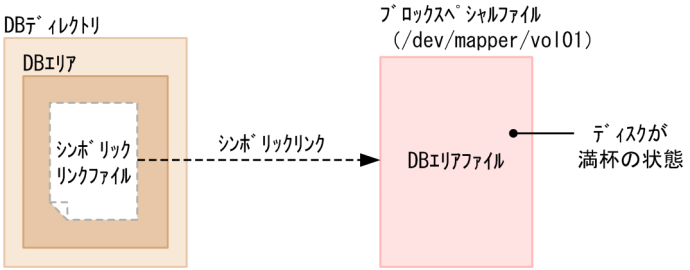
  LV を拡張すると，DB エリアファイルの上限値までは HADB サーバが自動的に増分します。DB エリアファイルの上限値については，「2.4.5　DB エリアの自動増分」を参照してください。

(2)　DB エリアファイルの格納場所を変更する

DB エリアファイルの格納場所を，容量の大きな別のディスクに変更してください。DB エリアファイルの格納場所を変更する方法を説明します。

格納場所を変更する前の DB エリアファイルの構成例を次の図に示します。

図 15-1　変更前の DB エリアファイルの構成例



DB ディレクトリ下の DB エリアからシンボリックリンクされているブロックスペシャルファイル（/dev/mapper/vol01）は，ディスクが満杯の状態です。空き容量がないため，格納場所を変更する必要があります。

DB エリアファイルの格納場所を変更する方法については，「11.10.5　データ用 DB エリアファイルの格納場所を変更する方法」の「(2)　データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する」を参照してください。

(3)　作業表用 DB エリアファイルのサイズを小さくする

レギュラーファイルで構成された作業表用 DB エリアファイルのサイズが大きくなったことで，ディスクが満杯になった場合は，次の手順で作業表用 DB エリアファイルのサイズを小さくしてください。

手順

1. adbstop コマンドで，HADB サーバを正常終了してください。
2. OS のコマンドなどで，作業表用 DB エリアファイルを削除してください。
3. adbstart コマンドで，HADB サーバを正常開始してください。

15.3.2　DB エリアの自動増分の失敗が原因の場合

DB エリアファイルを格納するディスクの空き容量が少ないときに自動増分が行われると，自動増分に失敗することがあります。このとき，KFAA41206-I メッセージが出力され，メッセージ中に DB エリアファイルを格納するディスクの空き容量がないことを示すエラー番号が表示されます。この場合，DB エリアファイルの格納領域を拡張することを検討してください。

なお，自動増分に失敗した DB エリアファイルは，一時的に満杯状態※となり，表データおよびインデクスデータの格納対象から除かれます。DB エリアを構成するすべての DB エリアファイルが満杯状態になると，DB エリアが満杯になったことを示すKFAA30756-E メッセージが出力されます。この場合，DB エリアファイルの格納領域を拡張してください。

DB エリアファイルの格納領域の拡張手順については，「15.3.1　DB エリアファイルの容量増加が原因の場合」を参照してください。

注※

一時的な満杯状態は，次に示す契機で解除されます。

- HADB サーバを再起動したとき
- DB エリアを構成するすべての DB エリアファイルが満杯状態になったとき

> **メモ**
>
> 作業表用 DB エリアファイルは，満杯状態にはなりません。

15.3.3　DB エリアファイル以外のファイルの容量増加が原因の場合

DB ディレクトリ下に格納されている DB エリアファイル以外のファイルの容量が増加し，ファイルを格納するディスクの空き容量が少なくなった場合の対処方法について説明します。

(1)　DB ディレクトリ下の実体ファイルが原因の場合

DB ディレクトリの格納場所を，容量の大きな別のディスクに変更してください。変更方法を次に示します。

変更方法

1. **HADB サーバを終了する**

   adbstop コマンドを実行して HADB サーバを終了します。

2. **DB ディレクトリ下のファイルをコピーする**

   OS のcp コマンドを実行して，DB ディレクトリ下にあるすべてのファイルを，容量の大きな別のディスクにコピーします。

3. **サーバ定義のadb_db_path オペランドを変更する**

   adb_db_path オペランドに，コピー先の DB ディレクトリパスを絶対パス名で指定します。サーバ定義ファイルは，サーバディレクトリ下（$ADBDIR/conf）に格納されているserver.def ファイルです。テキストエディタで開き，編集してください。

4. **HADB サーバを開始する**

   adbstart コマンドを実行して HADB サーバを開始します。

(2)　シンボリックリンクしているファイルが原因の場合

DB ディレクトリ下のシンボリックリンクしているファイルの格納場所を，容量の大きな別のディスクに変更してください。そのあとで，シンボリックリンクのリンク先を変更してください。変更方法を次に示します。

変更方法

1. **HADB サーバを終了する**

   adbstop コマンドを実行して HADB サーバを終了します。

2. **リンク先のファイルをコピーする**

OS のcp コマンドを実行して，リンク先にあるすべてのファイルを，容量の大きな別のディスクにコピーします。

3. シンボリックリンクを変更する

   OS のln コマンドを実行して，容量の大きな別のディスクにシンボリックリンクのリンク先を変更します。

4. HADB サーバを開始する

   adbstart コマンドを実行して HADB サーバを開始します。

(3) 統計ログファイルの容量増加が原因の場合

統計ログファイルの出力先を，容量の大きな別のディスクに変更してください。変更方法を次に示します。

変更方法

1. HADB サーバを正常終了する

   adbstop コマンドを実行して HADB サーバを正常終了します。

2. 統計ログファイルの出力先を変更する

   adb_sta_log_path オペランドで，統計ログファイルの出力先ディレクトリを指定します。容量の大きな別のディスクに対して，統計ログファイルが出力されるようにしてください。

   サーバ定義adb_sta_log_path オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式)」のadb_sta_log_path オペランドを参照してください。

   サーバ定義ファイルは，サーバディレクトリ下（$ADBDIR/conf）に格納されているserver.def ファイルです。テキストエディタで開き，編集してください。

3. HADB サーバを正常開始する

   adbstart コマンドを実行して HADB サーバを正常開始します。

統計ログファイルの出力先を変更した場合，変更前の出力先に，古い統計ログファイルが残ります。古い統計ログファイルが不要な場合は，手動で削除する必要があります。統計ログファイルの削除方法については，「10.10.5　統計ログファイルの運用」の「(4)　統計ログファイルを削除する場合」を参照してください。

15.4　ディスクに関するトラブル

ここでは，ディスクに関するトラブルが発生し，HADB が使用するディレクトリを回復する必要がある場合の対処方法を説明します。

15.4.1　データベースの回復

ディスク障害などのトラブルが発生しデータベースの回復が必要な場合，「10.3.1　バックアップの取得方法」に従って取得したデータベースのバックアップから回復してください。

データベースの回復方法については，「10.3.2　バックアップからの回復方法」を参照してください。

15.4.2　サーバディレクトリの回復

ディスクに対してトラブルが発生しサーバディレクトリの回復が必要な場合，HADB サーバを再インストールして，サーバディレクトリを回復してください。

なお，サーバディレクトリを回復する場合，次に示す条件を満たしていないと，トラブル発生前の状態に戻せません。

- HADB サーバの DB ディレクトリでは，ディスク障害などのトラブルが発生していない。
- サーバディレクトリ下（$ADBDIR/conf，および$ADBDIR/sample）に格納されているすべてのファイルのバックアップを取得している。
- サーバディレクトリ下（$ADBDIR）にユーザがファイルを作成している場合，そのファイルのバックアップを取得している。

回復手順

1. HADB サーバを再インストールする

   HADB サーバを再インストールして，サーバディレクトリを作成し直してください。HADB サーバのインストールについては，「8.2.2　インストール手順」を参照してください。

2. 定義ファイルとサンプル AP をバックアップから回復する

   バックアップから，サーバディレクトリ下（$ADBDIR/conf，および$ADBDIR/sample）に格納されているすべてのファイルを格納し直してください。

3. ユーザが作成したファイルをバックアップから回復する

   サーバディレクトリ下（$ADBDIR）にユーザがファイルを作成していた場合，バックアップから，そのファイルを格納し直してください。

4. HADB サーバを再開始する

   adbstart コマンドを実行してください。HADB サーバが再開始されます。

15.4.3　クライアントディレクトリの回復

ディスクに対してトラブルが発生しクライアントディレクトリの回復が必要な場合，HADB クライアントを上書きインストールして，クライアントディレクトリを回復してください。

なお，クライアントディレクトリを回復する場合，次に示す条件を満たしていないと，トラブル発生前の状態に戻せません。

- クライアントディレクトリ下（Windows 版の場合は%ADBCLTDIR%¥conf，Linux 版の場合は$ADBCLTDIR/conf）に格納されている定義ファイルのバックアップを取得している。
- クライアントディレクトリ下（Windows 版の場合は%ADBCLTDIR%，Linux 版の場合は$ADBCLTDIR）にユーザがファイルを作成している場合，そのファイルのバックアップを取得している。

回復手順

1. HADB クライアントを回復する

   HADB クライアントを上書きインストールして，クライアントディレクトリを回復してください。マニュアル『HADB　AP 開発ガイド』の『HADB クライアントのインストールおよびアンインストール』を参照してください。

2. 定義ファイルをバックアップから回復する

   バックアップから，クライアントディレクトリ（Windows 版の場合は%ADBCLTDIR%¥conf，Linux 版の場合は$ADBCLTDIR/conf）下に格納されている定義ファイルを格納し直してください。

3. ユーザが作成したファイルをバックアップから回復する

   クライアントディレクトリ下（Windows 版の場合は%ADBCLTDIR%，Linux 版の場合は$ADBCLTDIR）にユーザがファイルを作成していた場合，バックアップから，そのファイルを格納し直してください。

15.5　DB ディレクトリ下のファイルに関するトラブル

ここでは，DB ディレクトリ下に格納されるファイルで発生したトラブルの対処方法について説明します。

15.5.1　DB エリアファイルに障害が発生した場合の対処

DB エリアファイルに障害が発生した場合は，出力されたメッセージの対処方法に従ってください。

DB エリアファイルの障害を解消できない場合は，バックアップからデータベースを回復してください。

15.5.2　システムログファイルでトラブルが発生した場合の対処

システムログファイル（マスタログファイル，およびユーザログファイル）でトラブルが発生した場合，次に示すメッセージが出力されます。

- KFAA30959-E メッセージ
- KFAA41205-E メッセージ

上記のメッセージに出力されたシステムコールのリターンコードを確認しても発生したトラブルを解消できない場合，次に示す作業を行ってください。

作業手順

1. HADB サーバを再度開始できるか確認する

   HADB サーバをadbstop コマンドで終了，およびadbstart コマンドで再度開始できるか確認してください。HADB サーバを再度開始すると，前回稼働時に利用されていたマスタログファイルの領域は初期化されます。そのため，マスタログファイルの容量不足が原因の場合，HADB サーバを再度開始することでトラブルを解消できます。

   HADB サーバを再度開始できない場合は，手順 2.に進んでください。

2. DB ディレクトリが格納されているディスクの空き容量を確認する

   システムログファイルは，DB ディレクトリ下（$DBDIR/ADBSYS/ADBSLG）に格納されます。DB ディレクトリが格納されているディスクの空き容量があるかどうかを確認してください。ディスクの空き容量がない場合，「15.3.3　DB エリアファイル以外のファイルの容量増加が原因の場合」に従って，ディスクの空き容量を確保してください。

   ディスクの空き容量が十分にある場合，手順 3.に進んでください。

3. システムログファイルが壊れているかを確認する

   ディスクの空き容量は十分にあるがトラブルを解消できない場合，システムログファイルが壊れているおそれがあります。システムログファイルにアクセスしてもエラーとなる場合はシステムログファイルが壊れているため，手順 4.に進んでください。

4. システムログファイルを回復する

システムログファイルが壊れている場合，バックアップから回復する必要があります。「10.3.1 バックアップの取得方法」に従って取得したデータベースのバックアップがある場合は，「10.3.2 バックアップからの回復方法」に従って，バックアップから回復してください。

なお，データベースのバックアップを取得していない場合，システムログファイルを回復できないため，データベースを再作成する必要があります。

15.6　メモリに関するトラブル

KFAA30930-E メッセージまたはKFAA40007-E メッセージが出力された場合の対策について説明します。次に示す方法で，HADB サーバが使用できるメモリに余裕を持たせてください。

15.6.1　HADB サーバ開始時にメモリが不足した場合の対処方法

HADB サーバ開始時にKFAA40007-E メッセージが出力された場合の対処方法を次に示します。対処方法は，サーバ定義adb_sys_memory_limit オペランドの指定の有無によって異なります。

(1)　サーバ定義 adb_sys_memory_limit オペランドを指定している場合

HADB サーバが使用するメモリが不足しています。サーバ定義のadb_sys_memory_limit オペランドの指定値を大きくしてください。対処手順を次に示します。

手順

1. サーバ定義を変更する

   サーバ定義のadb_sys_memory_limit オペランドに，次に示す計算式で求めた値以上の値を指定してください。

   計算式（単位：メガバイト）

   ```
   HADBサーバが使用するメモリの最大サイズ＝
                   サーバ定義adb_sys_memory_limitオペランドの値(単位:メガバイト)
                   ＋↑get_size÷(100×1,024×1,024)↑×100
   ```

   変数の説明

   get_size：KFAA40007-E メッセージの変数 *bb....bb* の値（単位：バイト）

2. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_sys_memory_limit オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_memory_limit オペランドを参照してください。

(2)　サーバ定義 adb_sys_memory_limit オペランドを指定していない場合

KFAA40007-E メッセージに出力された，不足しているメモリの種類を示す変数 *aa....aa* を確認してください。*aa....aa* によって対策が異なります。

(a) aa....aa が PROCESS の場合

プロセス共通メモリが不足しています。サーバ定義のadb_sys_proc_area_max オペランドの指定値を大きくしてください。対処手順を次に示します。

手順

1. サーバ定義を変更する

   サーバ定義のadb_sys_proc_area_max オペランドに，次に示す計算式で求めた値以上の値を指定してください。

   計算式（単位：メガバイト）

   ```
   確保できるプロセス共通メモリの最大サイズ＝
     （↑max_size÷100↑×100）＋（↑get_size÷(100×1,024×1,024)↑×100）＋1
   ```

   変数の説明

   max_size：adb_sys_proc_area_max オペランドの確保できるプロセス共通メモリの最大サイズに指定していた値（単位：メガバイト）

   get_size：KFAA40007-E メッセージの変数 *bb....bb* の値（単位：バイト）

2. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_sys_proc_area_max オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_proc_area_max オペランドを参照してください。

(b) aa....aa が THREAD の場合

リアルスレッド固有メモリが不足しています。サーバ定義のadb_sys_rthd_area_max オペランドの指定値を大きくしてください。対処手順を次に示します。

手順

1. サーバ定義を変更する

   サーバ定義のadb_sys_rthd_area_max オペランドに，次に示す計算式で求めた値以上の値を指定してください。

   計算式（単位：メガバイト）

   ```
   確保できるリアルスレッド固有メモリの最大サイズ＝
     （↑max_size÷100↑×100）＋（↑get_size÷(100×1,024×1,024)↑×100）＋1
   ```

変数の説明

max_size：adb_sys_rthd_area_max オペランドの確保できるリアルスレッド固有メモリの最大サイズに指定していた値（単位：メガバイト）

get_size：KFAA40007-E メッセージの変数 *bb....bb* の値（単位：バイト）

2. HADB サーバを開始する

adbstart コマンドを実行して，HADB サーバを開始してください。

> **メモ**
>
> サーバ定義のadb_sys_rthd_area_max オペランドについては，「7.2.2　性能に関するオペランド（set 形式）」の adb_sys_rthd_area_max オペランドを参照してください。

(c)　aa....aa が HEAP の場合

ヒープメモリが不足しています。対処手順を次に示します。

手順

1. 不要なプロセスがないかどうかを確認してください。不要なプロセスがある場合は，そのプロセスを終了または削除してください。そのあとに，adbstart コマンドを実行して，HADB サーバを開始してください。
2. 手順 1.を実施してもメモリが不足する場合は，次の対処をしてください。
   - OS を再起動する
   - adbstart コマンドを実行して，HADB サーバを開始する
3. 手順 2.を実施してもメモリが不足する場合は，次の対処をしてください。
   - カーネルパラメタの設定を変更し，プロセスで使用できるメモリの上限を大きくする
   - OS を再起動する
   - adbstart コマンドを実行して，HADB サーバを開始する

(d)　aa....aa が SHARE の場合

共有メモリが不足しています。対処手順を次に示します。

手順

1. 不要なプロセス，および不要な共有メモリがないかどうかを確認してください。不要なプロセスがある場合は終了してください。不要な共有メモリがある場合は削除してください。そのあとに，adbstart コマンドを実行して，HADB サーバを開始してください。
2. 手順 1.を実施しても共有メモリが不足する場合は，次の対処をしてください。
   - OS を再起動する

- adbstart コマンドを実行して，HADB サーバを開始する

3. 手順 2.を実施しても共有メモリが不足する場合は，次の対処をしてください。
   - カーネルパラメタの設定を変更し，共有メモリの上限を大きくする

     また，HADB サーバの共有メモリにHugePages を適用している場合，カーネルパラメタがHugePages を使用できる設定になっていることを確認してください。詳細については，「6.2　カーネルパラメタの見積もり」を参照してください。
   - OS を再起動する
   - adbstart コマンドを実行して，HADB サーバを開始する

15.6.2　SQL 文実行時またはコマンド実行時にメモリが不足した場合の対処方法

SQL 文実行時またはコマンド実行時に次のメッセージが出力された場合の対処方法を示します。

出力されたメッセージ

- KFAA30930-E
- KFAA40007-E

対処方法は，サーバ定義adb_sys_memory_limit オペランドの指定の有無によって異なります。

(1)　サーバ定義 adb_sys_memory_limit オペランドを指定している場合

HADB サーバが使用するメモリが不足しています。サーバ定義のadb_sys_memory_limit オペランドの指定値を大きくしてください。対処手順を次に示します。

手順

1. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

2. サーバ定義を変更する

   サーバ定義のadb_sys_memory_limit オペランドに，次に示す計算式で求めた値以上の値を指定してください。

   計算式（単位：メガバイト）

   ```
   HADBサーバが使用するメモリの最大サイズ＝
                   サーバ定義adb_sys_memory_limitオペランドの値(単位:メガバイト)
                   ＋↑get_size÷(100×1,024×1,024)↑×100
   ```

   変数の説明

   get_size：KFAA30930-E メッセージおよびKFAA40007-E メッセージの変数 *bb....bb* の値（単位：バイト）

3. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

4. SQL 文またはコマンドを再実行する

> **メモ**
>
> サーバ定義のadb_sys_memory_limit オペランドについては，「7.2.2　性能に関するオペランド(set 形式)」の adb_sys_memory_limit オペランドを参照してください。

(2)　サーバ定義 adb_sys_memory_limit オペランドを指定していない場合

KFAA30930-E メッセージおよびKFAA40007-E メッセージに出力された，不足しているメモリの種類を示す変数 *aa....aa* を確認してください。*aa....aa* によって対策が異なります。

(a)　aa....aa が PROCESS の場合

プロセス共通メモリが不足しています。サーバ定義のadb_sys_proc_area_max オペランドの指定値を大きくしてください。対処手順を次に示します。

手順

1. HADB サーバが使用したメモリ容量の最大値を確認する

   adbls -d mem コマンドを実行してください。出力結果から，出力項目「THREAD-NO」が0 の行に表示されている「MAX」の値を確認してください。

   adbls -d mem コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d mem (メモリの使用状況の表示)』を参照してください。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadb_sys_proc_area_max オペランドに，次に示す計算式で求めた値以上の値を指定してください。

   計算式（単位：メガバイト）

   確保できるプロセス共通メモリの最大サイズ＝
   （↑max_size÷(100×1,024×1,024)↑×100）＋（↑get_size÷(100×1,024×1,024)↑×100）＋1

   変数の説明

   max_size：手順 1.で確認した「MAX」の値（単位：バイト）

   get_size：出力されたメッセージの変数 *bb....bb* の値（単位：バイト）

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. SQL 文またはコマンドを再実行する

> **メモ**
>
> サーバ定義のadb_sys_proc_area_max オペランドについては，「7.2.2　性能に関するオペランド(set 形式)」の adb_sys_proc_area_max オペランドを参照してください。

(b)　aa....aa が THREAD の場合

リアルスレッド固有メモリが不足しています。サーバ定義のadb_sys_rthd_area_max オペランドの指定値を大きくしてください。対処手順を次に示します。

手順

1. HADB サーバが使用したメモリ容量の最大値を確認する

   adbls -d mem コマンドを実行してください。出力結果から，出力項目「THREAD-NO」が1 以上の行に表示されている「MAX」のうち，最も大きな値を確認してください。

   adbls -d mem コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d mem (メモリの使用状況の表示)』を参照してください。

2. HADB サーバを終了する

   adbstop コマンドを実行して，HADB サーバを終了してください。

3. サーバ定義を変更する

   サーバ定義のadb_sys_rthd_area_max オペランドに，次に示す計算式で求めた値以上の値を指定してください。

   計算式（単位：メガバイト）

   ```
   確保できるリアルスレッド固有メモリの最大サイズ＝
     （↑max_size÷(100×1,024×1,024)↑×100）＋（↑get_size÷(100×1,024×1,024)↑×100）＋1
   ```

   変数の説明

   max_size：手順 1.で確認した最も大きな「MAX」の値（単位：バイト）

   get_size：出力されたメッセージの変数 *bb....bb* の値（単位：バイト）

4. HADB サーバを開始する

   adbstart コマンドを実行して，HADB サーバを開始してください。

5. SQL 文またはコマンドを再実行する

> **メモ**
>
> サーバ定義のadb_sys_rthd_area_max オペランドについては，「7.2.2　性能に関するオペランド(set 形式)」の adb_sys_rthd_area_max オペランドを参照してください。

(c) aa....aa が HEAP の場合

ヒープメモリが不足しています。対処手順を次に示します。

手順

1. 不要なプロセスがないかどうかを確認してください。不要なプロセスがある場合は，そのプロセスを終了または削除してください。
2. 手順 1.を実施してもメモリが不足する場合は，次の対処をしてください。
   - adbstop コマンドを実行して，HADB サーバを終了する
   - adbstart コマンドを実行して，HADB サーバを開始する
3. 手順 2.を実施してもメモリが不足する場合は，次の対処をしてください。
   - adbstop コマンドを実行して，HADB サーバを終了する
   - OS を再起動する
   - adbstart コマンドを実行して，HADB サーバを開始する

   クライアントで発生した場合は，クライアントの OS を再起動してください。
4. 手順 3.を実施してもメモリが不足する場合は，次の対処をしてください。
   - adbstop コマンドを実行して，HADB サーバを終了する
   - カーネルパラメタの設定を変更し，プロセスで使用できるメモリの上限を大きくする
   - OS を再起動する
   - adbstart コマンドを実行して，HADB サーバを開始する

   クライアントで発生した場合は，クライアントのカーネルパラメタの設定を変更し，プロセスで使用できるメモリの上限を大きくしてください。そのあとに，クライアントの OS を再起動してください。
5. SQL 文またはコマンドを再実行する

(d) aa....aa が SHARE の場合

共有メモリが不足しています。対処手順を次に示します。

手順

1. 不要なプロセス，および不要な共有メモリがないかどうかを確認してください。不要なプロセスがある場合は終了してください。不要な共有メモリがある場合は削除してください。
2. 手順 1.を実施しても共有メモリが不足する場合は，次の対処をしてください。
   - adbstop コマンドを実行して，HADB サーバを終了する
   - adbstart コマンドを実行して，HADB サーバを開始する
3. 手順 2.を実施しても共有メモリが不足する場合は，次の対処をしてください。
   - adbstop コマンドを実行して，HADB サーバを終了する

- OS を再起動する
- adbstart コマンドを実行して，HADB サーバを開始する

クライアントで発生した場合は，クライアントの OS を再起動してください。

4. 手順 3.を実施しても共有メモリが不足する場合は，次の対処をしてください。
   - adbstop コマンドを実行して，HADB サーバを終了する
   - カーネルパラメタの設定を変更し，共有メモリの上限を大きくする

     また，HADB サーバの共有メモリにHugePages を適用している場合，カーネルパラメタがHugePages を使用できる設定になっていることを確認してください。詳細については，「6.2　カーネルパラメタの見積もり」を参照してください。
   - OS を再起動する
   - adbstart コマンドを実行して，HADB サーバを開始する

   クライアントで発生した場合は，クライアントの共有メモリの上限を大きくしてください。そのあとに，クライアントの OS を再起動してください。
5. SQL 文またはコマンドを再実行する

15.7　DB エリアに関するトラブル

ここでは，データ用 DB エリアに関するトラブルがあった場合の対処方法について説明します。

15.7.1　データ用 DB エリアが満杯になった場合

ここでは，データ用 DB エリアが満杯になった場合の対処方法について説明します。

データ用 DB エリアが満杯になると，KFAA30756-E メッセージが出力されます。データ用 DB エリアが満杯になる場合を次に示します。

- データ用 DB エリアを構成するデータ用 DB エリアファイルを格納したディスクの空き容量がなくなった場合
- データ用 DB エリアを構成するデータ用 DB エリアファイルが自動増分の上限値に達したため，自動増分できない場合

データ用 DB エリアが満杯になった場合は，adbmodarea コマンドで，対象のデータ用 DB エリアを拡張（データ用 DB エリアファイルを追加）する必要があります。「11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照して，対象のデータ用 DB エリアを拡張してください。

なお，1 つのデータ用 DB エリアを構成するデータ用 DB エリアファイル数の上限は，1,024 個です。上限を超えて，データ用 DB エリアファイルを追加できません。そのため，1,024 個のデータ用 DB エリアファイルで構成されているデータ用 DB エリアの容量を拡張したい場合は，「15.7.3　データ用 DB エリアを拡張できなくなった場合」を参照してください。

また，データ用 DB エリアファイルを格納したディスクの空き容量がなくなった場合は，次の方法でも対処できます。

- DB エリアファイルの容量を拡張する
- DB エリアファイルの格納場所を変更する

詳細については，「15.3.1　DB エリアファイルの容量増加が原因の場合」を参照してください。

15.7.2　データ用 DB エリアを追加できなくなった場合

ここでは，新たにデータ用 DB エリアを追加できなくなった場合の対処方法について説明します。

HADB サーバが管理できるデータ用 DB エリアの数は，最大 1,014 個です。上限を超えて，データ用 DB エリアを追加できません。

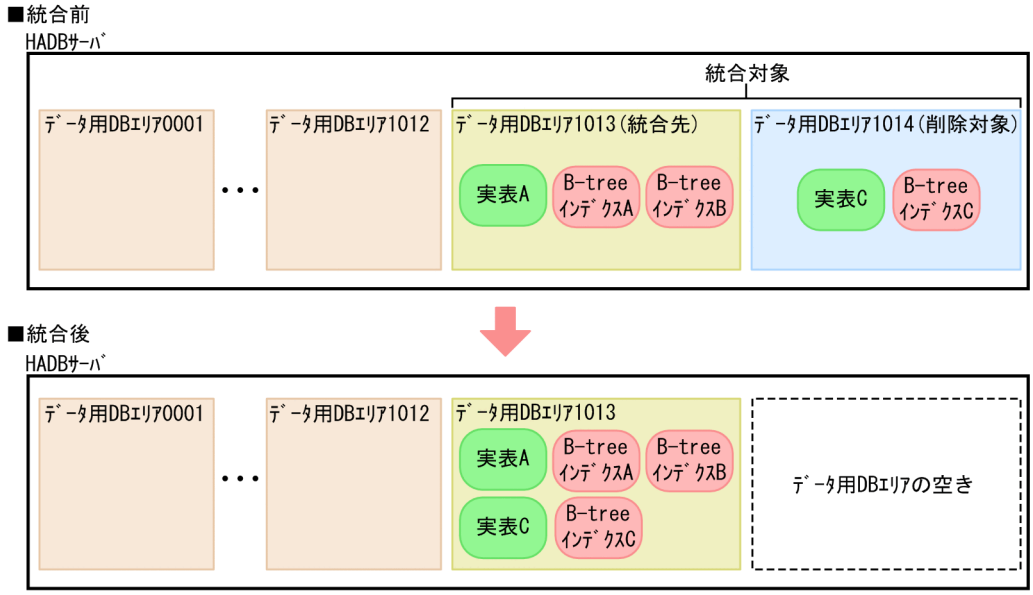
そのため，1,014 個のデータ用 DB エリアがある場合に，新たにデータ用 DB エリアを追加したいときは，既存のデータ用 DB エリアを統合して，管理できるデータ用 DB エリアの数の上限に余裕を持たせる必要があります。

> **メモ**
>
> データ用 DB エリアを統合する場合は，データ量が少ない（数メガバイト）実表やインデクスが格納されているデータ用 DB エリアを対象とすることを推奨します。詳細については，「5.6.1 データ用 DB エリアを設計する際の考慮点」を参照してください。

(1) 表とインデクスを格納したデータ用 DB エリアを削除して統合する（adbidxrebuild コマンドの実行が不要な場合）

表とインデクスを格納したデータ用 DB エリアを削除して統合する方法を，次に示します。

図 15-2 既存のデータ用 DB エリアの統合例 1



■実表と B-tree インデクスの関係

- 実表A に，B-tree インデクスA および B-tree インデクスB が定義されています。
- 実表C に，B-tree インデクスC が定義されています。

1. 削除対象のデータ用 DB エリアに格納された表のデータを出力する

   adbexport コマンドで，実表のデータを出力データファイルに出力してください。ここでは，「データ用 DB エリア1014」に格納された実表C のデータが対象になります。

2. 削除対象のデータ用 DB エリアに格納された表やインデクスを削除する

   削除対象のデータ用 DB エリアに定義されている実表およびインデクスを，DROP TABLE 文で削除してください。DROP TABLE 文を実行すると実表に定義されているインデクスもあわせて削除されます。ここでは，「データ用 DB エリア1014」に格納された実表C，および B-tree インデクスC が対象になります。

3. 削除対象のデータ用 DB エリアを削除する

   削除対象のデータ用 DB エリアを，adbmodarea コマンドで削除してください。ここでは，「データ用 DB エリア1014」が対象になります。

4. 統合先のデータ用 DB エリアに表やインデクスを再定義する

   手順 2.で削除した実表およびインデクスを，統合先のデータ用 DB エリアに，CREATE TABLE 文やCREATE INDEX 文で再定義してください。ここでは，「データ用 DB エリア1013」に対して実表C，および B-tree インデクスC を再定義してください。

5. 出力した表の全データを再格納する

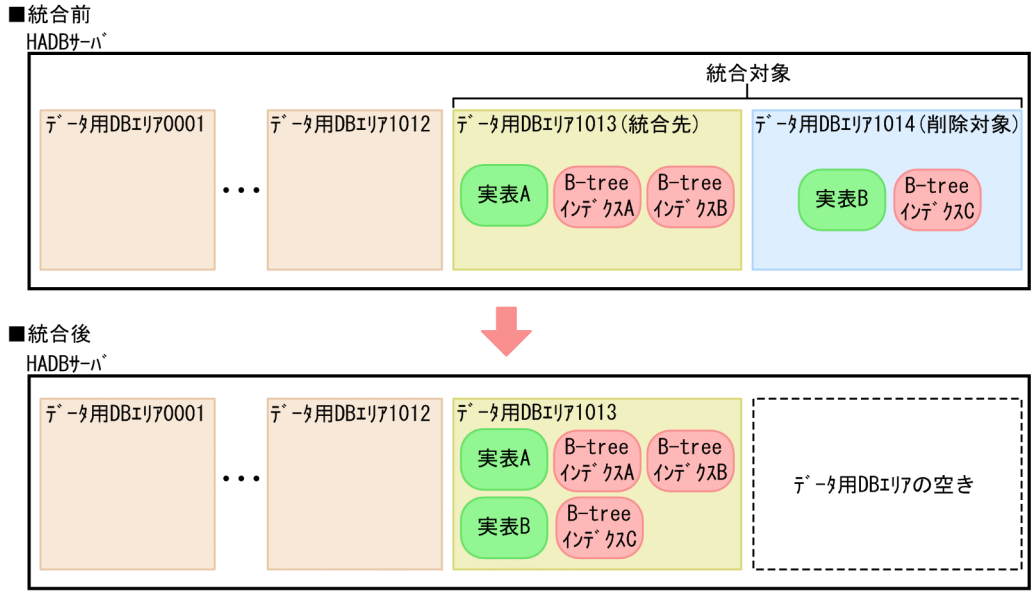
   手順 1.で出力した実表の全データを，手順 4.で再定義した実表にadbimport コマンドで再格納してください。ここでは，再定義した実表C に対してデータを再格納してください。

既存のデータ用 DB エリアを統合することで，データ用 DB エリアの数の上限に余裕ができたため，新たにデータ用 DB エリアを追加できます。

(2) 表とインデクスを格納したデータ用 DB エリアを削除して統合する（adbidxrebuild コマンドの実行が必要な場合）

表とインデクスを格納したデータ用 DB エリアを削除して統合する方法を，次に示します。

図 15-3 既存のデータ用 DB エリアの統合例 2



■実表と B-tree インデクスの関係

- 実表A に，B-tree インデクスA，B-tree インデクスB および B-tree インデクスC が定義されています。
- 実表B に，インデクスは定義されていません。

1. 削除対象のデータ用 DB エリアに格納された表のデータを出力する

   adbexport コマンドで，実表のデータを出力データファイルに出力してください。ここでは，「データ用 DB エリア1014」に格納された実表B のデータが対象になります。

2. 削除対象のデータ用 DB エリアに格納された表やインデクスを削除する

   削除対象のデータ用 DB エリアに定義されている実表をDROP TABLE 文で削除してください。また，インデクスをDROP INDEX 文で削除してください。ここでは，「データ用 DB エリア1014」に格納された実表B，および B-tree インデクスC が対象になります。

3. 削除対象のデータ用 DB エリアを削除する

   削除対象のデータ用 DB エリアを，adbmodarea コマンドで削除してください。ここでは，「データ用 DB エリア1014」が対象になります。

4. 統合先のデータ用 DB エリアに表やインデクスを再定義する

   手順 2.で削除した実表およびインデクスを，統合先のデータ用 DB エリアに，CREATE TABLE 文やCREATE INDEX 文で再定義してください。ここでは，「データ用 DB エリア1013」に対して実表B を再定義してください。また，実表A に対して B-tree インデクスC を再定義してください。

5. 出力した表の全データを再格納する

   手順 1.で出力した実表の全データを，手順 4.で再定義した実表にadbimport コマンドで再格納してください。ここでは，再定義した実表B に対してデータを再格納してください。

6. インデクスを再作成する

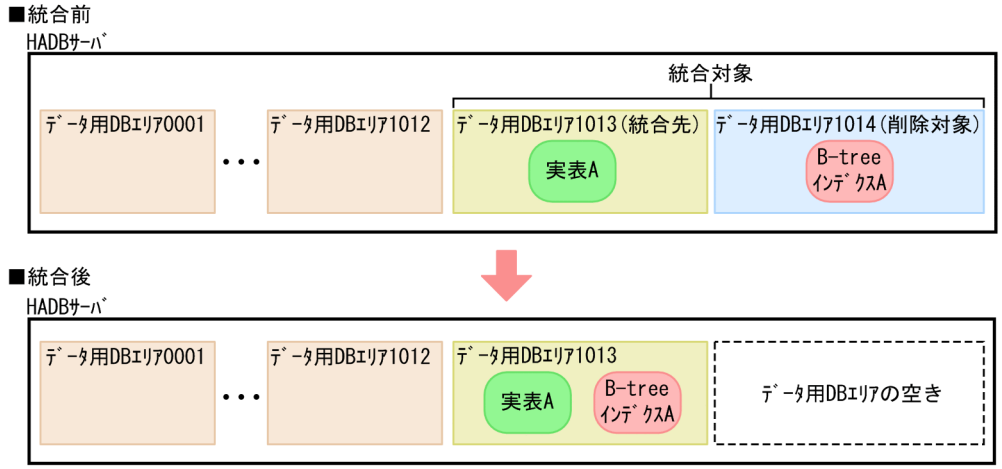
   手順 4.で再定義したインデクスは未完状態のため，インデクスの再作成が必要です。adbidxrebuild コマンドでインデクスを再作成してください。ここでは，B-tree インデクスC を再定義した実表A に対して，adbidxrebuild コマンドを実行してください。

既存のデータ用 DB エリアを統合することで，データ用 DB エリアの数の上限に余裕ができたため，新たにデータ用 DB エリアを追加できます。

(3)　インデクスだけを格納したデータ用 DB エリアを削除して統合する

インデクスだけを格納したデータ用 DB エリアを削除して統合する方法を，次に示します。

図 15-4　既存のデータ用 DB エリアの統合例 3



■実表と B-tree インデクスの関係

実表A に，B-tree インデクスA が定義されています。

1. 削除対象のデータ用 DB エリアに格納されたインデクスを削除する

   削除対象のデータ用 DB エリアに定義されているインデクスを，DROP INDEX 文で削除してください。ここでは，「データ用 DB エリア1014」に格納された B-tree インデクスA が対象になります。

2. 削除対象のデータ用 DB エリアを削除する

   削除対象のデータ用 DB エリアを，adbmodarea コマンドで削除してください。ここでは，「データ用 DB エリア1014」が対象になります。

3. 統合先のデータ用 DB エリアにインデクスを再定義する

   手順 1.で削除したインデクスを，統合先のデータ用 DB エリアに，CREATE INDEX 文で再定義してください。ここでは，「データ用 DB エリア1013」に格納されている実表A に対して，B-tree インデクスA を再定義してください。

4. インデクスを再作成する

   手順 3.で再定義したインデクスは未完状態のため，インデクスの再作成が必要です。adbidxrebuild コマンドでインデクスを再作成してください。ここでは，B-tree インデクスA を再定義した実表A に対して，adbidxrebuild コマンドを実行してください。

既存のデータ用 DB エリアを統合することで，データ用 DB エリアの数の上限に余裕ができたため，新たにデータ用 DB エリアを追加できます。

15.7.3 データ用 DB エリアを拡張できなくなった場合

ここでは，adbmodarea コマンドを使用して，既存のデータ用 DB エリアを拡張できなくなった場合の対処方法について説明します。

1 つのデータ用 DB エリアを構成するデータ用 DB エリアファイル数の上限は，1,024 個です。上限を超えて，データ用 DB エリアファイルを追加できません。

そのため，1,024 個のデータ用 DB エリアファイルで構成されているデータ用 DB エリアの容量を拡張したい場合は，データ用 DB エリアを再作成する必要があります。手順を次に示します。

1. 表の全データを出力する

   adbexport コマンドで，実表の全データを出力データファイルに出力してください。

2. 表やインデクスを削除する

   対象のデータ用 DB エリアに定義されている実表やインデクス（B-tree インデクス，テキストインデクスおよびレンジインデクス）を削除します。

   実表だけが定義されている場合，または実表とインデクスが定義されている場合は，DROP TABLE 文で削除してください。DROP TABLE 文を実行すると実表に定義されているインデクスもあわせて削除されます。インデクスだけが定義されている場合は，DROP INDEX 文で削除してください。

3. データ用 DB エリアを削除する

   対象のデータ用 DB エリアを，adbmodarea コマンドで削除してください。

4. 新たにデータ用 DB エリアを追加する

   adbmodarea コマンドで，新たにデータ用 DB エリアを追加してください。

   その際，DB エリア追加・変更オプションのadbaddarea オペランドの-i オプションに指定する初期確保サイズ，および作成するデータ用 DB エリアファイル数については，今後追加するデータ量を考慮して，十分な容量を指定してください。

5. 表やインデクスを再定義する

   手順 2.で削除した実表やインデクスを，手順 4.で新たに追加したデータ用 DB エリアに，CREATE TABLE 文やCREATE INDEX 文で再定義してください。

6. 出力した表の全データを再格納する

   手順 1.で出力した実表の全データを，手順 5.で再定義した実表にadbimport コマンドで再格納してください。

15.8　実表に関するトラブル

ここでは，実表に対して発生したトラブルの対処方法について説明します。

15.8.1　実表が更新不可状態になった場合

次に示すコマンドが中断した場合，表の情報とインデクスの情報との不整合が原因で，処理対象の実表が更新できない状態になることがあります。これを，**実表の更新不可状態**といいます。

■対象のコマンド

- adbimport コマンド
- adbidxrebuild コマンド
- adbunarchivechunk コマンド

更新不可状態の実表に対して，次に示す操作を行うとエラーになります。

■エラーになる操作

- ALTER TABLE 文の実行
- CREATE INDEX 文の実行
- INSERT 文の実行
- UPDATE 文の実行
- DELETE 文の実行
- PURGE CHUNK 文の実行
- adbmergechunk コマンドの実行
- adbchgchunkstatus コマンドの実行
- adbchgchunkcomment コマンドの実行
- adbimport コマンドの実行※1
- adbidxrebuild コマンドの実行※1
- adbarchivechunk コマンドの実行
- adbunarchivechunk コマンドの実行※2

注※1

adbunarchivechunk コマンドが中断して，実表が更新不可状態になった場合にエラーになります。adbimport コマンドまたはadbidxrebuild コマンドが中断して，実表が更新不可状態になった場合はエラーになりません。

注※2

adbimport コマンドまたはadbidxrebuild コマンドが中断して，実表が更新不可状態になった場合にエラーになります。adbunarchivechunk コマンドが中断して，実表が更新不可状態になった場合はエラーになりません。

> **メモ**
>
> TRUNCATE TABLE 文は実行できます。実表の行データはすべて削除されるため，実表の更新不可状態は解除されます。

(1) 実表の更新不可状態を解除する方法

実表の更新不可状態を解除する手順を次に示します。

手順

1. 対象の実表が更新不可状態かどうかを確認する

   実表の更新不可状態は，adbdbstatus コマンドで確認できます。「10.9.2 実表の状態や使用量を確認したい場合」を参照してください。

   実表の更新不可状態の場合，手順 2.に進んでください。

2. 実表の更新不可状態を解除する

   中断したコマンドを再実行してください。

   - **adbimport コマンドが中断した場合**

     adbimport コマンドを再実行してください。詳細については，マニュアル『HADB コマンドリファレンス』の『adbimport（データのインポート)』の『adbimport コマンドが異常終了したときの対処方法』を参照してください。

   - **adbidxrebuild コマンドが中断した場合**

     adbidxrebuild コマンドを再実行してください。詳細については，マニュアル『HADB コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドが異常終了したときの対処方法』を参照してください。

   - **adbunarchivechunk コマンドが中断した場合**

     adbunarchivechunk コマンドを再実行してください。詳細については，マニュアル『HADB コマンドリファレンス』の『adbunarchivechunk（チャンクのアンアーカイブ)』の『adbunarchivechunk コマンドが異常終了したときの対処方法』を参照してください。

   再実行したコマンドが正常終了すると，表の情報とインデクスの情報との不整合が解消されて，更新不可状態が解除されます。

   コマンドの再実行時にエラーが発生した場合は，「(2) コマンドの再実行時にエラーが発生したときの対処方法」を参照してください。

(2) コマンドの再実行時にエラーが発生したときの対処方法

コマンドを再実行する際，次の原因によってエラーが発生する場合があります。該当する原因に従って，対処してください。

■コマンドの再実行時に発生するエラーの原因

- 中断したコマンドによって作成された作業用一時ファイル，またはコマンドステータスファイルにアクセスできない場合
  「対処方法 1」を参照してください。
- 実表の更新不可状態を解除しないでバージョンアップした場合
  「対処方法 1」を参照してください。
- インデクスを格納しているデータ用 DB エリアの空き容量が不足している場合
  「対処方法 2」を参照してください。

■対処方法 1

次のメッセージが，メッセージログファイルに出力されているかどうかを確認してください。

- KFAA50244-E メッセージ
- KFAA50247-E メッセージ

上記のメッセージが出力されているかどうかによって対処が異なります。上記以外のメッセージが出力されている場合は，出力されたメッセージに従って対処してください。対処が完了すると，実表の更新不可状態が解除されます。

- KFAA50244-E メッセージが出力されている場合
  再実行できなかったコマンドによって，対処が異なります。
  - adbimport コマンドまたはadbidxrebuild コマンドの再実行がエラーになった場合
    --force オプションを指定して，adbidxrebuild コマンドを実行してください。
    なお，中断したコマンドが，バックグラウンドインポートを適用した（-b オプションを指定した）adbimport コマンドのときは，adbidxrebuild コマンドに--force オプションを指定して実行すると，adbimport コマンドで処理中だった表データの格納情報が削除されます。この場合は，adbidxrebuild コマンドを実行したあとに，-b オプションを指定して，中断したadbimport コマンドを再実行してください。
    adbidxrebuild コマンドの--force オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドの指定形式』を参照してください。
  - adbunarchivechunk コマンドの再実行がエラーになった場合
    --force オプションを指定して，adbunarchivechunk コマンドを実行してください。

adbunarchivechunk コマンドの--force オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbunarchivechunk（チャンクのアンアーカイブ)』の『adbunarchivechunk コマンドの指定形式』を参照してください。

- KFAA50247-E メッセージが出力されている場合

  --create-temp-file オプションを指定して，adbidxrebuild コマンドを実行してください。

  adbidxrebuild コマンドの--create-temp-file オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドの指定形式』を参照してください。

■対処方法 2

次のメッセージが，メッセージログファイルに出力されます。

- KFAA30756-E メッセージ
- KFAA61210-E メッセージ

まずは，データ用 DB エリアを拡張したあとで，コマンドを再実行してください。データ用 DB エリアの拡張については，「11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加)」を参照してください。

データ用 DB エリアを拡張できなかった場合は，データ用 DB エリアの空き容量を確保したあとで，コマンドを再実行してください。データ用 DB エリアの空き容量の確保については，「11.10.6　データ用 DB エリアの空き容量を確保する方法」を参照してください。

データ用 DB エリアの空き容量も確保できない場合は，次の方法で実表の更新不可状態を解除してください。ただし，次の方法では，実表の更新不可状態を解除する前に，データ用 DB エリアの空き容量の確保を目的として，インデクスをいったん削除します。実表の更新不可状態を解除したあとに，インデクスを再定義して再作成します。そのため，データ用 DB エリアには，インデクスを再作成するための空き容量が必要になります。空き容量がないとインデクスを再作成できないため，インデクスを再作成するための空き容量を確保できるかどうかを事前に確認してください。そのあとで，実表の更新不可状態を解除してください。データ用 DB エリアの空き容量を確保する方法については，「11.10.6　データ用 DB エリアの空き容量を確保する方法」を参照してください。確認した結果，インデクスを再作成するための空き容量が確保できない場合は，再度データ用 DB エリアの拡張を検討してください。

手順

1. 削除するインデクスを決定する

   更新不可状態を解除する実表に定義されていて，空き容量が不足しているデータ用 DB エリアに格納されているインデクスの中から，いったん削除するインデクスを決定してください。

2. インデクスの定義情報を取得する

   手順 1.で決定したインデクスをいったん削除したあとに，同じ定義情報でインデクスを再作成できるように，必要な定義情報を取得しておいてください。

インデクスの定義情報の取得については，「付録 B.22　ディクショナリ表の検索」の「(29)　インデクスの定義情報を調べる場合」を参照してください。

3. インデクスを削除する

手順 1.で決定したインデクスをいったん削除してください。

インデクスの削除については，「11.3.9　インデクスの削除」を参照してください。

4. 前回中断したコマンドを再実行する

更新不可状態を解除する実表に対して，前回中断したコマンドを再実行してください。コマンドが正常終了した場合は，実表の更新不可状態が解除されます。

中断したコマンドを確認する方法については，「10.9.2　実表の状態や使用量を確認したい場合」の「(1)　実表が更新不可状態かどうかを確認する場合」を参照してください。

コマンドがエラーになった場合は手順 5.に進んでください。コマンドが正常終了した場合は，手順 6.に進んでください。

5. 手順 4.で実行したコマンドがエラーになった場合の対処を行う

- **データ用 DB エリアの空き容量不足でエラーになった場合**

  データ用 DB エリアの空き容量を確保するために，ほかのインデクスも削除する必要があります。ほかのインデクスに対して，手順 1.から再度実行してください。

- **データ用 DB エリアの空き容量不足以外でエラーになった場合**

  出力されたエラーメッセージを確認し，エラー要因を取り除いてください。そのあとで，再度コマンドを実行してください。

  コマンドが正常終了した場合は，実表の更新不可状態が解除されます。コマンドが正常終了した場合は，手順 6.に進んでください。

6. インデクスを再作成するためにデータ用 DB エリアの空き容量を確保する

インデクスを再作成するために，手順 3.で削除したインデクスが格納されていたデータ用 DB エリアの空き容量を確保してください。

空き容量の確保については，「11.10.6　データ用 DB エリアの空き容量を確保する方法」を参照してください。

7. インデクスを再定義する

手順 2.で取得しておいた定義情報を基に，削除したインデクスを再定義してください。

8. インデクスを再作成する

手順 7.で再定義したインデクスを再作成してください。インデクスの再作成については，次のどれかを参照してください。

- 「11.3.2　B-tree インデクスの再作成」
- 「11.3.3　テキストインデクスの再作成」
- 「11.3.5　レンジインデクスの再作成」

(3)　更新不可状態の実表を検索した場合

更新不可状態の実表は，SELECT 文で検索できます。ただし，中断したコマンドによって検索結果が異なります。

■adbimport コマンドの場合

adbimport コマンド実行時の-d オプションの指定有無によって，検索結果が異なります。

-d オプションを指定しなかった場合

adbimport コマンド実行前の実表に対して検索した場合と同じ結果が返却されます。

-d オプションを指定した場合

インデクスを使用しないSELECT 文，およびレンジインデクスを使用したSELECT 文の検索結果は，0 件になります。

また，B-tree インデクスまたはテキストインデクスを使用したSELECT 文は，エラーとなります。

■adbidxrebuild コマンドの場合

再作成が完了していない B-tree インデクス，テキストインデクスまたはレンジインデクスを使用した SELECT 文を実行すると，インデクスの未完状態によって，エラーとなります。

■adbunarchivechunk コマンドの場合

adbunarchivechunk コマンド実行前の実表に対して検索した場合と同じ結果が返却されます。

15.8.2　実表に列を追加できない場合

定義系SQL のALTER TABLE 文を使用して，実表に列を追加できない場合の対処方法について説明します。

次に示すどれかに該当する場合は，実表に列を追加できません。

- 操作対象の実表が更新不可状態の場合
- 操作対象の実表が，行を格納するセグメントが割り当てられている状態であり，かつ FIX 表の場合
- 操作対象の実表が，行を格納するセグメントが割り当てられている状態であり，かつCREATE TABLE 文を実行した際にBRANCH ALL を指定した実表の場合
- 行を格納するセグメントが割り当てられている状態の実表に対して，ALTER TABLE 文の列定義でNOT NULL を指定した場合

このうち，「実表が更新不可状態になっている場合」は，「15.8.1　実表が更新不可状態になった場合」を参照して，更新不可状態を解除したあとで，ALTER TABLE 文を再実行してください。

それ以外の場合は，次に示す対処方法で，列を追加してください。

■実表に列を追加できない場合の対処方法

1. 実表のデータをファイルに出力する

adbexport コマンドを実行して，実表のすべてのデータをファイルに出力してください。出力データファイルが作成されます。ここで作成された出力データファイルは，手順 4.および手順 5.で使用する入力データファイルとなります。

2. 実表の全行データを削除する

   TRUNCATE TABLE 文を実行して，実表のすべての行データを削除してください。

3. 実表に列を追加する

   ALTER TABLE 文を再実行して，実表に列を追加してください。

4. 入力データファイルを編集する

   手順 1.で作成した入力データファイルに対して，手順 3.で追加した列に格納するデータを追加してください

5. 実表にデータを格納する

   adbimport コマンドを実行して，手順 4.で編集した入力データファイルのデータを，実表に格納してください。

メモ

行を格納するセグメントが割り当てられている状態については，「5.3.1　B-tree インデクスを定義する場合の注意点（B-tree インデクスの未完状態）」を参照してください。

15.9　B-tree インデクスに関するトラブル

ここでは，B-tree インデクスに対して発生したトラブルの対処方法について説明します。

15.9.1　B-tree インデクスが未完状態になったときの対処方法

行を格納するセグメントが割り当てられている状態の実表に対して B-tree インデクスを定義すると，B-tree インデクスが未完状態（B-tree インデクスのデータが作成されていない状態）になります。行を格納するセグメントが割り当てられている状態については，「5.3.1　B-tree インデクスを定義する場合の注意点（B-tree インデクスの未完状態）」を参照してください。

B-tree インデクスが未完状態になると，次に示す操作がエラーになります。

■エラーになる操作

- 未完状態の B-tree インデクスを使用するSELECT 文の実行
- 未完状態の B-tree インデクスが定義された表に対するINSERT 文，UPDATE 文，およびDELETE 文の実行
- 未完状態の B-tree インデクスが定義された実表に対する-d オプションを指定しないadbimport コマンドの実行
- 未完状態の B-tree インデクスが定義された実表に対するadbmergechunk コマンドの実行
- 未完状態の B-tree インデクスが定義されたアーカイブマルチチャンク表に対するadbunarchivechunk コマンドの実行

> **メモ**
>
> TRUNCATE TABLE 文は実行できます。TRUNCATE TABLE 文で実表の全行を削除した場合，B-tree インデクスのデータも削除されるため，B-tree インデクスの未完状態も解除されます。

B-tree インデクスが未完状態になる原因には，次に示す場合があります。

- 行を格納するセグメントが割り当てられている状態の実表に対して B-tree インデクスを定義した場合
- adbimport コマンドまたはadbidxrebuild コマンドが異常終了した場合

B-tree インデクスが未完状態になった原因によって，未完状態の解除方法が異なります。

(1)　行を格納するセグメントが割り当てられている状態の実表に対して B-tree インデクスを定義した場合

1. B-tree インデクスが未完状態かどうかを確認する

   B-tree インデクスが未完状態かどうかは，adbdbstatus コマンドで確認できます。「10.9.3　B-tree インデクスの状態や使用量を確認したい場合」を参照してください。

B-tree インデクスが未完状態の場合は，手順 2.に進んでください。

2. B-tree インデクスの未完状態を解除する

   未完状態の B-tree インデクスが定義されている実表に対して，adbidxrebuild コマンドを実行してください。

   adbidxrebuild コマンドを実行して，B-tree インデクスを再作成することで，B-tree インデクスの未完状態が解除されます。

   なお，adbimport コマンドが異常終了した状態でCREATE INDEX 文を実行した場合，B-tree インデクスが未完状態になることがあります。この場合，--force オプションを指定してadbidxrebuild コマンドを実行してください。--force オプションを指定しないでadbidxrebuild コマンドを実行すると，エラーとなります。

   adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild (インデクスの再作成)』を参照してください。

(2)　adbimport コマンドまたは adbidxrebuild コマンドが異常終了した場合

1. B-tree インデクスが未完状態かどうかを確認する

   B-tree インデクスが未完状態かどうかは，adbdbstatus コマンドで確認できます。「10.9.3　B-tree インデクスの状態や使用量を確認したい場合」を参照してください。

   B-tree インデクスが未完状態の場合は，手順 2.に進んでください。

2. B-tree インデクスの未完状態を解除する

   未完状態の B-tree インデクスが定義されている実表に対して，adbimport コマンドが異常終了した場合はadbimport コマンドを，adbidxrebuild コマンドが異常終了した場合はadbidxrebuild コマンドを再実行してください。

   adbimport コマンドまたはadbidxrebuild コマンドが正常終了したときに，B-tree インデクスの未完状態が解除されます。

15.9.2　一意性制約に違反した場合の対処方法（KFAA61205-W メッセージが出力された場合）

ユニークインデクスを定義した表に対して，adbimport コマンドまたはadbidxrebuild コマンドを実行した場合，キー値の重複があり，一意性制約に違反したときでも，各コマンドの処理は続行されます。adbimport コマンドまたはadbidxrebuild コマンドの終了時，一意性制約に違反した旨を示すKFAA61205-W メッセージが出力されます。

一意性制約に違反したユニークインデクスは，一意性制約がない B-tree インデクスとして扱われます。この状態を，**一意性制約違反状態**といいます。一意性制約違反状態になると，一意性制約を満たしているときと比べて，検索効率が悪くなるおそれがあります。

(1)　一意性制約違反状態かどうかを確認する方法

KFAA61205-W メッセージを確認する以外で，ユニークインデクスが一意性制約違反状態かどうかを確認する場合は，adbdbstatus コマンドを実行します。詳細については，「10.9.3　B-tree インデクスの状態や使用量を確認したい場合」を参照してください。

(2)　一意性制約違反状態を解除する方法

ユニークインデクスの一意性制約違反状態を解除する場合，次に示す手順で対処してください。

1. 重複したキー値を検索する

   対象のユニークインデクスが定義された表に対してSELECT 文を実行して，重複したキー値を検索してください。SELECT 文の指定例を次に示します。

   ■SELECT 文の指定例

   表T1 のC1 列およびC2 列をインデクス構成列とするユニークインデクスIX1 を定義している場合に，一意性制約に違反したキー値を検索します。

```
SELECT "C1","C2" FROM "T1"
   WHERE "C1" IS NOT NULL AND "C2" IS NOT NULL GROUP BY "C1","C2"
     HAVING COUNT(*) >= 2
```

2. 重複したキー値を持つ行を削除する

   一意性制約に違反したキー値を持つ行を，DELETE 文で削除してください。

   DELETE 文については，マニュアル『HADB　SQL リファレンス』の『操作系 SQL』の『DELETE（行の削除）』を参照してください。

3. B-tree インデクスを再作成する

   adbidxrebuild コマンドを実行して，B-tree インデクスを再作成してください。adbidxrebuild コマンドが完了すると，一意性制約違反状態を解除できます。

   adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成）』を参照してください。

メモ

一意性制約違反状態のユニークインデクスが定義された表に対して，次に示す操作で表のデータをすべて削除した場合も，一意性制約違反状態は解除されます。

- adbimport コマンドに-d オプション（作成モード）を指定して実行した場合
- TRUNCATE TABLE 文を実行した場合

15.10　テキストインデクスに関するトラブル

ここでは，テキストインデクスに対して発生したトラブルの対処方法について説明します。

15.10.1　テキストインデクスが未完状態になったときの対処方法

行を格納するセグメントが割り当てられている状態の実表に対してテキストインデクスを定義すると，テキストインデクスが未完状態（テキストインデクスのデータが作成されていない状態）になります。行を格納するセグメントが割り当てられている状態については，「5.4.3　テキストインデクスを定義する場合の注意点（テキストインデクスの未完状態）」を参照してください。

テキストインデクスが未完状態になると，次に示す操作がエラーになります。

■エラーになる操作

- 未完状態のテキストインデクスを使用するSELECT 文の実行
- 未完状態のテキストインデクスが定義された表に対するINSERT 文，UPDATE 文，およびDELETE 文の実行
- 未完状態のテキストインデクスが定義された実表に対する-d オプションを指定しないadbimport コマンドの実行
- 未完状態のテキストインデクスが定義された実表に対するadbmergechunk コマンドの実行
- 未完状態のテキストインデクスが定義されたアーカイブマルチチャンク表に対するadbunarchivechunk コマンドの実行

> **メモ**
>
> TRUNCATE TABLE 文は実行できます。TRUNCATE TABLE 文で実表の全行を削除した場合，テキストインデクスのデータも削除されるため，テキストインデクスの未完状態も解除されます。

テキストインデクスが未完状態になる原因には，次に示す場合があります。

- 行を格納するセグメントが割り当てられている状態の実表に対してテキストインデクスを定義した場合
- adbimport コマンドまたはadbidxrebuild コマンドが異常終了した場合

テキストインデクスが未完状態になった原因によって，未完状態の解除方法が異なります。

(1)　行を格納するセグメントが割り当てられている状態の実表に対してテキストインデクスを定義した場合

1. テキストインデクスが未完状態かどうかを確認する

   テキストインデクスが未完状態かどうかは，adbdbstatus コマンドで確認できます。「10.9.4　テキストインデクスの状態や使用量を確認したい場合」を参照してください。

テキストインデクスが未完状態の場合は，手順 2.に進んでください。

2. テキストインデクスの未完状態を解除する

   未完状態のテキストインデクスが定義されている実表に対して，adbidxrebuild コマンドを実行してください。

   adbidxrebuild コマンドを実行して，テキストインデクスを再作成することで，テキストインデクスの未完状態が解除されます。

   なお，adbimport コマンドが異常終了した状態でCREATE INDEX 文を実行した場合，テキストインデクスが未完状態になることがあります。この場合，--force オプションを指定してadbidxrebuild コマンドを実行してください。--force オプションを指定しないでadbidxrebuild コマンドを実行すると，エラーとなります。

   adbidxrebuild コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild (インデクスの再作成)』を参照してください。

(2)　adbimport コマンドまたは adbidxrebuild コマンドが異常終了した場合

1. テキストインデクスが未完状態かどうかを確認する

   テキストインデクスが未完状態かどうかは，adbdbstatus コマンドで確認できます。「10.9.4　テキストインデクスの状態や使用量を確認したい場合」を参照してください。

   テキストインデクスが未完状態の場合は，手順 2.に進んでください。

2. テキストインデクスの未完状態を解除する

   未完状態のテキストインデクスが定義されている実表に対して，異常終了したコマンド（adbimport コマンドまたはadbidxrebuild コマンド）を再実行してください。

   再実行したコマンドが正常終了したときに，テキストインデクスの未完状態が解除されます。

15.11　レンジインデクスに関するトラブル

ここでは，レンジインデクスに対して発生したトラブルの対処方法について説明します。

15.11.1　レンジインデクスが未完状態になったときの対処方法

行を格納するセグメントが割り当てられている状態の実表に対してレンジインデクスを定義すると，レンジインデクスが未完状態（レンジインデクスのデータが作成されていない状態）になります。行を格納するセグメントが割り当てられている状態については，「5.5.4　レンジインデクスを定義する場合の注意点（レンジインデクスの未完状態）」を参照してください。

レンジインデクスが未完状態になると，次に示す操作がエラーになります。

■エラーになる操作

- 未完状態のレンジインデクスを使用するSELECT 文の実行
- 未完状態のレンジインデクスが定義された表に対するINSERT 文，およびUPDATE 文の実行
- 未完状態のレンジインデクスが定義された実表に対する-d オプションを指定しないadbimport コマンドの実行
- 未完状態のレンジインデクスが定義された実表に対するadbmergechunk コマンドの実行
- 未完状態のレンジインデクスが定義されたアーカイブマルチチャンク表に対するadbunarchivechunk コマンドの実行

> **メモ**
>
> TRUNCATE TABLE 文は実行できます。TRUNCATE TABLE 文で実表の全行を削除した場合，レンジインデクスのデータも削除されるため，レンジインデクスの未完状態も解除されます。

レンジインデクスが未完状態かどうかは，adbdbstatus コマンドで確認できます。「10.9.5　レンジインデクスの状態や使用量を確認したい場合」を参照してください。

レンジインデクスが未完状態になる原因には，次に示す場合があります。

- 行を格納するセグメントが割り当てられている状態の実表に対してレンジインデクスを定義した場合
- adbidxrebuild コマンドが異常終了した場合

レンジインデクスの未完状態を解除する場合は，未完状態のレンジインデクスが定義されている実表に対して，adbidxrebuild コマンドを実行してください。

adbidxrebuild コマンドを実行して，レンジインデクスを再作成することで，レンジインデクスの未完状態が解除されます。

なお，adbimport コマンドが異常終了した状態でCREATE INDEX 文を実行した場合，レンジインデクスが未完状態になることがあります。この場合，--force オプションを指定してadbidxrebuild コマンドを実行してください。--force オプションを指定しないでadbidxrebuild コマンドを実行すると，エラーとなります。

15.12　バックグラウンドインポート機能に関するトラブル

ここでは，KFAA51246-E メッセージが出力された場合の対策について説明します。

KFAA51246-E メッセージは，バックグラウンドインポート実行時に作成されるチャンク数が，チャンク数の最大値（CREATE TABLE 文のCHUNK に指定した値）を超える場合に出力されます。

そのため，まずは「15.12.1　バックグラウンドインポートで作成されたチャンク数を減らす方法」を参照して，作成されたチャンク数を減らしてください。

また，作成されるチャンク数の最大値を大きくしたい場合は，「15.12.2　バックグラウンドインポートで作成されるチャンク数の最大値を大きくする方法」を参照してください。

作成されたチャンク数を変更できない場合は，「15.12.3　バックグラウンドインポートで作成されたチャンク数を変更できないときの対処方法」を参照してください。

15.12.1　バックグラウンドインポートで作成されたチャンク数を減らす方法

バックグラウンドインポート機能を適用したadbimport コマンド実行時に，KFAA51246-E メッセージが出力された場合，adbmergechunk コマンドを実行してください。

adbmergechunk コマンドで，作成された複数のチャンクを 1 つのチャンクにマージすると，作成されたチャンク数を減らせます。作成されたチャンク数を減らせば，バックグラウンドインポートを実行できます。

■対処方法

「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」を参照して，adbmergechunk コマンドを実行してください。

15.12.2　バックグラウンドインポートで作成されるチャンク数の最大値を大きくする方法

バックグラウンドインポート機能を適用したadbimport コマンド実行時に，KFAA51246-E メッセージが出力された場合で，作成されたチャンク数を減らすだけでなく，作成されるチャンク数の最大値を大きくしたい場合は，ALTER TABLE 文を実行してください。

ALTER TABLE 文を実行することで，実表の定義時に指定したCHUNK の値を変更できるため，作成されるチャンク数の最大値を大きくできます。

■対処方法

「11.4.10　チャンク数の最大値を変更する方法」を参照して，ALTER TABLE 文を実行してください。

15.12.3　バックグラウンドインポートで作成されたチャンク数を変更できないときの対処方法

バックグラウンドインポート機能を適用したadbimport コマンドの実行時に，KFAA51246-E メッセージが出力された場合で，次のどちらの方法も実行できないときの対処方法を説明します。

- 「15.12.1　バックグラウンドインポートで作成されたチャンク数を減らす方法」
- 「15.12.2　バックグラウンドインポートで作成されるチャンク数の最大値を大きくする方法」

■対処方法

最初に，「11.4.5　チャンク単位にデータをエクスポートする方法」を参照して，バックグラウンドインポートで格納したデータを出力してください。

そのあとで，次のどちらかの方法を実行してください。

- **不要なチャンクを削除する**

  「11.4.6　チャンク単位にデータを削除する方法」を参照して，PURGE CHUNK 文を実行してください。不要なチャンクを削除して，作成されたチャンク数が減ることで，バックグラウンドインポートを実行できます。

- **作成モードでデータインポートする**

  adbimport コマンドに-d オプション（作成モード）を指定して，事前に出力したデータを再度インポートしてください。adbimport コマンドを作成モードで実行すると，表の既存データ（チャンク）がすべて削除されて，データがインポートされます。作成されたチャンク数が 1 つになるため，バックグラウンドインポートを実行できます。

  adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

15.13　アーカイブディレクトリに関するトラブル

ここでは，アーカイブディレクトリに対して発生したトラブルの対処方法について説明します。

15.13.1　アーカイブファイルに障害が発生したときの対処方法

アーカイブファイルに障害が発生した場合，次のメッセージが出力されます。

- KFAA41205-E
- KFAA50225-E
- KFAA51271-E
- KFAA51272-E

出力されたメッセージの対策欄に従って対処してください。メッセージの対策欄については，マニュアル『HADB　メッセージ』を参照してください。

メッセージの対策欄に従って対処しても解決しない場合，またはメッセージの対策欄に従った対処が実行できない場合は，バックアップからデータベースを回復する必要があります。「10.3.1　バックアップの取得方法」に従って取得したデータベースのフルバックアップがある場合は，「10.3.2　バックアップからの回復方法」に従って，直近に取得したフルバックアップからデータベースを回復してください。

15.13.2　アーカイブディレクトリの空き容量が不足したときの対処方法

アーカイブディレクトリの空き容量が不足したため，adbarchivechunk コマンドが異常終了した場合，アーカイブディレクトリを容量の大きな別のディスクに変更してください。手順を次に示します。

手順

1. HADB サーバを正常終了する
   adbstop コマンドを実行して HADB サーバを正常終了します。
2. アーカイブディレクトリ下のファイルをコピーする
   OS のcp コマンドを実行して，アーカイブディレクトリ下にあるすべてのファイルを，容量の大きな別のディスクにコピーします。
3. 手順 2.のディスクをアーカイブディレクトリにマウントする
   OS のmount コマンドを実行して，手順 2.のディスクをアーカイブディレクトリにマウントします。
4. HADB サーバを正常開始する
   adbstart コマンドを実行して HADB サーバを正常開始します。

15.14　同義語辞書ファイルに関するトラブル

ここでは，同義語辞書ファイルに対して発生したトラブルの対処方法について説明します。

15.14.1　同義語辞書ファイルに障害が発生したときの対処方法

同義語辞書ファイルに障害が発生した場合は，次のどれかの方法で対処してください。最初は，対処方法 1 を参照してください。

- **対処方法 1**

  同義語辞書ファイルに障害が発生した場合，次のメッセージが出力されます。

  ・KFAA30959-E

  ・KFAA34008-E

  この場合，出力されたメッセージの対策欄に記載されている対処方法に従ってください。メッセージの対策欄に記載されている対処方法に従っても問題が解決しない場合，またはその対処方法が実行できない場合は，「対処方法 2」の方法で対処してください。

- **対処方法 2**

  同義語辞書ファイルのバックアップを使用して，同義語辞書ファイルを回復してください。同義語辞書ファイルのバックアップがない場合は，「対処方法 3」の方法で対処してください。

- **対処方法 3**

  障害が発生した同義語辞書ファイルをadbsyndict コマンドで再作成してください。保管している辞書情報登録ファイルを入力情報にして，adbsyndict コマンドを実行します。

  辞書情報登録ファイルがないなど，同義語辞書ファイルを再作成できない場合は，「対処方法 4」の方法で対処してください。

- **対処方法 4**

  直近に取得したデータベースのフルバックアップから DB をリストアしてください。

> **メモ**
>
> 同義語辞書ファイルは，サーバ定義のadb_syndict_storage_path オペランドに指定したディレクトリ下に格納されています。

15.14.2　同義語辞書ファイルの格納ディレクトリの空き容量が不足したときの対処方法

同義語辞書ファイルの格納ディレクトリの空き容量不足が原因で，adbsyndict コマンドが異常終了した場合，KFAA51514-E メッセージが出力されます。KFAA51514-E メッセージに出力される操作対象の情報が

「synonym-dictionary-file」または「temporary-work-file」で，かつエラー番号が28（ENOSPC）の場合，同義語辞書ファイルの格納ディレクトリの空き容量が不足していると考えられます。

この場合，まずは対処方法 1 を実施してください。対処方法 1 が実施できない場合，または対処方法 1 を実施しても容量不足が発生する場合は，対処方法 2 を実施してください。

- 対処方法 1

  adbsyndict コマンドの実行時，複数の同義語辞書を一度に更新している場合は，同義語辞書を 1 つずつ更新してください。一度に更新する同義語辞書の数を減らすと，同義語辞書ファイルの格納ディレクトリに必要な空き容量が少なくすむため，adbsyndict コマンドを実行できることがあります。

- 対処方法 2

  容量の大きい別のディスク上に同義語辞書ファイルの格納ディレクトリを作成し直してください。手順については，「11.16.13　同義語辞書ファイルの格納ディレクトリを変更する」を参照してください。

15.14.3　同義語辞書ファイルまたは同義語辞書ファイルの格納ディレクトリを誤って削除してしまったときの対処方法

同義語辞書ファイルまたは同義語辞書ファイルの格納ディレクトリを誤って削除してしまった場合は，同義語辞書ファイルを再作成する必要があります。

> **重要**
>
> 同義語辞書ファイルのバックアップからは回復できません。

(1)　同義語辞書ファイルを誤って削除してしまったときの対処方法

同義語辞書ファイルを誤って削除してしまった場合は，その同義語辞書ファイルを再作成する必要があります。再作成する手順を次に示します。

手順

1. 保管している同義語一覧定義ファイルを用意する

   誤って削除してしまった同義語辞書ファイルに対応する同義語一覧定義ファイルが必要になります。

2. 保管している辞書情報登録ファイルを用意する

   誤って削除してしまった同義語辞書ファイルに対応する辞書情報登録ファイルが必要になります。

3. adbsyndict コマンドを実行して同義語辞書を再登録する

   このとき，KFAA51504-W メッセージが出力されますが，対処の必要はありません。

メモ

- 手順の 3.で同義語辞書の登録が完了した場合，KFAA51509-I メッセージが出力されます（登録を完了した同義語辞書の数だけメッセージが出力されます）。メッセージ中の同義語辞書名を確認し，同義語辞書の登録が完了したことを確認してください。
- 同義語一覧定義ファイルを紛失してしまった場合は，「11.16.14　同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合）」を参照して，同義語一覧定義ファイルを再作成してください。

(2)　同義語辞書ファイルの格納ディレクトリを誤って削除してしまったときの対処方法

同義語辞書ファイルの格納ディレクトリを誤って削除してしまった場合は，すべての同義語辞書ファイルを再作成する必要があります。再作成する手順を次に示します。

手順

1. 同義語辞書ファイルの格納ディレクトリを再作成する
   誤って削除してしまった同義語辞書ファイルの格納ディレクトリを再作成してください。
2. 同義語辞書ファイルの格納ディレクトリに対する権限を付与する
   付与する権限については，「11.16.1　同義語検索を行うための準備をする」の「(5)　同義語辞書ファイルの格納ディレクトリに対する権限を付与する」を参照してください。
3. 保管している同義語一覧定義ファイルを用意する
   全同義語辞書の同義語一覧定義ファイルが必要になります。
4. 保管している辞書情報登録ファイルを用意する
   全同義語辞書の辞書情報登録ファイルが必要になります。
5. adbsyndict コマンドを実行してすべての同義語辞書を再登録する
   このとき，KFAA51504-W メッセージが出力されますが，対処の必要はありません。

メモ

- 手順の 5.で同義語辞書の登録が完了した場合，KFAA51509-I メッセージが出力されます（登録を完了した同義語辞書の数だけメッセージが出力されます）。メッセージ中の同義語辞書名を確認し，同義語辞書の登録が完了したことを確認してください。
- 同義語一覧定義ファイルを紛失してしまった場合は，「11.16.14　同義語一覧定義ファイルを再作成する（同義語一覧定義ファイルを紛失した場合）」を参照して，同義語一覧定義ファイルを再作成してください。

15.15　アンロードファイルに関するトラブル

ここでは，アンロードファイルに対して発生したトラブルの対処方法について説明します。

15.15.1　アンロードファイルを格納するディスクの空き容量が不足したときの対処方法

adbreorgsystemdata コマンドの実行中に，-f オプションに指定したディレクトリにアンロードファイルが作成されます。

adbreorgsystemdata コマンドの実行中に，次に示すメッセージのどれかが出力された場合は，アンロードファイルを格納するディスクの空き容量が不足したおそれがあります。

- KFAA30959-E
- KFAA40204-E
- KFAA40205-E
- KFAA40214-E
- KFAA41205-E
- KFAA41206-I

アンロードファイルを格納するディスクの空き容量が不足したかどうかは，メッセージに出力された内容から判断できます。詳細については，「(1)　アンロードファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法」を参照してください。

また，アンロードファイルを格納するディスクの空き容量が不足した場合，次に示す原因が考えられます。該当する原因に従って，対処してください。

- ディスクに不要なアンロードファイルが残っている
  「(2)　ディスクに不要なアンロードファイルが残っている場合」を参照してください。
- アンロードファイルを格納するディスク容量が，作成されるアンロードファイルよりも小さい
  「(3)　アンロードファイルを格納するディスク容量が小さい場合」を参照してください。

(1)　アンロードファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法

アンロードファイルを格納するディスクの空き容量が不足しているかどうかを確認するには，出力されたメッセージの内容を確認してください。出力されたメッセージの内容が，次に示す表の内容に一致している場合，アンロードファイルを格納するディスクの空き容量が不足しています。

表 15-2　アンロードファイルを格納するディスクの空き容量が不足した場合に出力されるメッセージと，メッセージに出力される内容

項番	メッセージ ID	メッセージに出力される内容	
		ファイル名	エラー番号
1	KFAA30959-E	ファイル名※1	28(ENOSPC)
2	KFAA40204-E	ファイルのパス名※2	－
3	KFAA40205-E	－	28(ENOSPC)
4	KFAA40214-E	－	28(ENOSPC)
5	KFAA41205-E	ファイルのパス名※2	28(ENOSPC)
6	KFAA41206-I	ファイルのパス名※2	28(ENOSPC)

（凡例）

－：該当するメッセージに，この内容は出力されません。

注※1

メッセージに出力されたファイル名のアンロードファイルが格納されているディレクトリを確認する必要があります。ファイルが格納されているディレクトリを確認する手順を次に示します。

1. メッセージに出力されたファイル名のファイルを，OS のfind コマンドで検索してください。

   OS のfind コマンドの実行例を次に示します。

   ■OS のfind コマンドの実行例

   ```
   find ~/ -name "xxxxx"
   ```

   [説明]

   メッセージに出力されたファイル名（*xxxxx*）を指定して，find コマンドを実行します。

2. adbreorgsystemdata コマンドの実行結果で，アンロードファイルが格納されているディレクトリが，-f オプションに指定したディレクトリと一致しているかどうかを確認してください。

   一致している場合は，アンロードファイルを格納するディスクの空き容量が不足しています。

注※2

メッセージに出力されたパス名から，アンロードファイルが格納されているディレクトリを確認する必要があります。

パス名に含まれるディレクトリが，adbreorgsystemdata コマンドの-f オプションに指定したディレクトリと一致しているかどうかを確認してください。

一致している場合は，アンロードファイルを格納するディスクの空き容量が不足しています。

(2)　ディスクに不要なアンロードファイルが残っている場合

ディスクに残っているアンロードファイルは，すべて不要なファイルです。OS のrm コマンドなどを実行して，不要なアンロードファイルを削除してください。

そのあとで，adbreorgsystemdata コマンドを再実行してください。

(3)　アンロードファイルを格納するディスク容量が小さい場合

アンロードファイルを格納するディスク容量が小さい場合は，アンロードファイルの格納先を，容量が大きい別のディスクに変更する必要があります。次に示す手順で対処してください。

手順

1. アンロードファイルの容量を再度見積もる

   adbreorgsystemdata コマンドを実行したときに作成されるアンロードファイルの容量を見積もってください。アンロードファイルの容量見積もりについては，「6.20　アンロードファイルの容量見積もり」を参照してください。

   そのあとで，見積もった値よりも容量が大きいディスクにアンロードファイルを格納します。

   adbreorgsystemdata コマンドの-f オプションに指定したディレクトリを変更する場合は，手順 2.に進んでください。-f オプションに指定したディレクトリを変更しない場合は，手順 3.に進んでください。

2. -f オプションに指定したディレクトリを変更する

   adbreorgsystemdata コマンドの-f オプションに指定したディレクトリパスファイル内の「アンロードファイルの格納先を指定したディレクトリ」を，容量が大きい別のディスクのディレクトリに変更してください。

   ディレクトリパスファイルに指定したディレクトリを変更する場合について，変更前および変更後のディレクトリパスファイルの指定例を次に示します。

   ■ディレクトリパスファイルの指定例（変更前）

   ```
   /mnt/disk1/xxxxx
   ```

   [説明]

   変更前のディレクトリパスファイルでは，容量不足のディスク（'/mnt/disk1'）のディレクトリを，アンロードファイルの格納先として指定しています。

   ■ディレクトリパスファイルの指定例（変更後）

   ```
   /mnt/largedisk1/xxxxx
   ```

   [説明]

   変更前に'/mnt/disk1'を指定していた個所に，容量が大きい別のディスク（'/mnt/largedisk1'）を指定して，アンロードファイルの格納先を変更します。

   そのあとで，adbreorgsystemdata コマンドを再実行してください。

3. -f オプションに指定したディレクトリに対して，容量が大きい別のディスクのディレクトリをリンク先とするシンボリックリンクを作成する

   adbreorgsystemdata コマンドの-f オプションに指定したディレクトリパスファイル内の「アンロードファイルの格納先を指定したディレクトリ」で，OS の ln コマンドを実行します。ln コマンドで，容量が大きい別のディスクのディレクトリをリンク先に設定したシンボリックリンクを作成してください。

このとき，シンボリックリンクのパス名として，-f オプションに指定したディレクトリパスファイル内の「アンロードファイルの格納先を指定したディレクトリ」を指定します。シンボリックリンクを作成する際の，OS の ln コマンドの実行例を次に示します。

■OS の ln コマンドの実行例

```
ln  -s /mnt/largedisk1/xxxxx  /mnt/disk1/xxxxx
```

[説明]

容量が大きい別のディスクのディレクトリ（'/mnt/largedisk1/xxxxx'）をリンク先に設定して，-f オプションに指定したディレクトリ（'/mnt/disk1/xxxxx'）と同名のシンボリックリンク（'/mnt/disk1/xxxxx'）を作成します。

そのあとで，adbreorgsystemdata コマンドを再実行してください。

15.16　メッセージログファイルに関するトラブル

ここでは，メッセージログファイルに対して発生したトラブルの対処方法について説明します。

15.16.1　メッセージログファイルの縮退状態を解除する方法

メッセージログファイルの縮退状態を解除する方法について説明します。メッセージログファイルの縮退状態については，「10.4.4　メッセージログファイルの縮退状態」を参照してください。

> **メモ**
>
> メッセージログファイルを格納しているディスクの空き容量が，「環境変数ADBMSGLOGSIZE の指定値（単位：メガバイト）×2」以上になったときに SQL 文またはコマンドが実行されると，メッセージログファイルの縮退状態は自動的に解除されます。

■HADB サーバのメッセージログファイルの縮退状態を解除する方法

対処手順を次に示します。

手順

1. 次のコマンドを実行して，メッセージログファイルを格納しているディスクの空き容量を確認してください。

```
df $ADBDIR/spool
```

2. 不要なファイルを削除するなどして，ディスクの空き容量を確保してください。次に示す以上の空き容量が必要になります。

```
環境変数ADBMSGLOGSIZEの指定値（単位：メガバイト）×2
```

■HADB クライアントのメッセージログファイルの縮退状態を解除する方法

対処手順を次に示します。

手順（Windows 版の HADB クライアントの場合）

1. コマンドプロンプトで次のコマンドを実行して，メッセージログファイルを格納しているドライブを確認してください。

```
echo %ADBCLTDIR%
```

2. メッセージログファイルを格納しているドライブの空き容量を確認してください。
3. 不要なファイルを削除するなどして，ドライブの空き容量を確保してください。次に示す以上の空き容量が必要になります。

```
環境変数ADBMSGLOGSIZEの指定値（単位：メガバイト）×2
```

手順（Linux 版の HADB クライアントの場合）

1. 次のコマンドを実行して，メッセージログファイルを格納しているディスクの空き容量を確認してください。

```
df $ADBCLTDIR/spool
```

2. 不要なファイルを削除するなどして，ディスクの空き容量を確保してください。次に示す以上の空き容量が必要になります。

```
環境変数ADBMSGLOGSIZEの指定値（単位：メガバイト）×2
```

15.17　HADB サーバの再開始に関するトラブル

ここでは，HADB サーバの再開始で発生したトラブルの対処方法について説明します。

15.17.1　HADB サーバの再開始に掛かる処理時間が長い場合の対処方法

HADB サーバの再開始に掛かる処理時間が長い場合に実施することについて説明します。

■HADB サーバの再開始中に実施すること

adbstart コマンドを実行したあとで，HADB サーバの再開始に想定以上の時間が掛かっている場合は，KFAA81215-I メッセージが出力されているかどうかを確認してください。KFAA81215-I メッセージから，データベースの回復処理の進捗状況を確認できます。

KFAA81215-I メッセージから，データベースの回復処理が完了して，後続の処理を再開できる時間を見積もってください。

なお，KFAA81215-I メッセージは，サーバ定義のadb_log_rec_msg_interval オペランドの指定値に従って，一定の間隔で出力されます。

■HADB サーバの再開始が完了したあとに実施すること

HADB サーバの再開始が完了したあとで，KFAA81211-I メッセージが出力されているかどうかを確認してください。KFAA81211-I メッセージから，トランザクションの回復処理の内容を確認できます。

KFAA81211-I メッセージの出力内容から，コミット回数を増やすなど，1 トランザクションの更新量を減らすことを検討してください。

なお，KFAA81211-I メッセージは，サーバ定義のadb_log_rec_msg_interval オペランドの指定値よりも，回復処理に時間が掛かったトランザクションが存在する場合に出力されます。

また，KFAA81211-I メッセージに出力されたトランザクション ID から，対象のトランザクションで実行された SQL 文を特定できます。トランザクション ID をキーにして，SQL トレース情報の「SQL 文の実行情報」に出力されたtran_id の値を確認してください。トランザクション ID とtran_id が一致する SQL 文が，対象のトランザクションで実行された SQL 文です。SQL 文の実行情報については，「10.11.2　SQL トレース情報に出力される情報」の「(2)　SQL 文の実行情報」を参照してください。

サーバ定義のadb_log_rec_msg_interval オペランドについては，「7.2.3　システムログに関するオペランド（set 形式）」の adb_log_rec_msg_interval オペランドを参照してください。

16 マルチノード機能の運用

この章では，マルチノード機能を使用したシステムの構築方法，およびマルチノード機能の運用方法について説明します。

マルチノード機能の概要を知りたい場合は，「2.19　マルチノード機能」を参照してください。

16.1　この章を読む際の留意事項

マルチノード機能の説明は，HA モニタについての知識があることを前提としています。HA モニタについては，マニュアル『HA モニタ Linux(R)(x86)編』を参照してください。

また，マニュアル『HA モニタ Linux(R)(x86)編』を参照する際，次の点に留意してください。

- プライマリノードおよびセカンダリノードの HADB サーバは，モニタモードで動作します。
- エイリアス IP アドレスを引き継ぐ構成になります。
- プライマリノードおよびセカンダリノードの HADB サーバは，HA モニタとのインタフェースを持たないプログラムとして動作します。
- サーバ障害の検出方法は，API を発行しない方法になります。
- HA モニタ Extension は使用できません。
- HA モニタのマニュアルで使用されている用語と，HADB のマニュアルで使用されている用語の対応を次の表に示します。

表 16-1　HA モニタのマニュアルで使用されている用語と，HADB のマニュアルで使用されている用語の対応

項番	HA モニタのマニュアルで使用されている用語	HADB のマニュアルで使用されている用語
1	系	ノード
2	実行系	プライマリノード
3	待機系	セカンダリノード
4	実行サーバ	プライマリノードの HADB サーバ
5	待機サーバ	セカンダリノードの HADB サーバ
6	系障害	ノード障害
7	サーバ障害	

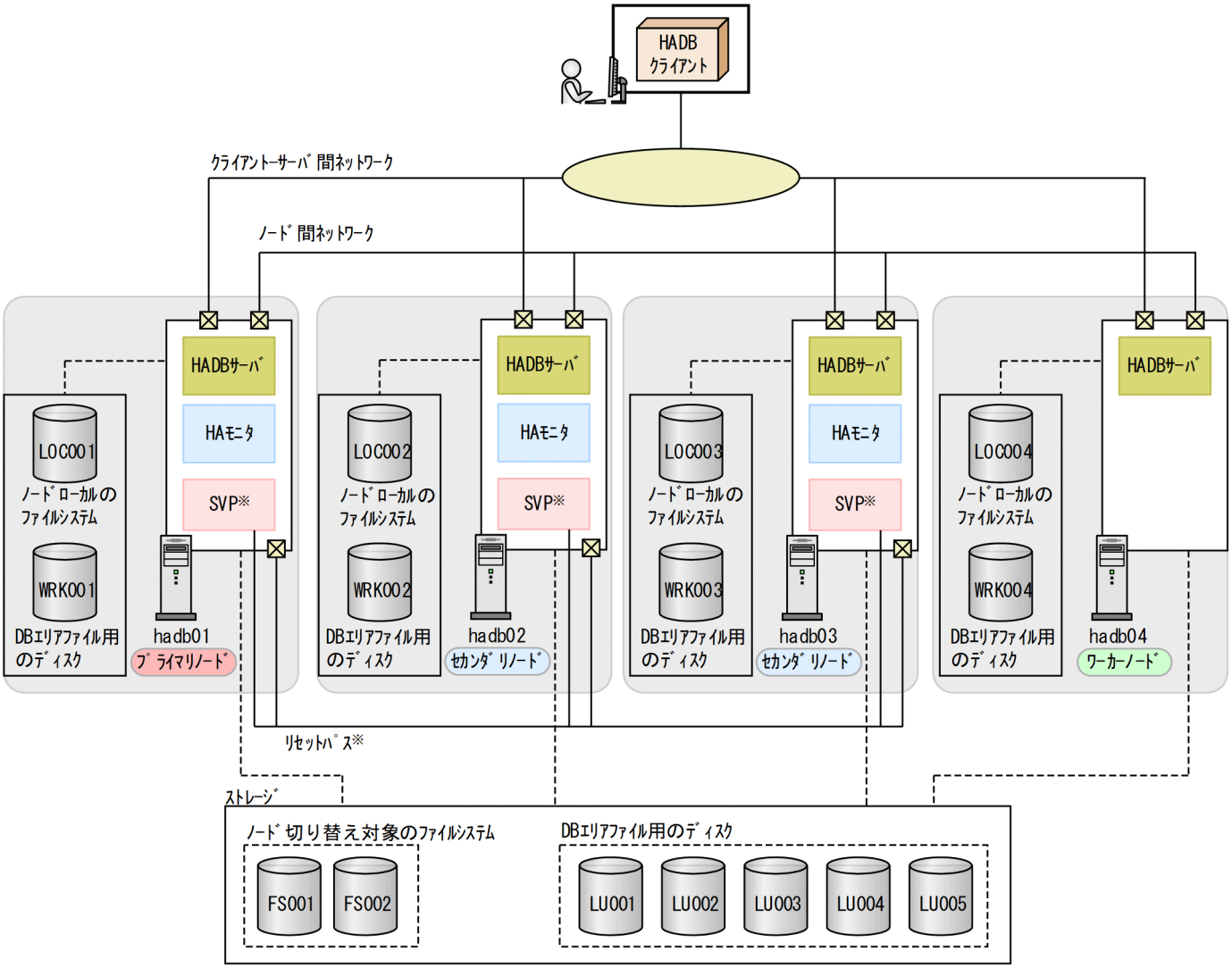
メモ

ワーカーノードは，HA モニタの管理対象外のため，上記の表中に記載がありません。

16.2　マルチノード機能を使用したシステム構成例

マルチノード機能を使用した場合のシステム構成例を次の図に示します。

図 16-1　マルチノード機能を使用した場合のシステム構成例



注※

共有ディスクの SCSI リザーブを使用する場合は，SVP およびリセットパスは必要ありません。

重要

HA モニタによる共有ディスクのデータ保護方法には，次のどちらかを選択できますが，系のリセットを使用することを推奨します。

- 系のリセット
- 共有ディスクの SCSI リザーブ

16.2.1　前提ソフトウェア

マルチノード機能を使用する場合，プライマリノードおよびセカンダリノードには HA モニタが必要になります。

HADB では，HA モニタのモニタモードの機能を使用し，プライマリノードおよびセカンダリノードの障害を監視しています。プライマリノードで障害が発生した場合，HA モニタの機能を使用して，セカンダリノードの 1 つをプライマリノードに切り替えて処理を続行します。セカンダリノードで障害が発生した場合，HA モニタの機能を使用して，セカンダリノードをマルチノード構成から切り離し，残りのノードで処理を続行します。

障害発生時のプライマリノードまたはセカンダリノードの切り離し処理については，「16.15　プライマリノードまたはセカンダリノードでノード障害が発生した場合の運用（ノードの復帰）」を参照してください。

なお，使用している Linux のバージョンによって，HA モニタの前提バージョンが次のように異なります。

- RHEL 6 の場合：01-56-01 以降
- RHEL 7 の場合：01-61 以降
- RHEL 8 の場合：01-71 以降

重要

- プライマリノードとセカンダリノードにインストールする HA モニタのバージョンは同じにしてください。
- ワーカーノードには HA モニタは必要ありません。

16.2.2　サーバ構成

「図 16-1　マルチノード機能を使用した場合のシステム構成例」で示す構成例の場合，4 つのノード（hadb01，hadb02，hadb03，およびhadb04）から構成されています。プライマリノードはhadb01，セカンダリノードはhadb02 およびhadb03，ワーカーノードはhadb04 になります。

マルチノード機能を使用する場合，各ノードのサーバマシンの性能（CPU，メモリサイズなど）は同一である必要はありません。ただし，サーバマシンの性能が異なる場合，接続するノードによって SQL 文の処理性能に差異が発生するおそれがあります。そのため，できる限り全ノードのサーバマシンの性能を同じにすることを推奨します。

また，マルチノード機能では，HA モニタの系切り替え機能を使用します。系を切り替える際の共有ディスクのデータ保護方法には，次のどちらかを選択してください。推奨する方法は，系のリセットです。

- 系のリセット
- 共有ディスクの SCSI リザーブ

プライマリノードおよびセカンダリノードのサーバマシンに障害管理プロセサ（「図 16-1　マルチノード機能を使用した場合のシステム構成例」の SVP）がある場合は，推奨する系のリセットを使用してください。

系のリセット，障害管理プロセサ，共有ディスクの SCSI リザーブについては，マニュアル『HA モニタ Linux(R)(x86)編』の次の個所を参照してください。

- 系のリセット：『系のリセット』
- 障害管理プロセサ：『必要なハードウェア』
- 共有ディスクの SCSI リザーブ：『共有ディスクの SCSI リザーブ』

メモ

ワーカーノードには HA モニタをインストールしないため，ワーカーノードの HADB サーバは系切り替えの対象にはなりません。

16.2.3　ネットワーク構成

マルチノード機能を使用する場合，次に示す 3 つのネットワークを使用します。各ネットワークは物理的に分けてください。

- クライアントーサーバ間ネットワーク
- ノード間ネットワーク
- リセットパス

(1)　クライアントーサーバ間ネットワーク

HADB クライアントと HADB サーバ間の通信で使用するネットワークです。

HADB クライアントは，エイリアス IP アドレスを使用して HADB サーバに接続します。そのため，マニュアル『HA モニタ Linux(R)(x86)編』の『LAN の引き継ぎ』を参照して，エイリアス IP アドレスを設定してください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」での IP アドレス，およびポート番号の設定例を次の表に示します。

表 16-2　クライアントーサーバ間ネットワークの IP アドレスとポート番号の設定例

項番	設定対象	IP アドレス	ポート番号
1	クライアントマシン	10.196.108.111	設定不要
2	サーバマシンhadb01（プライマリノード）	10.196.108.11	23650

項番	設定対象	IP アドレス	ポート番号
3	サーバマシンhadb02（セカンダリノード）	10.196.108.12	23650
4	サーバマシンhadb03（セカンダリノード）	10.196.108.13	23650
5	サーバマシンhadb04（ワーカーノード）	10.196.108.14	23650
6	エイリアス IP アドレス	10.196.108.143	23650

(2) ノード間ネットワーク

HADB サーバ間の通信，および HA モニタの監視パスとして使用するネットワークです。

ネットワーク構築する際，Linux の Bonding 機能を使用した NIC 冗長化を推奨します。また，Bonding 機能を使用する際には，複数の LAN カードを用意してください。

Bonding 機能の詳細については，OS のマニュアルを参照してください。HA モニタの監視パスの設定については，マニュアル『HA モニタ Linux(R)(x86)編』の『監視パスの設定』を参照してください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」での IP アドレス，およびポート番号の設定例を次の表に示します。

表 16-3　ノード間ネットワークの IP アドレスとポート番号の設定例

項番	設定対象	IP アドレス	ポート番号
1	サーバマシンhadb01（プライマリノード）の HADB サーバ間通信	172.16.0.11	23651
2	サーバマシンhadb01（プライマリノード）の HA モニタの監視パス	172.16.0.11	7777
3	サーバマシンhadb02（セカンダリノード）の HADB サーバ間通信	172.16.0.12	23651
4	サーバマシンhadb02（セカンダリノード）の HA モニタの監視パス	172.16.0.12	7777
5	サーバマシンhadb03（セカンダリノード）の HADB サーバ間通信	172.16.0.13	23651
6	サーバマシンhadb03（セカンダリノード）の HA モニタの監視パス	172.16.0.13	7777
7	サーバマシンhadb04（ワーカーノード※）の HADB サーバ間通信	172.16.0.14	23651

注※

ワーカーノードは HA モニタの管理対象外のため，HA モニタの監視パスの設定は不要です。

通常は，HADB サーバ間の通信で使用するネットワークと，HA モニタの監視パスとして使用するネットワークを分けます。ただし，この場合，HA モニタは，HADB サーバ間の通信で使用するネットワークの

障害を検知することができません。ここで説明しているとおり，HADB サーバ間の通信で使用するネットワークと，HA モニタの監視パスとして使用するネットワークを同じにすると，ネットワーク障害を HA モニタが検知することができます。そのため，ここで説明しているとおりにノード間ネットワークを構成することを推奨します。

ただし，対象のネットワークが単一障害点となってしまうため，必ず Linux のカーネル拡張パッケージである bonding パッケージなどを使用して LAN アダプタを冗長化し，ネットワークの耐障害性を向上させてください。

(3) リセットパス

系のリセットを使用する場合は，リセットパスが必要になります。共有ディスクの SCSI リザーブを使用する場合は，リセットパスは必要ありません。

リセットパスとは，HA モニタが障害を検知した際，障害が発生したノードの入出力をリセットするために使用するネットワークです。このネットワークは，リセットパス専用のネットワークである必要があります。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『リセットパスの構成』を参照してください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」での IP アドレス，およびポート番号の設定例を次の表に示します。

表 16-4　リセットパスの IP アドレスとポート番号の設定例

項番	設定対象	IP アドレス	ポート番号
1	サーバマシンhadb01（プライマリノード）のリセットパス	192.168.0.11	11111
2	サーバマシンhadb01（プライマリノード）のSVP	192.168.0.21	22222
3	サーバマシンhadb02（セカンダリノード）のリセットパス	192.168.0.12	11111
4	サーバマシンhadb02（セカンダリノード）のSVP	192.168.0.22	22222
5	サーバマシンhadb03（セカンダリノード）のリセットパス	192.168.0.13	11111
6	サーバマシンhadb03（セカンダリノード）のSVP	192.168.0.23	22222

注

ワーカーノードは HA モニタの管理対象外のため，リセットパスの設定は不要です。

HA モニタによる共有ディスクのデータ保護方法には，系のリセットの使用を推奨しています。この場合，リセットパスが必要になります。また，HADB サーバが稼働するサーバマシンに SVP などの障害管理プ

ロセサが搭載されている必要があります。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『必要なハードウェア』を参照してください。

16.2.4　ストレージ構成

マルチノード機能を使用する場合，次のファイルシステムおよびディスクを準備してください。

- ノードローカルのファイルシステム
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，LOC001～LOC004 が該当します。
- ノード切り替え対象のファイルシステム
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，FS001 およびFS002 が該当します。
- DB エリアファイル用のディスク
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，LU001～LU005 およびWRK001～WRK004 が該当します。

なお，一部のファイルシステムおよびディスクでは，各ノードでディスクを識別するためのデバイス名が必要になります。サーバとストレージ間がシングルパス構成の場合は，WWN の使用を推奨します。サーバとストレージのパスを冗長化する場合は，冗長化の方法によってデバイス名が異なります。「図 16-1 マルチノード機能を使用した場合のシステム構成例」の場合の，デバイス名の例を次に示します。

- シングルパス構成の場合のデバイス名の例

  FS001：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b53e0a

  FS002：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b6c44d

  LU001：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b828e9

  LU002：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b7d9fd

  LU003：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b87793

  LU004：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b8c6d3

  LU005：/dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b9160f

- マルチパスソフトウェア（DMMP）による冗長化構成の場合のデバイス名の例

  FS001：/dev/mapper/mpath1

  FS002：/dev/mapper/mpath2

  LU001：/dev/mapper/mpath11

  LU002：/dev/mapper/mpath12

  LU003：/dev/mapper/mpath13

  LU004：/dev/mapper/mpath14

  LU005：/dev/mapper/mpath15

(1) ノードローカルのファイルシステム

ノードローカルのファイルシステムには，次のディレクトリを配置します。

- サーバディレクトリ
  サーバディレクトリの作成には，adbinstall コマンドを使用します。
- DB ディレクトリ
  DB ディレクトリの作成については，「16.3.8　データベースの作成」を参照してください。
- マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ
  マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリについては，「16.26.1　同義語検索を行うための準備をする」の「(6)　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する」を参照してください。
- 監査証跡の出力先ディレクトリ
  監査証跡の出力先ディレクトリについては，「2.18.5　監査証跡の出力先（監査証跡ファイル）」を参照してください。
- 統一フォーマット用監査証跡の出力先ディレクトリ
  統一フォーマット用監査証跡の出力先ディレクトリについては，「2.18.9　監査証跡の変換（JP1/Audit Management - Manager との連携）」を参照してください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，LOC001～LOC004 がノードローカルのファイルシステムになります。

> **重要**
>
> ワーカーノードの場合，システムディレクトリもノードローカルのファイルシステムに配置してください。

- LOC001
  サーバマシンhadb01（プライマリノード）のローカルのファイルシステム
- LOC002
  サーバマシンhadb02（セカンダリノード）のローカルのファイルシステム
- LOC003
  サーバマシンhadb03（セカンダリノード）のローカルのファイルシステム
- LOC004
  サーバマシンhadb04（ワーカーノード）のローカルのファイルシステム

(2) ノード切り替え対象のファイルシステム

次のファイルシステムをノード切り替え対象のファイルシステムにしてください。

- DB ディレクトリ下のADBSYS の内容を格納するシステムディレクトリ用のファイルシステム
- 同義語辞書ファイルを格納するファイルシステム

このノード切り替え対象のファイルシステムは，プライマリノードおよびセカンダリノードから参照できるディスク（共有ディスク）から構成される VG 内の LV 上に作成してください。共有ディスクについては，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの構成』を参照してください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，FS001 とFS002 が，ノード切り替え対象のファイルシステムになります。

- FS001

  システムディレクトリ用のファイルシステム

  ここでは，VG の名称はvg_hadb，LV の名称はhadb_sys とします。

- FS002

  同義語辞書ファイルを格納するファイルシステム

  ここでは，VG の名称はvg_hadb02，LV の名称はhadb_syndict とします。 なお，同義語検索を行わない場合は，同義語辞書ファイルを格納するファイルシステムは必要ありません。

重要

共有ディスクの SCSI リザーブを使用する場合，ノード切り替え対象のファイルシステムを作成したディスクは，SCSI リザーブの対象としてください。また，SCSI リザーブの対象としたディスクには，ノード切り替え対象のファイルシステム以外の領域を作成しないでください。

(3)　DB エリアファイル用のディスク

DB エリアファイルごとにディスクを用意します。

次の DB エリアファイルには，ブロックスペシャルファイルを割り当ててください。また，ブロックスペシャルファイルに対応するディスクは，全ノードから参照可能である必要があります。

- マスタディレクトリ用 DB エリアファイル
- ディクショナリ用 DB エリアファイル
- システム表用 DB エリアファイル
- データ用 DB エリアファイル

重要

共有ディスクの SCSI リザーブを使用する場合，上記の DB エリアファイル用のディスクは，SCSI リザーブの対象としてください。

また，次の DB エリアファイルが使用するディスクは，ノードごとに準備してください。

- 作業表用 DB エリアファイル

重要

- ノードごとに準備が必要なディスク（作業表用 DB エリアファイルを格納するディスク）を複数のノードで共有して使用した場合，HADB サーバはその動作を保証しません。不正な検索結果が表示されたり，HADB サーバが異常終了したりするおそれがあります。
- 共有ディスクの SCSI リザーブを使用する場合，作業表用 DB エリアファイル用のディスクは，SCSI リザーブの対象としないでください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，LU001～LU005 および WRK001～WRK004 が DB エリアファイル用のディスクになります。

- LU001

  ディクショナリ用 DB エリアを構成する DB エリアディスク
- LU002

  マスタディレクトリ用 DB エリアを構成する DB エリアディスク
- LU003

  システム表用 DB エリアを構成する DB エリアディスク
- LU004

  データ用 DB エリア（ADBUTBL01）を構成する DB エリアディスク
- LU005

  データ用 DB エリア（ADBUIDX01）を構成する DB エリアディスク
- WRK001

  サーバマシンhadb01（プライマリノード）の HADB サーバの作業表用 DB エリアを構成する DB エリアディスク

  ノードローカルのディスクになります。対応するブロックスペシャルファイル名は，/dev/mapper/WRK001 とします。
- WRK002

  サーバマシンhadb02（セカンダリノード）の HADB サーバの作業表用 DB エリアを構成する DB エリアディスク

  ノードローカルのディスクになります。対応するブロックスペシャルファイル名は，/dev/mapper/WRK002 とします。
- WRK003

  サーバマシンhadb03（セカンダリノード）の HADB サーバの作業表用 DB エリアを構成する DB エリアディスク

  ノードローカルのディスクになります。対応するブロックスペシャルファイル名は，/dev/mapper/WRK003 とします。

- WRK004

  サーバマシンhadb04（ワーカーノード）の HADB サーバの作業表用 DB エリアを構成する DB エリアディスク

  ノードローカルのディスクになります。対応するブロックスペシャルファイル名は，/dev/mapper/WRK004 とします。

16.3　マルチノード機能を使用したシステムの構築

マルチノード機能を使用したシステムの構築手順について説明します。

この節の説明は，HADB の新規導入を前提としています。すでに HADB を導入している場合で，マルチノード機能を使用したい場合は，「16.21　マルチノード機能を使用するシステムへの移行」を参照してください。

16.3.1　HA モニタのインストール

プライマリノードおよびセカンダリノードのサーバマシンに HA モニタをインストールしてください。HA モニタのインストール方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの構築』を参照してください。

> **メモ**
>
> ワーカーノードは HA モニタの管理対象外のため，ワーカーノードのサーバマシンに HA モニタをインストールする必要はありません。

16.3.2　HADB サーバのインストールおよび環境設定

全ノードのサーバマシンに HADB サーバをインストールし，環境設定を行ってください。HADB サーバのインストール，および環境設定については，「8.　システム構築」を参照してください。

HADB サーバのインストールおよび環境設定時の注意事項を次に示します。

- HADB サーバのバージョンおよびリビジョンは，全ノードで同じにしてください。
- 全ノードで，通信情報ファイルを格納するディレクトリ，または設定ファイルを作成する必要があります。通信情報ファイルを格納するディレクトリ，または設定ファイルの作成方法については，「8.2.1 インストール前に実施する作業」の「(7)　通信情報ファイルを格納するディレクトリの作成」を参照してください。
- 環境変数の設定については，「8.4　環境変数の設定」を参照してください。ただし，次の環境変数については，全ノードで指定値を同じにしてください。
  - ADBLANG
  - TZ

メモ

HADB サーバのサーバディレクトリ（インストール時）のパス名，およびサーバディレクトリ（運用時）のパス名は，全ノードで同じにする必要はありません。

16.3.3 HADB クライアントのインストールおよび環境設定

HADB クライアントのインストール，および環境設定については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

16.3.4 HA モニタの環境設定

プライマリノードおよびセカンダリノードのサーバマシンにインストールした HA モニタの環境設定をしてください。HA モニタの環境設定方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの構築』を参照してください。

ここでは，HA モニタの環境設定をする際に留意することについて説明します。

(1) 環境変数の設定

HADB 管理者の環境変数に，次の環境変数を設定してください。

- PATH

  HA モニタのコマンドを実行するために，次のパスを追加してください。

  /opt/hitachi/HAmon/bin

(2) sysdef ファイルの指定

HA モニタのsysdef ファイルに指定するオペランドのうち，次のオペランドについては，指定値が決まっています。

- address

  ノードのリセット優先度を指定します。指定値が小さいほど，そのノードのリセット優先度が高くなります。各ノードで異なる値を次のように指定してください。

  - プライマリノードは，リセット優先度をいちばん高くしたいため，プライマリノードの指定値をいちばん小さくしてください。
  - セカンダリノードは，セカンダリノードごとに異なる値を指定してください。リセット優先度が高いセカンダリノードほど指定値を小さくしてください。プライマリノードの切り替えの際に優先順位が高いセカンダリノードほど，リセット優先度が高いセカンダリノードにしてください。

- monbegin_restart

  nouse を指定します。

- multistandby

  マルチノード機能は，HA モニタのマルチスタンバイ機能を使用するため，use を指定します。

- usrcommand

  HADB 用の設定を記述したユーザコマンドの絶対パスを指定します。

  ユーザコマンドはひな形を基に作成してください。作成方法については，「(4)　コマンド用環境変数定義およびコマンドの作成」の「(b)　ユーザコマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『ユーザコマンドの作成』を参照してください。

- termcmd_at_abort

  nouse を指定します。

- fence_reset

  系のリセットを使用する場合は，このオペランドを省略するか，またはデフォルト値のuse を指定してください。

  共有ディスクの SCSI リザーブを使用する場合は，nouse を指定してください。

- fence_scsi

  系のリセットを使用する場合は，このオペランドを省略するか，またはデフォルト値のnouse を指定してください。

  共有ディスクの SCSI リザーブを使用する場合は，use を指定してください。

- fence_lan

  このオペランドを省略するか，またはデフォルト値のnouse を指定してください。

上記のオペランドの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『HA モニタの環境設定（sysdef)』を参照してください。

(3)　servers ファイルの指定

HA モニタのservers ファイルに指定するオペランドのうち，次のオペランドについては，指定値が決まっています。

- name

  サーバディレクトリの絶対パスを指定します。

  ただし，name オペランドに指定するパス名として，空白を含む文字列は指定できません。そのため，マルチノード機能を使用する場合，パス名に空白を含まないディレクトリをサーバディレクトリとすることを推奨します。

- acttype

  マルチノード機能は，HA モニタのモニタモードで動作するため，monitor を指定します。

- termcommand

HADB 用の設定を記述したサーバの停止コマンドの絶対パスを指定します。

サーバの停止コマンドはひな形を基に作成してください。作成方法については，「(4)　コマンド用環境変数定義およびコマンドの作成」の「(d)　サーバの停止コマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバの停止コマンドの作成』を参照してください。

- initial

  プライマリノードの場合はonline を，セカンダリノードの場合はstandby を指定します。

- disk

  システムディレクトリのファイルシステム，および同義語辞書ファイルを格納するファイルシステムを含む VG の絶対パスを指定します。「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合は，/dev/vg_hadb および/dev/vg_hadb02 を指定します。

  同義語検索を行わない場合は，同義語辞書ファイルを格納するファイルシステムを含む VG の絶対パスを指定する必要はありません。

- fs_name

  システムディレクトリのファイルシステム，および同義語辞書ファイルを格納するファイルシステムに対応する論理ボリュームの絶対パスを指定します。「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合は，/dev/vg_hadb/sys_hadb および/dev/vg_hadb02/hadb_syndict を指定します。

  同義語検索を行わない場合は，同義語辞書ファイルを格納するファイルシステムに対応する論理ボリュームの絶対パスを指定する必要はありません。

- fs_mount_dir

  システムディレクトリのパス（$DBDIR/ADBSYS），および同義語辞書ファイルを格納するファイルシステムをマウントするディレクトリのパスを指定します。

  同義語辞書ファイルを格納するファイルシステムをマウントするディレクトリのパスは，プライマリノードおよびセカンダリノードで同じパスを指定してください。

  同義語検索を行わない場合は，同義語辞書ファイルを格納するファイルシステムをマウントするディレクトリのパスを指定する必要はありません。

- fs_mount_opt

  システムディレクトリのファイルシステム，および同義語辞書ファイルを格納するファイルシステムをマウントするためのマウントオプションを指定します。

  同義語検索を行わない場合は，同義語辞書ファイルを格納するファイルシステムをマウントするためのマウントオプションを指定する必要はありません。

- lan_updown

  LAN の状態設定ファイルを使用するため，use を指定します。

  LAN の状態設定ファイルには，エイリアス IP アドレスの設定をしてください。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『LAN の状態設定ファイルの設定』を参照してください。

- patrolcommand

  HADB 用の設定を記述したサーバの監視コマンドの絶対パスを指定します。

サーバの監視コマンドはひな形を基に作成してください。作成方法については，「(4)　コマンド用環境変数定義およびコマンドの作成」の「(e)　サーバの監視コマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバの監視コマンドの作成』を参照してください。

- servexec_retry

  0 を指定します。

- waitserv_exec

  yes を指定します。

- ip_neck

  use を指定します。

- uoc_neck

  nouse を指定します。

- vg_neck

  use を指定します。

  同義語検索を行う場合は，同義語辞書ファイルを格納するファイルシステムについてもuse を指定してください。

- fs_neck

  use を指定します。

  同義語検索を行う場合は，同義語辞書ファイルを格納するファイルシステムについてもuse を指定してください。

- actcommand

  HADB 用の設定を記述したサーバの起動コマンドの絶対パスを指定します。

  サーバの起動コマンドはひな形を基に作成してください。作成方法については，「(4)　コマンド用環境変数定義およびコマンドの作成」の「(c)　サーバの起動コマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバの起動コマンドの作成』を参照してください。

- standbypri

  プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。指定値が小さいほど，優先順位が高くなります。優先順位が高いノードほど指定値を小さくしてください。

  各セカンダリノードで異なる値を指定してください。

  セカンダリノードの HA モニタ（initial オペランドにstandby を指定した場合）で，このオペランドを指定します。

- scsi_device

  このオペランドには，SCSI リザーブの対象となるデバイス名を絶対パスで指定します。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクが次のどれかの条件を満たすときに，このオペランドを指定してください。

- シングルパス構成の場合
- VMware ESXi での仮想化環境の場合（DMMP を使用する場合を除く）
- マルチパスソフトウェア（HDLM）による冗長化構成の場合

このオペランドを指定する場合は，次のファイルシステムとディスクのデバイス名※を絶対パスで指定します。

- ノード切り替え対象のファイルシステム
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のFS001，FS002 が該当します。
- DB エリアファイル用のディスク（作業表用 DB エリアのディスクを除く）
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のLU001～LU005 が該当します。

なお，同義語検索を使用しない場合は，同義語辞書ファイルを格納するファイルシステムのデバイス名を指定する必要はありません。

注※

シングルパス構成，および VMware ESXi での仮想化環境（DMMP を使用する場合を除く）の場合，このオペランドには，「16.2.4　ストレージ構成」の「シングルパス構成の場合のデバイス名の例」に記載されているwwn-から始まるパスではなく，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの SCSI リザーブに必要なオペランドの指定値の求め方』の例のようにscsi-から始まるパスを指定してください。

重要

次のファイルシステムとディスクのデバイス名は，このオペランドに指定しないでください。

- ノードローカルのファイルシステム
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のLOC001～LOC003 が該当します。
- 作業表用 DB エリアのディスク
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のWRK001～WRK003 が該当します。

このオペランドを指定する場合は，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの SCSI リザーブに必要なオペランドの指定値の求め方』を参照してください。

- dmmp_device

  このオペランドには，SCSI リザーブの対象となるデバイスを DMMP の論理デバイスの絶対パスで指定します。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクがマルチパスソフトウェア（DMMP）による冗長化構成のときに，このオペランドを指定してください。

  このオペランドを指定する場合は，次のファイルシステムとディスクのデバイス名を絶対パスで指定します。

- ノード切り替え対象のファイルシステム
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のFS001，FS002 が該当します。
- DB エリアファイル用のディスクのうち，すべてのノードから参照可能であるディスク
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のLU001～LU005 が該当します。

なお，同義語検索を使用しない場合は，同義語辞書ファイルを格納するファイルシステムのデバイス名を指定する必要はありません。

重要

次のファイルシステムとディスクのデバイス名は，このオペランドに指定しないでください。

- ノードローカルのファイルシステム
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のLOC001～LOC003 が該当します。
- 作業表用 DB エリアのディスク
  「図 16-1　マルチノード機能を使用した場合のシステム構成例」のWRK001～WRK003 が該当します。

このオペランドを指定する場合は，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの SCSI リザーブに必要なオペランドの指定値の求め方』を参照してください。

servers ファイルに指定する上記のオペランドの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『サーバ対応の環境設定（servers)』を参照してください。

(4)　コマンド用環境変数定義およびコマンドの作成

HA モニタとの連携のために，コマンド用環境変数定義と次のコマンドを作成してください。

- ユーザコマンド
- サーバの起動コマンド
- サーバの停止コマンド
- サーバの監視コマンド

$ADBDIR/sample/scripts 下に，コマンド用環境変数定義，および上記のコマンドのひな形があります。ひな形を基にコマンドを作成してください。

(a)　コマンド用環境変数定義の作成

$ADBDIR/sample/scripts/multinode.env がひな形です。このひな形をコピーして，次の値を修正してください。

- 環境変数ADBMGR
  HADB 管理者（OS ユーザ）のユーザ名を指定してください。

- 環境変数ADBALIAS
  HA モニタのservers ファイルのalias オペランドの指定値を指定してください。
- 環境変数ADBDIR
  サーバディレクトリの絶対パスを指定してください。

(b) ユーザコマンドの作成

$ADBDIR/sample/scripts/multinode_user.sh がひな形です。このひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(c) サーバの起動コマンドの作成

$ADBDIR/sample/scripts/multinode_act.sh がひな形です。このひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(d) サーバの停止コマンドの作成

系のリセットを使用するか，または共有ディスクの SCSI リザーブを使用するかによって，サーバの停止コマンドのひな形が異なります。系のリセットを使用する場合は，$ADBDIR/sample/scripts/multinode_term.sh がひな形です。共有ディスクの SCSI リザーブを使用する場合は，$ADBDIR/sample/scripts/multinode_term_scsi.sh がひな形です。

どちらか該当するひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(e) サーバの監視コマンドの作成

$ADBDIR/sample/scripts/multinode_patrol.sh がひな形です。このひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(5) 各ファイルの指定例（系のリセットを使用する場合）

次のファイルの指定例を示します。系のリセットを使用する場合の指定例です。

- sysdef ファイル
- servers ファイル
- コマンド用環境変数定義ファイル（この例では/HADB/scripts/multinode.env）
- ユーザコマンドファイル（この例では/HADB/scripts/multinode_user.sh）
- サーバの起動コマンドファイル（この例では/HADB/scripts/multinode_act.sh）
- サーバの停止コマンドファイル（この例では/HADB/scripts/multinode_term.sh）
- サーバの監視コマンドファイル（この例では/HADB/scripts/multinode_patrol.sh）

ここで説明する指定例は，「図 16-1　マルチノード機能を使用した場合のシステム構成例」のシステム構成を前提としています。

(a)　sysdef ファイルの指定例

■プライマリノードhadb01 のsysdef ファイルの指定例

```
environment  name hadb01,
             address 1,           ...1
             patrol 30,           ...2
             lan path11,          ...3
             lanport HAmon1;      ...4
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             multistandby use,
             usrcommand /HADB/scripts/multinode_user.sh,  ...5
             termcmd_at_abort nouse;
```

[説明]

1. このノードのリセット優先度を指定します。このノードは，プライマリノードのため，リセット優先度をいちばん高くする必要があるので1 を指定します。
2. patrol オペランドには，30 以下の値を指定してください。
3. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.11 に対応するホスト名path11 を指定します。
4. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。
5. マルチノード機能で使用するユーザコマンドの絶対パスを指定します。

■セカンダリノードhadb02 のsysdef ファイルの指定例

```
environment  name hadb02,
             address 2,          ...1
             patrol 30,          ...2
             lan path21,         ...3
             lanport HAmon1;     ...4
function     pathpatrol 240,
             connect_retry 5:200,
```

```
            monbegin_restart nouse,
            multistandby use,
            usrcommand /HADB/scripts/multinode_user.sh,  ...5
            termcmd_at_abort nouse;
```

[説明]

1. このノードのリセット優先度を指定します。このノードは，プライマリノードの次にリセット優先度を高くしたいため，2 を指定します。
2. patrol オペランドには，30 以下の値を指定してください。
3. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.12 に対応するホスト名path21 を指定します。
4. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。
5. マルチノード機能で使用するユーザコマンドの絶対パスを指定します。

■セカンダリノードhadb03 のsysdef ファイルの指定例

```
environment  name  hadb03,
             address 3,          ...1
             patrol 30,          ...2
             lan path31,         ...3
             lanport HAmon1;     ...4
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             multistandby use,
             usrcommand /HADB/scripts/multinode_user.sh,  ...5
             term_at_abort  nouse;
```

[説明]

1. このノードのリセット優先度を指定します。このノードは，セカンダリノードhadb02 の次にリセット優先度を高くしたいため，3 を指定します。
2. patrol オペランドには，30 以下の値を指定してください。
3. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.13 に対応するホスト名path31 を指定します。
4. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。
5. マルチノード機能で使用するユーザコマンドの絶対パスを指定します。

(b) servers ファイルの指定例

ここで説明している例は，ファイルシステムに ext4 を使用していることを前提としています。ext4 以外のファイルシステムを使用する場合は，OS のmount コマンドのオプションがここで説明している例とは異なります。

■プライマリノードhadb01 のservers ファイルの指定例

```
server name /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                          ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:              ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:               ...4
                    /mnt/syndict,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":  ...5
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/multinode_act.sh",         ...6
       termcommand "/HADB/scripts/multinode_term.sh",       ...7
       patrolcommand "/HADB/scripts/multinode_patrol.sh",   ...8
       servexec_retry 0,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:use,   ...9
       fs_neck use:use,   ...10
       initial online;    ...11
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. プライマリノードとするため，online を指定します。

■セカンダリノードhadb02 のservers ファイルの指定例

```
server name /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
```

```
       disk /dev/vg_hadb:                        ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:            ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:             ...4
                    /mnt/syndict,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":  ...5
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/multinode_act.sh",         ...6
       termcommand "/HADB/scripts/multinode_term.sh",       ...7
       patrolcommand "/HADB/scripts/multinode_patrol.sh",   ...8
       servexec_retry 0,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:use,   ...9
       fs_neck use:use,   ...10
       initial standby,   ...11
       standbypri 1;      ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするための mount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. セカンダリノードとするため，standby を指定します。
12. プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。このノードの優先順位をいちばん高くしたいため，1 を指定します。

■セカンダリノードhadb03 のservers ファイルの指定例

```
server name /HADB/server,                        ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                        ...2
```

```
     /dev/vg_hadb02,
lan_updown use,
fs_name /dev/vg_hadb/sys_hadb:         ...3
        /dev/vg_hadb02/hadb_syndict,
fs_mount_dir /HADB/db/ADBSYS:          ...4
             /mnt/syndict,
fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":  ...5
             "-t ext4 -o defaults,noatime,_netdev",
actcommand "/HADB/scripts/multinode_act.sh",         ...6
termcommand "/HADB/scripts/multinode_term.sh",       ...7
patrolcommand "/HADB/scripts/multinode_patrol.sh",   ...8
servexec_retry 0,
waitserv_exec yes,
ip_neck use,
uoc_neck nouse,
vg_neck use:use,   ...9
fs_neck use:use,   ...10
initial standby,   ...11
standbypri 2;      ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. セカンダリノードとするため，standby を指定します。
12. プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。このノードの優先順位は，ノードhadb02 の次にしたいため，2 を指定します。

(c) コマンド用環境変数定義ファイルの指定例

この例では，コマンド用環境変数定義を/HADB/scripts/multinode.env に格納しています。指定内容は，プライマリノードおよびセカンダリノードで同じにしてください。

■コマンド用環境変数定義ファイルの指定例

```
#!/bin/sh

# The environment variables for HADB
export ADBMGR=adbmanager                ...1
export ADBALIAS=HADB                    ...2
export ADBDIR=/HADB/server              ...3
```

[説明]

1. HADB 管理者（OS ユーザ）のユーザ名adbmanager を指定します。

2. HA モニタのservers ファイルのalias オペランドの指定値であるHADB を指定します。

3. サーバディレクトリの絶対パスを指定します。

(d) ユーザコマンドファイルの指定例

この例では，ユーザコマンドを/HADB/scripts/multinode_user.sh に格納しています。指定内容は，プライマリノードおよびセカンダリノードで同じにしてください。

■ユーザコマンドファイルの指定例

```
#!/bin/sh

# Sample of the usrcommand for HADB

# Setting environment variables for HADB
source /HADB/scripts/multinode.env       ...1

###############################################################################
#  Main
###############################################################################

KIND_ONLINE="online"
KIND_STANDBY="standby"
SERV_START="-s"
SERV_END="-e"
SERV_PLANEND="-p"
SERV_ABORT="-a"
SERV_ABORT_NS="-o"
SERV_FAULT="-f"
SERV_HOSTDOWN="-h"
SERV_PLANSWAP="-w"
STATUS_START="start"
STATUS_END="end"
STATUS_SBYEND="sbyend"


patrol_sby_exe()
{
  $ADBDIR/bin/patrol_sby_exe       ...2
}

patrol_sby_term()
{
```

```
  $ADBDIR/bin/patrol_sby_term      ...3
}

stop_sby()
{
  $ADBDIR/bin/stop_sby             ...4
}

# Processing of the server
if [ "$2" = "$ADBALIAS" ]
then
  if [ "$4" = "$KIND_ONLINE" ]
  then
    case "$5" in
    "$SERV_START" )
      ;;
    "$SERV_END" )
      ;;
    "$SERV_PLANEND" )
      ;;
    "$SERV_ABORT" )
      ;;
    "$SERV_ABORT_NS" )
      ;;
    "$SERV_PLANSWAP" )
      ;;
    esac
  else
    case "$5" in
    "$SERV_START" )
      [ "$6" = "$STATUS_START" ] && patrol_sby_exe &
      ;;
    "$SERV_END" )
      [ "$6" = "$STATUS_START" -o "$6" = "$STATUS_SBYEND" ] && stop_sby
      ;;
    "$SERV_PLANEND" )
      [ "$6" = "$STATUS_START" ] && patrol_sby_term
      ;;
    "$SERV_ABORT" )
      [ "$6" = "$STATUS_START" ] && patrol_sby_term
      [ "$6" = "$STATUS_SBYEND" ] && stop_sby
      ;;
    "$SERV_FAULT" )
      [ "$6" = "$STATUS_START" ] && stop_sby
      ;;
    "$SERV_HOSTDOWN" )
      [ "$6" = "$STATUS_START" ] && patrol_sby_term
      ;;
    "$SERV_PLANSWAP" )
      [ "$6" = "$STATUS_START" ] && patrol_sby_term
      ;;
    esac
  fi
fi

exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/multinode.env を指定します。
2. セカンダリノードの監視を開始するためのコマンドを指定します。
3. セカンダリノードの監視を終了するためのコマンドを指定します。
4. セカンダリノードを停止するためのコマンドを指定します。

(e) サーバの起動コマンドファイルの指定例

この例では，サーバの起動コマンドを/HADB/scripts/multinode_act.sh に格納しています。指定内容は，プライマリノードおよびセカンダリノードで同じにしてください。

■サーバの起動コマンドファイルの指定例

```
#!/bin/sh

# Sample of the actcommand for HADB

SU=/bin/su

# Setting environment variables for HADB
source /HADB/scripts/multinode.env      ...1

# Execute adbchgnodetype command
$SU - $ADBMGR -c "$ADBDIR/bin/adbchgnodetype --primary -n $ADBALIAS"
CHGNODERES=$?
if [ "$CHGNODERES" = "0" -o "$CHGNODERES" = "4" ]
then
  exit 0
else
  exit $CHGNODERES
fi
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/multinode.env を指定します。

(f) サーバの停止コマンドファイルの指定例

この例では，サーバの停止コマンドを/HADB/scripts/multinode_term.sh に格納しています。指定内容は，プライマリノードおよびセカンダリノードで同じにしてください。

■サーバの停止コマンドファイルの指定例

```
#!/bin/sh

# Sample of the termcommand for HADB

STS_ACTIVE="ACTIVE"
STS_STOP="STOP"
STS_STARTING="STARTING"
STS_STOPPING="STOPPING"
```

```
STS_ABORT="ABORT"
STS_QUIESCE="QUIESCE"
STS_OFFLINE="OFFLINE"
STS_CHGMODE="CHGMODE"
STS_STOPWAIT="STOPWAIT"
STS_COREDUMP="COREDUMP"
STS_FORCE="FORCE"
STS_MAINTNCE="MAINTNCE"

AWK=/bin/awk
ECHO=/bin/echo
GREP=/bin/grep
SU=/bin/su

# Setting environment variables for HADB
source /HADB/scripts/multinode.env      ...1

# Execute adbstop command
while :
do

  GET_STS=`$SU - $ADBMGR -c "$ADBDIR/bin/adbls -d srv 2>/dev/null" | $GREP -v SVID | $AWK
'{ if ($1 ~ /[0-9]+/) { print $2 } else { print $1 } }'`

  case "$1" in
  "-e" )
    # Normal stop operations of primary node.
    # (when the monend command is executed.)
    if [ "$GET_STS" = "$STS_ACTIVE"   -o ¥
         "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
         "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
         "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_MAINTNCE" ]
    then
      $SU - $ADBMGR -c "$ADBDIR/bin/adbstop --cancel"
      STOPRES=$?
      if [ "$STOPRES" = "0" -o "$STOPRES" = "4" ]
      then
        break
      fi
    elif [ "$GET_STS" = "$STS_STARTING" ]
    then
      break
    else
      break
    fi

    ;;
  "-w" )
    # Plan stop operations of primary node.
    # (when the monswap command is executed.)
    if [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
         "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
         "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
         "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_MAINTNCE" ]
    then
      $SU - $ADBMGR -c "$ADBDIR/bin/adbstop --cancel --node"
      STOPRES=$?
      if [ "$STOPRES" = "0" -o "$STOPRES" = "4" ]
```

```
      then
        break
      fi
    else
      break
    fi

    ;;
  "-c" )
    break

    ;;
  esac

done

# stop HADB
if [ "$GET_STS" = "$STS_STARTING" ]
then
  ADB_ID=`$SU - $ADBMGR -c "ps x" | $GREP adbsrvd | $GREP -v $GREP | awk '{print $1}'`
  $SU - $ADBMGR -c "kill $ADB_ID"

  exit 0
fi

# Wait for end of HADB
while [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
        "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
        "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
        "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_COREDUMP" -o ¥
        "$GET_STS" = "$STS_MAINTNCE" ]
do
  GET_STS=`$SU - $ADBMGR -c "$ADBDIR/bin/adbls -d srv 2>/dev/null" | $GREP -v SVID | $AWK
 '{ if ($1 ~ /[0-9]+/) { print $2 } else { print $1 } }'`
  sleep 1
done

exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/multinode.env を指定します。

(g) サーバの監視コマンドファイルの指定例

この例では，サーバの監視コマンドを/HADB/scripts/multinode_patrol.sh に格納しています。指定内容は，プライマリノードおよびセカンダリノードで同じにしてください。

■サーバの監視コマンドファイルの指定例

```
#!/bin/sh

# Sample of the patrolcommand for HADB

AWK=/bin/awk
```

```
GREP=/bin/grep
PS=/bin/ps
PGREP=/usr/bin/pgrep
PKILL=/usr/bin/pkill
SU=/bin/su

# Setting environment variables for HADB
source /HADB/scripts/multinode.env      ...1

# Execute adbmonitor command for primary node
$SU - $ADBMGR -c "$ADBDIR/bin/adbmonitor -n" &

## Get su command process id
PID=$!

# Trap SIGTERM and terminate adbmonitor
trap "$PKILL -P $PID" 15


## Wait process terminated
CPID=""
CHKPID=`$PS aux | $AWK -v PID=$PID '{ if ($2 == PID) { print $2 } }'`
while [ "$CHKPID" != "" ]
do

  ## Get adbmonitor process id
  if [ "$CPID" = "" ]
  then
    CPID=`$PGREP -P $PID`
  fi

  ## Wait
  if [ "$CPID" != "" ]
  then
    CHKCPID=`$PS aux | $AWK -v CPID=$CPID '{ if ($2 == CPID) { print $2 } }'`
    while [ "$CHKCPID" != "" ]
    do
      sleep 1
      CHKCPID=`$PS aux | $AWK -v CPID=$CPID '{ if ($2 == CPID) { print $2 } }'`
    done
  fi

  sleep 1
  CHKPID=`$PS aux | $AWK -v PID=$PID '{ if ($2 == PID) { print $2 } }'`
done

exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/multinode.env を指定します。

(6) 各ファイルの指定例（共有ディスクの SCSI リザーブを使用する場合）

次のファイルの指定例を示します。共有ディスクの SCSI リザーブを使用する場合の指定例です。

- sysdef ファイル
- servers ファイル
- コマンド用環境変数定義ファイル（この例では/HADB/scripts/multinode.env）
- ユーザコマンドファイル（この例では/HADB/scripts/multinode_user.sh）
- サーバの起動コマンドファイル（この例では/HADB/scripts/multinode_act.sh）
- サーバの停止コマンドファイル（この例では/HADB/scripts/multinode_term_scsi.sh）
- サーバの監視コマンドファイル（この例では/HADB/scripts/multinode_patrol.sh）

ここで説明する指定例は，「図 16-1　マルチノード機能を使用した場合のシステム構成例」のシステム構成を前提としています。

(a)　sysdef ファイルの指定例

■プライマリノードhadb01 のsysdef ファイルの指定例

```
environment  name hadb01,
             address 1,         ...1
             patrol 30,         ...2
             lan path11,        ...3
             lanport HAmon1;    ...4
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             multistandby use,
             usrcommand /HADB/scripts/multinode_user.sh,              ...5
             termcmd_at_abort nouse,
             fence_reset nouse,                                       ...6
             fence_scsi use,                                          ...7
             fence_lan nouse;
```

[説明]

1. このノードのリセット優先度を指定します。このノードは，プライマリノードのため，リセット優先度をいちばん高くする必要があるので1 を指定します。
2. patrol オペランドには，30 以下の値を指定してください。
3. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.11 に対応するホスト名path11 を指定します。
4. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。
5. マルチノード機能で使用するユーザコマンドの絶対パスを指定します。
6. 系のリセットを使用しないため，nouse を指定します。
7. 共有ディスクの SCSI リザーブを使用するため，use を指定します。

■セカンダリノードhadb02 のsysdef ファイルの指定例

```
environment  name hadb02,
             address 2,         ...1
             patrol 30,         ...2
             lan path21,        ...3
             lanport HAmon1;    ...4
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             multistandby use,
             usrcommand /HADB/scripts/multinode_user.sh,              ...5
             termcmd_at_abort nouse,
             fence_reset nouse,                                       ...6
             fence_scsi use,                                          ...7
             fence_lan nouse;
```

[説明]

1. このノードのリセット優先度を指定します。このノードは，プライマリノードの次にリセット優先度を高くしたいため，2 を指定します。
2. patrol オペランドには，30 以下の値を指定してください。
3. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.12 に対応するホスト名path21 を指定します。
4. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。
5. マルチノード機能で使用するユーザコマンドの絶対パスを指定します。
6. 系のリセットを使用しないため，nouse を指定します。
7. 共有ディスクの SCSI リザーブを使用するため，use を指定します。

■セカンダリノードhadb03 のsysdef ファイルの指定例

```
environment  name hadb03,
             address 3,         ...1
             patrol 30,         ...2
             lan path31,        ...3
             lanport HAmon1;    ...4
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             multistandby use,
             usrcommand /HADB/scripts/multinode_user.sh,              ...5
             termcmd_at_abort nouse,
             fence_reset nouse,                                       ...6
             fence_scsi use,                                          ...7
             fence_lan nouse;
```

[説明]

1. このノードのリセット優先度を指定します。このノードは，セカンダリノードhadb02 の次にリセット優先度を高くしたいため，3 を指定します。

2. patrol オペランドには，30 以下の値を指定してください。

3. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.13 に対応するホスト名path31 を指定します。

4. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。

5. マルチノード機能で使用するユーザコマンドの絶対パスを指定します。

6. 系のリセットを使用しないため，nouse を指定します。

7. 共有ディスクの SCSI リザーブを使用するため，use を指定します。

(b) servers ファイルの指定例（シングルパス構成の場合）

シングルパス構成の場合のservers ファイルの指定例を説明します。

ここで説明している例は，ファイルシステムに ext4 を使用していることを前提としています。ext4 以外のファイルシステムを使用する場合は，OS のmount コマンドのオプションがここで説明している例とは異なります。

■プライマリノードhadb01 のservers ファイルの指定例

```
server name /HADB/server,                              ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                              ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:                  ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:                   ...4
                    /mnt/syndict,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":       ...5
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/multinode_act.sh",              ...6
       termcommand "/HADB/scripts/multinode_term_scsi.sh",       ...7
       patrolcommand "/HADB/scripts/multinode_patrol.sh",        ...8
       servexec_retry 0,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:use,                                                    ...9
       fs_neck use:use,                                                    ...10
       scsi_device /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b53e0a:  ...11
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b6c44d:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b828e9:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b7d9fd:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b87793:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b8c6d3:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b9160f,
       initial online;                                                     ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 次のファイルシステムおよびファイルがあるディスクの絶対パスを指定します。
    - システムディレクトリを作成するファイルシステム
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステム
    - DB エリアファイル（作業表用 DB エリアファイルを除く）

    プライマリノードおよびセカンダリノードで，絶対パスの指定順序を同じにしてください。
12. プライマリノードとするため，online を指定します。

■セカンダリノードhadb02 のservers ファイルの指定例

```
server name /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                          ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:              ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:               ...4
                    /mnt/syndict,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":      ...5
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/multinode_act.sh",             ...6
       termcommand "/HADB/scripts/multinode_term_scsi.sh",      ...7
       patrolcommand "/HADB/scripts/multinode_patrol.sh",       ...8
       servexec_retry 0,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:use,                                                ...9
```

```
       fs_neck use:use,                                                    ...10
       scsi_device /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b53e0a:  ...11
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b6c44d:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b828e9:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b7d9fd:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b87793:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b8c6d3:
                   /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b9160f,
       initial standby,                                                    ...12
       standbypri 1;                                                       ...13
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 次のファイルシステムおよびファイルがあるディスクの絶対パスを指定します。
    - システムディレクトリを作成するファイルシステム
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステム
    - DB エリアファイル（作業表用 DB エリアファイルを除く）

    プライマリノードおよびセカンダリノードで，絶対パスの指定順序を同じにしてください。
12. セカンダリノードとするため，standby を指定します。
13. プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。このノードの優先順位をいちばん高くしたいため，1 を指定します。

■セカンダリノードhadb03 のservers ファイルの指定例

```
server name /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                          ...2
```

```
     /dev/vg_hadb02,
lan_updown use,
fs_name /dev/vg_hadb/sys_hadb:              ...3
        /dev/vg_hadb02/hadb_syndict,
fs_mount_dir /HADB/db/ADBSYS:               ...4
             /mnt/syndict,
fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":       ...5
             "-t ext4 -o defaults,noatime,_netdev",
actcommand "/HADB/scripts/multinode_act.sh",              ...6
termcommand "/HADB/scripts/multinode_term_scsi.sh",       ...7
patrolcommand "/HADB/scripts/multinode_patrol.sh",        ...8
servexec_retry 0,
waitserv_exec yes,
ip_neck use,
uoc_neck nouse,
vg_neck use:use,                                                    ...9
fs_neck use:use,                                                    ...10
scsi_device /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b53e0a:  ...11
            /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b6c44d:
            /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b828e9:
            /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b7d9fd:
            /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b87793:
            /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b8c6d3:
            /dev/disk/by-id/scsi-3600605b0041db9c016ea34c3b1b9160f,
initial standby,                                                    ...12
standbypri 2;                                                       ...13
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 次のファイルシステムおよびファイルがあるディスクの絶対パスを指定します。
    - システムディレクトリを作成するファイルシステム
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステム

- DB エリアファイル（作業表用 DB エリアファイルを除く）

プライマリノードおよびセカンダリノードで，絶対パスの指定順序を同じにしてください。

12. セカンダリノードとするため，standby を指定します。
13. プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。このノードの優先順位は，ノードhadb02 の次にしたいため，2 を指定します。

(c) servers ファイルの指定例（マルチパスソフトウェアによる冗長化構成の場合）

マルチパスソフトウェアによる冗長化構成の場合のservers ファイルの指定例を説明します。

ここで説明している例は，ファイルシステムに ext4 を使用していることを前提としています。ext4 以外のファイルシステムを使用する場合は，OS のmount コマンドのオプションがここで説明している例とは異なります。

■プライマリノードhadb01 のservers ファイルの指定例

```
server name /HADB/server,                              ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                              ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:                  ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:                   ...4
                    /mnt/syndict,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":   ...5
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/multinode_act.sh",          ...6
       termcommand "/HADB/scripts/multinode_term_scsi.sh",   ...7
       patrolcommand "/HADB/scripts/multinode_patrol.sh",    ...8
       servexec_retry 0,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:use,                        ...9
       fs_neck use:use,                        ...10
       dmmp_device /dev/mapper/mpath1:         ...11
                   /dev/mapper/mpath2:
                   /dev/mapper/mpath11:
                   /dev/mapper/mpath12:
                   /dev/mapper/mpath13:
                   /dev/mapper/mpath14:
                   /dev/mapper/mpath15,
       initial online;                         ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ

- 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）

3. 2.で示したディレクトリを作成するファイルシステムを構築したLVの絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmountコマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがあるVGに対しては，useを指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，useを指定します。
11. 次のファイルシステムおよびファイルがあるディスクの絶対パスを指定します。
    - システムディレクトリを作成するファイルシステム
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステム
    - DBエリアファイル（作業表用DBエリアファイルを除く）

    プライマリノードおよびセカンダリノードで，絶対パスの指定順序を同じにしてください。
12. プライマリノードとするため，onlineを指定します。

■セカンダリノードhadb02のservers ファイルの指定例

```
server name /HADB/server,                              ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                              ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:                  ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:                   ...4
                    /mnt/syndict,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":   ...5
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/multinode_act.sh",          ...6
       termcommand "/HADB/scripts/multinode_term_scsi.sh",   ...7
       patrolcommand "/HADB/scripts/multinode_patrol.sh",    ...8
       servexec_retry 0,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:use,                        ...9
       fs_neck use:use,                        ...10
       dmmp_device /dev/mapper/mpath1:         ...11
                   /dev/mapper/mpath2:
                   /dev/mapper/mpath11:
                   /dev/mapper/mpath12:
                   /dev/mapper/mpath13:
```

```
                /dev/mapper/mpath14:
                /dev/mapper/mpath15,
      initial standby,                    ...12
      standbypri 1;                       ...13
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがあるVGの絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築したLVの絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmountコマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがあるVGに対しては，useを指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，useを指定します。
11. 次のファイルシステムおよびファイルがあるディスクの絶対パスを指定します。
    - システムディレクトリを作成するファイルシステム
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステム
    - DBエリアファイル（作業表用DBエリアファイルを除く）

    プライマリノードおよびセカンダリノードで，絶対パスの指定順序を同じにしてください。
12. セカンダリノードとするため，standbyを指定します。
13. プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。このノードの優先順位をいちばん高くしたいため，1を指定します。

■セカンダリノードhadb03のservers ファイルの指定例

```
server name /HADB/server,                 ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb:                 ...2
            /dev/vg_hadb02,
       lan_updown use,
       fs_name /dev/vg_hadb/sys_hadb:     ...3
               /dev/vg_hadb02/hadb_syndict,
       fs_mount_dir /HADB/db/ADBSYS:      ...4
                    /mnt/syndict,
```

```
fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":   ...5
             "-t ext4 -o defaults,noatime,_netdev",
actcommand "/HADB/scripts/multinode_act.sh",          ...6
termcommand "/HADB/scripts/multinode_term_scsi.sh",   ...7
patrolcommand "/HADB/scripts/multinode_patrol.sh",    ...8
servexec_retry 0,
waitserv_exec yes,
ip_neck use,
uoc_neck nouse,
vg_neck use:use,                    ...9
fs_neck use:use,                    ...10
dmmp_device /dev/mapper/mpath1:     ...11
            /dev/mapper/mpath2:
            /dev/mapper/mpath11:
            /dev/mapper/mpath12:
            /dev/mapper/mpath13:
            /dev/mapper/mpath14:
            /dev/mapper/mpath15,
initial standby,                    ...12
standbypri 2;                       ...13
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - システムディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするための mount コマンドのオプションを指定します。
6. マルチノード機能で使用するサーバの起動コマンドの絶対パスを指定します。
7. マルチノード機能で使用するサーバの停止コマンドの絶対パスを指定します。
8. マルチノード機能で使用するサーバの監視コマンドの絶対パスを指定します。
9. 2.で示したディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 2.で示したディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 次のファイルシステムおよびファイルがあるディスクの絶対パスを指定します。
    - システムディレクトリを作成するファイルシステム
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステム
    - DB エリアファイル（作業表用 DB エリアファイルを除く）

    プライマリノードおよびセカンダリノードで，絶対パスの指定順序を同じにしてください。
12. セカンダリノードとするため，standby を指定します。

13. プライマリノードの切り替えが発生した際，新たにプライマリノードにするノードの優先順位を指定します。このノードの優先順位は，ノードhadb02の次にしたいため，2を指定します。

(d)　コマンド用環境変数定義ファイルの指定例

コマンド用環境変数定義ファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合)」の「(c)　コマンド用環境変数定義ファイルの指定例」を参照してください。

(e)　ユーザコマンドファイルの指定例

ユーザコマンドファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合)」の「(d)　ユーザコマンドファイルの指定例」を参照してください。

(f)　サーバの起動コマンドファイルの指定例

サーバの起動コマンドファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合)」の「(e)　サーバの起動コマンドファイルの指定例」を参照してください。

(g)　サーバの停止コマンドファイルの指定例

この例では，サーバの停止コマンドを/HADB/scripts/multinode_term_scsi.shに格納しています。指定内容は，プライマリノードおよびセカンダリノードで同じにしてください。

■サーバの停止コマンドファイルの指定例

```
#!/bin/sh

# Sample of the termcommand for HADB

STS_ACTIVE=”ACTIVE”
STS_STOP=”STOP”
STS_STARTING=”STARTING”
STS_STOPPING=”STOPPING”
STS_ABORT=”ABORT”
STS_QUIESCE=”QUIESCE”
STS_OFFLINE=”OFFLINE”
STS_CHGMODE=”CHGMODE”
STS_STOPWAIT=”STOPWAIT”
STS_COREDUMP=”COREDUMP”
STS_FORCE=”FORCE”
STS_MAINTNCE=”MAINTNCE”

AWK=/bin/awk
ECHO=/bin/echo
GREP=/bin/grep
SU=/bin/su

# Setting environment variables for HADB
source /HADB/scripts/multinode.env      ...1

# Execute adbstop command
while :
do
```

```
  GET_STS=`$SU - $ADBMGR -c "$ADBDIR/bin/adbls -d srv 2>/dev/null" | $GREP -v SVID | $AWK
 '{ if ($1 ~ /[0-9]+/) { print $2 } else { print $1 } }'`

  case "$1" in
  "-e" )
    # Normal stop operations of primary node.
    # (when the monend command is executed.)
    if [ "$GET_STS" = "$STS_ACTIVE"   -o ¥
         "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
         "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
         "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_MAINTNCE" ]
    then
      $SU - $ADBMGR -c "$ADBDIR/bin/adbstop --cancel"
      STOPRES=$?
      if [ "$STOPRES" = "0" -o "$STOPRES" = "4" ]
      then
        break
      fi
    elif [ "$GET_STS" = "$STS_STARTING" ]
    then
      break
    else
      break
    fi

    ;;
  "-w" )
    # Plan stop operations of primary node.
    # (When the monswap command is executed or a monitoring path failure occurs.)
    if [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
         "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
         "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
         "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_MAINTNCE" ]
    then
      $SU - $ADBMGR -c "$ECHO y|$ADBDIR/bin/adbstop --force"
      STOPRES=$?
      if [ "$STOPRES" = "0" -o "$STOPRES" = "4" ]
      then
        break
      fi
    else
      break
    fi

    ;;
  "-c" )
    break

    ;;
  esac

done

# stop HADB
if [ "$GET_STS" = "$STS_STARTING" ]
then
  ADB_ID=`$SU - $ADBMGR -c "ps x" | $GREP adbsrvd | $GREP -v $GREP | awk '{print $1}'`
```

```
  $SU - $ADBMGR -c "kill $ADB_ID"

  exit 0
fi

# Wait for end of HADB
while [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
        "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
        "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
        "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_COREDUMP" -o ¥
        "$GET_STS" = "$STS_MAINTNCE" ]
do
  GET_STS=`$SU - $ADBMGR -c "$ADBDIR/bin/adbls -d srv 2>/dev/null" | $GREP -v SVID | $AWK
 '{ if ($1 ~ /[0-9]+/) { print $2 } else { print $1 } }'`
  sleep 1
done

exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/multinode.env を指定します。

(h)　サーバの監視コマンドファイルの指定例

サーバの監視コマンドファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合）」の「(g)　サーバの監視コマンドファイルの指定例」を参照してください。

(7)　HA モニタの起動設定

OS の起動時に，自動的に HA モニタが起動するように設定してください。設定方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの起動からサーバの起動までを自動化する』を参照してください。

16.3.5　各ノードでのサーバ定義の作成

マルチノード機能を使用する場合は，全ノードでサーバ定義を作成する必要があります。その際，adb_sys_multi_node_info オペランドは，必ず指定してください。

サーバ定義の作成方法については，「8.5　サーバ定義の作成および変更方法」を参照してください。サーバ定義のオペランドについては，「7.　サーバ定義の設計」を参照してください。

■全ノードで指定値を同じにする必要があるオペランド

次のオペランドは，全ノードで指定値を同じにしてください。オペランドを省略した場合は，全ノードでオペランドを省略してください。指定値が異なると，マルチノード構成の HADB サーバを開始できません。

- adb_dbarea_wrk_page_size

- adb_sys_multi_node_info（指定値の順序も同じにしてください）
- adb_sys_max_parallel_exec_num
- adb_sys_max_users
- adb_sys_max_users_wrn_pnt
- adb_sys_trn_iso_lv
- adb_sql_order_mode
- adb_sql_prep_delrsvd_words（指定値の順序も同じにしてください）
- adb_sql_prep_dec_div_rs_prior
- adbcltgrp（-g，-m，-u，および-w オプション）

■プライマリノードとセカンダリノードで指定値を同じにする必要があるオペランド

次のオペランドは，プライマリノードとセカンダリノードで指定値を同じにしてください。オペランドを省略する場合は，プライマリノードおよびセカンダリノードでオペランドを省略してください。指定値が異なると，マルチノード構成の HADB サーバを開始できません。

- adb_rpc_port
- adb_sql_default_dbarea_shared
- adb_syndict_storage_path

■ワーカーノードを使用する場合

ワーカーノードを使用する場合は，全ノードに次のオペランドを指定してください。また，指定値を全ノードで同じにしてください。

- adb_sys_multi_max_worker_nodes

また，ワーカーノードには，次のオペランドを指定してください。

- adb_sys_multi_worker_node_info

■SQL パラレル実行機能を使用する場合

SQL パラレル実行機能を使用する場合は，次のオペランドの指定値を全ノードで同じにしてください。オペランドを省略する場合は，全ノードでオペランドを省略してください。オペランドの指定値が SQL メインノードと一致しないノードは，SQL サブノードの候補になりません。そのため，オペランドの指定値を一致させておかないと，検索系 SQL の処理で使用されるノード数が少なくなります。SQL パラレル実行機能を適用する検索系 SQL の実行時に SQL サブノードが 1 つも確保できない場合，その検索系 SQL はエラーになります。

- adb_sys_rthd_num
- adb_sys_uthd_num
- adb_sys_shm_huge_page_size
- adb_sql_opt_drvtbl_grping_prior
- adb_sql_exe_max_rthd_num

- adb_sql_exe_hashgrp_area_size
- adb_sql_exe_hashtbl_area_size
- adb_sql_exe_hashflt_area_size
- adb_sql_tbldef_cache_size
- adb_sql_rngidx_preread
- adb_dbbuff_wrktbl_glb_blk_num
- adb_dbbuff_wrktbl_clt_blk_num
- adbbuff
- adbcltgrp（-r および-e オプション）

■監査証跡機能を使用する場合

監査証跡機能を使用する場合は，全ノードに次のオペランドを指定してください。オペランドの指定値は，ノードごとに異なっていてもかまいません。オペランドの指定有無がノードごとに異なると，マルチノード構成の HADB サーバを開始できません。

- adb_audit_log_path

■サーバ定義の指定例（プライマリノードおよびセカンダリノードの場合）

プライマリノードおよびセカンダリノードのサーバ定義の指定例を次に示します。指定値を同じにする必要があるオペランド以外は，ノードごとに指定値を変更できます。

```
set adb_db_path = /home/adbmanager/db
set adb_rpc_port = 23650
set adb_sys_max_users = 10
set adb_sys_rthd_num = 40
set adb_sys_uthd_num = 128
set adb_sql_exe_max_rthd_num = 4
set adb_sys_rthd_area_max = 1024
set adb_sys_proc_area_max = 8192
set adb_sys_multi_node_info =172.16.0.11:23651,¥    ...1
                             172.16.0.12:23651,¥    ...2
                             172.16.0.13:23651      ...3
set adb_sys_multi_max_worker_nodes = 8              ...4

adbbuff -g TBLBUF01 ¥
        -n ADBUTBL01 ¥
        -p 1000000 ¥
        -v 1024
adbbuff -g IDXBUF01 ¥
        -n ADBUIDX01 ¥
        -p 2500000
```

[説明]

1. プライマリノードhadb01 の HADB サーバ間通信用の IP アドレス172.16.0.11，およびポート番号23651 を指定します。

2. セカンダリノードhadb02 の HADB サーバ間通信用の IP アドレス172.16.0.12，およびポート番号23651 を指定します。

3. セカンダリノードhadb03 の HADB サーバ間通信用の IP アドレス172.16.0.13，およびポート番号23651 を指定します。

4. ワーカーノードの最大数を指定します。最大８つのワーカーノードを追加できるようになります。

■サーバ定義の指定例（ワーカーノードの場合）

ワーカーノードのサーバ定義の指定例を次に示します。指定値を同じにする必要があるオペランド以外は，ワーカーノードごとに指定値を変更できます。

```
set adb_db_path = /home/adbmanager/db
set adb_rpc_port = 23650
set adb_sys_max_users = 10
set adb_sys_rthd_num = 40
set adb_sys_uthd_num = 128
set adb_sql_exe_max_rthd_num = 4
set adb_sys_rthd_area_max = 1024
set adb_sys_proc_area_max = 8192
set adb_sys_multi_node_info =172.16.0.11:23651,¥            ...1
                             172.16.0.12:23651,¥            ...2
                             172.16.0.13:23651              ...3
set adb_sys_multi_max_worker_nodes = 8                      ...4
set adb_sys_multi_worker_node_info = 172.16.0.14:23651      ...5

adbbuff -g TBLBUF01 ¥
        -n ADBUTBL01 ¥
        -p 1000000 ¥
        -v 1024

adbbuff -g IDXBUF01 ¥
        -n ADBUIDX01 ¥
        -p 2500000
```

[説明]

1. プライマリノードhadb01 の HADB サーバ間通信用の IP アドレス172.16.0.11，およびポート番号23651 を指定します。

2. セカンダリノードhadb02 の HADB サーバ間通信用の IP アドレス172.16.0.12，およびポート番号23651 を指定します。

3. セカンダリノードhadb03 の HADB サーバ間通信用の IP アドレス172.16.0.13，およびポート番号23651 を指定します。

4. ワーカーノードの最大数を指定します。最大８つのワーカーノードを追加できるようになります。

5. ワーカーノードhadb04 の HADB サーバ間通信用の IP アドレス172.16.0.14，およびポート番号23651 を指定します。

重要

上記の 5.のadb_sys_multi_worker_node_info オペランドは，ワーカーノードごとに指定値が変わります。自ワーカーノード（このサーバ定義を指定するワーカーノード）のホスト名とポート番号を指定します。

16.3.6 クライアント定義の作成

検索系 SQL だけを実行する AP と，更新系 SQL も実行する AP とで，使用するクライアント定義を別々にし，トランザクションアクセスモードおよびトランザクション隔離性水準の指定を変えてください。JDBC ドライバを使用している場合は，システムプロパティなどでトランザクションアクセスモードおよびトランザクション隔離性水準の指定を変えてください。

クライアント定義の作成については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』の『クライアント定義の作成』を参照してください。クライアント定義のオペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』の『クライアント定義のオペランドの内容』を参照してください。

■クライアント定義の指定例（検索系 SQL だけを実行する AP の場合）

```
set adb_clt_rpc_srv_host = 10.196.108.143     ...1
set adb_clt_rpc_srv_port = 23650              ...2
set adb_clt_ap_name = ”AP01”                  ...3
set adb_clt_trn_access_mode = READ_ONLY       ...4
set adb_clt_trn_iso_lv = READ_COMMITTED       ...5
set adb_clt_sql_parallel_exec = Y             ...6
```

[説明]

1. 接続先のホスト名として，クライアントーサーバ間の通信で使用するエイリアス IP アドレス 10.196.108.143 を指定します。
2. 接続先のポート番号として，サーバ定義adb_rpc_port オペランドの指定値である23650 を指定します。
3. AP 識別子を指定します。
4. 検索系 SQL だけを実行する AP のため，READ_ONLY を指定し，トランザクションアクセスモードを読み取り専用モードにします。
5. トランザクション隔離性水準にREAD_COMMITTED を指定します。

   4.および 5.の指定によって，プライマリノードの HADB サーバが負荷の少ないノードを選択し，そのノードで検索系 SQL が実行されます。
6. 条件を満たす検索系 SQL に SQL パラレル実行機能を適用する場合に指定します。

   検索系 SQL に SQL パラレル実行機能が適用された場合，プライマリノードの HADB サーバが SQL メインノードを選択し，SQL メインノードと SQL サブノードで検索系 SQL を並列実行します。

■クライアント定義の指定例（更新系 SQL も実行する AP の場合）

```
set adb_clt_rpc_srv_host = 10.196.108.143     ...1
set adb_clt_rpc_srv_port = 23650              ...2
set adb_clt_ap_name = ”AP02”                  ...3
set adb_clt_trn_access_mode = READ_WRITE      ...4
```

［説明］

1. 接続先のホスト名として，クライアントーサーバ間の通信で使用するエイリアス IP アドレス 10.196.108.143 を指定します。
2. 接続先のポート番号として，サーバ定義adb_rpc_port オペランドの指定値である23650 を指定します。
3. AP 識別子を指定します。
4. 更新系 SQL も実行する AP のため，READ_WRITE を指定し，トランザクションアクセスモードを読み書き可能モードにします。これによって，この AP が実行する SQL 文はプライマリノードでだけ実行されます。

16.3.7　クライアント定義の集中管理機能を使用する場合の環境設定

クライアント定義の集中管理機能を使用する場合，クライアント管理定義ファイルおよびこの機能で使用するクライアント定義ファイルを，プライマリノードおよびセカンダリノードで作成してください（プライマリノードおよびセカンダリノードで同じファイルを作成してください)。プライマリノードとセカンダリノードでファイルの内容に不一致があると，マルチノード構成の HADB サーバを開始できません。

メモ

- プライマリノードの切り替えが発生した場合に，切り替えが発生する前と同じ状態でクライアント定義の集中管理機能を継続して使用するために，プライマリノードおよびセカンダリノードで同じファイルを作成する必要があります。
- ワーカーノードには，クライアント管理定義ファイル，およびこの機能で使用するクライアント定義ファイルを作成する必要はありません。

また，プライマリノードのクライアント管理定義ファイルまたはこの機能で使用するクライアント定義ファイルの内容を変更した場合は，全セカンダリノードのファイルの内容も変更してください。プライマリノードとセカンダリノードでファイルの内容に不一致があると，--update オプションを指定した adbclientdefmang コマンドがエラーになります。

クライアント定義の集中管理機能については，「2.13　クライアント定義の集中管理機能」を参照してください。

16.3.8　データベースの作成

マルチノード機能を使用する場合のデータベースの作成手順を説明します。

(1)　データベース作成の手順

データベースの作成手順の流れを次に示します。

1. ノード切り替え対象のファイルシステムを初期化する
2. プライマリノードでデータベースを作成する
3. セカンダリノードおよびワーカーノードで DB ディレクトリを作成する

(2)　ノード切り替え対象のファイルシステムの初期化

OS のコマンドを実行して，システムディレクトリ用のファイルシステムと，同義語辞書ファイルを格納するファイルシステムを初期化します。

システムディレクトリ用のファイルシステムの初期化

システムディレクトリ用の LV として作成した/dev/vg_hadb/hadb_sys を，ext4 のファイルシステムで初期化する場合の実行例を次に示します。

コマンドの実行例

```
mkfs -t ext4 /dev/vg_hadb/hadb_sys
```

同義語辞書ファイルを格納するファイルシステムの初期化

同義語辞書ファイルを格納する LV として作成した/dev/vg_hadb02/hadb_syndict を，ext4 のファイルシステムで初期化する場合の実行例を次に示します。

コマンドの実行例

```
mkfs -t ext4 /dev/vg_hadb02/hadb_syndict
```

同義語検索を行っていない場合は，同義語辞書ファイルを格納するファイルシステムの初期化は必要ありません。

(3)　プライマリノードでのデータベースの作成

プライマリノードでデータベースを作成します。

まず，OS のmount コマンドをプライマリノードで実行し，$DBDIR/ADBSYS にシステムディレクトリ用のファイルシステムをマウントします。

次に，adbinit コマンドをプライマリノードで実行し，データベースを作成します。adbinit コマンドの初期設定オプションの指定例を次に示します。

■初期設定オプションの指定例（プライマリノード）

- シングルパス構成の場合の例

```
set adb_init_dbarea_initialize = Y
set adb_init_wrk_blk_path = /dev/mapper/WRK001
set adb_init_mst_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b7d9fd
set adb_init_dic_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b828e9
set adb_init_stbl_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b87793
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b8c6d3
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b9160f
```

- マルチパスソフトウェア（DMMP）による冗長化構成の場合の例

```
set adb_init_dbarea_initialize = Y
set adb_init_wrk_blk_path = /dev/mapper/WRK001
set adb_init_mst_blk_path = /dev/mapper/mpath12
set adb_init_dic_blk_path = /dev/mapper/mpath11
set adb_init_stbl_blk_path = /dev/mapper/mpath13
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/mapper/mpath14
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/mapper/mpath15
```

最後に，adbinit コマンドの実行が完了したら，システムディレクトリ用のファイルシステムを$DBDIR/ADBSYS からアンマウントします。

(4)　セカンダリノードおよびワーカーノードでの DB ディレクトリの作成

adbinit コマンドをセカンダリノードおよびワーカーノードで実行します。プライマリノードで指定した初期設定オプションとは，次の点が異なります。

- adb_init_dbarea_initialize にN を指定して，DB ディレクトリの枠組みだけを作成します。
- adb_init_wrk_blk_path に，ノードごとに割り当てる作業表用 DB エリアファイルのブロックスペシャルファイル名を指定します。

■初期設定オプションの指定例（セカンダリノードhadb02）

- シングルパス構成の場合の例

```
# DBディレクトリの枠組みだけを作成する
set adb_init_dbarea_initialize = N
# WRK002 : hadb02の作業表用DBエリアを構成するDBエリアディスク
set adb_init_wrk_blk_path = /dev/mapper/WRK002
set adb_init_mst_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b7d9fd
set adb_init_dic_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b828e9
set adb_init_stbl_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b87793
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b8c6d3
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b9160f
```

- マルチパスソフトウェア（DMMP）による冗長化構成の場合の例

```
# DBディレクトリの枠組みだけを作成する
set adb_init_dbarea_initialize = N
# WRK002 : hadb02の作業表用DBエリアを構成するDBエリアディスク
set adb_init_wrk_blk_path = /dev/mapper/WRK002
set adb_init_mst_blk_path = /dev/mapper/mpath12
set adb_init_dic_blk_path = /dev/mapper/mpath11
set adb_init_stbl_blk_path = /dev/mapper/mpath13
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/mapper/mpath14
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/mapper/mpath15
```

■初期設定オプションの指定例（セカンダリノードhadb03）

- シングルパス構成の場合の例

```
# DBディレクトリの枠組みだけを作成する
set adb_init_dbarea_initialize = N
# WRK003 : hadb03の作業表用DBエリアを構成するDBエリアディスク
set adb_init_wrk_blk_path = /dev/mapper/WRK003
set adb_init_mst_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b7d9fd
set adb_init_dic_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b828e9
set adb_init_stbl_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b87793
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b8c6d3
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b9160f
```

- マルチパスソフトウェア（DMMP）による冗長化構成の場合の例

```
# DBディレクトリの枠組みだけを作成する
set adb_init_dbarea_initialize = N
# WRK003 : hadb03の作業表用DBエリアを構成するDBエリアディスク
set adb_init_wrk_blk_path = /dev/mapper/WRK003
set adb_init_mst_blk_path = /dev/mapper/mpath12
set adb_init_dic_blk_path = /dev/mapper/mpath11
set adb_init_stbl_blk_path = /dev/mapper/mpath13
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/mapper/mpath14
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/mapper/mpath15
```

■初期設定オプションの指定例（ワーカーノードhadb04）

- シングルパス構成の場合の例

```
# DBディレクトリの枠組みだけを作成する
set adb_init_dbarea_initialize = N
# WRK004 : hadb04の作業表用DBエリアを構成するDBエリアディスク
set adb_init_wrk_blk_path = /dev/mapper/WRK004
set adb_init_mst_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b7d9fd
set adb_init_dic_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b828e9
set adb_init_stbl_blk_path = /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b87793
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v  /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b8c6d3
```

```
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v  /dev/disk/by-id/wwn-0x600605b0041db9c016ea34c3b1b9160f
```

- マルチパスソフトウェア（DMMP）による冗長化構成の場合の例

```
# DBディレクトリの枠組みだけを作成する
set adb_init_dbarea_initialize = N
# WRK004 : hadb04の作業表用DBエリアを構成するDBエリアディスク
set adb_init_wrk_blk_path = /dev/mapper/WRK004
set adb_init_mst_blk_path = /dev/mapper/mpath12
set adb_init_dic_blk_path = /dev/mapper/mpath11
set adb_init_stbl_blk_path = /dev/mapper/mpath13
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v  /dev/mapper/mpath14
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v  /dev/mapper/mpath15
```

16.4　マルチノード構成の HADB サーバの開始と終了

ここでは，マルチノード構成の HADB サーバの開始および終了方法について説明します。

運用する際には，あわせて「10.2　HADB サーバの開始・終了と稼働モード」も参照してください。

16.4.1　マルチノード構成の HADB サーバの開始方法

(1)　マルチノード構成の HADB サーバの開始手順

マルチノード構成の HADB サーバの開始手順を次に示します。

重要

HA モニタの監視が停止していることを確認してから，マルチノード構成の HADB サーバを開始してください。誤って，adbstart コマンドを実行する前に HA モニタのmonbegin コマンドを実行していた場合は，monend コマンドを実行して，HA モニタの監視および起動中の HADB サーバを停止してください。そのあとで，次に示す正しい開始手順に従って，マルチノード構成の HADB サーバを開始してください。

手順

1. プライマリノードおよび全セカンダリノードで，HA モニタが起動していることを確認する
   確認方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの運用』を参照してください。
2. プライマリノードおよび全セカンダリノードで，adbstart コマンドを実行する
   adbstart コマンドを実行すると，HA モニタのmonbegin コマンドの実行を要求するKFAA91109-I メッセージが出力されます。
3. プライマリノードおよび全セカンダリノードで，HA モニタのmonbegin コマンドを実行する
   KFAA91109-I メッセージの出力を確認したあとに，別のターミナル（コマンド入力画面）から，monbegin コマンドを実行してください。

   重要

   - adbstart コマンドが終了する前に，monbegin コマンドを実行してください。
   - プライマリノードおよび全セカンダリノードのadbstart コマンドが正常終了すると，マルチノード構成の HADB サーバの開始処理が完了します。adbstart コマンドのリターンコードが0 または4 となっていることを，プライマリノードおよび全セカンダリノードで確認してください。確認後，次の手順に進んでください。
4. 全ワーカーノードでadbstart コマンドを実行する

ワーカーノードの HADB サーバの開始処理が完了すると，マルチノード構成にワーカーノードが追加されます。

> **メモ**
>
> ワーカーノードは HA モニタの管理対象外のため，HA モニタのmonbegin コマンドを実行する必要はありません。

(2) マルチノード構成の HADB サーバの開始モード

マルチノード構成の HADB サーバには，次の開始モードがあります。

表 16-5 マルチノード構成の HADB サーバの開始モード

項番	開始モード	実行するコマンド	説明	前回の終了形態
1	正常開始	adbstart	通常の開始モードです。 正常開始では，データベースの回復処理は行いません。	プライマリノードの HADB サーバが正常終了した場合
2	再開始		前回の終了モードが右記の場合，自動的に再開始となります。 再開始では，プライマリノードの HADB サーバがシステムログを使用してデータベースの回復処理を行います。	プライマリノードだけが残った状態のときに，そのプライマリノードの HADB サーバが異常終了または強制終了した場合

(3) マルチノード構成の HADB サーバの開始処理完了の確認

adbstart コマンドのリターンコードが0 または4 となっていることを，プライマリノードおよび全セカンダリノードで確認してください。

(4) マルチノード構成の HADB サーバ開始時の留意事項

- プライマリノードおよび全セカンダリノードの HADB サーバの開始処理が完了するまで，マルチノード構成の HADB サーバの開始処理は完了しません。マルチノード構成の HADB サーバの開始処理が完了しない場合は，プライマリノードまたはセカンダリノードでエラーや障害などが発生していないかを確認してください。
- マルチノード構成の HADB サーバの開始処理が完了したあとに，ワーカーノードの HADB サーバを開始してマルチノード構成にワーカーノードを追加します。エラーなどによってワーカーノードの HADB サーバが開始できない場合でも，開始処理が完了しているプライマリノード，セカンダリノード，そのほかのワーカーノードでマルチノード構成の HADB サーバを稼働して業務を開始することができます。

16.4.2　マルチノード構成の HADB サーバの終了方法

(1)　マルチノード構成の HADB サーバの終了手順

(a)　マルチノード構成の HADB サーバを正常終了させる場合

マルチノード構成の HADB サーバの正常終了手順を次に示します。

手順

1. プライマリノードでadbstop コマンドを実行する
   プライマリノードでadbstop コマンドを実行すると，全ノードの HADB サーバが正常終了します。
2. プライマリノードで HA モニタのmonend コマンドを実行する
   全ノードの HADB サーバが正常終了したことを確認してから，プライマリノードで HA モニタのmonend コマンドを実行してください。

> **メモ**
>
> 上記の手順で，ワーカーノードを含めた全ノードの HADB サーバが終了します。

(b)　マルチノード構成の HADB サーバを強制終了させる場合

マルチノード構成の HADB サーバの強制終了手順を次に示します。

手順

1. 全セカンダリノードおよび全ワーカーノードで，adbstop --force コマンドを実行する
   全セカンダリノードおよび全ワーカーノードで，adbstop --force コマンドが終了したことを確認してください。
2. プライマリノードでadbstop --force コマンドを実行する
   adbstop --force コマンドが終了したことを確認してください。
3. プライマリノードで HA モニタのmonend コマンドを実行する

(2)　特定のノードの HADB サーバの終了手順

(a)　プライマリノードの HADB サーバだけを正常終了させる場合

系のリセットを使用している場合と，共有ディスクの SCSI リザーブを使用している場合とで，プライマリノードの HADB サーバだけを正常終了させる手順が異なります。

- 系のリセットを使用している場合

プライマリノードでHAモニタのmonswapコマンドを実行すると，プライマリノードのHADBサーバだけが正常終了します。このとき，プライマリノードの切り替えが発生し，優先度の最も高いセカンダリノードがプライマリノードになります。

なお，monswapコマンドを実行すると，プライマリノードで実行中のトランザクションはすべてキャンセルされ，そのトランザクションを実行していたコネクションはすべて終了します。また，トランザクション実行前のコネクションもすべて終了します。コネクションの終了処理が完了したあとに，プライマリノードのHADBサーバの終了処理が開始されます。

重要

adbstop --nodeコマンドで，プライマリノードのHADBサーバを誤って終了してしまった場合も，そのノードでmonswapコマンドを実行してください。

- 共有ディスクのSCSIリザーブを使用している場合

  手順

  1. プライマリノードでadbstop --cancel --nodeコマンドを実行する
  2. プライマリノードでHAモニタのmonswapコマンドを実行する

  上記の手順で，プライマリノードのHADBサーバだけが正常終了します。このとき，プライマリノードの切り替えが発生し，優先度の最も高いセカンダリノードがプライマリノードになります。

(b) プライマリノードのHADBサーバだけを強制終了させる場合

プライマリノードでadbstop --forceコマンドを実行すると，プライマリノードのHADBサーバだけが強制終了します。このとき，プライマリノードの切り替えが発生し，優先度の最も高いセカンダリノードがプライマリノードになります。

メモ

強制終了したプライマリノードで実行されていた更新処理に対する回復処理（ロールバック）が，新しくプライマリノードになるセカンダリノードで実行されます。

(c) セカンダリノードのHADBサーバだけを正常終了させる場合

セカンダリノードでadbstop --nodeコマンドを実行すると，コマンドを実行したセカンダリノードのHADBサーバが正常終了します。

(d) セカンダリノードのHADBサーバだけを強制終了させる場合

セカンダリノードでadbstop --forceコマンドを実行すると，コマンドを実行したセカンダリノードのHADBサーバが強制終了します。

(e)　ワーカーノードの HADB サーバだけを正常終了させる場合

ワーカーノードでadbstop --node コマンドを実行すると，コマンドを実行したワーカーノードの HADB サーバが正常終了します。

(f)　ワーカーノードの HADB サーバだけを強制終了させる場合

ワーカーノードでadbstop --force コマンドを実行すると，コマンドを実行したワーカーノードの HADB サーバが強制終了します。

(3)　マルチノード構成の HADB サーバの終了形態

マルチノード構成の HADB サーバの終了形態を次の表に示します。

表 16-6　マルチノード構成の HADB サーバの終了形態

項番	終了形態	実行するコマンド	説明
1	マルチノード構成の HADB サーバの正常終了	adbstop	通常の終了形態です。プライマリノードでadbstop コマンドを実行すると，マルチノード構成の HADB サーバが正常終了します。 なお，正常終了には 4 種類あり，指定するオプションによって終了処理が異なります。詳細については，「10.2.2　HADB サーバの終了方法」の「(2)　オプションの指定と正常終了の種類」を参照してください。
2	特定のノードの HADB サーバの正常終了	• プライマリノードの場合：monswap※ • セカンダリノードまたはワーカーノードの場合：adbstop --node	コマンドを実行したノードの HADB サーバだけを正常終了します。 • プライマリノードの HADB サーバを正常終了した場合 そのプライマリノードはマルチノード構成から切り離され，セカンダリノードの 1 つがプライマリノードに切り替わります。 • セカンダリノードの HADB サーバを正常終了した場合 そのセカンダリノードは，マルチノード構成から切り離されます。 • ワーカーノードの HADB サーバを正常終了した場合 そのワーカーノードは，マルチノード構成から削除されます。
3	特定のノードの HADB サーバの強制終了	adbstop --force	処理中のトランザクションの完了を待たないで，adbstop コマンドを実行したノードの HADB サーバを直ちに終了します。 • プライマリノードの HADB サーバを強制終了した場合 そのプライマリノードはマルチノード構成から切り離され，セカンダリノードの 1 つがプライマリノードに切り替わります。このとき，プライマリノードで実行されていた更新処理に対する回復処理（ロールバック）が，新しくプライマリノードになるセカンダリノードで実行されます。 なお，セカンダリノードがない状態のときに，プライマリノードの HADB サーバを強制終了した場合，マルチノー

項番	終了形態	実行するコマンド	説明
			ド構成の HADB サーバの再開始時に，更新処理に対する回復処理（ロールバック）が実行されます。 • セカンダリノードの HADB サーバを強制終了した場合 そのセカンダリノードは，マルチノード構成から切り離されます。このとき，プライマリノードでトランザクションのロールバックが実行されます。 • ワーカーノードの HADB サーバを強制終了した場合 そのワーカーノードは，マルチノード構成から削除されます。このとき，プライマリノードでトランザクションのロールバックが実行されます。
4	特定のノードの HADB サーバの異常終了	－	• プライマリノードの HADB サーバが異常終了した場合 異常終了したプライマリノードはマルチノード構成から切り離され，セカンダリノードの 1 つがプライマリノードに切り替わります。このとき，プライマリノードで実行されていた更新処理に対する回復処理（ロールバック）が，新しくプライマリノードになるセカンダリノードで実行されます。 なお，セカンダリノードがない状態のときに，プライマリノードの HADB サーバが異常終了した場合は，マルチノード構成の HADB サーバの再開始時に，更新処理に対する回復処理（ロールバック）が実行されます。 • セカンダリノードの HADB サーバが異常終了した場合 異常終了したセカンダリノードはマルチノード構成から切り離されます。このとき，プライマリノードでトランザクションのロールバックが実行されます。 • ワーカーノードの HADB サーバが異常終了した場合 異常終了したワーカーノードはマルチノード構成から削除されます。このとき，プライマリノードでトランザクションのロールバックが実行されます。

（凡例）

－：該当しません。

注※

系のリセットを使用している場合に実行するコマンドです。

共有ディスクの SCSI リザーブを使用している場合は，adbstop --cancel --node コマンドを実行したあとに，monswap コマンドを実行します。

16.4.3　マルチノード機能使用時の HADB サーバの稼働モード

- マルチノード構成の HADB サーバを開始したときの稼働モード

プライマリノードの HADB サーバの場合は，adbstart コマンドにオプションを指定することで，HADB サーバ開始後の稼働モードを指定できます。セカンダリノードおよびワーカーノードの HADB サーバ

の場合は，稼働モードが常に静止モードとなるため，adbstart コマンドにオプションを指定しても稼働モードを変更できません。

- **稼働モードの変更**

  プライマリノードの場合，adbchgsrvmode コマンドを使用して HADB サーバの稼働モードを変更できます。

  セカンダリノードおよびワーカーノードの場合，HADB サーバの稼働モードは，常に静止モードとなります。そのため，adbchgsrvmode コマンドは実行できません。

- **稼働モードの確認**

  稼働モードを確認する場合は，稼働モードを確認したいノードでadbls -d srv コマンドを実行します。詳細については，「10.2.3　HADB サーバの稼働モード」を参照してください。

- **プライマリノードの切り替えが発生したときの稼働モード**

  プライマリノードの切り替えが発生した場合，稼働モードも新しいプライマリノードに引き継がれます。

16.5　AP の運用（マルチノード機能の使用時）

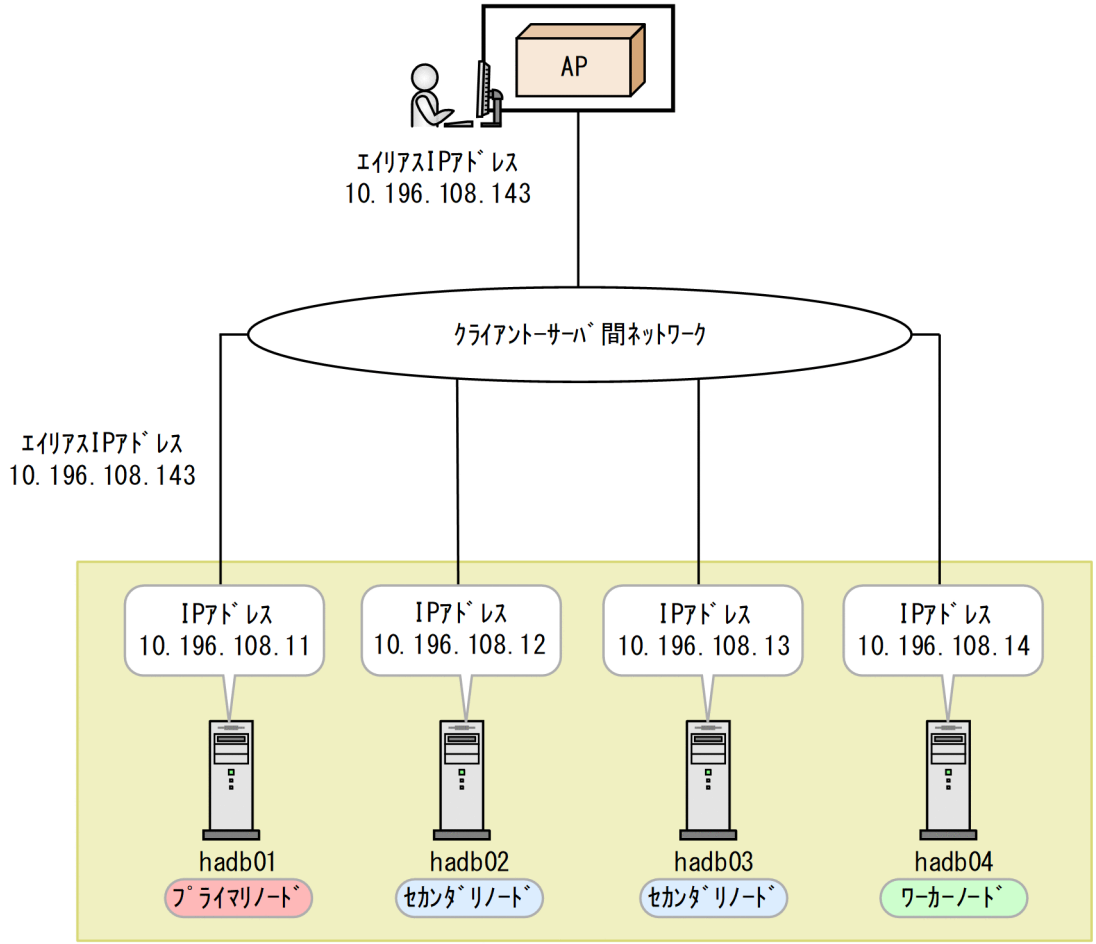
ここでは，マルチノード機能を使用したときの，AP の運用に関する注意点について説明します。AP の作成方法などについては，マニュアル『HADB　AP 開発ガイド』を参照してください。

16.5.1　HADB サーバへの接続

マルチノード機能を使用する場合，AP はエイリアス IP アドレスを使用して，プライマリノードの HADB サーバに接続するようにしてください。エイリアス IP アドレスの設定方法については，マニュアル『HA モニタ Linux(R)(x86)編』を参照してください。

マルチノード構成の HADB サーバと AP の構成例を次の図に示します。

図 16-2　マルチノード構成の HADB サーバと AP の構成例



16.5.2　マルチノード機能使用時の最大同時接続数

マルチノード機能使用時の最大同時接続数は，サーバ定義adb_sys_max_users オペランドの指定値になります（adb_sys_max_users オペランドの指定値は，全ノードの HADB サーバで同じ値にする必要があります)。

最大同時接続数を超えた場合，マルチノード構成の HADB サーバに AP が接続することはできません（エラーとなります)。

サーバ定義adb_sys_max_users オペランドについては，「7.2.1　システム構成に関するオペランド（set 形式)」の adb_sys_max_users オペランドを参照してください。

16.5.3　トランザクションを実行するノード

トランザクションを実行するノードは，トランザクションアクセスモードおよびトランザクション隔離性水準の指定によって決まります。トランザクションアクセスモードおよびトランザクション隔離性水準と，トランザクションを実行するノードの関係を次の表に示します。

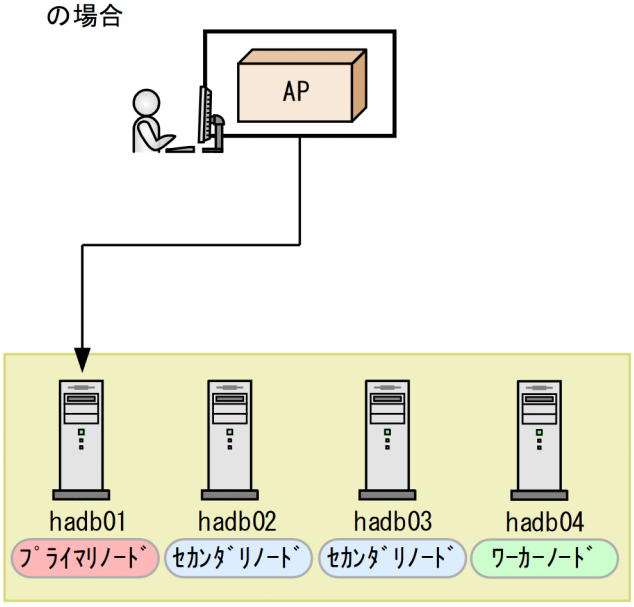
表 16-7　トランザクションアクセスモードおよびトランザクション隔離性水準と，トランザクションを実行するノードの関係

項番	トランザクションアクセスモードの指定	トランザクション隔離性水準の指定	トランザクションを実行するノード
1	読み書き可能モード	READ COMMITTED	プライマリノード
2		REPEATABLE READ	
3	読み取り専用モード	READ COMMITTED	全ノードのうちどれかのノード
4		REPEATABLE READ	プライマリノード

トランザクションを実行するノードを次の図に示します。

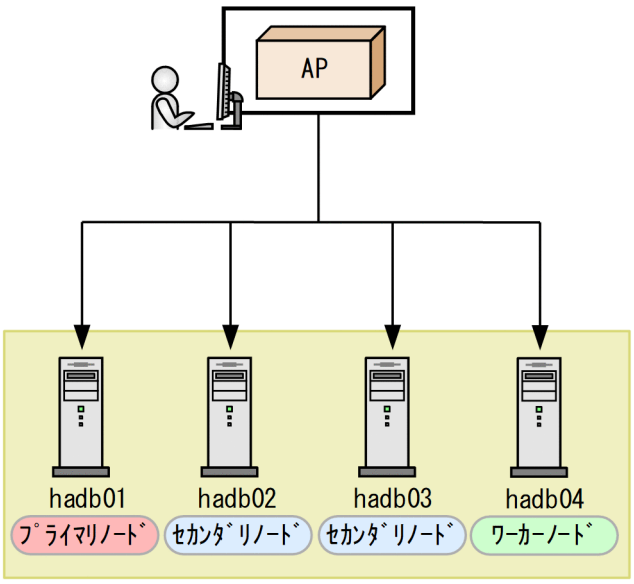
図 16-3　トランザクションを実行するノード

■「読み書き可能ﾓｰﾄﾞ」の場合
■「読み取り専用ﾓｰﾄﾞ」かつ「REPEATABLE READ」の場合



ﾄﾗﾝｻﾞｸｼｮﾝを実行するﾉｰﾄﾞは，ﾌﾟﾗｲﾏﾘﾉｰﾄﾞになります。

■「読み取り専用ﾓｰﾄﾞ」かつ「READ COMMITTED」の場合



ﾄﾗﾝｻﾞｸｼｮﾝを実行するﾉｰﾄﾞは，どれかのﾉｰﾄﾞになります。

検索系 SQL だけを実行する AP の場合，次の設定をすることで，セカンダリノードおよびワーカーノードのリソースを有効に活用できます。

- トランザクションアクセスモードを読み取り専用モードにする
- トランザクション隔離性水準をREAD COMMITTED にする

トランザクションアクセスモードおよびトランザクション隔離性水準の指定方法を次の表に示します。

表 16-8　トランザクションアクセスモードおよびトランザクション隔離性水準の指定方法

項番	AP の種別	トランザクションアクセスモードの指定方法	トランザクション隔離性水準の指定方法
1	JDBC ドライバを使用した AP	• システムプロパティの adb_clt_trn_access_mode • ユーザプロパティの adb_clt_trn_access_mode • 接続用の URL の adb_clt_trn_access_mode • Connection インタフェースの setReadOnly メソッド	• システムプロパティの adb_clt_trn_iso_lv • ユーザプロパティの adb_clt_trn_iso_lv • 接続用の URL のadb_clt_trn_iso_lv • Connection インタフェースの setTransactionIsolation メソッド
2	ODBC ドライバを使用した AP	• クライアント定義の adb_clt_trn_access_mode オペランド • SQLSetConnectAttr()または SQLSetConnectAttrW()	• クライアント定義の adb_clt_trn_iso_lv オペランド • SQLSetConnectAttr()または SQLSetConnectAttrW()

項番	APの種別	トランザクションアクセスモードの指定方法	トランザクション隔離性水準の指定方法
3	CLI関数を使用したAP	• クライアント定義のadb_clt_trn_access_modeオペランド • a_rdb_SQLSetConnectAttr()	• クライアント定義のadb_clt_trn_iso_lvオペランド • a_rdb_SQLSetConnectAttr()

指定方法の詳細については，マニュアル『HADB　AP開発ガイド』を参照してください。

また，トランザクションを実行するノードは，次に示す要素などを考慮して，プライマリノードのHADBサーバが自動的に決定します。

- トランザクションアクセスモード
- トランザクション隔離性水準
- 各ノードの負荷

そのため，トランザクションごとに，実行するノードが変わることがあります。ただし，ホールダブルカーソルでカーソルをオープン中の場合は，実行するノードは変わりません。

重要

読み書き可能モードのトランザクション，または一部のコマンドが実行中のときは，読み取り専用モードのトランザクションもプライマリノードで実行されます（セカンダリノードおよびワーカーノードを使用しません）。そのため，セカンダリノードおよびワーカーノードのリソースを有効に活用するために，読み書き可能モードのトランザクションはできる限り細かい頻度で決着させるようにしてください。

なお，ここでいう一部のコマンドについては，「2.19.2　トランザクションおよびコマンドが実行されるノード」の「(4)　トランザクションとコマンドの同時実行についての制限」を参照してください。

■SQLパラレル実行機能を使用する場合

- SQLパラレル実行機能を使用する場合も，トランザクションアクセスモードを読み取り専用モードにし，トランザクション隔離性水準をREAD COMMITTEDにする必要があります。この指定をしないと，検索系SQLにSQLパラレル実行機能が適用されません。
- プライマリノードのHADBサーバが，各ノードの負荷などを考慮してSQLサブノードを決定します。したがって，SQLパラレル実行機能が適用された検索系SQLを実行するたびにSQLサブノードが変わることがあります。ただし，SQLパラレル実行機能が適用された検索系SQLを実行しているホールダブルカーソルがオープン中の場合は，トランザクションを決着しても，SQLメインノードおよびSQLサブノードは変わりません。

16.5.4　ノード障害が発生した場合

ノード障害が発生した場合，障害が発生したノードで実行中のトランザクションはロールバックします。

プライマリノードでノード障害が発生した場合は，プライマリノードで実行されていたデータベースの更新処理に対する回復処理が，新たにプライマリノードになるセカンダリノードで実行されます。この回復処理中に，回復処理対象のページにセカンダリノードまたはワーカーノードで実行中の AP がアクセスした場合，その AP の処理がエラーになることがあります。このとき，KFAA31733-E メッセージが出力されます。この場合，メッセージに記載されている対処方法に従ってください。

また，障害が発生したノードでトランザクションを実行していた AP は，マルチノード構成の HADB サーバから切断されます。トランザクションを再実行する際は，改めてマルチノード構成の HADB サーバに接続してください。

なお，プライマリノードでノード障害が発生した場合，障害が発生した時点でどのノードでもトランザクションが実行されていない AP も，マルチノード構成の HADB サーバから切断されます。

■SQL パラレル実行機能を使用している場合

SQL パラレル実行機能を適用した検索系 SQL を実行していたノードで障害が発生したときの HADB サーバの処理を説明します。

- SQL メインノードで障害が発生した場合

  トランザクションはロールバックし，トランザクションを実行していた AP はマルチノード構成の HADB サーバから切断されます。トランザクションを再実行する際は，改めてマルチノード構成の HADB サーバに接続してください。

- SQL サブノードで障害が発生した場合

  検索系 SQL の実行がエラーとなり，トランザクションはロールバックします。ただし，トランザクションを実行していた AP はマルチノード構成の HADB サーバからは切断されません。必要に応じてトランザクションを再実行してください。

16.6　排他の運用（マルチノード機能の使用時）

マルチノード構成の HADB サーバでは，「2.10.2　排他資源」にある排他資源を，全ノードの HADB サーバで共有しています。そのため，あるノードの HADB サーバで実行中のトランザクションが，特定の排他資源を占有モードで確保した場合，排他資源が解放されるまで，全ノードの HADB サーバで，ほかのトランザクションはその排他資源を確保できません。

上記以外の排他制御については，「2.10　排他制御」を参照してください。

16.7　コマンドによるプライマリノードの切り替え

コマンドを実行して，プライマリノードを切り替えることができます。系のリセットを使用している場合と，共有ディスクの SCSI リザーブを使用している場合とで手順が異なります。

- 系のリセットを使用している場合

  HA モニタのmonswap コマンドをプライマリノードで実行すると，プライマリノードを切り替えることができます。

- 共有ディスクの SCSI リザーブを使用している場合

  次の手順で，プライマリノードを切り替えることができます。

  1. adbstop --cancel --node コマンドをプライマリノードで実行する
  2. HA モニタのmonswap コマンドをプライマリノードで実行する

新しくプライマリノードになるセカンダリノードは，HA モニタのservers ファイルのinitial およびstandbypri オペランドの指定によって決まります。initial およびstandbypri オペランドについては，「16.3.4　HA モニタの環境設定」の「(3)　servers ファイルの指定」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバ対応の環境設定（servers)』を参照してください。

なお，コマンドを実行したノードは，マルチノード構成から切り離されます。

メモ

- monswap コマンドについては，マニュアル『HA モニタ Linux(R)(x86)編』の『コマンド』を参照してください。
- コマンドによるプライマリノードの切り替えは，HA モニタの計画系切り替え機能を使用しています。

16.8　データベースのバックアップ（マルチノード機能の使用時）

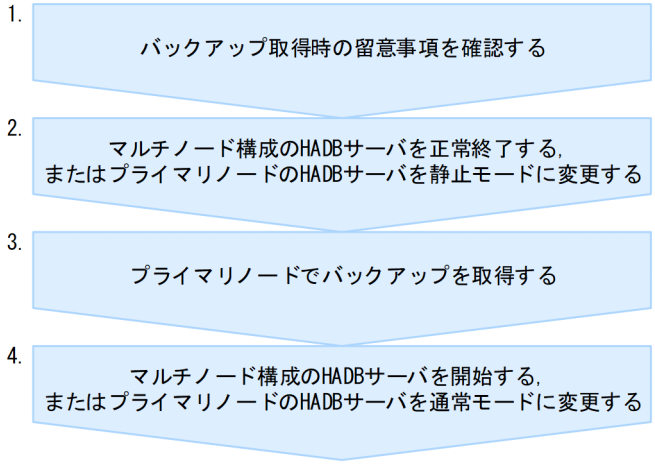
ここでは，マルチノード機能を使用している場合のバックアップの取得方法，およびバックアップからのデータベースの回復方法について説明します。

マルチノード機能を使用していない場合のバックアップの取得方法，およびバックアップからのデータベースの回復方法については，「10.3　データベースのバックアップ」を参照してください。

16.8.1　バックアップの取得方法

データベースのバックアップを取得する手順を次に示します。

図 16-4　バックアップを取得する手順（マルチノード機能を使用している場合）



各手順の詳細を以降で説明します。

(1)　バックアップ取得時の留意事項を確認する

「10.3.1　バックアップの取得方法」を参照して，バックアップ取得時の留意事項を確認してください。

(2)　マルチノード構成の HADB サーバを正常終了する，またはプライマリノードの HADB サーバを静止モードに変更する

次のどちらかの操作を実行してください。

- adbstop コマンドおよび HA モニタのmonend コマンドを実行して，マルチノード構成の HADB サーバを正常終了する
- プライマリノードでadbchgsrvmode コマンドを実行して，プライマリノードの HADB サーバを静止モードに変更する

上記のどちらかの操作を実行しないと，データベースを回復する際にデータベースの整合性が取れなくなるおそれがあります。

各ノードの HADB サーバの状態を確認する場合は，状態を確認したいノードでadbls -d srv コマンドを実行してください。

(3) プライマリノードでバックアップを取得する

プライマリノードで，バックアップを取得するファイルの一覧を次の表に示します。

ファイルがシンボリックリンクの場合は，リンク先のファイル，またはブロックスペシャルファイルのコピーが必要です。

なお，マルチノード構成の HADB サーバを正常終了している場合は，コピーを実行する前に，システムディレクトリが$DBDIR/ADBSYS にマウントされていることを確認してください。マウントされていない場合は，OS のmount コマンドを実行して，システムディレクトリを$DBDIR/ADBSYS にマウントしてください。

また，バックアップファイルはスパースファイルにしないでください。

表 16-9 バックアップを取得するファイルおよびディレクトリの一覧

項番	バックアップを取得するファイル	ファイルの格納先	ファイルの種類
1	コマンドステータスファイル	$DBDIR/ADBSYS/ADBUTL	レギュラーファイル
2	システムログファイル	$DBDIR/ADBSYS/ADBSLG	レギュラーファイル
3	ステータスファイル	$DBDIR/ADBSYS/ADBSTS	レギュラーファイル
4	マスタディレクトリ用 DB エリアファイル	$DBDIR/ADBMST	ブロックスペシャルファイル
5	ディクショナリ用 DB エリアファイル	$DBDIR/ADBDIC	ブロックスペシャルファイル
6	システム表用 DB エリアファイル	$DBDIR/ADBSTBL	ブロックスペシャルファイル
7	データ用 DB エリアを構成するすべてのファイル	$DBDIR/DBAREA※1	ブロックスペシャルファイル
8	アーカイブファイル	アーカイブディレクトリ※2	レギュラーファイル
9	同義語辞書ファイル	同義語辞書ファイルの格納ディレクトリ※3	レギュラーファイル

注※1

adbinit コマンドまたはadbmodarea コマンドで，ユーザが指定した名称になります。

注※2

CREATE TABLE 文のチャンクアーカイブ指定のARCHIVEDIR で指定したディレクトリのことです。

注※3

サーバ定義のadb_syndict_storage_path オペランドで指定したディレクトリのことです。

メモ

次に示すファイルについては，バックアップの取得は不要です。

- 作業表用 DB エリアファイル（$DBDIR/ADBWRK）
- 作業用ディレクトリ下（$DBDIR/ADBWORK）のファイル
- 障害情報（コアファイル）の出力先ディレクトリ下（$DBDIR/SPOOL）のファイル

  サーバ定義のadb_core_path オペランドを指定した場合は，adb_core_path オペランドに指定したディレクトリになります。

なお，これらのファイルも，上記の表に示すファイルと一緒にバックアップを取得して，バックアップから回復しても問題はありません。

メモ

監査証跡の出力先ディレクトリ（サーバ定義のadb_audit_log_path オペランドに指定したディレクトリ）下の監査証跡ファイルのバックアップを取得する必要はありません。監査証跡の出力先ディレクトリ下の監査証跡ファイルは，バックアップを取得するのではなく，監査証跡の保存先ディレクトリに移動してください。監査証跡ファイルの移動については，「12.3.1　監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）」を参照してください。

(4)　マルチノード構成の HADB サーバを開始する，またはプライマリノードの HADB サーバを通常モードに変更する

次のどちらかの操作を実行してください。

- バックアップの取得前に，adbstop コマンドでマルチノード構成の HADB サーバを正常終了した場合

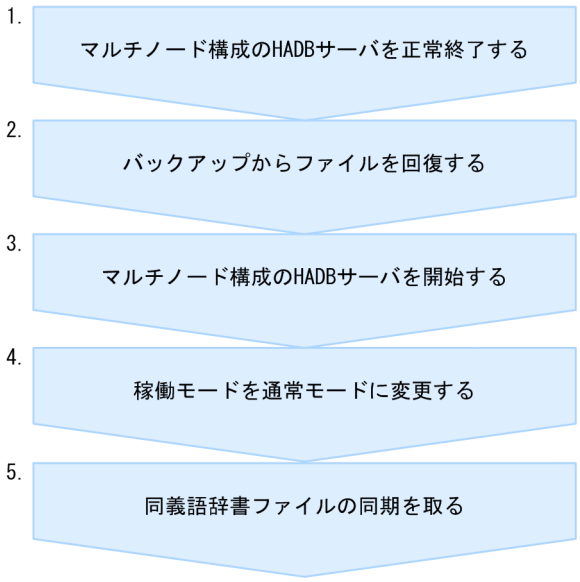
  adbstart コマンドおよび HA モニタのmonbegin コマンドを実行して，マルチノード構成の HADB サーバを開始してください。
- バックアップの取得前に，adbchgsrvmode コマンドでプライマリノードの HADB サーバを静止モードにした場合

  プライマリノードでadbchgsrvmode コマンドを実行して，プライマリノードの HADB サーバを通常モードに変更してください。

16.8.2　バックアップからの回復方法

バックアップからの回復手順を次に示します。

図 16-5　バックアップからの回復手順（マルチノード機能を使用している場合）



なお，バックアップからの回復を行う際は，「10.3.2　バックアップからの回復方法」の内容も参照してください。

各手順の詳細を以降で説明します。

(1)　マルチノード構成の HADB サーバを正常終了する

adbstop コマンドおよび HA モニタのmonend コマンドを実行して，マルチノード構成の HADB サーバを正常終了してください。

マルチノード構成の HADB サーバを正常終了しないで，データベースの回復を実行した場合，データベースの内容が不整合となるおそれがあります。

(2)　バックアップからファイルを回復する

「16.8.1　バックアップの取得方法」の「(3)　プライマリノードでバックアップを取得する」で取得したバックアップからファイルを回復してください。

なお，バックアップからファイルを回復する前に，システムディレクトリが$DBDIR/ADBSYS にマウントされていることを確認してください。マウントされていない場合は，OS のmount コマンドを実行して，システムディレクトリを$DBDIR/ADBSYS にマウントしてください。

(3)　マルチノード構成の HADB サーバを開始する

adbstart コマンドおよび HA モニタのmonbegin コマンドを実行して，マルチノード構成の HADB サーバを開始してください。

(4) 稼働モードを通常モードに変更する

静止モードでバックアップを取得した場合，データベースを回復したあとに HADB サーバを開始すると，HADB サーバが静止モードのままとなっています。この場合，プライマリノードでadbchgsrvmode コマンドを実行して，プライマリノードの HADB サーバの稼働モードを通常モードに変更してください。

(5) 同義語辞書ファイルの同期を取る

adbsyndict コマンドを実行して，同義語辞書ファイルの同期を取ってください。同義語辞書ファイルの同期を取る方法については，「16.26.5 同義語辞書ファイルの同期を取る」を参照してください。

16.8.3 バックアップの運用例（OS コマンドを使用）

ここでは，OS のコマンドを使用したバックアップの運用例（フルバックアップを取得する運用例）について説明します。

(1) システムの構成例

4 つのノード（hadb01，hadb02，hadb03，hadb04）で構成されるマルチノード構成で，プライマリノードはhadb01 とします。DB ディレクトリ，アーカイブディレクトリ，同義語辞書ファイルの格納ディレクトリ，および監査証跡の出力先ディレクトリは，次の構成となっているとします。

■hadb01 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル（サイズは512MB）
 ├ADBMST：ブロックスペシャルファイル（サイズは10MB）
 ├ADBSTBL：ブロックスペシャルファイル（サイズは512MB）
 ├ADBWORK：ローカルファイルシステム上のディレクトリ
 ├ADBWRK：ブロックスペシャルファイル（サイズは2GB）
 ├ADBUTBL01：ブロックスペシャルファイル（サイズは4GB）
 ├ADBUIDX01：ブロックスペシャルファイル（サイズは2GB）
 ├SPOOL：ローカルファイルシステム上のディレクトリ
 └ADBSYS：システムディレクトリ
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

  アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリの構成

```
/mnt/syndict
```

  同義語辞書ファイルの格納ディレクトリは，同義語検索を行っている場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■hadb02 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル
 ├ADBMST：ブロックスペシャルファイル
 ├ADBSTBL：ブロックスペシャルファイル
 ├ADBWORK：ローカルファイルシステム上のディレクトリ
 ├ADBWRK：ブロックスペシャルファイル
 ├ADBUTBL01：ブロックスペシャルファイル
 ├ADBUIDX01：ブロックスペシャルファイル
 ├SPOOL：ローカルファイルシステム上のディレクトリ
 └ADBSYS：システムディレクトリのファイルシステムのマウントポイント
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリのファイルシステムのマウントポイント

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索を行っている場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■hadb03 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル
 ├ADBMST：ブロックスペシャルファイル
 ├ADBSTBL：ブロックスペシャルファイル
 ├ADBWORK：ローカルファイルシステム上のディレクトリ
 ├ADBWRK：ブロックスペシャルファイル
 ├ADBUTBL01：ブロックスペシャルファイル
 ├ADBUIDX01：ブロックスペシャルファイル
 ├SPOOL：ローカルファイルシステム上のディレクトリ
 └ADBSYS：システムディレクトリのファイルシステムのマウントポイント
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリのファイルシステムのマウントポイント

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索を行っている場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■hadb04 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル
 ├ADBMST：ブロックスペシャルファイル
 ├ADBSTBL：ブロックスペシャルファイル
 ├ADBWORK：ローカルファイルシステム上のディレクトリ
 ├ADBWRK：ブロックスペシャルファイル
 ├ADBUTBL01：ブロックスペシャルファイル
 ├ADBUIDX01：ブロックスペシャルファイル
 ├SPOOL：ローカルファイルシステム上のディレクトリ
 └ADBSYS：ローカルファイルシステム上のシステムディレクトリ
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

(2) バックアップの取得

OS のコマンドを使用して，バックアップを取得する手順を説明します。

手順

1. マルチノード構成の HADB サーバを終了する※1

```
adbstop
KFAA90000-I adbstop processing started.
KFAA91154-I The HADB system was terminated normally.
KFAA90001-I adbstop processing ended. (return code = 0)

monend
```

プライマリノード（hadb01）でadbstop コマンドを実行し，全ノードの HADB サーバが正常終了したあとに，HA モニタのmonend コマンドを実行してください。

2. OS のコマンドでバックアップを取得する

OS のcp コマンドおよびdd コマンドをプライマリノード（hadb01）で実行し，次のディレクトリのバックアップを取得します。

- DB ディレクトリのバックアップの取得

```
cp -r /HADB/db/ADBSYS /HADB_bkup/db/ADBSYS
dd if=/HADB/db/ADBMST of=/HADB_bkup/db/ADBMST bs=524288
dd if=/HADB/db/ADBDIC of=/HADB_bkup/db/ADBDIC bs=524288
dd if=/HADB/db/ADBSTBL of=/HADB_bkup/db/ADBSTBL bs=524288
dd if=/HADB/db/ADBUTBL01 of=/HADB_bkup/db/ADBUTBL01 bs=524288
dd if=/HADB/db/ADBUIDX01 of=/HADB_bkup/db/ADBUIDX01 bs=524288
```

バックアップの取得先ディレクトリは/HADB_bkup/db です。

システムディレクトリが$DBDIR/ADBSYS にマウントされていない場合は，OS のmount コマンドを実行してマウントしてください。

なお，ブロックスペシャルファイルのデータをバックアップする場合，ボリューム全体をコピーするため，実際の使用量よりもバックアップ容量が増えます。

- アーカイブディレクトリのバックアップの取得

```
cp -r /HADB/archive /HADB_bkup/archive
```

アーカイブマルチチャンク表を定義している場合は，アーカイブディレクトリのバックアップを取得する必要があります。

アーカイブディレクトリを NFS サーバ上に作成している場合は，アーカイブディレクトリのバックアップをセカンダリノードおよびワーカーノードで取得する必要はありません。

- 同義語辞書ファイルの格納ディレクトリのバックアップの取得

```
cp -r /mnt/syndict /HADB_bkup/syndict
```

同義語検索を行っている場合は，同義語辞書ファイルの格納ディレクトリのバックアップを取得する必要があります。

> **メモ**
>
> 監査証跡の出力先ディレクトリ下の監査証跡ファイルは，監査証跡の保存先ディレクトリに移動する運用のため，監査証跡の出力先ディレクトリのバックアップを取得する必要はありません。

3. マルチノード構成の HADB サーバを開始する※2

```
adbstart
```

プライマリノードおよび全セカンダリノード（hadb01～hadb03）で，adbstart コマンドおよび HA モニタのmonbegin コマンドを実行してください。adbstart コマンドの実行後，別ターミナル（コマンド入力画面）から，HA モニタのmonbegin コマンドを実行してください。

重要

adbstart コマンドが終了する前に，monbegin コマンドを実行してください。

```
monbegin
```

4. マルチノード構成にワーカーノードを追加する

```
adbstart
```

ワーカーノード（hadb04）で，adbstart コマンドを実行してください。

注※1

手順 1.は，次の方法で代替することもできます。

プライマリノードのhadb01 でadbchgsrvmode コマンドを実行し，プライマリノードの HADB サーバの稼働モードを静止モードに変更します。

```
adbchgsrvmode --quiescence
```

注※2

手順 1.で注※1 の方法を実施した場合は，プライマリノード（hadb01）でadbchgsrvmode コマンドを実行し，プライマリノードの HADB サーバの稼働モードを通常モードに変更してください。

```
adbchgsrvmode --normal
```

(3) バックアップからの回復

バックアップからの回復手順を説明します。

手順

1. マルチノード構成の HADB サーバを終了する

```
adbstop
KFAA90000-I adbstop processing started.
KFAA91154-I The HADB system was terminated normally.
KFAA90001-I adbstop processing ended. (return code = 0)

monend
```

プライマリノード（hadb01）でadbstop コマンドを実行し，マルチノード構成の HADB サーバが正常終了したあとに，HA モニタのmonend コマンドを実行してください。

2. OS のコマンドでデータベースを回復する

OS のcp コマンドおよびdd コマンドをプライマリノード（hadb01）で実行し，バックアップから次のディレクトリを回復します。

- DB ディレクトリの回復

```
cp -r /HADB_bkup/db/ADBSYS /HADB/db/ADBSYS
dd if=/HADB_bkup/db/ADBMST of=/HADB/db/ADBMST bs=524288
dd if=/HADB_bkup/db/ADBDIC of=/HADB/db/ADBDIC bs=524288
dd if=/HADB_bkup/db/ADBSTBL of=/HADB/db/ADBSTBL bs=524288
dd if=/HADB_bkup/db/ADBUTBL01 of=/HADB/db/ADBUTBL01 bs=524288
dd if=/HADB_bkup/db/ADBUIDX01 of=/HADB/db/ADBUIDX01 bs=524288
```

バックアップファイルの格納ディレクトリは，/HADB_bkup/db です。

システムディレクトリが$DBDIR/ADBSYS にマウントされていない場合は，OS のmount コマンドを実行してマウントしてください。

- アーカイブディレクトリの回復

```
rm -r /HADB/archive/*
cp -r /HADB_bkup/archive/* /HADB/archive
```

アーカイブマルチチャンク表を定義している場合は，アーカイブディレクトリを回復する必要があります。

アーカイブディレクトリを NFS サーバ上に作成している場合は，セカンダリノードおよびワーカーノードでアーカイブディレクトリを回復する必要はありません。

- 同義語辞書ファイルの格納ディレクトリの回復

```
rm -r /mnt/syndict/*
cp -r /HADB_bkup/syndict/* /mnt/syndict
```

同義語検索を行っている場合は，同義語辞書ファイルの格納ディレクトリを回復する必要があります。

3. マルチノード構成の HADB サーバを開始する

```
adbstart
```

プライマリノードおよび全セカンダリノード（hadb01～hadb03）で，adbstart コマンドおよび HA モニタのmonbegin コマンドを実行してください。adbstart コマンドの実行後，別ターミナル（コマンド入力画面）から，HA モニタのmonbegin コマンドを実行してください。

重要

adbstart コマンドが終了する前に，monbegin コマンドを実行してください。

```
monbegin
```

マルチノード構成の HADB サーバの開始後，プライマリノードの HADB サーバの稼働モードが静止モードの場合，プライマリノード（hadb01）でadbchgsrvmode コマンドを実行し，プライマリノードの HADB サーバの稼働モードを通常モードに変更してください。

```
adbchgsrvmode --normal
```

4. マルチノード構成にワーカーノードを追加する

```
adbstart
```

ワーカーノード（hadb04）で，adbstart コマンドを実行してください。

5. 同義語辞書ファイルの同期を取る

```
adbsyndict -s
```

プライマリノード（hadb01）でadbsyndict コマンドを実行し，同義語辞書ファイルの同期を取ってください。同義語検索を行っている場合に限り，この操作を実行してください。

16.8.4 バックアップの運用例（ShadowImage を使用）

ここでは，ShadowImage を使用したバックアップの運用例について説明します。

(1) システムの構成例

4 つのノード（hadb01，hadb02，hadb03，hadb04）で構成されるマルチノード構成で，プライマリノードはhadb01 とします。DB ディレクトリ，アーカイブディレクトリ，同義語辞書ファイルの格納ディレクトリ，および監査証跡の出力先ディレクトリは，次の構成となっているとします。

■hadb01 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル（LU002を使用）
 ├ADBMST：ブロックスペシャルファイル（LU001を使用）
 ├ADBSTBL：ブロックスペシャルファイル（LU003を使用）
 ├ADBWORK：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 ├ADBWRK：ブロックスペシャルファイル（/dev/mapper/WRK001を使用）
 ├ADBUTBL01：ブロックスペシャルファイル（LU004を使用）
 ├ADBUIDX01：ブロックスペシャルファイル（LU005を使用）
 ├SPOOL：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 └ADBSYS：システムディレクトリ（FS001上のファイルシステム）
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリの構成（FS002 上のファイルシステム）

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索を行っている場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■hadb02 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル（LU002を使用）
 ├ADBMST：ブロックスペシャルファイル（LU001を使用）
 ├ADBSTBL：ブロックスペシャルファイル（LU003を使用）
 ├ADBWORK：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 ├ADBWRK：ブロックスペシャルファイル（/dev/mapper/WRK002を使用）
 ├ADBUTBL01：ブロックスペシャルファイル（LU004を使用）
 ├ADBUIDX01：ブロックスペシャルファイル（LU005を使用）
 ├SPOOL：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 └ADBSYS：システムディレクトリのマウントポイント
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリのファイルシステムのマウントポイント

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索を行っている場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■hadb03 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル（LU002を使用）
 ├ADBMST：ブロックスペシャルファイル（LU001を使用）
 ├ADBSTBL：ブロックスペシャルファイル（LU003を使用）
 ├ADBWORK：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 ├ADBWRK：ブロックスペシャルファイル（/dev/mapper/WRK003を使用）
 ├ADBUTBL01：ブロックスペシャルファイル（LU004を使用）
 ├ADBUIDX01：ブロックスペシャルファイル（LU005を使用）
 ├SPOOL：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 └ADBSYS：システムディレクトリのマウントポイント
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリのファイルシステムのマウントポイント

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索を行っている場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■hadb04 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル（LU002を使用）
 ├ADBMST：ブロックスペシャルファイル（LU001を使用）
 ├ADBSTBL：ブロックスペシャルファイル（LU003を使用）
 ├ADBWORK：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 ├ADBWRK：ブロックスペシャルファイル（/dev/mapper/WRK004を使用）
 ├ADBUTBL01：ブロックスペシャルファイル（LU004を使用）
 ├ADBUIDX01：ブロックスペシャルファイル（LU005を使用）
 ├SPOOL：ローカルファイルシステム上のディレクトリ（/dev/sdb1内）
 └ADBSYS：ローカルファイルシステム上のシステムディレクトリ
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

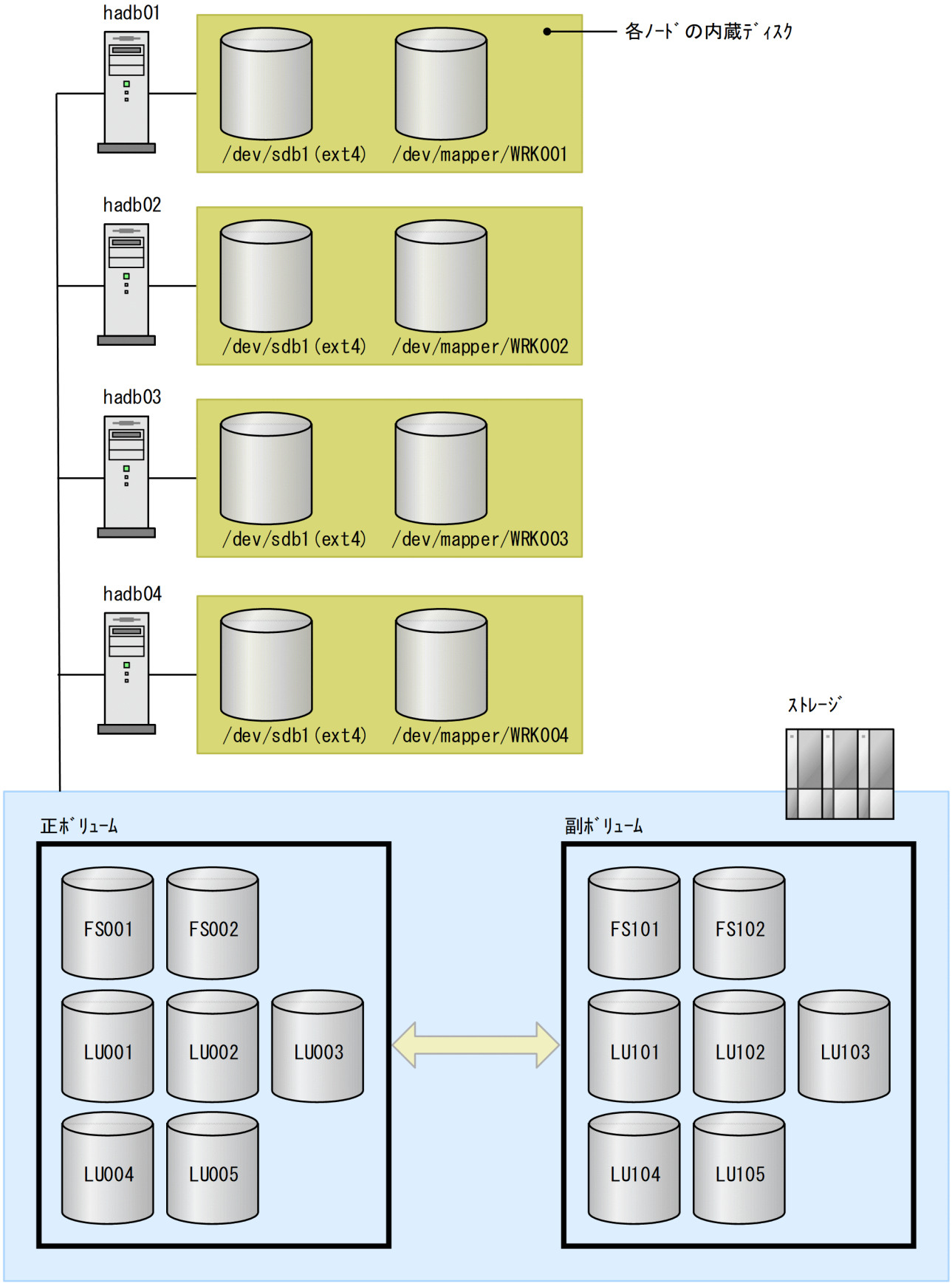
- 監査証跡の出力先ディレクトリの構成

```
/HADB/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用している場合に作成するディレクトリです。

■Logical Unit の構成

マルチノード構成のときのストレージの構成例を次の図に示します。



[説明]

- FS001 には，システムディレクトリのファイルシステムを作成しています。

- FS002 には，同義語辞書ファイルを格納しているファイルシステムを作成しています。
- LU001～LU005 は，DB エリアファイルに使用しています。
- 正ボリュームFS00X，LU00X に対応する副ボリュームはFS10X，LU10X です。

(2)　バックアップの取得

FS001，FS002，およびLU001～LU005 のすべての Logical Unit を，「10.3.4　バックアップの運用例(ShadowImage を使用)」の「(3)　バックアップの取得」の手順に従ってバックアップを取得してください。

(3)　バックアップからの回復

FS001，FS002，およびLU001～LU005 のすべての Logical Unit を，「10.3.4　バックアップの運用例(ShadowImage を使用)」の「(4)　バックアップからの回復」の手順に従って回復してください。

同義語検索を行っている場合は，データベースを回復したあとに，プライマリノードのhadb01 でadbsyndict コマンドを実行し，同義語辞書ファイルの同期を取ってください。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

16.9　状態監視（マルチノード機能の使用時）

ここでは，マルチノード機能使用時特有のシステムの監視方法を説明します。

16.9.1　全ノードの情報を確認する場合

稼働中のプライマリノードまたはセカンダリノードでadbls -d node コマンドを実行すると，全ノードの種別や，ステータス情報などを確認できます。adbls -d node コマンドを実行した際に出力される情報については，マニュアル『HADB　コマンドリファレンス』の『adbls -d node（各ノードの HADB サーバの状態表示)』を参照してください。

メモ

- プライマリノードまたはセカンダリノードでadbls -d node コマンドを実行した場合
  全ノードの情報が表示されます。
- ワーカーノードでadbls -d node コマンドを実行した場合
  コマンドを実行したワーカーノード，プライマリノード，および全セカンダリノードの情報だけが表示されます。ほかのワーカーノードの情報は表示されません。

16.9.2　HADB サーバの状態を確認する場合

マルチノード機能を使用している場合，自ノードで出力されたメッセージは，自ノードのメッセージログファイルおよび syslog に出力されます。各ノードに出力されたメッセージを時刻などで突き合わせて参照することで，全ノードの状況を確認することができます。そのため，NTP などを使用して各ノード間の時刻を合わせるようにしてください。

メッセージの確認方法については，「10.4　メッセージの確認」を参照してください。

16.9.3　AP またはコマンドの実行時間が長い場合

AP またはコマンドの処理時間が予想より長い場合は，次の手順で AP やコマンドを実行しているノードを特定してください。

手順

1. adbls -d cnct コマンドを実行する
   コマンドの実行結果を参照して，AP やコマンドを実行しているノードのノード番号を確認してください。
2. adbls -d node コマンドを実行する

コマンドの実行結果を参照して，ノード番号に対応するノードを特定してください。

3. 原因を調査する

ノードを特定したあとに，「10.8　トランザクションの処理状況の確認」を参照して，原因を調査してください。

adbls -d cnct コマンドを実行した際に出力される情報については，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示)』を参照してください。

adbls -d node コマンドを実行した際に出力される情報については，マニュアル『HADB　コマンドリファレンス』の『adbls -d node（各ノードの HADB サーバの状態表示)』を参照してください。

16.9.4　HADB サーバの開始処理または終了処理に掛かる時間が長い場合

HADB サーバの開始処理または終了処理に掛かる時間が予想より長い場合は，次のどちらかの方法で調査してください。

- 出力メッセージから調査する

  HADB サーバの開始または終了時に出力されるメッセージを，各ノードのメッセージログファイルを参照して調査してください。

- adbls -d srv コマンドで調査する

  adbls -d srv を実行した際に出力される情報については，マニュアル『HADB　コマンドリファレンス』の『adbls -d srv（HADB サーバの状態表示)』を参照してください。

16.9.5　各ノードの HADB サーバの状態を確認する場合

adbls -d srv コマンドで，各ノードの HADB サーバの状態を確認できます。

adbls -d srv コマンドを実行した際に出力される情報については，マニュアル『HADB　コマンドリファレンス』の『adbls -d srv（HADB サーバの状態表示)』を参照してください。

16.9.6　メモリの使用状況を確認する場合

HADB サーバが使用しているメモリの使用状況を確認したい場合は，adbstat コマンドを実行してください。確認方法については，「10.6.1　全メモリの使用状況を確認したい場合」を参照してください。

adbstat コマンドを実行すると，全メモリの最大使用量（Total_memory_max_size）が，ノードごとに出力されます。ノードごとにTotal_memory_max_size の値は異なるため，全ノードのTotal_memory_max_size の値を確認してください。

16.9.7　SQL メインノードと SQL サブノードを確認する場合

SQL パラレル実行機能を適用した検索系 SQL を実行した際に，SQL メインノードおよび SQL サブノードがどのノードに割り当てられたのかを確認したい場合は，次に示す手順でノードを特定してください。手順は，４つのノード（hadb01，hadb02，hadb03，hadb04）から構成されるマルチノード構成を例にして説明します。

手順

1. adbls -d cnct コマンドを実行する

   HADB サーバに接続しているコネクションの状態が表示されます。

```
CID  CNUMBER  CONNECT_TIME         PROGRAM  C-PID      IP-ADDRESS          STATUS
1    1        2022-01-01 00:00:00  HADB     12345      (10.196.108.111 )   STARTED

TRN_ISO_LV      ACCESS_MODE  CONNECTION_INFORMATION            NODE_NO  CLIENT_TYPE
READ_COMMITTED  READ_ONLY    000010000000001-0000000000000000  4        AP(JDBC)

CLIENT_GROUP SQL_ELAPSED_TIME
```

   NODE_NO に SQL メインノードのノード番号が表示されます。この例では，SQL メインノードのノード番号は４になります。

2. adbls -d node コマンドを実行する

```
NODE_NO NODE_TYPE SVID       IP-ADDRESS        STATUS   START_TIME
1       PRIMARY   11111      (172.16.0.11    ) ACTIVE   2022-01-01 00:00:00
2       SECONDARY 22222      (172.16.0.12    ) ACTIVE   2022-01-01 00:00:00
3       SECONDARY 33333      (172.16.0.13    ) ACTIVE   2022-01-01 00:00:00
4       WORKER    44444      (172.16.0.14    ) ACTIVE   2022-01-01 00:00:00
```

   ノード番号が４のノードは，ワーカーノードになります。このワーカーノードが SQL メインノードです。

3. SQL メインノードのノードでadbls -d prl コマンドを実行する

```
CNUMBER    SQL_ID              NODE_NO SQL_NODE_TYPE
1          2                   4       MAIN
1          2                   2       SUB
1          2                   3       SUB
```

   NODE_NO に，SQL メインノードおよび SQL サブノードのノード番号が表示されます。この例では，SQL サブノードのノード番号は，２と３になります。

4. adbls -d node コマンドを実行する

```
NODE_NO NODE_TYPE SVID       IP-ADDRESS        STATUS   START_TIME
1       PRIMARY   11111      (172.16.0.11    ) ACTIVE   2022-01-01 00:00:00
2       SECONDARY 22222      (172.16.0.12    ) ACTIVE   2022-01-01 00:00:00
3       SECONDARY 33333      (172.16.0.13    ) ACTIVE   2022-01-01 00:00:00
4       WORKER    44444      (172.16.0.14    ) ACTIVE   2022-01-01 00:00:00
```

   ノード番号が２，３のノードは，両方ともセカンダリノードになります。このセカンダリノードが SQL サブノードです。

メモ

adbls -d cnct コマンド，adbls -d node コマンド，およびadbls -d prl コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d cnct（コネクションの状態表示)』，『adbls -d node（各ノードの HADB サーバの状態表示)』，および『adbls -d prl（SQL パラレル実行機能の情報表示)』を参照してください。

16.10　DB エリアの運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用する場合の，DB エリアの運用について説明します。

16.10.1　グローバルバッファの割り当て

マルチノード機能を使用する場合，グローバルバッファは，各ノードで独立しています。そのため，各ノードの HADB サーバに対して，サーバ定義adbbuff オペランドで，それぞれグローバルバッファを割り当ててください。

サーバ定義adbbuff オペランドについては，「7.2.11　グローバルバッファに関するオペランドおよびオプション（コマンド形式）」を参照してください。

16.10.2　データベースの状態や使用量の確認

マルチノード機能を使用している場合に，データベースの状態や使用量を確認するときは，プライマリノードでadbdbstatus コマンドを実行してください。

セカンダリノードまたはワーカーノードでadbdbstatus コマンドを実行した場合は，エラーとなります(KFAA31898-E メッセージが出力されます)。

adbdbstatus コマンドの実行結果から，データベースの状態や使用量を確認する方法については，「10.9 データベースの状態や使用量の確認」を参照してください。

16.10.3　データ用 DB エリアの追加・削除・拡張（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合に，adbmodarea コマンドを使用して，データ用 DB エリアの追加，削除および拡張する方法について説明します。

(1)　データ用 DB エリアを追加する方法

データ用 DB エリアを追加する方法については，「11.10.1　データ用 DB エリアを追加する方法」を参照してください。

なお，データ用 DB エリアを追加するadbmodarea コマンドは，プライマリノードで実行してください。

重要

マルチノード構成から切り離されているノードがある場合，データ用 DB エリアを追加できません。切り離されているノードをマルチノード構成に復帰させたあとに，データ用 DB エリアを追加してください。

サーバマシンの故障などによって，マルチノード構成からノードを除外してから，セカンダリノードとしてマルチノード構成に再度追加するまでの間に，データ用 DB エリアを追加した場合は，DB ディレクトリを再作成する必要があります。この場合の手順については，「16.18.1　セカンダリノードの新規追加」を参照してください。

同様に，データ用 DB エリアを追加したときにマルチノード構成になかったワーカーノードをマルチノード構成に追加する場合も，DB ディレクトリを再作成する必要があります。DB ディレクトリを再作成する際，データ用 DB エリアの構成変更の差分情報を反映した初期設定オプションを作成する必要があります。作成方法については，「16.18.2　セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項」を参照してください。その際，セカンダリノードをワーカーノードに読み替えてください。

メモ

データ用 DB エリアの追加中に HADB サーバが異常終了した場合，そのノードの DB ディレクトリには，追加しようとしたデータ用 DB エリアファイルへのシンボリックリンクが残ったままになることがあります。シンボリックリンクが残ったままでも，動作上の問題はありません。

■共有ディスクの SCSI リザーブを使用している場合

共有ディスクの SCSI リザーブを使用している場合は，データ用 DB エリアを追加する前に，HA モニタのservers ファイル中のオペランドの指定を変更する必要があります。データ用 DB エリアを追加する手順を次に示します。

手順

1. マルチノード構成の HADB サーバを正常終了する

   終了方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照してください。

2. HA モニタのservers ファイルの指定を変更する

   scsi_device オペランドまたはdmmp_device オペランドに，追加するディスクのデバイス名を追加してください。プライマリノードおよびセカンダリノードのservers ファイルの指定を変更してください。

3. マルチノード構成の HADB サーバを正常開始する

   開始方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

4. adbmodarea コマンドを実行して，データ用 DB エリアを追加する

   データ用 DB エリアを追加する方法については，「11.10.1　データ用 DB エリアを追加する方法」を参照してください。

(2) データ用 DB エリアを削除する方法

データ用 DB エリアを削除する方法については，「11.10.2 データ用 DB エリアを削除する方法」を参照してください。

なお，データ用 DB エリアを削除するadbmodarea コマンドは，プライマリノードで実行してください。

> **メモ**
>
> データ用 DB エリアの削除時に停止中のノードがある場合，そのノードの DB ディレクトリには，削除したデータ用 DB エリアのデータ用 DB エリアファイルへのシンボリックリンクが残ったままになります。シンボリックリンクが残ったままでも，動作上の問題はありません。

■共有ディスクの SCSI リザーブを使用している場合

共有ディスクの SCSI リザーブを使用している場合は，データ用 DB エリアを削除したあとに，HA モニタのservers ファイル中のオペランドの指定を変更する必要があります。データ用 DB エリアを削除する手順を次に示します。

手順

1. adbmodarea コマンドを実行して，データ用 DB エリアを削除する

   データ用 DB エリアを削除する方法については，「11.10.2 データ用 DB エリアを削除する方法」を参照してください。

2. マルチノード構成の HADB サーバを正常終了する

   終了方法については，「16.4.2 マルチノード構成の HADB サーバの終了方法」を参照してください。

3. HA モニタのservers ファイルの指定を変更する

   scsi_device オペランドまたはdmmp_device オペランドから，削除したディスクのデバイス名を削除してください。プライマリノードおよびセカンダリノードのservers ファイルの指定を変更してください。

4. マルチノード構成の HADB サーバを正常開始する

   開始方法については，「16.4.1 マルチノード構成の HADB サーバの開始方法」を参照してください。

(3) データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）

データ用 DB エリアを拡張する方法については，「11.10.3 データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照してください。

なお，データ用 DB エリアを拡張するadbmodarea コマンドは，プライマリノードで実行してください。

重要

マルチノード構成から切り離されているノードがある場合，データ用 DB エリアを拡張できません。切り離されているノードをマルチノード構成に復帰させたあとに，データ用 DB エリアを拡張してください。

サーバマシンの故障などによって，マルチノード構成からノードを除外してから，セカンダリノードとしてマルチノード構成に再度追加するまでの間に，データ用 DB エリアを拡張した場合は，DB ディレクトリを再作成する必要があります。この場合の手順については，「16.18.1　セカンダリノードの新規追加」を参照してください。

同様に，データ用 DB エリアを拡張したときにマルチノード構成になかったワーカーノードをマルチノード構成に追加する場合も，DB ディレクトリを再作成する必要があります。DB ディレクトリを再作成する際，データ用 DB エリアの構成変更の差分情報を反映した初期設定オプションを作成する必要があります。作成方法については，「16.18.2　セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項」を参照してください。その際，セカンダリノードをワーカーノードに読み替えてください。

メモ

データ用 DB エリアの拡張中に HADB サーバが異常終了した場合，そのノードの DB ディレクトリには，追加しようとしたデータ用 DB エリアファイルへのシンボリックリンクが残ったままになることがあります。シンボリックリンクが残ったままでも，動作上の問題はありません。

■共有ディスクの SCSI リザーブを使用している場合

共有ディスクの SCSI リザーブを使用している場合は，データ用 DB エリアを拡張する前に，HA モニタのservers ファイル中のオペランドの指定を変更する必要があります。データ用 DB エリアを拡張する手順を次に示します。

手順

1. マルチノード構成の HADB サーバを正常終了する

   終了方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照してください。

2. HA モニタのservers ファイルの指定を変更する

   scsi_device オペランドまたはdmmp_device オペランドに，追加するディスクのデバイス名を追加してください。プライマリノードおよびセカンダリノードのservers ファイルの指定を変更してください。

3. マルチノード構成の HADB サーバを正常開始する

   開始方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

4. adbmodarea コマンドを実行して，データ用 DB エリアを拡張する

   データ用 DB エリアを拡張する方法については，「11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照してください。

16.10.4　DB エリアの格納位置の変更

マルチノード機能を使用している場合で，データ用 DB エリアおよび作業表用 DB エリアを構成する DB エリアファイルの格納位置を変更する方法を次に示します。

(1)　データ用 DB エリアの格納位置の変更

マルチノード機能を使用している場合で，データ用 DB エリアを構成するデータ用 DB エリアファイル（ブロックスペシャルファイル）の格納位置を変更する手順を次に示します。

手順

1. マルチノード構成の HADB サーバを正常終了する
   終了方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」を参照してください。
2. 格納位置を変更したいデータ用 DB エリアファイルをコピーする
   格納位置を変更したいデータ用 DB エリアファイル（シンボリックリンク先）を，別の領域にコピーしてください。このとき，コピー先は，全ノードの HADB サーバから参照できる必要があります。
3. 全ノードでシンボリックリンク先をコピー先の領域に変更する
4. マルチノード構成の HADB サーバを開始する
   開始方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

上記の手順 2.および手順 3.の詳細な方法については，「11.10.5　データ用 DB エリアファイルの格納場所を変更する方法」を参照してください。

> **重要**
>
> 変更前のブロックスペシャルファイルから，変更後のブロックスペシャルファイルへのdd コマンドによるコピーは，1 つのノードで実施してください。また，シンボリックリンクの変更は，すべてのノードで実施してください。

■共有ディスクの SCSI リザーブを使用している場合

SCSI リザーブの対象となる DB エリアの格納位置を変更する場合は，HA モニタのservers ファイル中のオペランドの指定を変更する必要があります。そのため，上記の手順 4.でマルチノード構成の HADB サーバを開始する前に，次のことを実施してください。

- servers ファイル中のscsi_device オペランドまたはdmmp_device オペランドの指定を修正します。変更元のディスクのデバイス名を削除し，変更先のディスクのデバイス名を追加してください。

プライマリノードおよびセカンダリノードのservers ファイルの指定を変更してください。

(2)　作業表用 DB エリアの格納位置の変更

マルチノード機能を使用している場合で，作業表用 DB エリアを構成する作業表用 DB エリアファイルの格納位置を変更する手順を次に示します。

手順

1. マルチノード構成の HADB サーバを正常終了する
2. サーバ定義adb_blk_path_wrk オペランドに作業表用 DB エリアファイルの変更後の領域を指定する
3. マルチノード構成の HADB サーバを開始する

16.10.5　作業表用 DB エリアの運用（マルチノード機能の使用時）

マルチノード機能を使用する場合，作業表用 DB エリアは，各ノードでそれぞれ異なる作業表用 DB エリアファイルで構成されます。そのため，複数のノードの HADB サーバから，同じ作業表用 DB エリアファイルにアクセスすることはありません。

作業表用 DB エリアの容量については，「5.9　作業表用 DB エリアの容量見積もり」を参照し，各ノードでそれぞれ見積もってください。

16.10.6　ディスクを増設した場合の運用

増設したディスクに DB エリアファイルを作成する場合，増設したディスクを各ノードの OS に事前に認識させておく必要があります。OS に増設したディスクを認識させるには，マルチノード構成の HADB サーバを終了して全ノードの OS を再起動するか，次に示す手順で 1 ノードずつ OS を再起動してください。

(例)

次に示すマルチノード構成の場合にディスクを増設し，増設したディスクを各ノードの OS に認識させます。

- ノード 1（プライマリノード）
- ノード 2（セカンダリノード）
- ノード 3（セカンダリノード）
- ノード 4（ワーカーノード）

手順

1. ノード 2～ノード 4 の HADB サーバをadbstop --node コマンドで正常終了する
2. ノード 2～ノード 4 の OS を再起動する
3. ノード 2 とノード 3 をマルチノード構成に復帰させる

   マルチノード構成に復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。
4. ノード 4 をマルチノード構成に追加する

   マルチノード構成へのワーカーノードの追加方法については，「16.16.3　マルチノード構成から削除されたワーカーノードを再度追加する手順」を参照してください。

5. コマンドによるプライマリノードの切り替えを実施し，ノード 2 またはノード 3 のどちらかをプライマリノードに切り替える

   コマンドによるプライマリノードの切り替え方法については，「16.7　コマンドによるプライマリノードの切り替え」を参照してください。

   このとき，ノード 1 の HADB サーバは自動的に停止します。

6. ノード 1 の OS を再起動する

7. ノード 1 をマルチノード構成に復帰させる

   マルチノード構成に復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

   このとき，ノード 1 は，セカンダリノードとしてマルチノード構成に復帰します。ノード 1 をプライマリノードにしたい場合は，「16.15.3　マルチノード構成へのノードの復帰」の「(3)　復帰させたノードをプライマリノードに切り替える手順」を参照してください。

16.11　スキーマ，表およびインデクスの運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用する場合の，スキーマ，表およびインデクスの運用について説明します。

16.11.1　スキーマの運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合の，スキーマの運用について説明します。

基本的には，マルチノード機能を使用しない場合と同じです。スキーマの定義と削除は，CREATE SCHEMA 文とDROP SCHEMA 文で実行できます。マニュアル『HADB　SQL リファレンス』の『定義系 SQL』を参照してください。

16.11.2　実表の運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合の，実表の運用について説明します。

マルチノード機能を使用している場合，実表の運用のうち，次の運用項目ではマルチノード機能特有の運用が必要となります。

- 実表にデータを格納する方法
- 実表の更新不可状態を解除する方法

上記以外については，マルチノード機能を使用しない場合と同じです。「11.1　実表の運用」を参照してください。

(1)　実表にデータを格納する方法（マルチノード機能の使用時）

マルチノード機能を使用している場合，adbimport コマンドを実行できるノードはプライマリノードだけです。セカンダリノードおよびワーカーノードでは，adbimport コマンドを実行できません。

adbimport コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbimport（データのインポート)』を参照してください。

adbimport コマンドとトランザクションの同時実行可否については，「2.19.2　トランザクションおよびコマンドが実行されるノード」の「(4)　トランザクションとコマンドの同時実行についての制限」を参照してください。

■SQL パラレル実行機能の検索対象表にデータを格納する場合

SQL パラレル実行機能の検索対象表は，複数の DB エリアファイルで構成されるデータ用 DB エリアに格納されています。SQL パラレル実行機能を適用して検索系 SQL を実行する場合，DB エリアファイルごとに検索系 SQL の処理を並列実行します。そのため，各 DB エリアファイルに格納されているデータ量に偏りがあると，各ノードの負荷にばらつきが出てしまい，並列に処理にするメリットが小さ

くなります。したがって，SQL パラレル実行機能の検索対象表にデータを格納する際は，各 DB エリアファイルに均等にデータが格納されるようにしてください（次の条件をすべて満たすようにして adbimport コマンドを実行してください）。

- 「*adbimport コマンドが使用するスレッド数*－1」を，インポート対象表が格納されているデータ用 DB エリアの DB エリアファイル数の倍数にする。
- 入力データファイル数を「*adbimport コマンドが使用するスレッド数*－1」の倍数にする。
- 各入力データファイルのデータ件数をなるべく同じにする。

(2)　実表の更新不可状態を解除する方法（マルチノード機能の使用時）

- **adbimport コマンドまたはadbidxrebuild コマンドが異常終了した場合**

  adbimport コマンドまたはadbidxrebuild コマンドが異常終了し，処理対象の実表が更新不可状態になった場合，異常終了したコマンドを再実行して，実表の更新不可状態を解除してください。

  なお，プライマリノードの HADB サーバが異常終了してプライマリノードが切り替わった場合は，新たなプライマリノードでadbidxrebuild コマンドを実行し，実表の更新不可状態を解除してください。このとき，--create-temp-file オプションを指定してadbidxrebuild コマンドを実行してください。

  --create-temp-file オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成）』の『adbidxrebuild コマンドの指定形式』を参照してください。

- **adbunarchivechunk コマンドが異常終了した場合**

  adbunarchivechunk コマンドが異常終了し，処理対象の実表が更新不可状態になった場合，adbunarchivechunk コマンドを再実行して，実表の更新不可状態を解除してください。

> **メモ**
>
> 実表の更新不可状態については，「15.8.1　実表が更新不可状態になった場合」を参照してください。

16.11.3　ビュー表の運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合の，ビュー表の運用について説明します。

基本的には，マルチノード機能を使用しない場合と同じです。「11.2　ビュー表の運用」を参照してください。

16.11.4　インデクスの運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合の，B-tree インデクス，テキストインデクスおよびレンジインデクスの運用について説明します。

マルチノード機能を使用している場合，B-tree インデクス，テキストインデクスおよびレンジインデクスの運用のうち，次の運用項目ではマルチノード機能特有の運用が必要となります。

- インデクスを作成する方法
- B-tree インデクスの未完状態を解除する方法
- テキストインデクスの未完状態を解除する方法
- レンジインデクスの未完状態を解除する方法

上記以外については，マルチノード機能を使用しない場合と同じです。「11.3　インデクスの運用」を参照してください。

(1)　インデクスを作成する方法（マルチノード機能の使用時）

マルチノード機能を使用している場合，次に示すコマンドを実行できるノードは，プライマリノードだけです。セカンダリノードおよびワーカーノードでは，次に示すコマンドを実行できません。

対象のコマンド

- adbimport コマンド
- adbidxrebuild コマンド
- adbmergechunk コマンド
- adbunarchivechunk コマンド

対象のコマンドについては，マニュアル『HADB　コマンドリファレンス』を参照してください。

(2)　B-tree インデクスの未完状態を解除する方法（マルチノード機能の使用時）

adbimport コマンドまたはadbidxrebuild コマンドが異常終了し，B-tree インデクスが未完状態になった場合，異常終了したコマンドを再実行して，B-tree インデクスの未完状態を解除してください。

なお，プライマリノードの HADB サーバが異常終了してプライマリノードが切り替わった場合は，新たなプライマリノードでadbidxrebuild コマンドを実行して，B-tree インデクスの未完状態を解除してください。このとき，--create-temp-file オプションを指定してadbidxrebuild コマンドを実行してください。

--create-temp-file オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成）』の『adbidxrebuild コマンドの指定形式』を参照してください。

B-tree インデクスの未完状態については，「15.9.1　B-tree インデクスが未完状態になったときの対処方法」を参照してください。

(3)　テキストインデクスの未完状態を解除する方法（マルチノード機能の使用時）

adbimport コマンドまたはadbidxrebuild コマンドが異常終了し，テキストインデクスが未完状態になった場合，異常終了したコマンドを再実行して，テキストインデクスの未完状態を解除してください。

なお，プライマリノードの HADB サーバが異常終了してプライマリノードが切り替わった場合は，新たなプライマリノードでadbidxrebuild コマンドを実行して，テキストインデクスの未完状態を解除してください。このとき，--create-temp-file オプションを指定してadbidxrebuild コマンドを実行してください。

--create-temp-file オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドの指定形式』を参照してください。

テキストインデクスの未完状態については，「15.10.1　テキストインデクスが未完状態になったときの対処方法」を参照してください。

(4)　レンジインデクスの未完状態を解除する方法（マルチノード機能の使用時）

adbidxrebuild コマンドが異常終了し，レンジインデクスが未完状態になった場合，adbidxrebuild コマンドを再実行して，レンジインデクスの未完状態を解除してください。

なお，プライマリノードの HADB サーバが異常終了してプライマリノードが切り替わった場合は，新たなプライマリノードでadbidxrebuild コマンドを再実行して，レンジインデクスの未完状態を解除してください。このとき，--create-temp-file オプションを指定してadbidxrebuild コマンドを再実行してください。

--create-temp-file オプションについては，マニュアル『HADB　コマンドリファレンス』の『adbidxrebuild（インデクスの再作成)』の『adbidxrebuild コマンドの指定形式』を参照してください。

レンジインデクスの未完状態については，「15.11.1　レンジインデクスが未完状態になったときの対処方法」を参照してください。

16.11.5　マルチチャンク表の運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合の，マルチチャンク表の運用について説明します。

マルチノード機能を使用している場合，マルチチャンク表の運用のうち，次の運用項目ではマルチノード機能特有の運用が必要となります。

- チャンクをマージする方法
- チャンクのコメントを変更する方法
- チャンクの状態を変更する方法

また，マルチノード機能を使用している場合の，実表の運用については，「16.11.2　実表の運用（マルチノード機能の使用時)」を参照してください。

上記以外については，マルチノード機能を使用しない場合と同じです。「11.4　マルチチャンク表の運用」を参照してください。

(1)　チャンクをマージする方法（マルチノード機能の使用時）

複数のチャンクを 1 つのチャンクにマージする場合，adbmergechunk コマンドを実行します。詳細については，「11.4.9　チャンクをマージする方法（チャンク数を減らす方法）」を参照してください。

なお，マルチノード機能を使用している場合，adbmergechunk コマンドが実行できるノードは，プライマリノードだけです。セカンダリノードおよびワーカーノードでは，adbmergechunk コマンドを実行できません。

(2)　チャンクのコメントを変更する方法（マルチノード機能の使用時）

チャンクのコメントを設定，変更および削除する場合，adbchgchunkcomment コマンドを実行します。詳細については，「11.4.11　チャンクのコメントを変更する方法」を参照してください。

なお，マルチノード機能を使用している場合，adbchgchunkcomment コマンドが実行できるノードは，プライマリノードだけです。セカンダリノードおよびワーカーノードでは，adbchgchunkcomment コマンドを実行できません。

(3)　チャンクの状態を変更する方法（マルチノード機能の使用時）

チャンクの状態を変更する場合，adbchgchunkstatus コマンドを実行します。詳細については，「11.4.12 チャンクの状態を変更する方法」を参照してください。

なお，マルチノード機能を使用している場合，adbchgchunkstatus コマンドが実行できるノードは，プライマリノードだけです。セカンダリノードおよびワーカーノードでは，adbchgchunkstatus コマンドを実行できません。

16.11.6　アーカイブマルチチャンク表の運用（マルチノード機能の使用時）

ここでは，マルチノード機能を使用している場合の，アーカイブマルチチャンク表の運用について説明します。

マルチノード機能を使用している場合，アーカイブマルチチャンク表の運用のうち，次の運用項目ではマルチノード機能特有の運用が必要となります。

- アーカイブディレクトリを設定する方法
- チャンクをアーカイブする方法
- チャンクをアンアーカイブする方法

上記以外については，マルチノード機能を使用しない場合と同じです。「11.4　マルチチャンク表の運用」を参照してください。

(1)　アーカイブディレクトリを設定する方法

すべてのノードから，同一の絶対パスでアーカイブディレクトリを参照できるようにしてください。

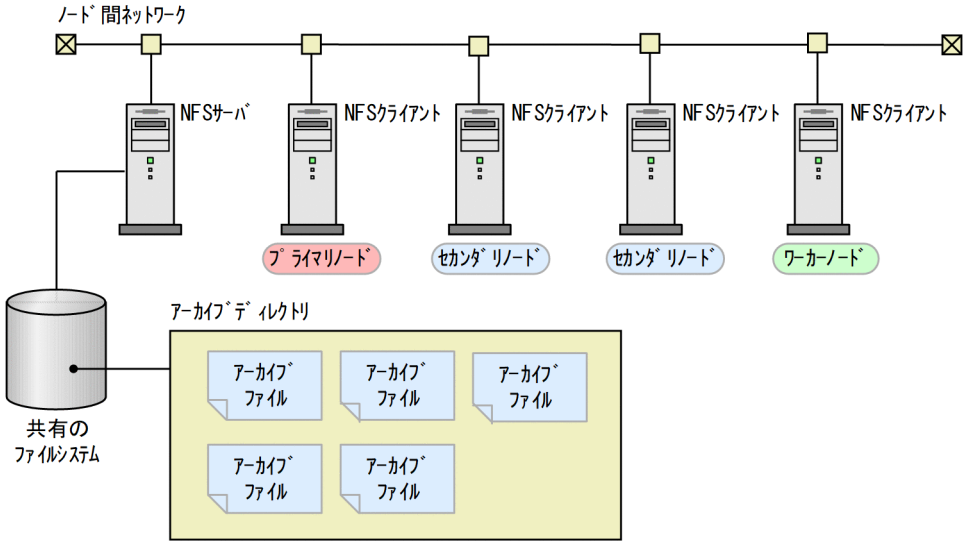
例えば，分散ファイルシステムの一つである NFS を使用する場合は，次の手順でアーカイブディレクトリを準備します。

手順

1. NFS サーバを用意し，NFS サーバ上でアーカイブディレクトリを作成してください。
2. 手順の 1.で作成したアーカイブディレクトリをエクスポートしてください。
   この操作は，NFS サーバで実行します。
3. 手順の 2.でエクスポートしたディレクトリを，全ノードで同じパスとなるようにマウントしてください。
   この操作は，全ノードで実行します。

NFS を使用したシステム構成例を次の図に示します。

図 16-6　NFS を使用したシステム構成例



(2)　チャンクをアーカイブする方法

マルチノード機能を使用している場合，チャンクをアーカイブするadbarchivechunk コマンドを実行できるノードはプライマリノードだけです。セカンダリノードおよびワーカーノードでは，adbarchivechunk コマンドを実行できません。

adbarchivechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbarchivechunk（チャンクのアーカイブ)』を参照してください。

(3)　チャンクをアンアーカイブする方法

マルチノード機能を使用している場合，チャンクをアンアーカイブするadbunarchivechunk コマンドを実行できるノードはプライマリノードだけです。セカンダリノードおよびワーカーノードでは，adbunarchivechunk コマンドを実行できません。

adbunarchivechunk コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbunarchivechunk（チャンクのアンアーカイブ)』を参照してください。

16.11.7　システム表の再編成（マルチノード機能の使用時）

システム表を再編成するadbreorgsystemdata コマンドは，プライマリノードで実行してください。セカンダリノードおよびワーカーノードでは，adbreorgsystemdata コマンドを実行できません。

システム表を再編成する手順については，「11.17　システム表の再編成」を参照してください。

16.12　システムログの運用（マルチノード機能の使用時）

マルチノード機能を使用している場合，システムログファイルは全ノードで 1 つだけ存在し，プライマリノードの HADB サーバが管理しています。プライマリノードの HADB サーバは，各ノードの HADB サーバの操作履歴情報，およびデータベースの更新履歴情報をシステムログとして，システムログファイルに出力します。

システムログファイルは，プライマリノードの DB ディレクトリ下のADBSYS/ADBSLG ディレクトリにマウントしているファイルシステム内に作成されます。

システムログについては，「2.11.1　再開始による回復処理の流れ」の「(2)　システムログ」を参照してください。

■プライマリノードの切り替えが発生した場合

切り替え前のプライマリノードの HADB サーバが管理していたシステムログを，新しくプライマリノードになる HADB サーバが引き継ぎます。新しくプライマリノードになる HADB サーバが，引き継いだシステムログを使用して，プライマリノードで実行されていた更新処理に対する回復処理（ロールバック）を行います。

プライマリノードの切り替えが発生した場合，システムログが格納されているファイルシステムは，切り替え前のプライマリノードからアンマウントされます。そのあと，新たなプライマリノードの DB ディレクトリ下のADBSYS ディレクトリにマウントされます。これらのアンマウントおよびマウント処理は，HA モニタが自動的に実行します。

16.13　統計情報の運用（マルチノード機能の使用時）

マルチノード機能を使用している場合，統計情報はノードごとに管理されます。そのため，統計情報を出力したいノードでadbstat コマンドを実行してください。

また，トランザクションを実行するノードをシステムが自動的に決定することがあります。そのため，コネクションの稼働情報，またはSQL 文の統計情報を出力する場合は，adbstat コマンドを実行したノードで実行されたトランザクションの情報だけが出力されます。

トランザクションを実行するノードについては，「16.5.3　トランザクションを実行するノード」を参照してください。

統計情報の確認方法については，「10.10　統計解析の運用（HADB サーバの稼働情報の確認）」を参照してください。

■アクセスパスの統計情報を取得している場合

アクセスパスの統計情報はSQL トレース情報中に出力されます。マルチノード機能を使用している場合，SQL トレース情報はノードごとに管理されます。そのため，各ノードのSQL トレース情報には，自ノードで実行したトランザクションのSQL 文の情報だけが出力されます。したがって，すべてのトランザクションのアクセスパスの統計情報を取得する場合は，全ノードでSQL トレース情報を取得するようにしてください。SQL トレース情報の取得方法については，「10.11.5　SQL トレース情報を出力するための準備」を参照してください。

16.14　チューニング（マルチノード機能の使用時）

マルチノード機能を使用している場合，HADB サーバのチューニングを実施するときは，すべてのノードの HADB サーバに対して，それぞれチューニングを行ってください。

チューニングの内容については，「13.　チューニング」を参照してください。

16.15　プライマリノードまたはセカンダリノードでノード障害が発生した場合の運用（ノードの復帰）

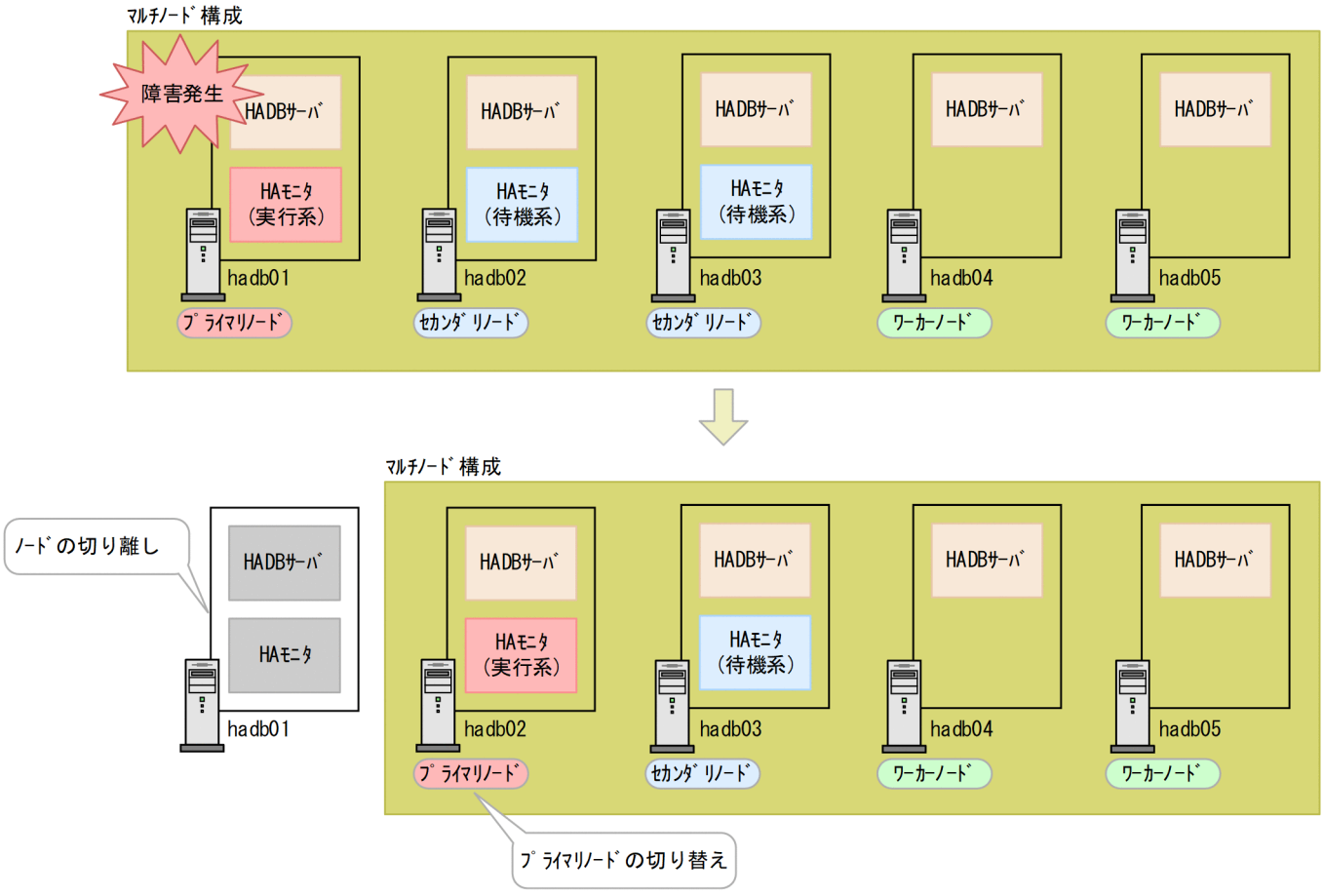
プライマリノードまたはセカンダリノードでノード障害が発生した場合，障害が発生したノードはマルチノード構成から自動的に切り離され，残ったノードで業務が継続されます。ここでは，プライマリノードまたはセカンダリノードでノード障害が発生した場合の運用について説明します。

16.15.1　プライマリノードでノード障害が発生した場合

(1)　プライマリノードでノード障害が発生した場合のシステムの処理

プライマリノードでノード障害が発生した場合，セカンダリノードの 1 つがプライマリノードに切り替わり，障害が発生したプライマリノードはマルチノード構成から切り離されます。プライマリノードでノード障害が発生した場合のシステムの処理を次の図に示します。

図 16-7　プライマリノードでノード障害が発生した場合のシステムの処理



[説明]

プライマリノード（hadb01）でノード障害が発生した場合，セカンダリノードの 1 つをプライマリノードに切り替えて，障害が発生したノードはマルチノード構成から切り離されます。その後，プライマリ

ノード（hadb02），セカンダリノード（hadb03），およびワーカーノード（hadb04，hadb05）で処理が継続されます。

なお，ワーカーノード（hadb04，hadb05）は，プライマリノードの切り替え先にはなりません。

HADB では，HA モニタのマルチスタンバイ機能を使用して，プライマリノードを切り替えています。マルチスタンバイ機能とは，1 つの実行系（プライマリノード）に対して，複数の待機系（セカンダリノード）を準備する機能のことです。セカンダリノードの優先順位（どのセカンダリノードがプライマリノードになるか）は，HA モニタのservers ファイルのinitial およびstandbypri オペランドで指定します。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『マルチスタンバイ機能を使用する場合のサーバと系の管理』を参照してください。

(2) ノード障害が発生したときの HADB 管理者の対処

ノード障害の原因を特定するために，「14.1 障害発生時の対処の流れ」を参照して対処してください。そこで説明している手順を，ノード障害が発生したノードで実施してください。

また，障害が発生したノードで，次のトラブルシュート情報を取得してください。

- HADB サーバのトラブルシュート情報をadbinfoget コマンドで取得してください。
- HA モニタのトラブルシュート情報を HA モニタのmonts コマンドで取得してください。

障害の原因を対処したあとに，マルチノード構成から切り離されたノードをマルチノード構成に復帰させることができます。マルチノード構成への復帰手順については，「16.15.3 マルチノード構成へのノードの復帰」を参照してください。

なお，マルチノード構成に復帰させたノードは必ずセカンダリノードになります。

(3) 注意事項

マルチノード構成にセカンダリノードがない状態のときにプライマリノードでノード障害が発生した場合，切り替え先のノードがないため，マルチノード構成の HADB サーバは異常終了します。このとき，ノード障害が発生したプライマリノードで HA モニタのmonshow コマンドを実行し，自系のサーバの状態を確認してください。

自系のサーバの状態が ONL（実行処理中）の場合は，HA モニタのmonend コマンドを実行し，HA モニタを終了してください。

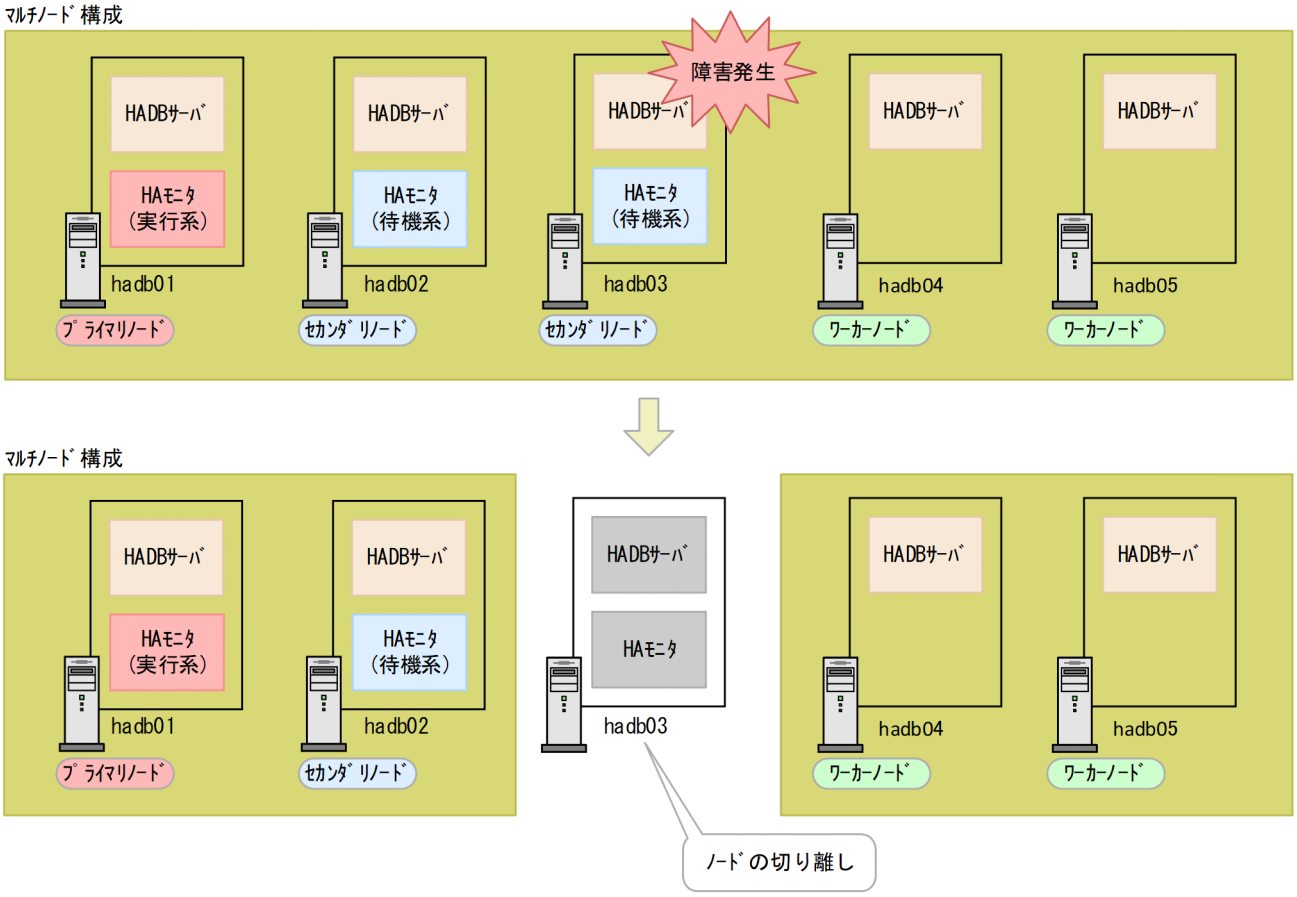
> **メモ**
>
> ワーカーノードは，プライマリノードの切り替え先になりません。そのため，プライマリノードとワーカーノードだけの状態（セカンダリノードがない状態）のときに，プライマリノードでノード障害が発生した場合，ワーカーノードを含めたマルチノード構成の HADB サーバが異常終了します。

16.15.2　セカンダリノードでノード障害が発生した場合

(1)　セカンダリノードでノード障害が発生した場合のシステムの処理

セカンダリノードでノード障害が発生した場合，そのセカンダリノードはマルチノード構成から切り離されます。セカンダリノードでノード障害が発生した場合のシステムの処理を次の図に示します。

図 16-8　セカンダリノードでノード障害が発生した場合のシステムの処理



[説明]

セカンダリノード（hadb03）でノード障害が発生した場合，そのセカンダリノードをマルチノード構成から切り離します。その後，残った４つのノードで処理が継続されます。

(2)　ノード障害が発生したときの HADB 管理者の対処

ノード障害の原因を特定するために，「14.1　障害発生時の対処の流れ」を参照して対処してください。そこで説明している手順を，ノード障害が発生したノードで実施してください。

また，障害が発生したノードで，次のトラブルシュート情報を取得してください。

- HADB サーバのトラブルシュート情報をadbinfoget コマンドで取得してください。
- HA モニタのトラブルシュート情報を HA モニタのmonts コマンドで取得してください。

障害の原因を対処したあとに，マルチノード構成から切り離されたノードをマルチノード構成に復帰させることができます。マルチノード構成への復帰手順については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

なお，マルチノード構成に復帰させたノードは必ずセカンダリノードになります。

16.15.3　マルチノード構成へのノードの復帰

マルチノード構成から切り離されたノードを，マルチノード構成に復帰させる手順を説明します。

(1)　マルチノード構成にノードを復帰させる際の留意事項

1. サーバ定義の次のオペランドの指定値を変更することはできません。

   - adb_sys_rthd_num

   また，全ノードで指定値を同じにする必要があるオペランド，およびプライマリノードとセカンダリノードで指定値を同じにする必要があるオペランドについても，指定値を変更することはできません。全ノードで指定値を同じにする必要があるオペランド，およびプライマリノードとセカンダリノードで指定値を同じにする必要があるオペランドについては，「16.3.5　各ノードでのサーバ定義の作成」を参照してください。

   上記のオペランド以外についても，必要のない場合は指定値を変更しないでください。

2. 次に示す条件を 1 つでも満たす場合は，マルチノード構成にノードを復帰させることはできません。次に示す処理の完了後にノードの復帰操作を実行してください。

   - マルチノード構成の HADB サーバの中に，終了処理中の HADB サーバが存在する場合
   - プライマリノードの切り替え処理中の場合
   - マルチノード構成に復帰処理中のノードが存在する場合
   - マルチノード構成に追加処理中のワーカーノードが存在する場合
   - HADB サーバに接続中の AP またはコマンドが存在する場合

(2)　マルチノード構成にノードを復帰させる手順

マルチノード構成にノードを復帰させる手順を次に示します。

手順

1. HA モニタのmonshow コマンドを復帰対象のノードで実行する

   HA モニタの監視が停止していることを確認してください。

> **重要**
>
> 以降の手順は，HA モニタの監視が停止していることを確認してから実施してください。誤って，手順 3.のadbstart コマンドを実行する前に，手順 4.の HA モニタのmonbegin コマンドを実行してしまった場合は，HA モニタのmonend コマンドを実行してください（HA モニタの監視，および起動中の HADB サーバを停止してください）。そのあとに，マルチノード構成にノードを復帰させる手順を最初から実施してください。

2. adbchgsrvmode --maintenance コマンドをプライマリノードで実行する

   プライマリノードの HADB サーバの稼働モードをメンテナンスモードに変更してください。または，実行中のトランザクションが存在しない状態にしてください。

3. adbstart コマンドを復帰対象のノードで実行する

   adbstart コマンドの実行後，HA モニタのmonbegin コマンドの実行を要求するKFAA91109-I メッセージが出力されます。

4. HA モニタのmonbegin コマンドを復帰対象のノードで実行する

   KFAA91109-I メッセージの出力を確認したあとに，別のターミナル（コマンド入力画面）から，monbegin コマンドを実行してください。

   > **重要**
   >
   > adbstart コマンドが終了する前に，monbegin コマンドを実行してください。

   adbstart コマンドが正常終了すると，マルチノード構成へのノードの復帰処理が完了します。

5. adbchgsrvmode --normal コマンドをプライマリノードで実行する

   手順 2.で，プライマリノードの HADB サーバの稼働モードをメンテナンスモードに変更した場合は，稼働モードを通常モードに戻してください。

6. adbls -d node コマンドを任意のノードで実行する

   コマンドの実行結果のNODE_TYPE 欄を参照して，プライマリノードが 1 つ存在することを確認してください。もし，すべてのノードがセカンダリノードになっている場合は，プライマリノードにしたいノードで HA モニタのmonact コマンドを実行してください。monact コマンドを実行したノードがプライマリノードになります。

   > **メモ**
   >
   > 手順 2.で，KFAA91109-I メッセージの出力前にmonbegin コマンドを実行し，かつその瞬間にプライマリノードが停止した場合，すべてのノードがセカンダリノードになる可能性があります。

なお，マルチノード構成に復帰させたノードは必ずセカンダリノードになります。復帰させたノードをプライマリノードにしたい場合は，「(3)　復帰させたノードをプライマリノードに切り替える手順」を参照してください。

重要

同義語検索を行っている場合は，ノードを復帰したあとに同義語辞書ファイルの同期を取る必要があります。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

メモ

監査証跡機能が有効な場合は，ノードの復帰が完了したあとに，そのノードでの監査証跡の出力が自動的に再開されます。監査証跡機能が無効な場合は，復帰対象ノードの HADB サーバが前回稼働時に監査証跡機能が有効であっても，監査証跡機能が無効な状態でマルチノード構成に復帰します。

(3)　復帰させたノードをプライマリノードに切り替える手順

復帰させたノードをプライマリノードに切り替える手順を説明します。系のリセットを使用している場合と，共有ディスクの SCSI リザーブを使用している場合とで手順が異なります。

- 手順（系のリセットを使用している場合）

  1. マルチノード構成から切り離されていたノード（元のプライマリノード）を，マルチノード構成に復帰させる

     マルチノード構成に復帰させたノードはセカンダリノードになります。

  2. HA モニタの`monswap` コマンドを現在のプライマリノードで実行する

     現在のプライマリノードの HADB サーバは終了し，手順 1.で復帰させたノードがプライマリノードに切り替わります。

  3. `adbls -d node` コマンドを任意のノードで実行する

     コマンドの実行結果の`NODE_TYPE` 欄を参照して，プライマリノードが切り替わったことを確認してください。プライマリノードの切り替えが完了しないと，次の操作ができません。

  4. 手順 2.で終了したノードをマルチノード構成に復帰させる

     復帰手順については，「(2)　マルチノード構成にノードを復帰させる手順」を参照してください。

- 手順（共有ディスクの SCSI リザーブを使用している場合）

  1. マルチノード構成から切り離されていたノード（元のプライマリノード）を，マルチノード構成に復帰させる

     マルチノード構成に復帰させたノードはセカンダリノードになります。

  2. `adbstop --cancel --node` コマンドをプライマリノードで実行する

     プライマリノードが正常終了します。

  3. HA モニタの`monswap` コマンドを実行する

手順 2.で終了したノードで上記のコマンドを実行すると，手順 1.で復帰させたノードがプライマリノードに切り替わります。

4. adbls -d node コマンドを任意のノードで実行する

   コマンドの実行結果のNODE_TYPE 欄を参照して，プライマリノードが切り替わったことを確認してください。プライマリノードの切り替えが完了しないと，次の操作ができません。

5. 手順 2.で終了したノードをマルチノード構成に復帰させる

   復帰手順については，「(2)　マルチノード構成にノードを復帰させる手順」を参照してください。

16.16　ワーカーノードでノード障害が発生した場合の運用（ワーカーノードの追加）

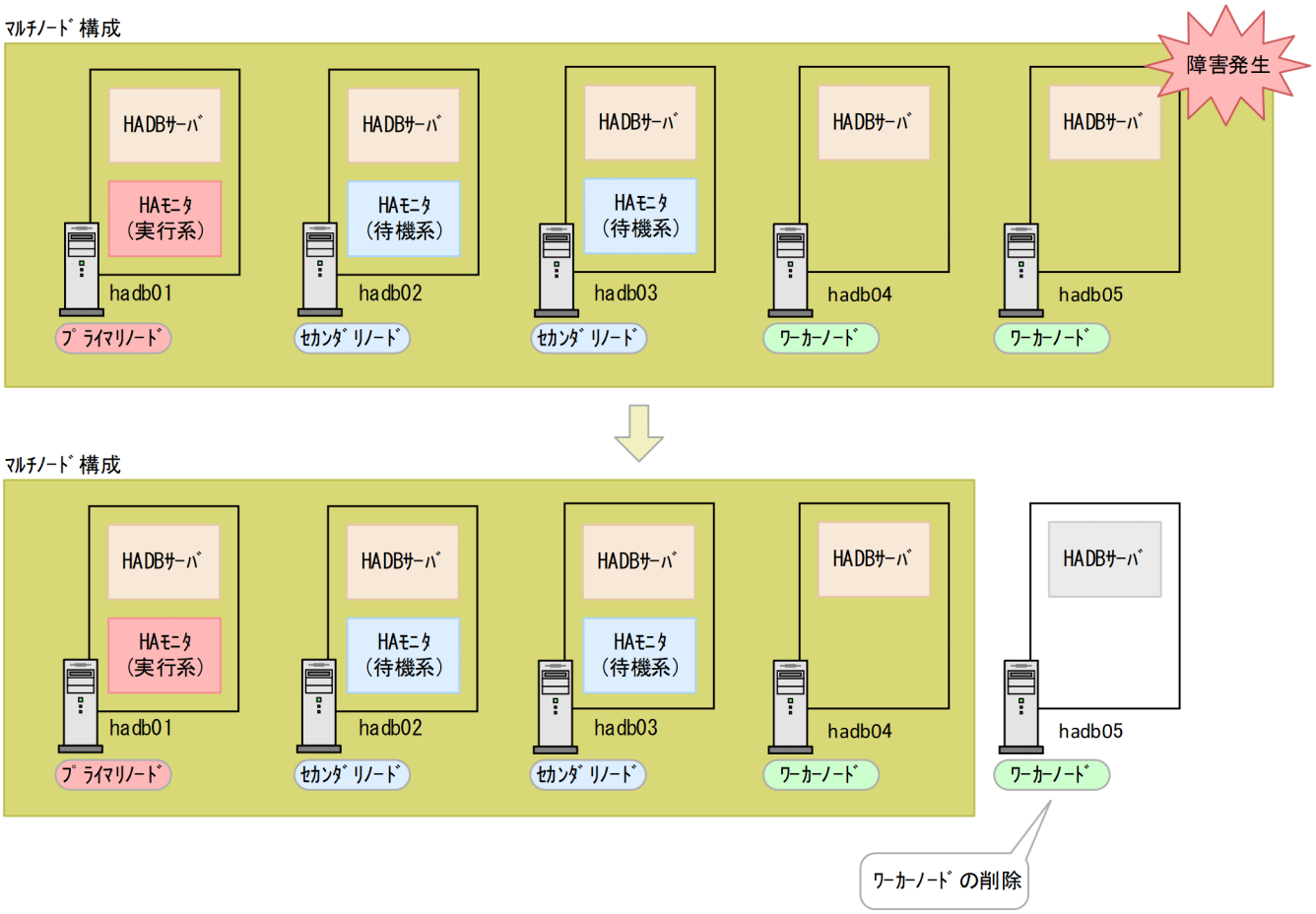
ワーカーノードでノード障害が発生して HADB サーバが異常終了した場合，そのワーカーノードはマルチノード構成から削除されます。ここでは，ワーカーノードでノード障害が発生した場合の運用について説明します。

16.16.1　ワーカーノードでノード障害が発生した場合

(1)　ワーカーノードでノード障害が発生した場合のシステムの処理

ワーカーノードでノード障害が発生して HADB サーバが異常終了した場合，そのワーカーノードはマルチノード構成から削除されます。ワーカーノードでノード障害が発生した場合のシステムの処理を次の図に示します。

図 16-9　ワーカーノードでノード障害が発生した場合のシステムの処理



[説明]

ワーカーノード（hadb05）でノード障害が発生した場合，ワーカーノード（hadb05）をマルチノード構成から削除します。その後，残った４つのノードで処理が継続されます。

(2)　ノード障害が発生したときの HADB 管理者の対処

ノード障害の原因を特定するために，「14.1　障害発生時の対処の流れ」を参照して対処してください。そこで説明している手順を，ノード障害が発生したワーカーノードで実施してください。

また，障害が発生したワーカーノードで，次のトラブルシュート情報を取得してください。

- HADB サーバのトラブルシュート情報をadbinfoget コマンドで取得してください。

障害の原因を対処したあとに，マルチノード構成から削除されたワーカーノードをマルチノード構成に再び追加することができます。マルチノード構成へのワーカーノードの追加手順については，「16.16.3　マルチノード構成から削除されたワーカーノードを再度追加する手順」を参照してください。

16.16.2　マルチノード構成から削除されたワーカーノードを再度追加する際の留意事項

1. 全ノードで指定値を同じにする必要があるオペランドについては，指定値を変更することはできません。全ノードで指定値を同じにする必要があるオペランドについては，「16.3.5　各ノードでのサーバ定義の作成」を参照してください。

   全ノードで指定値を同じにする必要があるオペランド以外についても，必要のない場合は指定値を変更しないでください。

2. 次の条件を 1 つでも満たす場合は，マルチノード構成にワーカーノードを追加することはできません。次の条件に示す処理の完了後に，ワーカーノードの追加操作を実行してください。
   - プライマリノードおよびセカンダリノードの HADB サーバの中に，終了処理中の HADB サーバが存在する場合
   - プライマリノードの切り替え処理中の場合
   - マルチノード構成に復帰処理中のノードがある場合
   - adbmodarea コマンドの実行中の場合
3. サーバ定義のadb_sys_multi_max_worker_nodes オペランドに指定した*ワーカーノードの最大数*を超える数のワーカーノードを追加することはできません。

16.16.3　マルチノード構成から削除されたワーカーノードを再度追加する手順

マルチノード構成から削除されたワーカーノードを，マルチノード構成に再度追加する際の手順を次に示します。

手順

1. ワーカーノードの HADB サーバを開始する

   次のコマンドを追加対象のワーカーノードで実行してください。

   ```
   adbstart
   ```

   ワーカーノードの HADB サーバの開始処理が完了すると，マルチノード構成にワーカーノードが追加されます。

重要

同義語検索を行っている場合は，ワーカーノードを追加したあとに同義語辞書ファイルの同期を取る必要があります。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

16.17　トラブルシュート（マルチノード機能の使用時）

ここでは，マルチノード機能の使用時にトラブルが発生した場合の対処方法について説明します。

16.17.1　マルチノード構成の HADB サーバの開始または終了に関するトラブル

マルチノード構成の HADB サーバの開始または終了に関するトラブルが発生したときの対処方法について説明します。

(1)　マルチノード構成の HADB サーバを正常開始できない場合

出力されたメッセージの対処方法に従って原因を特定してください。そのあとに，マルチノード構成の HADB サーバを正常開始してください。

原因がすぐに特定できない場合は，障害が発生したプライマリノードまたはセカンダリノードをマルチノード構成から除外してから，マルチノード構成の HADB サーバを正常開始してください。障害が発生したプライマリノードまたはセカンダリノードをマルチノード構成から除外するには，全ノードの HADB サーバのサーバ定義adb_sys_multi_node_info オペランドから，障害が発生したプライマリノードまたはセカンダリノードの指定を削除します。

なお，障害が発生していないノード（プライマリノードおよびセカンダリノード）が 1 台だけの場合は，そのノードの HADB サーバのサーバ定義adb_sys_multi_node_info オペランドをコメントにしてください。そして，マルチノード機能を使用しないシステムとして HADB サーバを開始してください。このとき，システムディレクトリが$DBDIR/ADBSYS にマウントされていることを確認してください。マウントされていない場合は，OS のmount コマンドを実行して，システムディレクトリを$DBDIR/ADBSYS にマウントしてください。

> **メモ**
>
> オペランドの指定行の先頭にシャープ（#）を記述すると，その行全体の指定がコメントになります。

また，障害が発生したノードがプライマリノードの場合は，次のどちらかの方法でプライマリノードを変更してください。

- HA モニタの設定を変更してプライマリノードを変更する
- マルチノード構成の HADB サーバの開始時に，monact コマンドを実行してセカンダリノードの 1 つをプライマリノードに変更する

代替マシンがある場合は，代替マシンの環境を，障害が発生したノードのマシンとまったく同じ環境に設定したあとに，マルチノード構成の HADB サーバを正常開始してください。

■KFAA41205-E メッセージが出力された場合

次に示す内容のKFAA41205-E メッセージが出力された場合，DB エリアファイルのオープンに失敗しています。

- ファイル名に DB エリアファイルへのシンボリックリンク名が表示されている
- エラーとなったシステムコール名にopen が表示されている
- エラー番号にENOENT が表示されている

セカンダリノードの新規追加前や，ワーカーノードの停止中にデータ用 DB エリアを追加または拡張したため，このメッセージが出力されたノードの DB ディレクトリが古い状態（データ用 DB エリアを追加または拡張する前の状態）である可能性があります。

マルチノード構成の HADB サーバの開始時にこのメッセージが出力された場合は，このメッセージが出力されたセカンダリノードでadbinit コマンドを実行して DB ディレクトリを再作成※してください。そのあとに，マルチノード構成の HADB サーバを正常開始してください。

マルチノード構成へのワーカーノードの追加時にこのメッセージが出力された場合は，このメッセージが出力されたワーカーノードでadbinit コマンドを実行して DB ディレクトリを再作成※してください。そのあとに，adbstart コマンドでワーカーノードの HADB サーバを開始し，ワーカーノードをマルチノード構成に追加してください。

注※

DB ディレクトリを再作成する際，ほかのノードで使用した初期設定オプションをそのまま使用できないことがあります。データ用 DB エリアの追加または拡張をしている場合，その変更内容を初期設定オプションの指定に反映する必要があります。反映方法については，「16.18.2　セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項」を参照してください。ワーカーノードの DB ディレクトリを再作成する場合は，セカンダリノードをワーカーノードに読み替えてください。

(2)　マルチノード構成の HADB サーバを再開始できない場合

出力されたメッセージの対処方法に従って原因を特定してください。そのあとに，マルチノード構成の HADB サーバを再開始してください。

原因がすぐに特定できない場合は，障害が発生したプライマリノードまたはセカンダリノードをマルチノード構成から除外してから，マルチノード構成の HADB サーバを再開始してください。障害が発生したプライマリノードまたはセカンダリノードをマルチノード構成から除外するには，全ノードの HADB サーバのサーバ定義adb_sys_multi_node_info オペランドから，障害が発生したプライマリノードまたはセカンダリノードの指定を削除します。

また，障害が発生したノードがプライマリノードの場合は，次のどちらかの方法でプライマリノードを変更してください。

- HA モニタの設定を変更してプライマリノードを変更する
- マルチノード構成の HADB サーバの開始時に，monact コマンドを実行してセカンダリノードの 1 つをプライマリノードに変更する

■KFAA41205-E メッセージが出力された場合

KFAA41205-E メッセージが出力された場合の対処方法については，「(1)　マルチノード構成の HADB サーバを正常開始できない場合」の「■KFAA41205-E メッセージが出力された場合」を参照してください。

(3)　マルチノード構成の HADB サーバを終了できない場合

AP またはコマンドが HADB サーバに接続中の場合は，マルチノード構成の HADB サーバを終了できません。AP またはコマンドの処理が終わるのを待ってください。早急に終了する必要がある場合は，マルチノード構成の HADB サーバを強制終了してください。マルチノード構成の HADB サーバを強制終了する方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(1)　マルチノード構成の HADB サーバの終了手順」の「(b)　マルチノード構成の HADB サーバを強制終了させる場合」を参照してください。

HADB サーバに接続中の AP またはコマンドがないのに，マルチノード構成の HADB サーバを終了できない場合は，次の手順で終了してください。

手順

1. プライマリノードでadbstop --force コマンドを実行する

   プライマリノードの HADB サーバが強制終了すると，プライマリノードの切り替えが発生します。

2. マルチノード構成の HADB サーバを正常終了する

   マルチノード構成の HADB サーバを正常終了する方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(1)　マルチノード構成の HADB サーバの終了手順」の「(a)　マルチノード構成の HADB サーバを正常終了させる場合」を参照してください。

上記の方法でも，マルチノード構成の HADB サーバが終了できない場合は，すべてのノードでadbstop --force コマンドを実行し，マルチノード構成の HADB サーバを強制終了してください。

16.17.2　AP に関するトラブル（マルチノード機能の使用時）

AP に関するトラブルが発生した場合，最初に「15.1　AP に関するトラブル」を参照して対処してください。対処できない場合に，ここの説明をお読みください。

(1)　AP が実行できない場合の対処方法

AP が実行できない場合の考えられる原因と対処方法を次の表に示します。

表 16-10　AP が実行できない場合の考えられる原因と対処方法

項番	考えられる原因	対処方法
1	接続先のホスト名が，HADB クライアントと HADB サーバ間の通信で使用するエイリアス IP アドレスに対応するホスト名でない可能性があります。	HADB クライアントと HADB サーバ間の通信で使用しているエイリアス IP アドレスに対応するホスト名を指定しているか，確認してください。 クライアント定義など，AP の接続先のホスト名を指定している個所を確認して修正してください。
2	AP の同時実行数が，マルチノード構成の HADB サーバの最大同時接続数を超えている可能性があります。	手順 1. マルチノード構成の HADB サーバを終了してください。 2. サーバ定義の adb_sys_max_users オペランドの指定値を大きくしてください。 3. マルチノード構成の HADB サーバを開始してください。

16.17.3　コマンドに関するトラブル（マルチノード機能の使用時）

コマンドに関するトラブルが発生した場合，最初に「15.2　コマンドに関するトラブル」を参照して対処してください。対処できない場合に，ここの説明をお読みください。

(1)　コマンドが実行できない場合の対処方法（KFAA31733-E メッセージが出力された場合）

プライマリノードの切り替えが発生した場合，プライマリノードで実行されていた更新処理に対する回復処理（ロールバック）が，新しくプライマリノードになるセカンダリノードで実行されます。その回復処理中のページにコマンドがアクセスすると，コマンドがエラーになります。このとき，KFAA31733-E メッセージが出力されます。この場合，メッセージに記載されている対処方法に従ってください。

(2)　コマンドが実行できない場合の対処方法（KFAA50036-E メッセージが出力された場合）

セカンダリノードまたはワーカーノードで実行できないコマンドを，セカンダリノードまたはワーカーノードで実行しようとしました。プライマリノードでコマンドを実行してください。

16.17.4　DB エリアに関するトラブル（マルチノード機能の使用時）

データ用 DB エリアに関するトラブルが発生したときの対処方法について説明します。

(1)　データ用 DB エリアが満杯になった場合

データ用 DB エリアを拡張してください。拡張方法については，「16.10.3　データ用 DB エリアの追加・削除・拡張（マルチノード機能の使用時）」の「(3)　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照してください。

16.17.5　DB ディレクトリのファイルに関するトラブル（マルチノード機能の使用時）

DB ディレクトリ下のファイルにトラブルが発生したときの対処方法について説明します。

(1)　システムログファイルに障害が発生したときの対処方法

出力されたメッセージの対処に従ってください。

また，システムディレクトリのファイルシステムが$DBDIR/ADBSYS にマウントされているかどうかを確認してください。

「15.5.2　システムログファイルでトラブルが発生した場合の対処」もあわせて参照してください。

(2)　システムログファイルの容量不足によって，HADB サーバが異常終了したときの対処方法

出力されたメッセージの対処に従ってください。

また，「15.5.2　システムログファイルでトラブルが発生した場合の対処」もあわせて参照してください。

16.17.6　サーバディレクトリ（インストール時）のファイルに関するトラブル（マルチノード機能の使用時）

出力されたメッセージの対策欄の説明に従ってください。

16.17.7　サーバディレクトリ（運用時）のファイルに関するトラブル（マルチノード機能の使用時）

サーバディレクトリ下のファイルにトラブルが発生したときの対処方法について説明します。

(1)　サーバディレクトリに障害が発生したときの対処方法

サーバディレクトリに障害が発生したときの対処方法については，「15.4.2　サーバディレクトリの回復」を参照してください。

(2)　サーバ定義ファイルに障害が発生したときの対処方法

サーバ定義ファイルに障害が発生したときの対処方法については，出力されたメッセージの対策欄の説明に従ってください。

(3)　メッセージログファイルに障害が発生したときの対処方法

メッセージログファイルに障害が発生したときの対処方法については，出力されたメッセージの対策欄の説明に従ってください。

16.17.8　クライアントディレクトリのファイルに関するトラブル（マルチノード機能の使用時）

クライアントディレクトリの回復方法については，「15.4.3　クライアントディレクトリの回復」を参照してください。

16.17.9　OS に関するトラブル（マルチノード機能の使用時）

セカンダリノードの OS が異常終了した場合，そのセカンダリノードはマルチノード構成から切り離されます。

プライマリノードの OS が異常終了した場合，プライマリノードの切り替えが発生します。OS が異常終了したプライマリノードは，マルチノード構成から切り離されます。

ワーカーノードの OS が異常終了した場合，そのワーカーノードはマルチノード構成から削除されます。

OS のリブート後，障害を対処してください。

16.17.10　ハードウェアに関するトラブル（マルチノード機能の使用時）

ハードウェアに関するトラブルが発生したときの対処方法を説明します。

(1)　ディスク障害が発生したときの対処方法

ディスク障害が発生したときの対処方法については，「15.4　ディスクに関するトラブル」を参照してください。

(2)　ディスクが満杯になったときの対処方法

(a)　DB エリアファイルが原因の場合

DB エリアファイルの容量増加によってディスクが満杯になった場合は，次の個所を参照して対処してください。

- 「15.3.1　DB エリアファイルの容量増加が原因の場合」
- 「15.3.2　DB エリアの自動増分の失敗が原因の場合」

(b)　DB エリアファイル以外が原因の場合

- システムログファイルまたはステータスファイルが原因の場合

  「15.3.3　DB エリアファイル以外のファイルの容量増加が原因の場合」の「(1)　DB ディレクトリ下の実体ファイルが原因の場合」を参照して対処してください。
- システムログファイルまたはステータスファイル以外のファイルが原因の場合

  「15.3.3　DB エリアファイル以外のファイルの容量増加が原因の場合」の「(2)　シンボリックリンクしているファイルが原因の場合」を参照して対処してください。

(3)　通信障害，CPU 障害，または電源障害が発生したときの対処方法

通信障害，CPU 障害，または電源障害が発生したときの対処方法について説明します。

(a)　HADB クライアントとプライマリノードの HADB サーバ間で通信障害が発生した場合

この場合，該当するトランザクションが，通信障害が発生した旨のエラーリターンします。対処手順を次に示します。

手順

1. 通信障害が発生したプライマリノードをマルチノード構成から切り離す

   コマンドを実行してプライマリノードを切り替えてください。通信障害が発生したプライマリノードは，マルチノード構成から切り離されます。コマンドによるプライマリノードの切り替え方法については，「16.7　コマンドによるプライマリノードの切り替え」を参照してください。
2. 通信障害の原因を調査して対策する
3. 手順 1.で切り離したノードをマルチノード構成に復帰させる

   マルチノード構成に復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

(b)　HADB クライアントとセカンダリノードまたはワーカーノードの HADB サーバ間で通信障害が発生した場合

この場合，該当するトランザクションが，通信障害が発生した旨のエラーリターンします。

通信障害の原因を調査してください。すぐに対処できない場合は，障害が発生したセカンダリノードまたはワーカーノードの HADB サーバを終了してください。

(c) ノード間で通信障害が発生した場合

ノード間で通信障害が発生した場合の対処方法を説明します。

■プライマリノードまたはセカンダリノードで通信障害が発生した場合

系のリセットを使用している場合と，共有ディスクの SCSI リザーブを使用している場合とで対処方法が異なります。

- **系のリセットを使用している場合**

  ノード間で通信障害が発生した場合，HA モニタはノード障害が発生したと判断します。このとき，マルチノード構成から切り離されるノードを HA モニタが決定します。切り離されるノードにプライマリノードが含まれている場合，プライマリノードの切り替えが発生します。

  なお，マルチノード構成から切り離されたノードのサーバマシンは，HA モニタによって停止されます（サーバマシンの電源断となります）。サーバマシンが停止しなかった場合は，HA モニタの `monswap` コマンドを実行したあとに，サーバマシンを手動で停止してください。

  そのあと，サーバマシンを再起動し，通信障害の原因を調査して対策してください。

  対策後，必要に応じて，そのノードをマルチノード構成に復帰させてください。マルチノード構成に復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

- **共有ディスクの SCSI リザーブを使用している場合**

  ノード間で通信障害が発生した場合，HA モニタはノード障害が発生したと判断します。このとき，マルチノード構成から切り離されるノードを HA モニタが決定します。切り離されるノードにプライマリノードが含まれている場合，プライマリノードの切り替えが発生します。なお，マルチノード構成から切り離されたノードのサーバマシンは，停止しません。

  この場合，通信障害の原因を調査して対策してください。対策後，必要に応じて，そのノードをマルチノード構成に復帰させてください。マルチノード構成に復帰させる方法については，「16.15.3 マルチノード構成へのノードの復帰」を参照してください。

  なお，HA モニタの監視パスに障害が発生した場合，HA モニタの`monlink` コマンドを実行して，監視パスを復旧する必要があります。対処方法の詳細については，マニュアル『HA モニタ Linux(R) (x86)編』の『障害発生による系切り替え時の運用』を参照してください。

■ワーカーノードで通信障害が発生した場合

ワーカーノードとほかのノードの間で通信障害が発生した場合，マルチノード構成からそのワーカーノードが削除されます。

通信障害の原因を調査して対策したあとに，ワーカーノードで`adbstart` コマンドを実行し，ワーカーノードをマルチノード構成に追加してください。

(d)　CPU 障害が発生した場合

■プライマリノードまたはセカンダリノードで CPU 障害が発生した場合

プライマリノードで CPU 障害が発生した場合，プライマリノードの切り替えが発生します。CPU 障害が発生したプライマリノードは，マルチノード構成から切り離されます。

セカンダリノードで CPU 障害が発生した場合，そのセカンダリノードはマルチノード構成から切り離されます。

OS のリブート後，原因を調査して対策してください。対策後，切り離されたノードをマルチノード構成に復帰させてください。マルチノード構成に復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

■ワーカーノードで CPU 障害が発生した場合

ワーカーノードで CPU 障害が発生した場合，そのワーカーノードはマルチノード構成から削除されます。

OS のリブート後，原因を調査して対策してください。対策後，ワーカーノードで`adbstart` コマンドを実行し，ワーカーノードをマルチノード構成に追加してください。

(e)　電源障害が発生した場合

■プライマリノードまたはセカンダリノードで電源障害が発生した場合

プライマリノードで電源障害が発生した場合，プライマリノードの切り替えが発生します。電源障害が発生したプライマリノードは，マルチノード構成から切り離されます。

セカンダリノードで電源障害が発生した場合，そのセカンダリノードはマルチノード構成から切り離されます。

OS のリブート後，原因を調査して対策してください。対策後，切り離されたノードをマルチノード構成に復帰させてください。マルチノード構成に復帰させる方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

■ワーカーノードで電源障害が発生した場合

ワーカーノードで電源障害が発生した場合，そのワーカーノードはマルチノード構成から削除されます。

OS のリブート後，原因を調査して対策してください。対策後，ワーカーノードで`adbstart` コマンドを実行し，ワーカーノードをマルチノード構成に追加してください。

16.17.11　HA モニタに関するトラブル（マルチノード機能の使用時）

HA モニタに関するトラブルが発生した場合は，マニュアル『HA モニタ Linux(R)(x86)編』を参照して対処してください。

16.17.12　作業表を作成するSQL文に関するトラブル（マルチノード機能の使用時）

HADBサーバが異常終了したときに実行していたSQL文が，作業表を作成するSQL文の場合は，作業表用DBエリアファイルが使用するディスクを，複数のノードで共有して使用していないことを確認してください。作業表用DBエリアファイルが使用するディスクを，複数のノードで共有して使用していた場合，HADBサーバはその動作を保証しません。この場合，不正な検索結果が表示されたり，HADBサーバが異常終了したりするおそれがあります。

作業表用DBエリアファイルとして使用するディスクを，複数のノードで共有して使用していた場合は，次に示す手順で対処してください。

手順

1. マルチノード構成のHADBサーバを終了させてください。
2. 各ノードに対して，作業表用DBエリアファイルが使用するディスクを準備してください。
3. 各ノードのサーバ定義adb_blk_path_wrkオペランドに対して，手順2.で準備したディスクのパスを指定してください。
4. マルチノード構成のHADBサーバを再度開始してください。

メモ

実行したSQL文が作業表を作成するSQL文かどうかは，マニュアル『HADB　AP開発ガイド』の『作業表が作成されるSQLを実行する際の考慮点』を参照してください。

16.17.13　同義語辞書ファイルに関するトラブル（マルチノード機能の使用時）

ここでは，同義語辞書ファイルに対して発生したトラブルの対処方法について説明します。

(1)　マルチノード機能で使用する同義語辞書ファイルに障害が発生したときの対処方法

マルチノード機能で使用する同義語辞書ファイルに障害が発生した場合は，次の方法で対処してください。

メモ

マルチノード機能で使用する同義語辞書ファイルは，サーバ定義のadb_syndict_node_storage_pathオペランドに指定したディレクトリ下に格納されています。

対処方法 1

同義語辞書ファイルに障害が発生した場合，障害が発生したセカンダリノードまたはワーカーノードのメッセージログファイルに次のメッセージが出力されます。

- KFAA30959-E
- KFAA34008-E
- KFAA51537-W

上記のメッセージの対策欄に記載されている対処方法に従ってください。上記のメッセージの対策欄に記載されている対処方法に従っても問題が解決しない場合，またはその対処方法が実行できない場合は，対処方法 2 を実施してください。

対処方法 2

障害が発生した同義語辞書ファイルを削除してください。そのあとに，adbsyndict コマンドを実行して，同義語辞書ファイルの同期を取ってください。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

(2)　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの空き容量が不足したときの対処方法

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの空き容量不足が原因で，adbsyndict コマンドが異常終了した場合，次のメッセージが出力されます。

- プライマリノードにKFAA51519-E メッセージが出力されます。
- 容量不足が発生したノードにKFAA51514-E メッセージが出力されます。

KFAA51514-E メッセージに出力される操作対象の情報がsynonym-dictionary-file で，かつエラー番号が28（ENOSPC）の場合，同義語辞書ファイルの格納ディレクトリの空き容量が不足していると考えられます。この場合，まずは対処方法 1 を実施してください。対処方法 1 が実施できない場合，または対処方法 1 を実施しても容量不足が発生する場合は，対処方法 2 を実施してください。

> **メモ**
>
> マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリは，サーバ定義の adb_syndict_node_storage_path オペランドに指定したディレクトリのことです。

対処方法 1

adbsyndict コマンドの実行時，複数の同義語辞書を一度に更新している場合は，同義語辞書を 1 つずつ更新してください。一度に更新する同義語辞書の数を減らすと，同義語辞書ファイルの格納ディレクトリに必要な空き容量が少なく済むため，adbsyndict コマンドを実行できることがあります。

対処方法 2

容量の大きい別のディスク上に，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成し直してください。手順を次に示します。

手順

1. 容量の大きいディスクに，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを新たに作成してください。ディレクトリの作成方法については，「16.26.1　同義語検索を行うための準備をする」の「(6)　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する」を参照してください。
2. 容量不足が発生したノードの HADB サーバを正常終了してください。特定のノードの HADB サーバを正常終了させる方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(2)　特定のノードの HADB サーバの終了手順」を参照してください。
3. HADB サーバを正常終了したノードのサーバ定義のadb_syndict_node_storage_path オペランドの指定を変更してください。手順の 1.で作成したディレクトリを指定します。
4. 手順の 2.で正常終了した HADB サーバを正常開始してください。
   - 手順の 2.で，プライマリノードまたはセカンダリノードの HADB サーバを正常終了した場合
     マルチノード構成にノードを復帰させてください。復帰手順については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。
   - 手順の 2.で，ワーカーノードの HADB サーバを正常終了した場合
     ワーカーノードでadbstart コマンドを実行して，マルチノード構成にワーカーノードを追加してください。
5. adbsyndict コマンドを実行して，同義語辞書ファイルの同期を取ってください。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。
6. 変更前のマルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを削除してください。

16.18　セカンダリノードの新規追加，または不要なセカンダリノードの削除

ここでは，サーバマシンの追加によるセカンダリノードの新規追加方法について説明します。また，不要となったセカンダリノードを削除する方法についても説明します。

16.18.1　セカンダリノードの新規追加

サーバマシンの追加によるセカンダリノードの新規追加手順を次に示します。

手順

1. サーバマシンを追加して，HADB サーバのインストールおよび環境設定を行う

   HADB サーバのインストールおよび環境設定については，「16.3.2　HADB サーバのインストールおよび環境設定」を参照してください。

   > **重要**
   >
   > adbinit コマンドを実行する前に，「16.18.2　セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項」で説明している内容を確認してください。

2. マルチノード構成の HADB サーバを正常終了する

   マルチノード構成の HADB サーバの正常終了手順については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(1)　マルチノード構成の HADB サーバの終了手順」の「(a)　マルチノード構成の HADB サーバを正常終了させる場合」を参照してください。

3. HA モニタの定義を修正する

   プライマリノードおよびセカンダリノードの HA モニタの定義に，追加するセカンダリノードの情報を指定します。

4. サーバ定義を修正する

   全ノードの HADB サーバのサーバ定義adb_sys_multi_node_info オペランドに，追加するセカンダリノードの情報を追加します。

5. マルチノード構成の HADB サーバを開始する

   マルチノード構成の HADB サーバの開始手順については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

> **重要**
>
> 同義語検索を行っている場合，セカンダリノードを追加したあとに同義語辞書ファイルの同期を取る必要があります。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

16.18.2　セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項

セカンダリノードを新規に追加する場合，追加するセカンダリノードでadbinit コマンドを実行しますが，ほかのノードで使用した初期設定オプションをそのまま利用できないことがあります。DB エリアの構成を変更している場合，その変更内容を初期設定オプションの指定に反映して，adbinit コマンドを実行する必要があります。そのため，新規に追加するセカンダリノードで使用する初期設定オプションに，DB エリアの構成変更の差分情報を反映する必要があります。それができない場合は，ここで説明する手順に従って初期設定オプションを作成してください。

ここで説明する手順のおおまかな流れは，最初にコマンドを実行して，現在の DB エリアの情報をファイルに抽出します。そのファイルを編集して，初期設定オプションファイルを作成します。

重要

- 説明中の$DBDIR は，DB ディレクトリを意味しています。実際にコマンドを実行する際は，DB ディレクトリのパスを指定してください。
- 以降の手順は，プライマリノードで実行してください。

(1)　初期設定オプションファイルの基を作成する

```
echo ”set adb_init_mst_blk_path = ¥”$(ls -l $DBDIR/ADBMST ¥
| sed -e ’s/.*ADBMST ->/ADBMST ->/g’ | awk ’{print $3}’)¥”” >adbinit.opt
```

上記のコマンドを実行し，初期設定オプションファイル（adbinit.opt ファイル）の基を作成してください。この例では，カレントディレクトリにadbinit.opt ファイルを作成しています。

上記のコマンドを実行すると，次に示すadbinit.opt ファイルが作成されます。

■この時点でのadbinit.opt ファイルの内容

```
set adb_init_mst_blk_path = ”/dev/disk/by-id/wwn-0x60....93bc”
```

(2)　adbinit.opt ファイルに adbinitdbarea オペランドなどの情報を追加する

```
adbdbstatus -d used | grep Used | awk -F’,’ ’{print $3, $9}’ | uniq ¥
| awk ’{print $1, $2}’ ¥
| sed -e ’s/DBarea_name DBarea_filename/set adb_init_dbarea_initialize = N/g’ ¥
      -e ’s/¥”ADBDIC¥”/set adb_init_dic_blk_path =/g’ ¥
      -e ’s/¥”ADBSTBL¥”/set adb_init_stbl_blk_path =/g’ ¥
| sed -e ’s/^¥”¥(.*¥)¥” ¥”¥(.*¥)¥”/adbinitdbarea -n ¥¥¥”¥1¥¥¥” -i 0K,x -v ¥2,¥¥/g’ ¥
>>adbinit.opt
```

上記のadbdbstatus コマンドを実行すると，adbinitdbarea オペランドなどの情報が，(1)で作成したadbinit.opt ファイルに追加されます。上記のコマンドを実行すると，ある程度整った形でadbinit.opt ファイルが作成されるため，(3)以降でadbinit.opt ファイルを編集する手間が軽減できます。

■この時点でのadbinit.opt ファイルの内容

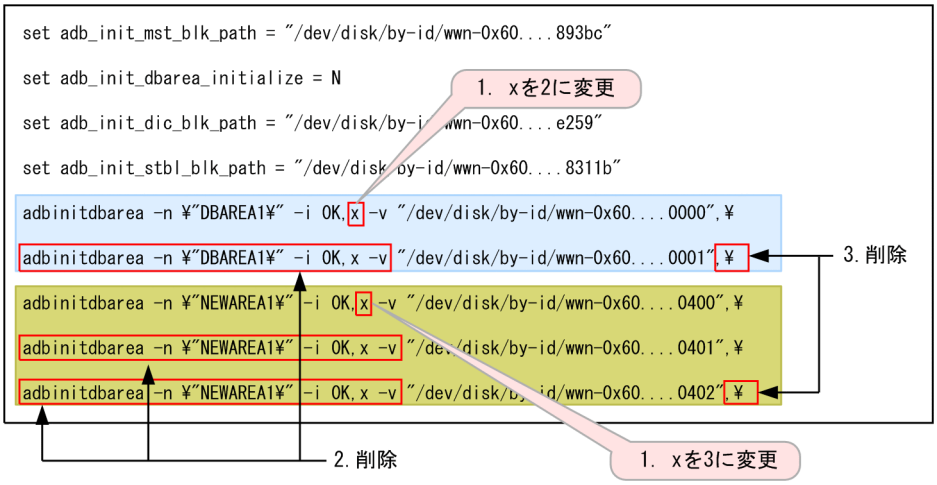
```
set adb_init_mst_blk_path = ”/dev/disk/by-id/wwn-0x60....93bc”
set adb_init_dbarea_initialize = N                                          ...1
set adb_init_dic_blk_path = ”/dev/disk/by-id/wwn-0x60....e259”              ...2
set adb_init_stbl_blk_path = ”/dev/disk/by-id/wwn-0x60....311b”             ...3
adbinitdbarea -n ¥”DBAREA1¥” -i 0K,x -v ”/dev/disk/by-id/wwn-0x60....0000”,¥  ...4
adbinitdbarea -n ¥”DBAREA1¥” -i 0K,x -v ”/dev/disk/by-id/wwn-0x60....0001”,¥  ...5
adbinitdbarea -n ¥”NEWAREA1¥” -i 0K,x -v ”/dev/disk/by-id/wwn-0x60....0400”,¥ ...6
adbinitdbarea -n ¥”NEWAREA1¥” -i 0K,x -v ”/dev/disk/by-id/wwn-0x60....0401”,¥ ...7
adbinitdbarea -n ¥”NEWAREA1¥” -i 0K,x -v ”/dev/disk/by-id/wwn-0x60....0402”,¥ ...8
```

上記の 1～8 のオペランドが追加されます。

(3)　adbinit.opt ファイルの内容を修正する

adbinit.opt ファイルをテキストエディタで開いて，初期設定オプションの指定を修正してください。初期設定オプションの指定については，マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』の『adbinit コマンドの指定形式』の『初期設定オプションの形式』を参照してください。

この例の場合，adbinitdbarea オペランドのオプションの指定を修正する必要があります。



[説明]

adbinitdbarea オペランドの指定を次のように修正します。

- adbinitdbarea オペランドの-n オプションの DB エリア名が重複している行がある場合
  上記の図の 1.～3.を実施します。
- adbinitdbarea オペランドの-n オプションの DB エリア名が重複している行がない場合
  上記の図の 1.および 3.を実施します。

上記の図の番号と次の説明の番号が対応しています。

1. -i オプションのファイル数の指定値を，x から重複行の数に変更してください。この例の場合，2と3 に変更します。
2. 囲まれた個所の指定を削除してください。
3. 各重複行の最終行の末尾の「,¥」を削除してください。

■adbinit.opt ファイルの修正後の内容

```
set adb_init_mst_blk_path = ”/dev/disk/by-id/wwn-0x60....93bc”
set adb_init_dbarea_initialize = N
set adb_init_dic_blk_path = ”/dev/disk/by-id/wwn-0x60....e259”
set adb_init_stbl_blk_path = ”/dev/disk/by-id/wwn-0x60....311b”
adbinitdbarea -n ¥”DBAREA1¥” -i 0K,2 -v ”/dev/disk/by-id/wwn-0x60....0000”,¥
                                        ”/dev/disk/by-id/wwn-0x60....0001”
adbinitdbarea -n ¥”NEWAREA1¥” -i 0K,3 -v ”/dev/disk/by-id/wwn-0x60....0400”,¥
                                         ”/dev/disk/by-id/wwn-0x60....0401”,¥
                                         ”/dev/disk/by-id/wwn-0x60....0402”
```

上記の網掛け部分が修正されています。

(4) DB ディレクトリ直下のファイルを確認する

DB ディレクトリのサブディレクトリに，DB エリアファイルのシンボリックリンクを格納している場合は，その DB エリアの情報を指定しているadbinitdbarea オペランドに，-f オプションの指定を追加する必要があります。この場合は，以降の手順を実施してください。

```
ls $DBDIR
```

■実行結果の表示例

```
ADBDIC   ADBMST   ADBSTBL   ADBSYS   ADBWORK   ADBWRK   AREAS   DBAREA1   DBAREA1.00001   SPOOL
```

OS のls コマンドで，DB ディレクトリ直下のファイル一覧を表示してください。DB エリア名と同じ名称のファイルがあるかどうかを確認してください。

上記の例の場合，DBAREA1 はありますが，NEWAREA1 はありません。このため，NEWAREA1 を指定しているadbinitdbarea オペランドに-f オプションを追加する必要があります。

(5) NEWAREA1 の格納ディレクトリを調査する

```
find $DBDIR -name NEWAREA1 2>/dev/null
```

上記のコマンドを実行して，DB エリアファイルNEWAREA1 の，DB ディレクトリ内の格納ディレクトリを調査します。

■実行結果の表示例

```
$DBDIR/AREAS/NEW/NEWAREA1       ...1
$DBDIR/AREAS/OLD/NEWAREA1       ...2
```

以降の説明で，1 を 1 のファイル，2 を 2 のファイルと表記します。

上記の例のように，パスは異なるが，同じファイル名が複数表示された場合は，OS の ls コマンドでリンク先を調査する必要があります。

(6)　1 のファイルのリンク先を確認する

```
ls -l $DBDIR/AREAS/NEW/NEWAREA1* | sed -e 's/.* -> //g'
```

OS の ls -l コマンドを実行して，1 のファイルのリンク先を確認してください。

■実行結果の表示例

```
/dev/disk/by-id/wwn-0x60....0400
/dev/disk/by-id/wwn-0x60....0401
/dev/disk/by-id/wwn-0x60....0402
```

上記 3 つのリンク先が，adbinit.opt ファイルの内容と一致しています。

■adbinit.opt ファイルの内容

```
adbinitdbarea -n ¥"NEWAREA1¥" -i 0K,3 -v "/dev/disk/by-id/wwn-0x60....0400",¥
                                         "/dev/disk/by-id/wwn-0x60....0401",¥
                                         "/dev/disk/by-id/wwn-0x60....0402"
```

(7)　2 のファイルのリンク先を確認する

```
ls -l $DBDIR/AREAS/OLD/NEWAREA1* | sed -e 's/.* -> //g'
```

OS の ls -l コマンドを実行して，2 のファイルのリンク先を確認してください。

■実行結果の表示例

```
/dev/disk/by-id/wwn-0x60....a000
```

上記のリンク先は，adbinit.opt ファイルの内容と一致していません。

(8)　初期設定オプションの指定内容を修正する

(6)および(7)の確認結果から，1 のファイルが該当することがわかりました。

1 のファイルのパス名のうち，下記の下線部分を-f オプションに指定して追加してください。

```
$DBDIR/AREAS/NEW/NEWAREA1
```

上記の例の場合，次のように-f オプションを追加します。

```
-f ¥"AREAS/NEW/¥"
```

■adbinit.opt ファイルの修正後の内容

```
set adb_init_mst_blk_path = "/dev/disk/by-id/wwn-0x60....93bc"
set adb_init_dbarea_initialize = N
set adb_init_dic_blk_path = "/dev/disk/by-id/wwn-0x60....e259"
set adb_init_stbl_blk_path = "/dev/disk/by-id/wwn-0x60....311b"
adbinitdbarea -n ¥"DBAREA1¥" -i 0K,2 -v "/dev/disk/by-id/wwn-0x60....0000",¥
                                        "/dev/disk/by-id/wwn-0x60....0001"
adbinitdbarea -n ¥"NEWAREA1¥" -f ¥"AREAS/NEW/¥" -i 0K,3¥
              -v "/dev/disk/by-id/wwn-0x60....0400",¥
                 "/dev/disk/by-id/wwn-0x60....0401",¥
                 "/dev/disk/by-id/wwn-0x60....0402"
```

上記の下線部分を追加します。

(9)　作業表用 DB エリアに関するオペランドを追加する

次のオペランドを必要に応じて追加してください。

- adb_init_wrk_page_size
- adb_init_wrk_blk_path

adb_init_wrk_blk_path オペランドには，ノードごとに割り当てる作業表用 DB エリアファイルのブロックスペシャルファイル名を指定します。詳細については，「16.3.8　データベースの作成」の「(4)　セカンダリノードおよびワーカーノードでの DB ディレクトリの作成」を参照してください。

以上で初期設定オプションの修正が完了しました。セカンダリノードを新規に追加する際，ここで作成した初期設定オプションを使用してadbinit コマンドを実行してください。

16.18.3　セカンダリノードの削除

不要となったセカンダリノードの削除手順を次に示します。

手順

1. マルチノード構成の HADB サーバを正常終了する

   マルチノード構成の HADB サーバの正常終了手順については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(1)　マルチノード構成の HADB サーバの終了手順」の「(a)　マルチノード構成の HADB サーバを正常終了させる場合」を参照してください。

2. サーバ定義を修正する

   全ノードの HADB サーバのサーバ定義adb_sys_multi_node_info オペランドから，削除するセカンダリノードの情報を削除します。

3. セカンダリノードの情報を削除する

   プライマリノードおよびセカンダリノードの HA モニタの定義から，削除するセカンダリノードの情報を削除します。

4. マルチノード構成の HADB サーバを開始する

   マルチノード構成の HADB サーバの開始手順については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

16.19　ワーカーノードの新規追加，または不要となったワーカーノードの削除

ここでは，サーバマシンの追加によるワーカーノードの新規追加方法について説明します。また，不要となったワーカーノードを削除する方法についても説明します。

16.19.1　ワーカーノードの新規追加

サーバマシンの追加によるワーカーノードの新規追加手順を次に示します。

なお，次の手順を実施する前に，サーバ定義のadb_sys_multi_max_worker_nodes オペランドの指定値を確認してください。このオペランドの指定値を超える数のワーカーノードをマルチノード構成に追加することはできません。また，このオペランドの指定がない場合は，このオペランドを追加してください。このオペランドの指定値を変更する場合，またはこのオペランドを追加する場合は，マルチノード構成の HADB サーバをいったん終了してください。

手順

1. サーバマシンを追加して，HADB サーバのインストールおよび環境設定を行う

   HADB サーバのインストールおよび環境設定については，「16.3.2　HADB サーバのインストールおよび環境設定」を参照してください。

2. 追加するワーカーノードのサーバ定義を作成する

   ワーカーノードのサーバ定義には，adb_sys_multi_worker_node_info オペランドを必ず指定してください。

   ワーカーノードのサーバ定義の指定例については，「16.3.5　各ノードでのサーバ定義の作成」を参照してください。

3. adbinit コマンドを実行して，追加するワーカーノードの DB ディレクトリを作成する

   重要

   adbinit コマンドを実行する前に，「16.18.2　セカンダリノードの新規追加時に adbinit コマンドを実行する際の留意事項」で説明している内容を確認してください。その際，セカンダリノードをワーカーノードに読み替えてください。

4. adbstart コマンドを実行して，ワーカーノードの HADB サーバを開始する

   adbstart コマンドが正常終了すると，マルチノード構成にワーカーノードが追加されます。

重要

同義語検索を行っている場合，ワーカーノードを追加したあとに同義語辞書ファイルの同期を取る必要があります。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

16.19.2　不要となったワーカーノードの削除

不要となったワーカーノードの削除手順を次に示します。

手順

1. 削除するワーカーノードで`adbstop --node`コマンドを実行する
   ワーカーノードの HADB サーバが正常終了し，マルチノード構成からワーカーノードが削除されます。

16.20　サーバマシンの OS のホスト名・IP アドレスの変更（マルチノード機能使用時）

ここでは，マルチノード機能を使用している場合に，HADB サーバがインストールされているサーバマシンの OS のホスト名または IP アドレスを変更する場合の手順について説明します。

■マルチノード構成の HADB サーバで実施する作業

サーバマシンの OS のホスト名または IP アドレスを変更する場合に，マルチノード構成の HADB サーバで実施する作業を次に示します。

1. マルチノード構成の HADB サーバを正常終了する

   全ノードの HADB サーバを正常終了してください。終了方法については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(1)　マルチノード構成の HADB サーバの終了手順」の「(a)　マルチノード構成の HADB サーバを正常終了させる場合」を参照してください。

   また，プライマリノードおよびセカンダリノードの HA モニタについても，monstop コマンドで正常停止してください。

2. プライマリノードおよびセカンダリノードの HA モニタの設定を変更する

   「16.3.4　HA モニタの環境設定」を参照して，プライマリノードおよびセカンダリノードの HA モニタの設定を変更してください。

   - HA モニタの監視パスとして使用する LAN のホスト名を変更する場合

     HA モニタの環境設定ファイル（sysdef ファイル）の設定を変更してください。

   - エイリアス IP アドレスを変更する場合

     LAN 接続時の状態設定ファイル（*サーバ識別名*.up ファイル），および LAN 切り離し時の状態設定ファイル（*サーバ識別名*.down ファイル）の設定を変更してください。

3. 全ノードのサーバ定義を変更する

   「16.3.5　各ノードでのサーバ定義の作成」を参照して，変更が必要なオペランドについて，全ノードのサーバ定義を変更してください。

4. マルチノード構成の HADB サーバを開始する

   プライマリノードおよびセカンダリノードの HA モニタを，monstart コマンドで起動してください。そのあと，マルチノード構成の HADB サーバを開始してください。開始方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

■HADB クライアントで実施する作業

サーバマシンの OS のホスト名または IP アドレスを変更する場合に，HADB クライアントで実施する作業を次に示します。HADB クライアントと HADB サーバ間の通信で使用するエイリアス IP アドレスを変更する場合に実施してください。

1. AP を停止する

   マルチノード構成の HADB サーバに接続する AP をすべて停止してください。

2. クライアント定義に指定しているエイリアス IP アドレスを変更する

HADB クライアントと HADB サーバ間の通信で使用するエイリアス IP アドレスを変更する場合は，「16.3.6　クライアント定義の作成」を参照して，クライアント定義adb_clt_rpc_srv_host オペランドの値を変更してください。

クライアント定義adb_clt_rpc_srv_host オペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』の『クライアント定義のオペランドの内容』を参照してください。

- JDBC ドライバを使用している場合

  クライアント定義adb_clt_rpc_srv_host オペランドに対応するプロパティで指定している HADB サーバのホスト名または IP アドレスを変更してください。JDBC ドライバで使用するプロパティは，システムプロパティ，ユーザプロパティ，または接続用の URL のプロパティがあります。詳細については，マニュアル『HADB　AP 開発ガイド』の『JDBC ドライバの環境設定』を参照してください。

- ODBC ドライバまたは CLI 関数を使用している場合

  クライアント定義adb_clt_rpc_srv_host オペランドに指定している HADB サーバのホスト名または IP アドレスを変更してください。詳細については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

3. AP を開始する

マルチノード構成の HADB サーバに接続する AP を開始してください。

重要

マルチノード機能を使用していない場合は，「8.11　サーバマシンの OS のホスト名・IP アドレスの変更」を参照してください。

16.21　マルチノード機能を使用するシステムへの移行

ここでは，マルチノード機能を使用するシステムへの移行手順を説明します。

> **重要**
>
> 移行を行う際は，以降の手順の説明にあるように，サーバディレクトリのバックアップ，およびDBディレクトリのバックアップを必ず取得してください。手順を誤ってHADBサーバが開始できなくなった場合，サーバディレクトリおよびDBディレクトリをバックアップから回復する必要があります。

16.21.1　HADBサーバを正常終了する

マルチノード機能を使用するシステムに移行する前に，HADBサーバを正常終了してください。

HADBサーバがすでに終了している場合は，次に示す2点を確認して，HADBサーバが正常終了していることを確認してください。

1. adbls -d srv コマンドの出力項目「STATUS」の表示がSTOPになっているかどうか
2. メッセージログファイルまたはsyslogに出力されているKFAA91154-Iメッセージの終了モードが，normally（正常終了）になっているかどうか

HADBサーバが正常終了していない場合は，adbstartコマンドを実行してHADBサーバを開始したあとで，adbstopコマンドを実行して正常終了させてください。

16.21.2　サーバディレクトリのバックアップを取得する

HADBサーバが異常終了している状態で，誤ってマルチノード機能を使用するシステムに移行してしまうと，HADBサーバを開始できなくなります。そのため，事前にサーバディレクトリ下のファイルのバックアップを取得してください。

取得したバックアップは，移行後のシステムでの動作確認後に削除してください。

16.21.3　DBディレクトリのバックアップを取得する

マルチノード機能を使用するシステムに移行する前には，必ずフルバックアップを取得してください。取得したフルバックアップは，移行後のシステムでの動作確認後に削除してください。

フルバックアップの取得方法については，「10.3.1　バックアップの取得方法」を参照してください。

16.21.4　HADB サーバをバージョンアップする

移行前の HADB サーバが，マルチノード機能がサポートされる前のバージョン（02-02 以前）の場合は，マルチノード機能を使用するために，HADB サーバをバージョンアップしてください。

HADB サーバのバージョンアップは，マルチノード機能を使用するシステムに移行する前に必ず実施してください。誤って，バージョンアップを実施しないで HADB サーバのライブラリを入れ替えて，マルチノード機能を使用するシステムに移行してしまうと，HADB サーバを開始できません。

HADB サーバのバージョンアップについては，「8.6　HADB サーバのバージョンアップ」を参照してください。

16.21.5　HA モニタおよび HADB サーバをインストールする

プライマリノードおよびセカンダリノードのサーバマシンに，HA モニタをインストールしてください。また，全ノードのサーバマシンに HADB サーバをインストールしてください。HA モニタのインストールについては，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの構築』を，HADB サーバのインストールについては，「8.2　インストール」を参照してください。

> **メモ**
>
> ワーカーノードは HA モニタの管理対象外のため，ワーカーノードのサーバマシンには HA モニタをインストールする必要はありません。

16.21.6　各ノードの設定をする

次のことを行ってください。

- 全ノードで環境変数の指定を変更する
  詳細については，「16.3.2　HADB サーバのインストールおよび環境設定」を参照してください。
- プライマリノードおよびセカンダリノードで HA モニタの環境設定をする
  詳細については，「16.3.4　HA モニタの環境設定」を参照してください。
- 全ノードでサーバ定義を作成する
  詳細については，「16.3.5　各ノードでのサーバ定義の作成」を参照してください。
- クライアント定義を作成する
  詳細については，「16.3.6　クライアント定義の作成」を参照してください。

16.21.7　DB ディレクトリを準備する

次に示す手順に従って，DB ディレクトリを準備します。

なお，移行前の HADB サーバで，作業表用 DB エリアファイル以外の DB エリアファイルでレギュラーファイルを使用していた場合は，ブロックスペシャルファイルに変更する必要があります。変更方法については，「16.10.4　DB エリアの格納位置の変更」を参照してください。

1. システムディレクトリ用のファイルシステムを準備する

   システムディレクトリとして使用するファイルシステムを準備します。システムディレクトリとして使用するファイルシステムについては，「16.2.4　ストレージ構成」の「(2)　ノード切り替え対象のファイルシステム」を参照してください。

2. 同義語辞書ファイルを格納するファイルシステムを準備する

   同義語辞書ファイルを格納するファイルシステムについては，「16.2.4　ストレージ構成」の「(2)　ノード切り替え対象のファイルシステム」を参照してください。

   同義語検索を行っている場合に限り，この作業を実施してください。

3. 追加したノードで使用する作業表用 DB エリア用のディスクを準備する

   ノードごとに，作業表用 DB エリアとして使用するディスクを準備してください。作業表用 DB エリアと使用するディスクについては，「16.2.4　ストレージ構成」の「(3)　DB エリアファイル用のディスク」を参照してください。

4. 追加したノードでadbinit コマンドを実行する

   追加したノードで，adbinit コマンドを実行して DB ディレクトリを作成します。

   このとき，次に示す点に留意してください。

   - 初期設定オプションのadb_init_dbarea_initialize オペランドにN を指定し，DB ディレクトリの枠組みだけを作成してください。
   - 初期設定オプションのadb_init_wrk_blk_path オペランドに指定するブロックスペシャルファイルのパスは，手順 2.で準備した作業表用 DB エリアとして使用するディスクを指定してください。
   - 初期設定オプションのadb_init_mst_blk_path オペランド，adb_init_dic_blk_path オペランド，adb_init_stbl_blk_path オペランド，およびadbinitdbarea に指定するブロックスペシャルファイルのパスは，移行前の HADB サーバで使用していた DB エリアファイルのブロックスペシャルファイルのパスを指定してください。

16.21.8　監査証跡機能を使用するための準備を行う

移行前のシステムで監査証跡機能を使用していた場合は，ここで説明する作業を実施してください。

1. 監査証跡の出力先ディレクトリを作成するファイルシステムを準備する

監査証跡の出力先ディレクトリを作成するファイルシステムを，ノードごとに準備してください。監査証跡の出力先ディレクトリを作成するファイルシステムについては，「16.27.1　監査証跡機能の環境設定」を参照してください。

2. 共有のファイルシステムに監査証跡の保存先ディレクトリを作成する

   監査証跡を検索する際に使用するADB_AUDITREAD関数中に指定する監査証跡ファイルが，全ノードから同じ絶対パスで参照できるように環境を構築する必要があります。そのため，全ノードで共有のファイルシステムに監査証跡の保存先ディレクトリを作成してください。詳細については，「16.27.1　監査証跡機能の環境設定」を参照してください。

16.21.9　マルチノード構成の HADB サーバを開始する

マルチノード構成の HADB サーバを開始してください。開始方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

16.21.10　同義語検索の準備を行う

移行前のシステムで同義語検索を行っていた場合は，ここで説明する作業を実施してください。

1. 同義語辞書ファイルを再作成する

   サーバ定義のadb_syndict_storage_pathオペランドの指定値を変更した場合は，すべての同義語辞書ファイルを再作成してください。同義語辞書ファイルを再作成する方法については，「15.14.3　同義語辞書ファイルまたは同義語辞書ファイルの格納ディレクトリを誤って削除してしまったときの対処方法」を参照してください。

   なお，adb_syndict_storage_pathオペランドの指定に変更はなくても，ファイルシステム（ディスク）のマウントによって，同義語辞書ファイルの格納ディレクトリを作成したファイルシステムを変更した場合は，新しいディレクトリ下にすべての同義語辞書ファイルをコピーしてください。

2. 同義語辞書ファイルの同期を取る

   adbsyndictコマンドを実行して，同義語辞書ファイルの同期を取ってください。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

16.22　HADB サーバのバージョンアップ（マルチノード機能の使用時）

マルチノード機能を使用している HADB サーバをバージョンアップする場合は，「8.6　HADB サーバのバージョンアップ」を参照してください。

16.23　修正版 HADB サーバとの入れ替え（マルチノード機能の使用時）

次に示すどちらかの方法で，修正版 HADB サーバに入れ替えることができます。

- マルチノード構成の HADB サーバを正常終了して入れ替えを実施する方法
- ノードごとに入れ替えを実施する方法

16.23.1　マルチノード構成の HADB サーバを正常終了して入れ替えを実施する方法

マルチノード構成の HADB サーバを正常終了してから，修正版 HADB サーバに入れ替える手順を次に示します。

手順

1. マルチノード構成の HADB サーバを正常終了する

   マルチノード構成の HADB サーバを正常終了する手順については，「16.4.2　マルチノード構成の HADB サーバの終了方法」の「(1)　マルチノード構成の HADB サーバの終了手順」の「(a)　マルチノード構成の HADB サーバを正常終了させる場合」を参照してください。
2. 修正版 HADB サーバとの入れ替えを全ノードで実施する

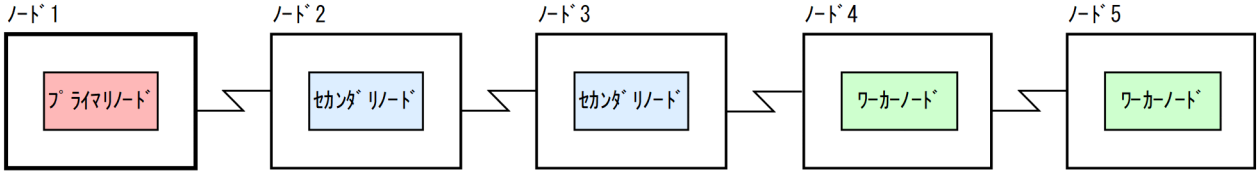
   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。
3. マルチノード構成の HADB サーバを正常開始する

   マルチノード構成の HADB サーバを正常開始する手順については，「16.4.1　マルチノード構成の HADB サーバの開始方法」の「(1)　マルチノード構成の HADB サーバの開始手順」を参照してください。

16.23.2　ノードごとに入れ替えを実施する方法

ノードごとに修正版 HADB サーバに入れ替える手順を次に示します。この方法の場合，マルチノード構成の HADB サーバを停止する必要がないため，業務を停止する必要がありません。

なお，入れ替えを実施するマルチノード構成は，次のとおりとします。



入れ替え作業の流れは，最初にプライマリノードで入れ替え作業を実施し，次にセカンダリノードで入れ替え作業を実施します。最後にワーカーノードで入れ替え作業を実施します。

手順

1. コマンドによるプライマリノードの切り替えを実施する

   コマンドによるプライマリノードの切り替え方法については，「16.7　コマンドによるプライマリノードの切り替え」を参照してください。

   プライマリノードの切り替えが完了すると，ノード 1 の HADB サーバが正常終了し，ノード 2 がプライマリノードになります。

2. 修正版 HADB サーバとの入れ替えをノード 1 で実施する

   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。

3. ノード 1 をマルチノード構成に復帰させる

   マルチノード構成への復帰方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

   ノード 1 は，セカンダリノードとしてマルチノード構成に復帰します。

4. コマンドによるプライマリノードの切り替えを実施する

   コマンドによるプライマリノードの切り替え方法については，「16.7　コマンドによるプライマリノードの切り替え」を参照してください。

   プライマリノードの切り替えが完了すると，ノード 2 の HADB サーバが正常終了し，ノード 1 がプライマリノードになります。

5. 修正版 HADB サーバとの入れ替えをノード 2 で実施する

   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。

6. ノード 2 をマルチノード構成に復帰させる

   マルチノード構成への復帰方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

   ノード 2 は，セカンダリノードとしてマルチノード構成に復帰します。

7. ノード 3 の HADB サーバを正常終了する

   次のコマンドをノード 3 のセカンダリノードで実行してください。

   ```
   adbstop --node
   ```

   ノード 3 のセカンダリノードがマルチノード構成から切り離されます。

8. 修正版 HADB サーバとの入れ替えをノード 3 で実施する

   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。

9. ノード 3 をマルチノード構成に復帰させる

マルチノード構成への復帰方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

ノード３は，セカンダリノードとしてマルチノード構成に復帰します。

> **メモ**
>
> セカンダリノードが３つ以上ある場合は，上記の手順の 7.～9.を繰り返し実施してください。

10. ノード４の HADB サーバを正常終了する

次のコマンドをノード４のワーカーノードで実行してください。

```
adbstop --node
```

ノード４のワーカーノードがマルチノード構成から削除されます。

11. 修正版 HADB サーバとの入れ替えをノード４で実施する

修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。

12. ノード４のワーカーノードをマルチノード構成に追加する

次のコマンドをノード４のワーカーノードで実行してください。

```
adbstart
```

ノード４の HADB サーバが開始し，ノード４のワーカーノードがマルチノード構成に追加されます。

> **メモ**
>
> - ノード５のワーカーノードについても，上記の手順の 10.～12.を実施してください。
> - ワーカーノードの追加および削除は同時実行可能です。そのため，上記の手順の 10.～12.をノード４とノード５で同時実行しても問題ありません。

以上の手順で，修正版 HADB サーバとの入れ替えが完了します。

■修正版 HADB サーバとの入れ替えに失敗した場合

修正版 HADB サーバとの入れ替えに失敗した場合は，次に示す手順で入れ替え前の状態に戻してください。この手順は，修正版 HADB サーバとの入れ替えに失敗したすべてのノードで実施する必要があります。

手順

1. 入れ替えに失敗したノードの HADB サーバを強制終了する

入れ替えに失敗したノードの HADB サーバが稼働中の場合は，`adbstop --force` コマンドを実行して HADB サーバを強制終了してください。

2. 修正版 HADB サーバをアンインストールする

修正版との入れ替えに失敗したノードで，修正版 HADB サーバをアンインストールしてください。

3. サーバディレクトリを回復する

入れ替え前に取得しておいた，サーバディレクトリのバックアップを使用して，サーバディレクトリを回復してください。

4. **環境変数およびカーネルパラメタの指定値を元に戻す**

   環境変数およびカーネルパラメタの指定値を変更した場合は，変更前の指定値に戻してください。

5. **ノードをマルチノード構成に復帰させる**

   手順 4.までが完了したノードがプライマリノードまたはセカンダリノードの場合は，そのノードをマルチノード構成に復帰させます。マルチノード構成への復帰方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

   手順 4.までが完了したノードがワーカーノードの場合は，そのワーカーノードをマルチノード構成に追加します。マルチノード構成へのワーカーノードの追加手順については，「16.16.3　マルチノード構成から削除されたワーカーノードを再度追加する手順」を参照してください。

6. **コマンドによるプライマリノードの切り替えを実施する**

   プライマリノードを元のノードに戻したい場合は，コマンドによるプライマリノードの切り替えを実施してください。コマンドによるプライマリノードの切り替え方法については，「16.7　コマンドによるプライマリノードの切り替え」を参照してください。

7. **マルチノード構成から切り離されたノードをマルチノード構成に復帰させる**

   手順 6.でプライマリノードの切り替えを実施した結果，マルチノード構成から切り離されたノードを，マルチノード構成に復帰させてください。マルチノード構成への復帰方法については，「16.15.3　マルチノード構成へのノードの復帰」を参照してください。

16.24　SQL トレース機能を使用する際の留意事項（マルチノード機能使用時）

SQL トレース機能を使用する際の留意事項については，「10.11.11　マルチノード機能を使用している場合の留意事項」を参照してください。

16.25　CSV ファイル中のデータを検索するときの運用（マルチノード機能使用時）

CSV ファイル中のデータを検索するときの運用については，「11.15　CSV ファイル中のデータを検索するときの運用」を参照してください。ここでは，マルチノード機能使用時の留意事項について説明します。

マルチノード機能を使用している場合，全ノードから CSV ファイルを参照できるようにしてください。また，CSV ファイルの絶対パスが全ノードで同じになるようにしてください。

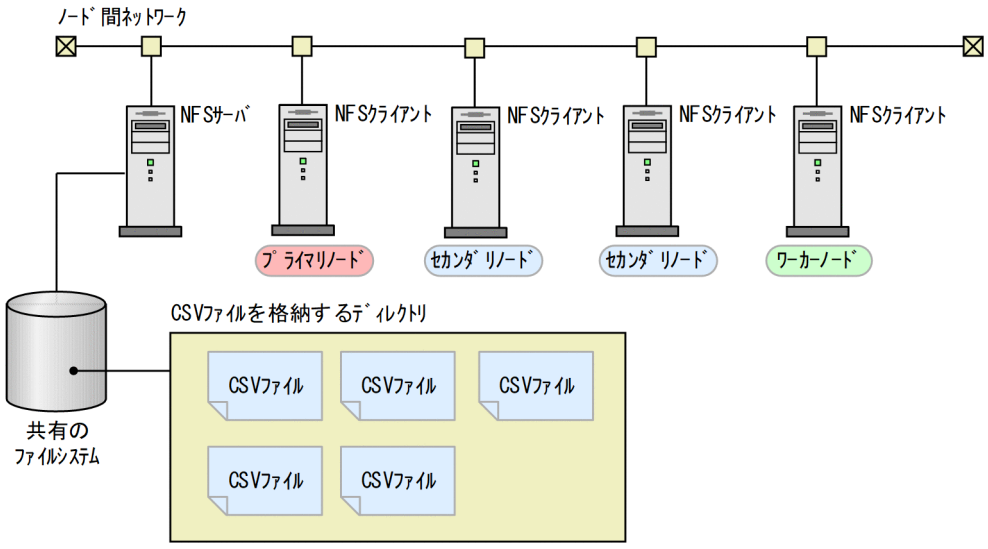
例えば，分散ファイルシステムの一つである NFS を使用する場合は，次の手順で CSV ファイルを格納するディレクトリを準備します。

手順

1. NFS サーバを用意し，NFS サーバ上で CSV ファイルを格納するディレクトリを作成してください。
2. 手順の 1.で作成した CSV ファイルを格納するディレクトリをエクスポートしてください。
   この操作は，NFS サーバで実行します。
3. 手順の 2.でエクスポートしたディレクトリを，全ノードで同じパスとなるようにマウントしてください。
   この操作は，全ノードで実行します。

NFS を使用したシステム構成例を次の図に示します。

図 16-10　NFS を使用したシステム構成例



16.26　同義語検索を行う際の運用（マルチノード機能使用時）

ここでは，同義語検索を行う際の準備作業と運用方法について説明します。

16.26.1　同義語検索を行うための準備をする

ここでは，同義語検索を行うときの準備作業について説明します。ここの説明は，すでにマルチノード機能を使用していることが前提となっています。

(1)　テキストインデクスの定義を検討する

同義語検索を行う場合，テキストインデクスを使用すると性能向上が見込まれるため，テキストデータ（文書データ）を格納している列にテキストインデクスを定義することを推奨します。詳細については，「11.16.1　同義語検索を行うための準備をする」の「(1)　テキストインデクスの定義を検討する」を参照してください。

(2)　同義語辞書ファイルの格納ディレクトリに必要な容量を見積もる

マルチノード機能の使用時に同義語検索を行う場合は，同義語辞書ファイルの格納ディレクトリを，ノード切り替え対象のファイルシステムに作成する必要があります。ファイルシステムを用意する前に次のことを実施してください。

同義語辞書ファイルの格納ディレクトリの容量見積もり

同義語辞書ファイルの格納ディレクトリの容量見積もりを実施し，その結果を考慮してファイルシステムを用意してください。同義語辞書ファイルの格納ディレクトリの容量見積もりについては，「11.16.1　同義語検索を行うための準備をする」の「(3)　同義語辞書ファイルの格納ディレクトリに必要な容量を見積もる」を参照してください。

> **重要**
>
> 同義語辞書ファイルの格納ディレクトリの空き容量が不足した場合，同義語辞書ファイルの格納ディレクトリを変更する必要があります。同義語辞書ファイルの格納ディレクトリを変更した場合，すべての同義語辞書を再登録する必要があるため，ディレクトリを作成しているディスクの容量不足などが発生しないように，見積もりを行ってください。

> **メモ**
>
> 同義語辞書ファイルの格納ディレクトリとは，サーバ定義のadb_syndict_storage_path オペランドに指定するディレクトリのことです。

(3)　同義語辞書ファイルの格納ディレクトリを作成するファイルシステムを準備する

同義語辞書ファイルの格納ディレクトリは，ノード切り替え対象のファイルシステムに作成する必要があります。ノード切り替え対象のファイルシステムは，プライマリノードおよびセカンダリノードから参照可能なディスクで構成される VG 内の LV 上に作成してください。共有ディスクについては，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの構成』を参照してください。

(4)　同義語辞書ファイルの格納ディレクトリを作成するファイルシステムを初期化する

同義語辞書ファイルの格納ディレクトリを作成するファイルシステムを初期化してください。同義語辞書ファイルの格納ディレクトリを作成する LV として作成した/dev/vg_hadb02/hadb_syndict を，ext4 のファイルシステムで初期化する場合の実行例を次に示します。

```
mkfs -t ext4 /dev/vg_hadb02/hadb_syndict
```

(5)　同義語辞書ファイルの格納ディレクトリを作成する

ファイルシステムの準備が完了したあとに，同義語辞書ファイルの格納ディレクトリを作成し，作成したディレクトリに対する権限を付与してください。詳細については，「11.16.1　同義語検索を行うための準備をする」の「(4)　同義語辞書ファイルの格納ディレクトリを作成する」～「(5)　同義語辞書ファイルの格納ディレクトリに対する権限を付与する」を参照してください。

重要

- 同義語辞書ファイルの格納ディレクトリを削除しないでください。また，同義語辞書ファイルの格納ディレクトリ下のファイルを，adbsyndict コマンド以外で削除しないでください。削除した場合，同義語検索の実行がエラーになります。
- 同義語辞書ファイルの格納ディレクトリ下のファイルを，adbsyndict コマンド以外で更新しないでください。更新した場合，同義語検索の実行結果が不正となるか，または同義語検索の実行がエラーになります。

メモ

同義語辞書ファイルの格納ディレクトリはフルバックアップの対象になります。詳細については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

(6) マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する

マルチノード機能の使用時に同義語検索を行う場合は，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを，ノードローカルのファイルシステムに作成してください。マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリは，全ノードで作成する必要があります。

> **メモ**
>
> マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリとは，サーバ定義の adb_syndict_node_storage_path オペランドに指定するディレクトリのことです。

同義語検索を行う SQL 文がセカンダリノードまたはワーカーノードで実行される場合，ここで作成したディレクトリ下にある同義語辞書ファイルが使用されます。adbsyndict コマンドによる同義語辞書の登録，変更，または削除をした際，プライマリノードの同義語辞書ファイルが自動的にセカンダリノードおよびワーカーノードにコピーされ，ここで作成したディレクトリ下に格納されます。

> **重要**
>
> セカンダリノードまたはワーカーノードに同義語辞書ファイルが格納されていなかったり，古い同義語辞書ファイルが格納されていたりした場合でも，同義語検索は実行できます。ただし，この場合，そのセカンダリノードまたはワーカーノードに同義語検索を行う SQL 文の実行処理が割り振られても，そのセカンダリノードまたはワーカーノードでは SQL 文の実行処理は行われません。SQL 文の実行処理はプライマリノードで行われます。この状態が続くと，プライマリノードの負荷が高まるおそれがあるため，同義語辞書ファイルの同期を取って，セカンダリノードまたはワーカーノードに最新の同義語辞書ファイルを格納するようにしてください。同義語辞書ファイルの同期については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

次に示す手順で，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成してください。

手順

1. マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量を見積もる

   マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量見積もりを実施し，その結果を考慮してディレクトリを作成してください。マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量見積もりについては，「6.17　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの容量見積もり」を参照してください。

   > **重要**
   >
   > マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの空き容量が不足した場合，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを変更する必要

があります。マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを変更した場合，同義語辞書ファイルの同期を実行する必要があるため，ディレクトリを作成しているディスクの容量不足などが発生しないように，見積もりを行ってください。

2. マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する

マルチノード機能で使用する同義語辞書ファイルを格納するディレクトリを作成してください。ディレクトリを作成する際，次の点に注意してください。

- 次のディレクトリは，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにできません。
  - サーバディレクトリ
  - サーバディレクトリの下位のディレクトリ
  - 下位のディレクトリにサーバディレクトリがあるディレクトリ
  - DB ディレクトリ
  - DB ディレクトリの下位のディレクトリ
  - 下位のディレクトリに DB ディレクトリがあるディレクトリ
  - ルートディレクトリ
  - adb_syndict_storage_path オペランドに指定したディレクトリ
  - adb_syndict_storage_path オペランドに指定したディレクトリの下位のディレクトリ
  - 下位のディレクトリに，adb_syndict_storage_path オペランドに指定したディレクトリがあるディレクトリ

DB ディレクトリが/HADB/db の場合，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにできるディレクトリの例と，できないディレクトリの例を次に示します。

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの例		理由
マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにできるディレクトリの例	/HADB/syndict	なし。
マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにできないディレクトリの例	/HADB/db	左記のディレクトリは，DB ディレクトリと同一のため，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにできません。
	/HADB/db/synonym	左記のディレクトリは，DB ディレクトリの下位のディレクトリのため，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにできません。
	/HADB	左記のディレクトリは，下位のディレクトリに DB ディレクトリがあるため，マルチノー

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリの例		理由
		ド機能で使用する同義語辞書ファイルの格納ディレクトリにできません。

- HADB サーバのインストール時にインストールデータを格納したディレクトリを，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリにしないでください。

3. マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリに対する権限を付与する

次に示す手順で，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリに対して必要な権限を付与してください。

手順

1. マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリに，HADB 管理者に対する読み取り権限，書き込み権限，および実行権限を付与する
2. マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリのパスに含まれるすべてのディレクトリに，HADB 管理者に対する実行権限を付与する

（例）マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリが/HADB/syndict の場合

- /HADB/syndict には，読み取り権限，書き込み権限，および実行権限が必要です。
- /，および/HADB には，実行権限が必要です。

重要

- マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを変更した場合，adbsyndict コマンドを実行して同義語辞書ファイルの同期を取る必要があります。同期を取らなかった場合，同義語検索を行う SQL 文の実行ノードがセカンダリノードまたはワーカーノードからプライマリノードに切り替わります。その結果，プライマリノードの負荷増加や，実行ノードを切り替える処理によって，検索性能が低下するおそれがあります。
- マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを削除しないでください。また，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ下のファイルを，adbsyndict コマンド以外で削除しないでください。削除した場合，同義語検索を行う SQL 文の実行ノードがセカンダリノードまたはワーカーノードからプライマリノードに切り替わります。その結果，プライマリノードの負荷増加や，実行ノードを切り替える処理によって，検索性能が低下するおそれがあります。
- マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ下のファイルを，adbsyndict コマンド以外で更新しないでください。更新した場合，同義語検索の実行結果が不正となるか，または同義語検索の実行がエラーになります。

(7) HA モニタの servers ファイルの指定を変更する

マルチノード構成の HADB サーバを終了して，HA モニタのservers ファイルの指定を変更してください。

「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合，FS002 が同義語辞書ファイルの格納ディレクトリを作成するファイルシステムです。VG の名称はvg_hadb02 で，LV の名称はhadb_syndict とします。

次に示すオペランドの指定値を変更します。

- disk

  同義語辞書ファイルの格納ディレクトリを作成するファイルシステムを含む VG の絶対パスの指定を追加してください。「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合は，/dev/vg_hadb02 を追加します。

- fs_name

  同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対応する論理ボリュームの絶対パスの指定を追加してください。「図 16-1　マルチノード機能を使用した場合のシステム構成例」の場合は，/dev/vg_hadb02/hadb_syndict を追加します。

- fs_mount_dir

  同義語辞書ファイルの格納ディレクトリを作成するファイルシステムをマウントするディレクトリのパスの指定を追加してください。同義語辞書ファイルの格納ディレクトリを作成するファイルシステムをマウントするディレクトリのパスは，プライマリノードおよびセカンダリノードで同じパスとなるように指定してください。

- fs_mount_opt

  同義語辞書ファイルの格納ディレクトリを作成するファイルシステムをマウントするためのマウントオプションを追加してください。

- vg_neck

  use の指定を追加してください。

- fs_neck

  use の指定を追加してください。

- scsi_device

  このオペランドには，同義語辞書ファイルを格納するファイルシステムのデバイス名（SCSI リザーブの対象となるデバイス名）を絶対パスで追加してください。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクが次のどれかの条件を満たすときに，このオペランドを指定してください。

  - シングルパス構成の場合
  - VMware ESXi での仮想化環境の場合（DMMP を使用する場合を除く）
  - マルチパスソフトウェア（HDLM）による冗長化構成の場合

- dmmp_device

  このオペランドには，同義語辞書ファイルを格納するファイルシステムのデバイス名（SCSI リザーブの対象となるデバイス名）を，DMMP の論理デバイスの絶対パスで指定します。

なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクがマルチパスソフトウェア（DMMP）による冗長化構成のときに，このオペランドを指定してください。

servers ファイルの指定例については，次のどちらかの個所を参照してください。

- 系のリセットを使用している場合

  「16.3.4　HA モニタの環境設定」の「(5)　各ファイルの指定例（系のリセットを使用する場合）」の「(b)　servers ファイルの指定例」を参照してください。

- 共有ディスクの SCSI リザーブを使用している場合

  シングルパス構成の場合は，「16.3.4　HA モニタの環境設定」の「(6)　各ファイルの指定例（共有ディスクの SCSI リザーブを使用する場合）」の「(b)　servers ファイルの指定例（シングルパス構成の場合）」を参照してください。

  マルチパスソフトウェアによる冗長化構成の場合は，「16.3.4　HA モニタの環境設定」の「(6)　各ファイルの指定例（共有ディスクの SCSI リザーブを使用する場合）」の「(c)　servers ファイルの指定例（マルチパスソフトウェアによる冗長化構成の場合）」を参照してください。

servers ファイルの各オペランドの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『サーバ対応の環境設定（servers）』を参照してください。

(8)　サーバ定義を修正する

サーバ定義に次に示すオペランドを指定してください。

- adb_syndict_storage_path

  サーバ定義のadb_syndict_storage_path オペランドに，同義語辞書ファイルの格納ディレクトリ名を指定してください。このオペランドは，プライマリノードおよびセカンダリノードのサーバ定義に指定してください。

  adb_syndict_storage_path オペランドについては，「7.2.8　同義語検索に関するオペランド（set 形式）」の adb_syndict_storage_path を参照してください。

  adb_syndict_storage_path オペランドに同義語辞書ファイルの格納ディレクトリのシンボリックリンクを指定した場合，シンボリックリンクを解決したあとの絶対パス名が，「11.16.1　同義語検索を行うための準備をする」の「(4)　同義語辞書ファイルの格納ディレクトリを作成する」で説明している規則に従っているかどうかがチェックされます。

- adb_syndict_node_storage_path

  サーバ定義のadb_syndict_node_storage_path オペランドに，マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ名を指定してください。このオペランドは，全ノードのサーバ定義に指定してください。

  adb_syndict_node_storage_path オペランドについては，「7.2.8　同義語検索に関するオペランド（set 形式）」の adb_syndict_node_storage_path を参照してください。

  adb_syndict_node_storage_path オペランドにマルチノード機能で使用する同義語辞書ファイルの格納ディレクトリのシンボリックリンクを指定した場合，シンボリックリンクを解決したあとの絶対パス名

が，「(6)　マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する」の「2. マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリを作成する」で説明している規則に従っているかどうかがチェックされます。

> **メモ**
>
> サーバ定義の変更手順については，「8.5.2　サーバ定義の変更方法」を参照してください。

(9)　マルチノード構成の HADB サーバを開始する

マルチノード構成の HADB サーバを開始してください。

マルチノード構成の HADB サーバの開始方法については，「16.4.1　マルチノード構成の HADB サーバの開始方法」を参照してください。

(10)　テキストインデクスを定義する

「(1)　テキストインデクスの定義を検討する」の結果，テキストインデクスを定義する場合は，次に示す手順でテキストインデクスを定義してください。

手順

1. CREATE INDEX 文を実行して，テキストインデクスを定義する
2. adbidxrebuild コマンドを実行して，テキストインデクスのインデクスデータを作成する

(11)　同義語辞書を登録する

同義語辞書を登録するには，「11.16.1　同義語検索を行うための準備をする」の「(7)　同義語一覧定義ファイルを作成する」～「(10)　同義語辞書ファイルの格納ディレクトリのバックアップを取得する」の作業を実施してください。

> **重要**
>
> 同義語一覧定義ファイル，辞書情報登録ファイル，辞書情報削除ファイルは，プライマリノードから参照可能なファイルシステムに作成することを推奨します。同義語辞書ファイルの格納ディレクトリと同様に，ノード切り替え対象のファイルシステム上に作成するか，またはプライマリノードおよびセカンダリノードで同一パスのディレクトリにマウントした，各ノードから参照可能なディスク上に作成することを検討してください。

プライマリノードでadbsyndict コマンドを実行して同義語辞書を登録した場合，自動的に同義語辞書ファイルの同期が行われます。そのため，-s オプションを指定したadbsyndict コマンドの実行は不要です。

16.26.2　同義語検索を行う

同義語検索の方法は，マルチノード機能を使用しない場合と同じです。同義語検索の方法については，「11.16.2　同義語検索を行う」を参照してください。

マルチノード機能の使用時に同義語検索を行う場合，SQL 文の実行処理を割り振られたノードにある同義語辞書ファイルが使用されます。マルチノード機能で使用する同義語辞書ファイルは，サーバ定義の adb_syndict_node_storage_path オペランドに指定したディレクトリ下に格納されています。このディレクトリ下に最新の同義語辞書ファイルがないセカンダリノード（またはワーカーノード）に SQL 文の実行処理が割り振られた場合，そのセカンダリノード（またはワーカーノード）では SQL 文の実行処理は行われないで，プライマリノードで SQL 文の実行処理が行われます。このとき，KFAA51537-W メッセージが出力されます。

この状態が続くと，次に示す要因によって検索性能が低下するおそれがあります。

- 同義語検索を実行する SQL 文の実行処理がプライマリノードに集中してプライマリノードの負荷が高まる
- 同義語検索を実行する SQL 文の実行ノードをセカンダリノード（またはワーカーノード）からプライマリノードに変更する処理が行われる

そのため，同義語辞書ファイルの同期を取って，セカンダリノードおよびワーカーノードの同義語辞書ファイルを最新のファイルにしてください。同義語辞書ファイルの同期を取る方法については，「16.26.5　同義語辞書ファイルの同期を取る」を参照してください。

16.26.3　同義語辞書を登録または削除する

同義語辞書の登録または削除方法は，マルチノード機能を使用しない場合と同じです。同義語辞書の登録方法については，「11.16.8　同義語辞書を登録する」を参照してください。同義語辞書の削除方法については，「11.16.9　同義語辞書を削除する」を参照してください。

> **メモ**
>
> プライマリノードでadbsyndict コマンドを実行して同義語辞書を登録または削除した場合，同義語辞書ファイルの同期が自動的に行われます。そのため，-s オプションを指定したadbsyndict コマンドの実行は必要ありません。

16.26.4　同義語を追加または削除する

同義語の追加または削除方法は，マルチノード機能を使用しない場合と同じです。同義語の追加または削除方法については，「11.16.5　同義語を追加する」または「11.16.6　同義語を削除する」を参照してください。

プライマリノードでadbsyndictコマンドを実行して同義語を追加または削除した場合，同義語辞書ファイルの同期が自動的に行われます。そのため，-sオプションを指定したadbsyndictコマンドの実行は必要ありません。

> **メモ**
>
> 同義語グループの追加および削除方法も，マルチノード機能を使用しない場合と同じです。同義語グループの追加方法については，「11.16.7　同義語グループを追加する」を参照してください。

16.26.5　同義語辞書ファイルの同期を取る

同義語辞書ファイルの同期を取る場合は，次のコマンドを実行します。

```
adbsyndict -s
```

同義語辞書ファイルの同期とは

マルチノード機能を使用している場合，同義語辞書ファイルが全ノードで必要になります。全ノードの同義語辞書ファイルを同じファイルにすることを同義語辞書ファイルの同期といいます。プライマリノードの同義語辞書ファイルをセカンダリノードおよびワーカーノードにコピーして，同義語辞書ファイルの同期を取っています。

セカンダリノードおよびワーカーノードの同義語辞書ファイルは，adb_syndict_node_storage_pathオペランドに指定したディレクトリ下に格納されています。セカンダリノードまたはワーカーノードで同義語検索のSQL文が実行される場合，そのセカンダリノードまたはワーカーノードの同義語辞書ファイルが使用されます。

同義語辞書ファイルの同期が取られていない場合

同義語辞書ファイルの同期が取られていない場合でも，同義語検索は実行できます。ただし，同義語辞書ファイルの同期が取られていないセカンダリノード（またはワーカーノード）にSQL文の実行処理が割り振られた場合，そのセカンダリノード（またはワーカーノード）ではSQL文の実行処理は行われません。この場合，SQL文の実行処理はプライマリノードで行われます。この状態が続くと，プライマリノードの負荷が高まるおそれがあるため，同義語辞書ファイルの同期を取って，セカンダリノード（またはワーカーノード）の同義語辞書ファイルを最新のファイルにしてください。

同義語辞書ファイルの同期を取る必要があるケース

次の場合，セカンダリノードまたはワーカーノードに最新の同義語辞書ファイルが格納されていないため，同義語辞書ファイルの同期を取る必要があります。

- ノードを新規に追加した場合
- マルチノード構成にノードを復帰した場合※
- マルチノード構成にワーカーノードを追加した場合※

注※

ノードの停止中に同義語辞書の登録や更新を行った場合に，同義語辞書ファイルの同期を取る必要があります。

セカンダリノードまたはワーカーノードに最新の同義語辞書ファイルが格納されていない場合，HADB サーバの開始時，または同義語検索の実行時に警告メッセージKFAA51537-W が出力されます。この場合，adbsyndict コマンドを実行して同義語辞書ファイルの同期を取ってください。

メモ

同義語時辞書の登録，更新，または削除をしたときは，HADB サーバが自動的に同義語時辞書ファイルの同期を取ります。

16.27　監査証跡機能の運用（マルチノード機能使用時）

ここでは，監査証跡機能を使用する場合の環境設定方法と運用方法について説明します。

なお，JP1/Audit と連携して，監査証跡を収集および一元管理する場合は，「16.27.3　JP1/Audit Management - Manager との連携（マルチノード機能使用時）」を参照してください。

16.27.1　監査証跡機能の環境設定

ここでは，監査証跡機能とマルチノード機能の両方を使用する際の環境設定に関する留意事項について説明します。監査証跡機能の環境設定方法については，「12.2　監査証跡機能の環境設定」を参照してください。

■監査証跡の出力先ディレクトリの作成

- 各ノードに，監査証跡の出力先ディレクトリを作成してください。
- 監査証跡の出力先ディレクトリは，各ノードのノードローカルのファイルシステムに作成することを推奨します。
- 各ノードのサーバ定義に，adb_audit_log_path オペランドを指定してください。このオペランドには，監査証跡の出力先ディレクトリを指定します。

> **メモ**
>
> 監査証跡は，操作が実行されたノードに出力されます。そのため，監査証跡の出力先ディレクトリをノードごとに作成し，そのディレクトリ名をadb_audit_log_path オペランドに指定する必要があります。

■監査証跡の保存先ディレクトリを作成するファイルシステムの準備

監査証跡を検索する際に使用するADB_AUDITREAD 関数中に指定する監査証跡ファイルのパスが，全ノードから同じ絶対パスで参照できるように環境を構築してください。

> **重要**
>
> - 全ノードから同じ絶対パスで監査証跡ファイルを参照できない場合，監査証跡を検索する SQL 文を実行するトランザクションが割り当てられたノードによって，検索結果が異なることがあります。
> - ADB_AUDITREAD 関数中に，検索対象の監査証跡ファイルの絶対パスを指定する必要があります。

全ノードで共有のファイルシステムに監査証跡の保存先ディレクトリを作成してください。例として，分散ファイルシステムの一つである NFS を使用したときの環境設定方法を説明します。

手順

1. NFS サーバを用意し，NFS サーバ上で監査証跡の保存先ディレクトリを作成してください。
2. 手順の 1.で作成した監査証跡の保存先ディレクトリをエクスポートしてください。

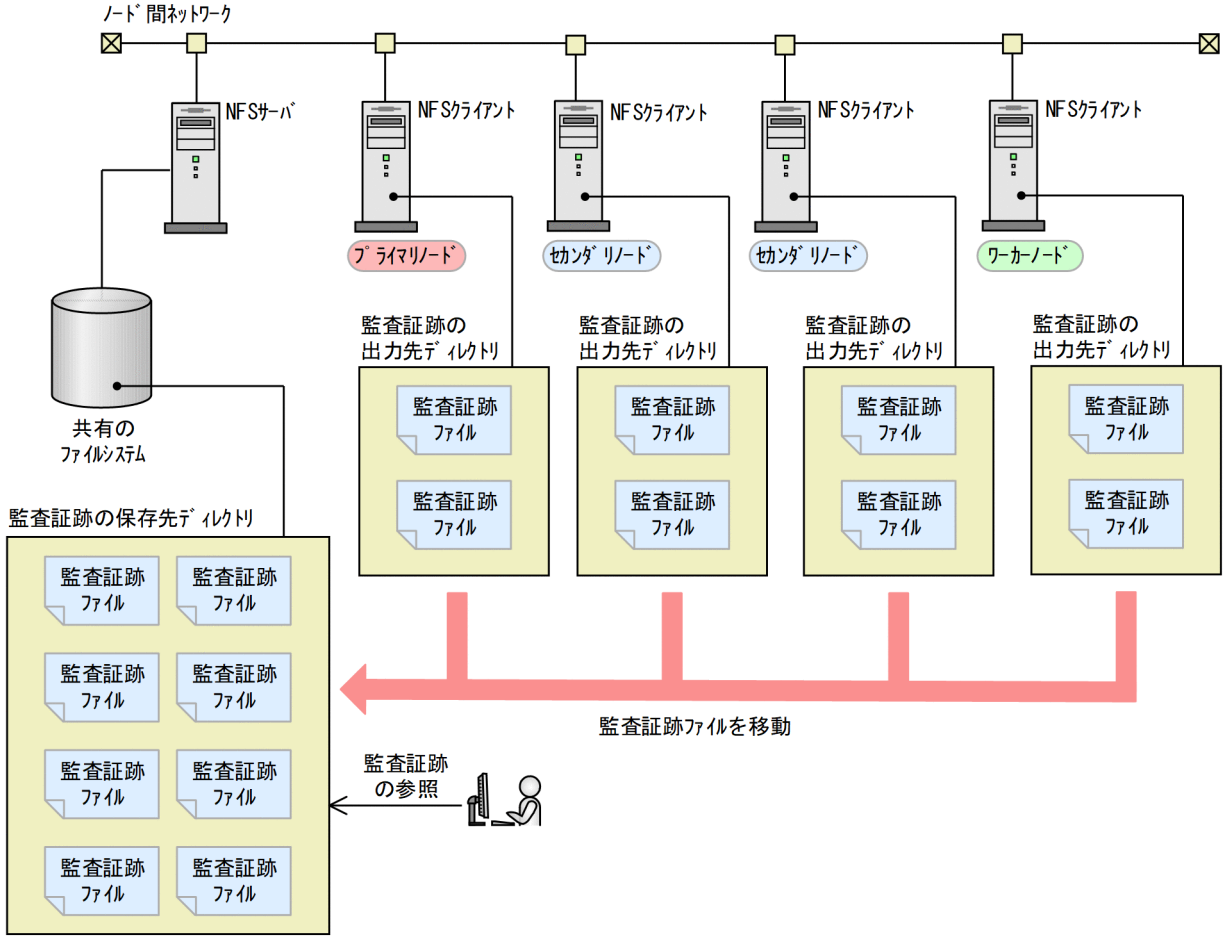
   この操作は，NFS サーバで実行します。
3. 手順の 2.でエクスポートしたディレクトリを，全ノードで同じパスとなるようにマウントしてください。

   この操作は，全ノードで実行します。
4. 各ノードの監査証跡の出力先ディレクトリ下に出力された監査証跡ファイルを，NFS サーバ上の監査証跡の保存先ディレクトリに移動してください。

   この操作は，全ノードで実行します。
5. 監査証跡の保存先ディレクトリ下に格納されている監査証跡ファイルのパスを，ADB_AUDITREAD 関数に指定して監査証跡を検索します。

NFS を使用したシステム構成例を次の図に示します。

図 16-11　NFS を使用したシステム構成例



16.27.2　監査証跡機能の運用方法

ここでは，監査証跡機能とマルチノード機能の両方を使用する際の運用方法に関する留意事項について説明します。監査証跡機能の運用方法については，「12.　監査証跡機能の運用」を参照してください。

■監査証跡機能を有効にする場合，または無効にする場合

監査証跡機能を有効にする場合は，プライマリノードでadbaudittrail --start コマンドを実行してください。全ノードで監査証跡機能が有効になります。

監査証跡機能を無効にする場合は，プライマリノードでadbaudittrail --stop コマンドを実行してください。全ノードで監査証跡機能が無効になります。

> **メモ**
>
> ノードごとに監査証跡機能を有効にしたり，無効にしたりすることはできません。

■監査証跡ファイルを切り替える場合

監査証跡ファイルを切り替えたいノードでadbaudittrail --swap コマンドを実行してください。

また，監査証跡ファイルを切り替えたいノードのノード番号を-n オプションに指定して，プライマリノードでadbaudittrail --swap コマンドを実行しても監査証跡ファイルを切り替えることができます。

なお，監査証跡ファイルを切り替えたいノードのノード番号を確認する場合，adbls -d node コマンドを実行してください。adbls -d node コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d node（各ノードの HADB サーバの状態表示)』を参照してください。

■監査証跡機能に関連する情報を表示する場合

監査証跡機能に関連する情報を調べたいノードでadbaudittrail -d コマンドを実行してください。

また，監査証跡機能に関連する情報を調べたいノードのノード番号を-n オプションに指定して，プライマリノードでadbaudittrail -d コマンドを実行しても監査証跡機能に関連する情報を調べることができます。

なお，監査証跡機能に関連する情報を調べたいノードのノード番号を確認する場合，adbls -d node コマンドを実行してください。adbls -d node コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbls -d node（各ノードの HADB サーバの状態表示)』を参照してください。

16.27.3　JP1/Audit Management - Manager との連携（マルチノード機能使用時）

JP1/Audit と連携して，監査証跡を収集および一元管理する場合は，adbconvertauditrailfile コマンドで監査証跡を変換し，統一フォーマット用監査証跡ファイルに出力してください。監査証跡の変換については，「2.18.9　監査証跡の変換（JP1/Audit Management - Manager との連携)」を参照してください。

adbconvertauditrailfile コマンドは，全ノードで実行できます。また，監査証跡ファイルを出力したノードと，adbconvertauditrailfile コマンドを実行するノードを一致させる必要はありません。そのため，次の 2 つの運用方法が考えられます。

- プライマリノード，セカンダリノード，ワーカーノードが，それぞれ自ノードで出力した監査証跡ファイルをadbconvertaudittrailfile コマンドで変換する運用方法
- プライマリノードが，すべての監査証跡ファイルをadbconvertaudittrailfile コマンドで変換する運用方法

どちらの運用でも JP1/Audit で確認できる内容は同じです。なお，統一フォーマット用監査証跡の出力先ディレクトリは，ノードローカルのファイルシステム上に作成してください。

それぞれの運用方法の留意事項について説明します。

■プライマリノード，セカンダリノード，ワーカーノードが，それぞれ自ノードで出力した監査証跡ファイルをadbconvertaudittrailfile コマンドで変換する場合

adbconvertaudittrailfile コマンドを実行する全ノードで，JP1/Audit の環境設定を行ってください。

JP1/Audit の環境設定については，「12.8.3　監査証跡機能と JP1/Audit を連携する際の環境設定」の「(2)　JP1/Audit の環境設定」を参照してください。また，adbconvertaudittrailfile コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbconvertaudittrailfile（監査証跡ファイルの変換)』を参照してください。

なお，JP1/Base をクラスタ環境として構築する必要はありません。

> **重要**
>
> 統一フォーマット用監査証跡の出力先ディレクトリは，それぞれのノードに作成してください。同一のディレクトリに対して，複数のノードが統一フォーマット用監査証跡ファイルを同時に出力した場合，統一フォーマット用監査証跡ファイルは正常に出力されません。

運用手順の例を次に示します。

1. 各ノードで，監査証跡の出力先ディレクトリに出力された監査証跡ファイルを，adbconvertaudittrailfile コマンドで統一フォーマット用監査証跡ファイルに変換する
2. 手順 1.が完了したら，各ノードで監査証跡の出力先ディレクトリに出力された監査証跡ファイルを，全ノードで共有している監査証跡の保存先ディレクトリに移動する

■プライマリノードが，すべての監査証跡ファイルをadbconvertaudittrailfile コマンドで変換する場合

adbconvertaudittrailfile コマンドを実行するプライマリノードで，JP1/Audit の環境設定を行ってください。

なお，プライマリノードの切り替えが発生したときに，新たなプライマリノードで adbconvertaudittrailfile コマンドを実行したい場合は，全ノードで JP1/Audit の環境設定を行ってください。その際，JP1/Base をクラスタ環境として構築する必要はありません。

JP1/Audit の環境設定については，「12.8.3　監査証跡機能と JP1/Audit を連携する際の環境設定」の「(2)　JP1/Audit の環境設定」を参照してください。また，adbconvertaudittrailfile コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbconvertaudittrailfile（監査証跡ファイルの変換)』を参照してください。

運用手順の例を次に示します。

1. 各ノードで，監査証跡の出力先ディレクトリに出力された監査証跡ファイルを，全ノードで共有している監査証跡の保存先ディレクトリに移動する
2. プライマリノードでadbconvertaudittrailfileコマンドを実行し，監査証跡の保存先ディレクトリに移動された監査証跡ファイルを，統一フォーマット用監査証跡ファイルに変換する

16.28　更新行のカラム化機能の運用（マルチノード機能使用時）

更新行のカラム化機能の運用については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。ここでは，マルチノード機能使用時の留意事項について説明します。

- adbcolumnize コマンドはプライマリノードで実行してください。
- プライマリノードの切り替えが発生したあとも，更新行のカラム化機能の状態（有効または無効）は引き継がれます。
- 更新行のカラム化機能のメンテナンス処理は，プライマリノードで実行されます。
- マルチノード構成の HADB サーバで，トランザクションまたは HADB サーバに接続するコマンドが実行されている間は，メンテナンス処理は実行されません。
- 更新行のカラム化機能の有効時にプライマリノードの切り替えが発生した場合，プライマリノードの切り替えが完了したあとにメンテナンス処理が実行されます。
- 更新行のカラム化機能に関するエラーの対処をしないで，更新行のカラム化機能を有効な状態のままにしておくと，プライマリノードの切り替えが完了したあとに実行されるメンテナンス処理がエラーとなり，エラーメッセージが出力されます。
- メンテナンス処理が 24 時間実行されない状態が続くと，KFAA51277-I メッセージが出力されますが，プライマリノードの切り替えが発生すると，時間経過の監視がリセットされます。そのため，プライマリノードの切り替え後，メンテナンス処理が実行されない状態がさらに 24 時間続いたときにKFAA51277-I メッセージが出力されます。
- 更新行のカラム化機能で使用するファイルの入出力制御の開始処理はノードごとに実行されます。そのため，切り替え先のプライマリノードで，更新行のカラム化機能で使用するファイルの入出力制御の開始処理に失敗していた場合，プライマリノードの切り替えが完了したあとに実行されるメンテナンス処理がエラーになります。

16.29　SQL パラレル実行機能の運用

ここでは，SQL パラレル実行機能を使用する際の準備作業と運用方法について説明します。

16.29.1　SQL パラレル実行機能を使用するときの準備作業

SQL パラレル実行機能を使用する場合，ここで説明する準備作業をしてください。

(1)　検索対象表の定義

SQL パラレル実行機能を適用する検索系 SQL の検索対象表（SELECT 文のFROM 句に指定する実表）を，次のように定義してコスト情報を収集してください。

- 検索対象表は，カラムストア表かつマルチチャンク表として定義してください。
- バージョン 04-03 以降の HADB サーバで，検索対象表のコスト情報を収集してください。

(2)　データ用 DB エリアの定義

検索対象表を格納するデータ用 DB エリアを定義する際，2 つ以上の DB エリアファイルで構成するように定義してください。詳細については，「5.6.1　データ用 DB エリアを設計する際の考慮点」の「(3) データ用 DB エリアを構成するデータ用 DB エリアファイルの考慮点」を参照してください。

なお，SQL パラレル実行機能を適用した検索系 SQL を分割処理するノードの最大数は，次のどちらかの小さい方の値になります。

- 検索系 SQL の検索対象表が格納されているデータ用 DB エリアの DB エリアファイル数
- マルチノード構成の全ノード数

> **重要**
>
> - 検索系 SQL を分割処理するノード数を増やしたい場合は，ノード数を増やすだけではなく，データ用 DB エリアの DB エリアファイル数も増やす必要があります。例えば，全ノード数が 8 であっても，DB エリアファイル数が 4 の場合，検索系 SQL を分割処理するノードの最大数は 4 になります（検索系 SQL を処理する際に全ノードを活用できません）。
> - 検索系 SQL の分割処理が各ノードに均等に割り振られるようにするには，DB エリアファイル数を全ノード数の倍数となるようにしてください。

SQL パラレル実行機能を適用した検索系 SQL を実行する際，全ノードを使用して分割処理をする必要がない場合は，DB エリアファイル数を全ノード数より少なくするようにしてください。

(3) サーバ定義の指定

サーバ定義の次のオペランドの指定値を全ノードで同じにしてください。

- adb_sys_rthd_num
- adb_sys_uthd_num
- adb_sys_shm_huge_page_size
- adb_sql_opt_drvtbl_grping_prior
- adb_sql_exe_max_rthd_num
- adb_sql_exe_hashgrp_area_size
- adb_sql_exe_hashtbl_area_size
- adb_sql_exe_hashflt_area_size
- adb_sql_tbldef_cache_size
- adb_sql_rngidx_preread
- adb_dbbuff_wrktbl_glb_blk_num
- adb_dbbuff_wrktbl_clt_blk_num
- adbbuff
- adbcltgrp（-r オプションおよび-e オプション）

重要

上記のオペランドの指定値が SQL メインノードと異なるノードは，SQL サブノードの候補になりません。

(4) クライアント定義または SQL 文の指定

SQL パラレル実行機能を使用する場合は，次のどちらかの指定をしてください。

- クライアント定義のadb_clt_sql_parallel_exec オペランドにY を指定する
- SQL 文に SQL パラレル実行指定を指定する

adb_clt_sql_parallel_exec オペランドについては，マニュアル『HADB　AP 開発ガイド』の『性能に関するオペランド』を参照してください。

SQL パラレル実行指定については，マニュアル『HADB　SQL リファレンス』の『SELECT 文の指定形式および規則』を参照してください。

(5) トランザクションの実行条件の設定

検索系 SQL を実行するトランザクションは，次の条件をすべて満たすようにしてください。

- トランザクションアクセスモードを読み取り専用モードにする
- トランザクション隔離性水準をREAD COMMITTED にする

上記の設定は，サーバ定義，クライアント定義，または JDBC のプロパティで行います。

> **メモ**
>
> SQL パラレル実行機能が適用される検索系 SQL については条件があります。条件については，マニュアル『HADB　AP 開発ガイド』の『SQL パラレル実行機能が適用される条件』を参照してください。

16.29.2　運用

(1)　SQL メインノードと SQL サブノードを確認する場合

adbls -d prl コマンドを実行すると，SQL メインノードと SQL サブノードを確認できます（検索系 SQL が実行されているノードを確認できます）。詳細については，「16.9.7　SQL メインノードと SQL サブノードを確認する場合」を参照してください。

(2)　検索系 SQL に SQL パラレル実行機能が適用されたかどうかを確認する場合

SQL トレース機能によって出力される SQL トレース情報を確認すると，実行した検索系 SQL に SQL パラレル実行機能が適用されたかどうかを確認できます。詳細については，「10.11.2　SQL トレース情報に出力される情報」の「(7)　SQL パラレル実行機能に関する情報」を参照してください。

検索系 SQL を実行する前に SQL パラレル実行機能が適用されるかどうかを確認したい場合は，アクセスパス情報を参照してください。詳細については，マニュアル『HADB　AP 開発ガイド』の『ツリー表示に出力される情報』の『SQL パラレル実行機能の適用情報』を参照してください。

(3)　検索対象表にデータをインポートをする場合

adbimport コマンドを実行する際に注意が必要です。詳細については，「16.11.2　実表の運用（マルチノード機能の使用時）」の「(1)　実表にデータを格納する方法（マルチノード機能の使用時）」を参照してください。

(4)　注意事項

次のどちらかの条件に該当する場合，SQL パラレル実行機能が適用された検索系 SQL はエラーになります。このとき，エラーメッセージKFAA31898-E が出力されます。

- 同一トランザクションで別の文ハンドルを使用して検索系 SQL を実行しているときに，SQL パラレル実行機能を適用した検索系 SQL を実行しようとした場合
- SQL パラレル実行機能を適用した検索系 SQL の最大同時実行数（サーバ定義の adb_sys_max_parallel_exec_num オペランドの値）を超えて，SQL パラレル実行機能を適用した検索系 SQL を実行しようとした場合

16.30　マルチノード機能使用時の注意事項

- プライマリノードおよび全セカンダリノードの HADB サーバで処理が実行できなくなった場合，マルチノード構成の HADB サーバが異常終了します。

17 コールドスタンバイ構成の運用

この章では，コールドスタンバイ構成でのシステムの構築方法，およびコールドスタンバイ構成の運用方法について説明します。

17.1　この章を読む際の留意事項

コールドスタンバイ構成の説明は，HA モニタについての知識があることを前提としています。HA モニタについては，マニュアル『HA モニタ Linux(R)(x86)編』を参照してください。

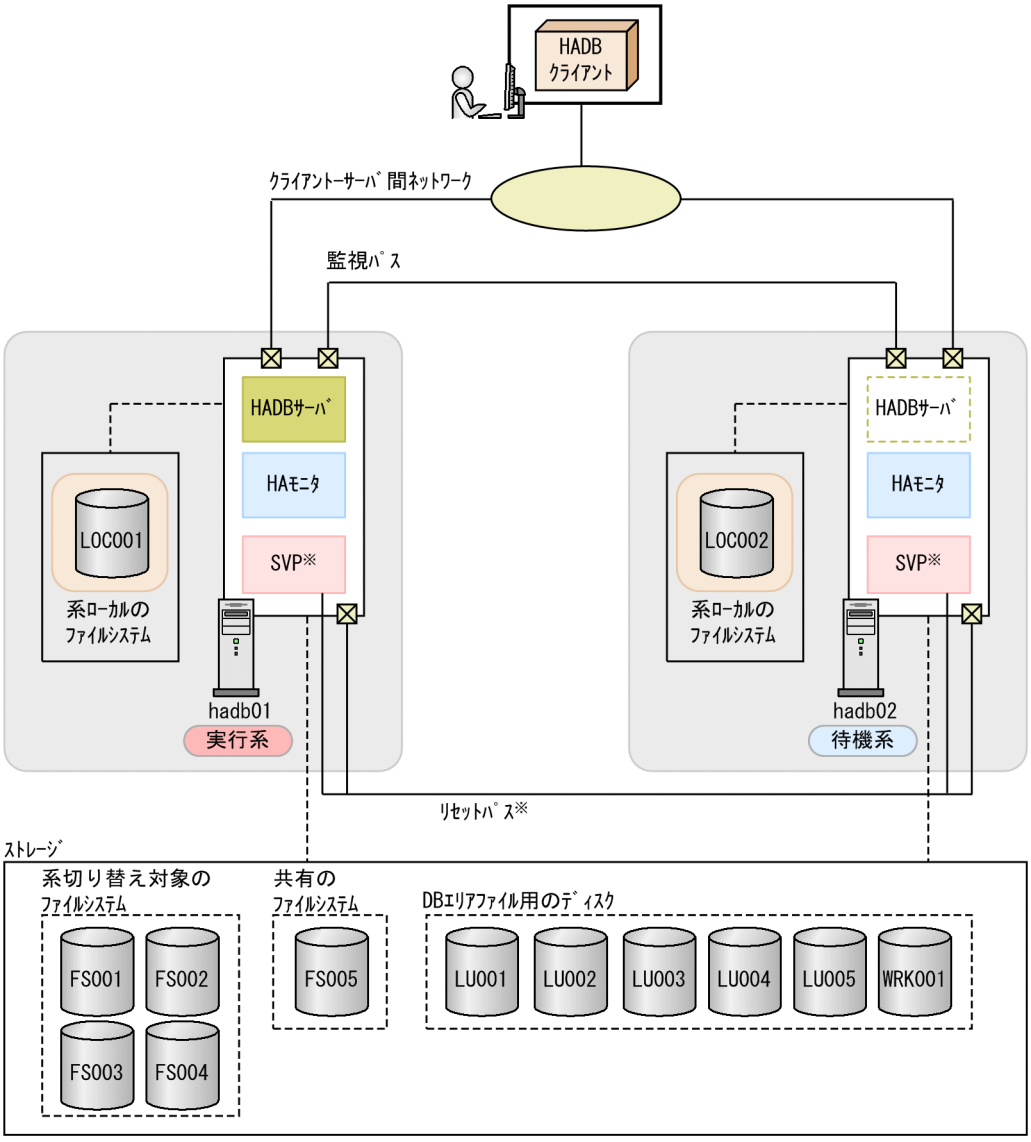
また，マニュアル『HA モニタ Linux(R)(x86)編』を参照する際，次の点に留意してください。

- HADB サーバは，モニタモードで動作します。
- エイリアス IP アドレスを引き継ぐ構成になります。
- HADB サーバは，HA モニタとのインタフェースを持たないプログラムとして動作します。
- サーバ障害の検出方法は，API を発行しない方法になります。
- HA モニタ Extension は使用できません。

17.2　コールドスタンバイ構成でのシステム構成例

コールドスタンバイ構成でのシステム構成例を次の図に示します。

図 17-1　コールドスタンバイ構成でのシステム構成例



注※

共有ディスクの SCSI リザーブを使用する場合は，SVP およびリセットパスは必要ありません。

重要

HA モニタによる共有ディスクのデータ保護方法には，次のどちらかを選択できますが，系のリセットを使用することを推奨します。

- 系のリセット
- 共有ディスクの SCSI リザーブ

17.2.1　前提ソフトウェア

コールドスタンバイ構成の場合，HA モニタが必要になります。

HADB では，HA モニタのモニタモードの機能を使用し，実行系の障害を監視しています。実行系で障害が発生した場合，HA モニタの系切り替え機能を使用して，待機系に系切り替えをします。

障害発生時の系切り替え処理については，「17.12　障害が発生した場合の運用」を参照してください。

なお，使用している Linux のバージョンによって，HA モニタの前提バージョンが次のように異なります。

- RHEL 6 の場合：01-56-01 以降
- RHEL 7 の場合：01-61 以降
- RHEL 8 の場合：01-71 以降

17.2.2　サーバ構成

「図 17-1　コールドスタンバイ構成でのシステム構成例」で示す構成例の場合，2 つの系（hadb01 および hadb02）から構成されています。実行系はhadb01，待機系はhadb02 になります。

コールドスタンバイ構成の場合，実行系と待機系のサーバマシンの性能（CPU，メモリサイズなど）は同一である必要はありません。ただし，サーバマシンの性能が異なる場合，系が切り替わったあとに SQL 文の処理性能に差異が発生するおそれがあります。そのため，できる限り実行系と待機系のマシン性能を同じにすることを推奨します。

また，コールドスタンバイ構成では，HA モニタの系切り替え機能を使用します。系を切り替える際の共有ディスクのデータの保護方法には，次のどちらかを選択してください。推奨する方法は，系のリセットです。

- 系のリセット
- 共有ディスクの SCSI リザーブ

各系のサーバマシンに障害管理プロセサ（「図 17-1　コールドスタンバイ構成でのシステム構成例」の SVP）がある場合は，推奨する系のリセットを使用してください。

障害管理プロセサは，使用するマシンの機種によって，SVP，BMC，または管理サーバのように名称が異なります。系のリセット，障害管理プロセサ，共有ディスクの SCSI リザーブについては，マニュアル『HA モニタ Linux(R)(x86)編』の次の個所を参照してください。

- 系のリセット：『系のリセット』
- 障害管理プロセサ：『必要なハードウェア』
- 共有ディスクの SCSI リザーブ：『共有ディスクの SCSI リザーブ』

17.2.3　ネットワーク構成

コールドスタンバイ構成の場合，次に示す 3 つのネットワークを使用します。各ネットワークは物理的に分けてください。

- クライアントーサーバ間ネットワーク
- 監視パス
- リセットパス

(1)　クライアントーサーバ間ネットワーク

HADB クライアントと HADB サーバ間の通信で使用するネットワークです。

HADB クライアントは，エイリアス IP アドレスを使用して HADB サーバに接続します。そのため，マニュアル『HA モニタ Linux(R)(x86)編』の『LAN の引き継ぎ』を参照して，エイリアス IP アドレスを設定してください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」での IP アドレス，およびポート番号の設定例を次の表に示します。

表 17-1　クライアントーサーバ間ネットワークの IP アドレスとポート番号の設定例

項番	設定対象	IP アドレス	ポート番号
1	クライアントマシン	10.196.108.111	設定不要
2	サーバマシンhadb01（実行系）	10.196.108.11	23650
3	サーバマシンhadb02（待機系）	10.196.108.12	23650
4	エイリアス IP アドレス	10.196.108.143	23650

(2)　監視パス

HA モニタの監視パスとして使用するネットワークです。

ネットワーク構築する際，Linux の Bonding 機能を使用した NIC 冗長化を推奨します。また，Bonding 機能を使用する際には，複数の LAN カードを用意してください。

Bonding 機能の詳細については，OS のマニュアルを参照してください。HA モニタの監視パスの設定については，マニュアル『HA モニタ Linux(R)(x86)編』の『監視パスの設定』を参照してください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」での IP アドレス，およびポート番号の設定例を次の表に示します。

表 17-2　監視パスの IP アドレスとポート番号の設定例

項番	設定対象	IP アドレス	ポート番号
1	サーバマシンhadb01（実行系）の HA モニタの監視パス	172.16.0.11	7777
2	サーバマシンhadb02（待機系）の HA モニタの監視パス	172.16.0.12	7777

(3)　リセットパス

系のリセットを使用する場合は，リセットパスが必要になります。共有ディスクの SCSI リザーブを使用する場合は，リセットパスは必要ありません。

リセットパスとは，HA モニタが障害を検知した際，障害が発生した系の入出力をリセットするために使用するネットワークです。このネットワークは，リセットパス専用のネットワークである必要があります。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『リセットパスの構成』を参照してください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」での IP アドレス，およびポート番号の設定例を次の表に示します。

表 17-3　リセットパスの IP アドレスとポート番号の設定例

項番	設定対象	IP アドレス	ポート番号
1	サーバマシンhadb01（実行系）のリセットパス	192.168.0.11	11111
2	サーバマシンhadb01（実行系）の SVP	192.168.0.21	22222
3	サーバマシンhadb02（待機系）のリセットパス	192.168.0.12	11111
4	サーバマシンhadb02（待機系）の SVP	192.168.0.22	22222

HA モニタによる共有ディスクのデータ保護方法には，系のリセットの使用を推奨しています。この場合，リセットパスが必要になります。また，HADB サーバが稼働するサーバマシンに SVP などの障害管理プロセサが搭載されている必要があります。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『必要なハードウェア』を参照してください。

17.2.4　ストレージ構成

コールドスタンバイ構成の場合，次のファイルシステムおよびディスクを準備してください。

- 系ローカルのファイルシステム
  「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，LOC001 およびLOC002 が該当します。
- 系切り替え対象のファイルシステム
  「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，FS001～FS004 が該当します。
- 共有のファイルシステム

「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，FS005 が該当します。

- DB エリアファイル用のディスク

  「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，LU001～LU005 およびWRK001 が該当します。

なお，一部のファイルシステムおよびディスクでは，実行系と待機系でディスクを識別するためのデバイス名が必要になります。サーバとストレージ間がシングルパス構成の場合は，WWN の使用を推奨します。サーバとストレージのパスを冗長化する場合は，冗長化の方法によってデバイス名が異なります。「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合の，デバイス名の例を次に示します。

- シングルパス構成の場合のデバイス名の例

  FS001：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000005

  FS002：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000006

  FS003：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000007

  FS004：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000008

  LU001：/dev/disk/by-id/wwn-0x60060e8010205850051104c50000000f

  LU002：/dev/disk/by-id/wwn-0x60060e8010205850051104c50000000e

  LU003：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000010

  LU004：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000011

  LU005：/dev/disk/by-id/wwn-0x60060e8010205850051104c500000013

  WRK001：/dev/disk/by-id/wwn-0x60060e8010205850051104c50000000d

- マルチパスソフトウェア（DMMP）による冗長化構成の場合のデバイス名の例

  FS001：/dev/mapper/mpath1

  FS002：/dev/mapper/mpath2

  FS003：/dev/mapper/mpath3

  FS004：/dev/mapper/mpath4

  LU001：/dev/mapper/mpath11

  LU002：/dev/mapper/mpath12

  LU003：/dev/mapper/mpath13

  LU004：/dev/mapper/mpath14

  LU005：/dev/mapper/mpath15

  WRK001：/dev/mapper/mpath16

(1)　系ローカルのファイルシステム

系ローカルのファイルシステムには，次のディレクトリを配置します。

- サーバディレクトリ

  サーバディレクトリの作成には，adbinstall コマンドを使用します。

重要

実行系と待機系の両方にサーバディレクトリを作成してください。サーバディレクトリを共有ディスク（系切り替え対象のファイルシステム）上に作成したり，共有のファイルシステム上に作成したりしないでください。

- 統一フォーマット用監査証跡の出力先ディレクトリ

  統一フォーマット用監査証跡の出力先ディレクトリについては，「2.18.9　監査証跡の変換（JP1/Audit Management - Manager との連携）」を参照してください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，LOC001 およびLOC002 が系ローカルのファイルシステムになります。

- LOC001

  サーバマシンhadb01（実行系）のローカルのファイルシステム

- LOC002

  サーバマシンhadb02（待機系）のローカルのファイルシステム

(2)　系切り替え対象のファイルシステム

次に示すファイルシステムを系切り替え対象にします。

- DB ディレクトリを格納するファイルシステム
- 作業用一時ファイルを格納するファイルシステム
- 同義語辞書ファイルを格納するファイルシステム
- 監査証跡の出力先ディレクトリを作成するファイルシステム

系切り替え対象のファイルシステムは，実行系と待機系から参照できるディスク（共有ディスク）から構成される VG 内の LV 上に作成してください。共有ディスクについては，マニュアル『HA モニタ Linux(R) (x86)編』の『共有ディスクの構成』を参照してください。

メモ

DB ディレクトリの作成については，「17.3.7　データベースの作成」を参照してください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，FS001～FS004 が系切り替え対象のファイルシステムになります。

- FS001

  DB ディレクトリを格納するファイルシステム

  ここでは，VG の名称はvg_hadb01，LV の名称はhadb_db とします。

- FS002

作業用一時ファイルを格納するファイルシステム
ここでは，VG の名称はvg_hadb02，LV の名称はhadb_workarea とします。

- FS003
  同義語辞書ファイルを格納するファイルシステム
  ここでは，VG の名称はvg_hadb03，LV の名称はhadb_syndict とします。
  なお，同義語辞書ファイルを格納するファイルシステムは，同義語検索をする場合に必要です。
- FS004
  監査証跡の出力先ディレクトリを作成するファイルシステム
  ここでは，VG の名称はvg_hadb04，LV の名称はhadb_audit とします。
  なお，監査証跡の出力先ディレクトリを作成するファイルシステムは，監査証跡機能を使用する場合に必要です。

重要

共有ディスクの SCSI リザーブを使用する場合，系切り替え対象のファイルシステムを作成したディスクは，SCSI リザーブの対象としてください。また，SCSI リザーブの対象としたディスクには，系切り替え対象のファイルシステム以外の領域を作成しないでください。

(3) 共有のファイルシステム

共有のファイルシステムには，次のファイルシステムを配置します。

- データインポートで使用する入力データファイルを格納するファイルシステム
- ADB_CSVREAD 関数で使用する CSV ファイルを格納するファイルシステム
- 監査証跡の保存先ディレクトリを作成するファイルシステム

共有のファイルシステムは，実行系と待機系から参照できるディスク（共有ディスク）上に構築し，実行系と待機系で同一パスのディレクトリにマウントしてください。共有ディスクについては，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの構成』を参照してください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，FS005 が共有のファイルシステムになります。

- FS005
  データインポートの入力データファイルを格納するファイルシステム，CSV ファイルを格納するファイルシステム，および監査証跡の保存先ディレクトリを作成するファイルシステム

共有のファイルシステムは，分散ファイルシステムの一つである NFS などを用いて構築できます。

(4) DB エリアファイル用のディスク

DB エリアファイルごとにディスクを用意します。

次の DB エリアファイルには，ブロックスペシャルファイルを割り当ててください。

- マスタディレクトリ用 DB エリアファイル
- ディクショナリ用 DB エリアファイル
- システム表用 DB エリアファイル
- データ用 DB エリアファイル

また，次の DB エリアファイルに対応するディスクは，実行系と待機系から参照できる必要があります。

- マスタディレクトリ用 DB エリアファイル
- ディクショナリ用 DB エリアファイル
- システム表用 DB エリアファイル
- データ用 DB エリアファイル
- 作業表用 DB エリアファイル

> **重要**
>
> 共有ディスクの SCSI リザーブを使用する場合，上記の DB エリアファイル用のディスクは，SCSI リザーブの対象としてください。

「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，LU001～LU005 およびWRK001 が DB エリアファイル用のディスクになります。

- LU001
  ディクショナリ用 DB エリアを構成する DB エリアディスク
- LU002
  マスタディレクトリ用 DB エリアを構成する DB エリアディスク
- LU003
  システム表用 DB エリアを構成する DB エリアディスク
- LU004
  データ用 DB エリア（ADBUTBL01）を構成する DB エリアディスク
- LU005
  データ用 DB エリア（ADBUIDX01）を構成する DB エリアディスク
- WRK001
  作業表用 DB エリア（ADBWRK）を構成する DB エリアディスク

17.3　コールドスタンバイ構成でのシステムの構築

コールドスタンバイ構成でのシステムの構築手順について説明します。

17.3.1　HA モニタのインストール

実行系および待機系に HA モニタをインストールしてください。HA モニタのインストール方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの構築』を参照してください。

17.3.2　HADB サーバのインストールおよび環境設定

実行系および待機系に HADB サーバをインストールし，環境設定を行ってください。HADB サーバのインストール，および環境設定については，「8.　システム構築」を参照してください。

HADB サーバのインストールおよび環境設定時の注意事項を次に示します。

- HADB サーバのバージョンおよびリビジョンは，実行系と待機系で同じにしてください。
- 環境変数の設定については，「8.4　環境変数の設定」を参照してください。ただし，次の環境変数については，実行系と待機系で指定値を同じにしてください。
  - ADBLANG
  - TZ
- コールドスタンバイ構成では，実行系の HADB サーバが異常終了した場合，HADB サーバの再開始を行います。HADB サーバの再開始の前にトラブルシュート情報が取得されるため，実行系と待機系に，トラブルシュート情報ファイルを格納するディレクトリを作成してください。

  系ローカルのファイルシステムに，/HADB/adbinfo というディレクトリを作成する例を次に示します。

```
mkdir -p /HADB/adbinfo                  ...1
chmod 755 /HADB/adbinfo                 ...2
```

  1. HADB 管理者が，OS の mkdir コマンドを使用してディレクトリを作成します。
  2. HADB 管理者が書き込めるように，OS の chmod コマンドを使用して権限を付与します。

  トラブルシュート情報ファイルを格納するディレクトリに作成されたファイルは，自動では削除されません。不要になった情報は定期的に手動で削除してください。

> **メモ**
>
> HADB サーバのサーバディレクトリ（インストール時）およびサーバディレクトリ（運用時）は，実行系と待機系の HADB サーバで同じにする必要はありません。

17.3.3　HADB クライアントのインストールおよび環境設定

HADB クライアントのインストール，および環境設定については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

17.3.4　HA モニタの環境設定

実行系および待機系で HA モニタの環境設定をしてください。HA モニタの環境設定方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの構築』を参照してください。

ここでは，HA モニタの環境設定をする際に留意することについて説明します。

(1)　環境変数の設定

HADB 管理者の環境変数に，次の環境変数を設定してください。

- PATH

  HA モニタのコマンドを実行するために，次のパスを追加してください。

  /opt/hitachi/HAmon/bin

(2)　sysdef ファイルの指定

HA モニタのsysdef ファイルに指定するオペランドのうち，次のオペランドについては，指定値が決まっています。

- address

  系のリセット優先度を指定します。指定値が小さいほど，その系のリセット優先度が高くなります。各系で異なる値を指定してください。実行系の指定値が，待機系の指定値より小さくなるようにしてください。

- monbegin_restart

  nouse を指定します。

- termcmd_at_abort

  nouse を指定します。

- fence_reset

  系のリセットを使用する場合は，このオペランドを省略するか，またはデフォルト値のuse を指定してください。

  共有ディスクの SCSI リザーブを使用する場合は，nouse を指定してください。

- fence_scsi

系のリセットを使用する場合は，このオペランドを省略するか，またはデフォルト値のnouse を指定してください。

共有ディスクの SCSI リザーブを使用する場合は，use を指定してください。

- fence_lan

  このオペランドを省略するか，またはデフォルト値のnouse を指定してください。

上記のオペランドの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『HA モニタの環境設定（sysdef)』を参照してください。

(3) servers ファイルの指定

HA モニタのservers ファイルに指定するオペランドのうち，次のオペランドについては，指定値が決まっています。

- name

  サーバディレクトリの絶対パスを指定します。

  ただし，name オペランドに指定するパス名として，空白を含む文字列は指定できません。そのため，コールドスタンバイ構成の場合，パス名に空白を含まないディレクトリをサーバディレクトリとすることを推奨します。

- acttype

  コールドスタンバイ構成は，HA モニタのモニタモードで動作するため，monitor を指定します。

- termcommand

  HADB 用の設定を記述したサーバの停止コマンドの絶対パスを指定します。

  サーバの停止コマンドはひな形を基に作成してください。作成方法については，「(4) コマンド用環境変数定義およびコマンドの作成」の「(c) サーバの停止コマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバの停止コマンドの作成』を参照してください。

- initial

  実行系の場合はonline を，待機系の場合はstandby を指定します。

- disk

  次に示すファイルシステムを含む VG の絶対パスを指定します。

  - DB ディレクトリを格納するファイルシステム
  - 作業用一時ファイルを格納するファイルシステム
  - 同義語辞書ファイルを格納するファイルシステム（同義語検索を行わない場合は，不要なファイルシステムです）
  - 監査証跡の出力先ディレクトリを作成するファイルシステム（監査証跡機能を使用しない場合は，不要なファイルシステムです）

  「図 17-1 コールドスタンバイ構成でのシステム構成例」の場合は，/dev/vg_hadb01，/dev/vg_hadb02，/dev/vg_hadb03，および/dev/vg_hadb04 を指定します。

- fs_name

  次に示すファイルシステムに対応する論理ボリュームの絶対パスを指定します。

  - DB ディレクトリを格納するファイルシステム
  - 作業用一時ファイルを格納するファイルシステム
  - 同義語辞書ファイルを格納するファイルシステム（同義語検索を行わない場合は，不要なファイルシステムです）
  - 監査証跡の出力先ディレクトリを作成するファイルシステム（監査証跡機能を使用しない場合は，不要なファイルシステムです）

  「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合は，/dev/vg_hadb01/hadb_db，/dev/vg_hadb02/hadb_workarea，/dev/vg_hadb03/hadb_syndict，および/dev/vg_hadb04/hadb_audit を指定します。

- fs_mount_dir

  次に示すファイルシステムをマウントするディレクトリのパスを指定します。実行系と待機系で同じパスを指定します。

  - DB ディレクトリを格納するファイルシステム
  - 作業用一時ファイルを格納するファイルシステム
  - 同義語辞書ファイルを格納するファイルシステム（同義語検索を行わない場合は，不要なファイルシステムです）
  - 監査証跡の出力先ディレクトリを作成するファイルシステム（監査証跡機能を使用しない場合は，不要なファイルシステムです）

- fs_mount_opt

  次に示すファイルシステムをマウントするためのマウントオプションを指定します。

  - DB ディレクトリを格納するファイルシステム
  - 作業用一時ファイルを格納するファイルシステム
  - 同義語辞書ファイルを格納するファイルシステム（同義語検索を行わない場合は，不要なファイルシステムです）
  - 監査証跡の出力先ディレクトリを作成するファイルシステム（監査証跡機能を使用しない場合は，不要なファイルシステムです）

- lan_updown

  LAN の状態設定ファイルを使用するため，use を指定します。

  LAN の状態設定ファイルには，エイリアス IP アドレスの設定をしてください。詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『LAN の状態設定ファイルの設定』を参照してください。

- patrolcommand

  HADB 用の設定を記述したサーバの監視コマンドの絶対パスを指定します。

サーバの監視コマンドはひな形を基に作成してください。作成方法については，「(4)　コマンド用環境変数定義およびコマンドの作成」の「(d)　サーバの監視コマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバの監視コマンドの作成』を参照してください。

- servexec_retry

  2 を指定します。

- waitserv_exec

  yes を指定します。

- ip_neck

  use を指定します。

- uoc_neck

  nouse を指定します。

- vg_neck

  DB ディレクトリを格納するファイルシステム，および監査証跡の出力先ディレクトリを作成するファイルシステムを含む VG には，use を指定します。

  そのほかの VG には，nouse を指定します。

- fs_neck

  DB ディレクトリを格納するファイルシステム，および監査証跡の出力先ディレクトリを作成するファイルシステムには，use を指定します。

  そのほかのファイルシステムには，nouse を指定します。

- actcommand

  HADB 用の設定を記述したサーバの起動コマンドの絶対パスを指定します。

  サーバの起動コマンドはひな形を基に作成してください。作成方法については，「(4)　コマンド用環境変数定義およびコマンドの作成」の「(b)　サーバの起動コマンドの作成」，およびマニュアル『HA モニタ Linux(R)(x86)編』の『サーバの起動コマンドの作成』を参照してください。

- scsi_device

  このオペランドには，SCSI リザーブの対象となるデバイス名を絶対パスで指定します。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクが次のどれかの条件を満たすときに，このオペランドを指定してください。

  - シングルパス構成の場合
  - VMware ESXi での仮想化環境の場合（DMMP を使用する場合を除く）
  - マルチパスソフトウェア（HDLM）による冗長化構成の場合

  このオペランドを指定する場合は，次のファイルシステムとディスクのデバイス名※を絶対パスで指定します。

  - 系切り替え対象のファイルシステム

    「図 17-1　コールドスタンバイ構成でのシステム構成例」のFS001～FS004 が該当します。

- DB エリアファイル用のディスク

  「図 17-1　コールドスタンバイ構成でのシステム構成例」のLU001～LU005，WRK001 が該当します。

なお，同義語検索を使用しない場合は，同義語辞書ファイルを格納するファイルシステムのデバイス名を指定する必要はありません。また，監査証跡機能を使用しない場合は，監査証跡の出力先ディレクトリを作成するファイルシステムのデバイス名を指定する必要はありません。

注※

シングルパス構成，および VMware ESXi での仮想化環境（DMMP を使用する場合を除く）の場合，このオペランドには，「17.2.4　ストレージ構成」の「シングルパス構成の場合のデバイス名の例」に記載されているwwn-から始まるパスではなく，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの SCSI リザーブに必要なオペランドの指定値の求め方』の例のようにscsi-から始まるパスを指定してください。

重要

次のファイルシステムのデバイス名は，このオペランドに指定しないでください。

- 系ローカルのファイルシステム

  「図 17-1　コールドスタンバイ構成でのシステム構成例」のLOC001，LOC002 が該当します。

- 共有のファイルシステム

  「図 17-1　コールドスタンバイ構成でのシステム構成例」のFS005 が該当します。

このオペランドを指定する場合は，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの SCSI リザーブに必要なオペランドの指定値の求め方』を参照してください。

- dmmp_device

  このオペランドには，SCSI リザーブの対象となるデバイスを DMMP の論理デバイスの絶対パスで指定します。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクがマルチパスソフトウェア（DMMP）による冗長化構成のときに，このオペランドを指定してください。

  このオペランドを指定する場合は，次のファイルシステムとディスクのデバイス名を絶対パスで指定します。

  - 系切り替え対象のファイルシステム

    「図 17-1　コールドスタンバイ構成でのシステム構成例」のFS001～FS004 が該当します。

  - DB エリアファイル用のディスク

    「図 17-1　コールドスタンバイ構成でのシステム構成例」のLU001～LU005，WRK001 が該当します。

  なお，同義語検索を使用しない場合は，同義語辞書ファイルを格納するファイルシステムのデバイス名を指定する必要はありません。また，監査証跡機能を使用しない場合は，監査証跡の出力先ディレクトリを作成するファイルシステムのデバイス名を指定する必要はありません。

重要

次のファイルシステムのデバイス名は，このオペランドに指定しないでください。

- 系ローカルのファイルシステム
  「図 17-1　コールドスタンバイ構成でのシステム構成例」のLOC001，LOC002 が該当します。
- 共有のファイルシステム
  「図 17-1　コールドスタンバイ構成でのシステム構成例」のFS005 が該当します。

このオペランドを指定する場合は，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの SCSI リザーブに必要なオペランドの指定値の求め方』を参照してください。

servers ファイルに指定する上記のオペランドの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『サーバ対応の環境設定（servers)』を参照してください。

(4)　コマンド用環境変数定義およびコマンドの作成

HA モニタとの連携のために，コマンド用環境変数定義と次のコマンドを作成してください。

- サーバの起動コマンド
- サーバの停止コマンド
- サーバの監視コマンド

$ADBDIR/sample/scripts 下に，コマンド用環境変数定義，および上記のコマンドのひな形があります。ひな形を基にコマンドを作成してください。

(a)　コマンド用環境変数定義の作成

$ADBDIR/sample/scripts/coldstandby.env がひな形です。このひな形をコピーして，次の値を修正してください。

- 環境変数ADBMGR
  HADB 管理者（OS ユーザ）のユーザ名を指定してください。
- 環境変数ADBDIR
  サーバディレクトリの絶対パスを指定してください。
- 環境変数ADBINFODIR
  トラブルシュート情報ファイルを格納するディレクトリの絶対パスを指定してください。

(b)　サーバの起動コマンドの作成

$ADBDIR/sample/scripts/coldstandby_act.sh がひな形です。このひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(c)　サーバの停止コマンドの作成

$ADBDIR/sample/scripts/coldstandby_term.sh がひな形です。このひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(d)　サーバの監視コマンドの作成

$ADBDIR/sample/scripts/coldstandby_patrol.sh がひな形です。このひな形をコピーして，次の個所を修正してください。

- source 文
  source 文の引数に，コマンド用環境変数定義を指定したファイルの絶対パスを指定してください。

(5)　各ファイルの指定例（系のリセットを使用する場合）

次のファイルの指定例を示します。系のリセットを使用する場合の指定例です。

- sysdef ファイル
- servers ファイル
- コマンド用環境変数定義ファイル（この例では/HADB/scripts/coldstandby.env）
- サーバの起動コマンドファイル（この例では/HADB/scripts/coldstandby_act.sh）
- サーバの停止コマンドファイル（この例では/HADB/scripts/coldstandby_term.sh）
- サーバの監視コマンドファイル（この例では/HADB/scripts/coldstandby_patrol.sh）

ここで説明する指定例は，「図 17-1　コールドスタンバイ構成でのシステム構成例」のシステム構成を前提としています。

(a)　sysdef ファイルの指定例

■サーバマシンhadb01（実行系）のsysdef ファイルの指定例

```
environment  name hadb01,
             address 1,                ...1
             patrol 60,
             lan path11,               ...2
             lanport HAmon1;           ...3
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             termcmd_at_abort nouse;
```

[説明]

1. この系のリセット優先度を指定します。この系は，実行系のため，リセット優先度をいちばん高くする必要があるので1 を指定します。
2. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.11 に対応するホスト名path11 を指定します。
3. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。

■サーバマシンhadb02（待機系）のsysdef ファイルの指定例

```
environment  name hadb02,
             address 2,                  ...1
             patrol 60,
             lan path21,                 ...2
             lanport HAmon1;             ...3
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             termcmd_at_abort nouse;
```

[説明]

1. この系のリセット優先度を指定します。この系は，待機系のため，2 を指定します。
2. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.12 に対応するホスト名path21 を指定します。
3. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。

(b) servers ファイルの指定例

ここで説明している例は，ファイルシステムに ext4 を使用していることを前提としています。ext4 以外のファイルシステムを使用する場合は，OS のmount コマンドのオプションがここで説明している例とは異なります。

■サーバマシンhadb01（実行系）のservers ファイルの指定例

```
server name  /HADB/server,                    ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb01:                  ...2
            /dev/vg_hadb02:
            /dev/vg_hadb03:
            /dev/vg_hadb04,
       lan_updown use,
       fs_name /dev/vg_hadb01/hadb_db:       ...3
               /dev/vg_hadb02/hadb_workarea:
               /dev/vg_hadb03/hadb_syndict:
               /dev/vg_hadb04/hadb_audit,
       fs_mount_dir /HADB/db:                ...4
                    /mnt/workarea:
                    /mnt/syndict:
```

```
                /mnt/audit,
        fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":    ...5
                     "-t ext4 -o defaults,noatime,_netdev":
                     "-t ext4 -o defaults,noatime,_netdev":
                     "-t ext4 -o defaults,noatime,_netdev",
        actcommand "/HADB/scripts/coldstandby_act.sh",         ...6
        termcommand "/HADB/scripts/coldstandby_term.sh",       ...7
        patrolcommand "/HADB/scripts/coldstandby_patrol.sh",   ...8
        servexec_retry 2,
        waitserv_exec yes,
        ip_neck use,
        uoc_neck nouse,
        vg_neck use:nouse:nouse:use,   ...9
        fs_neck use:nouse:nouse:use,   ...10
        initial online;                ...11
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - DB ディレクトリ
   - 作業用一時ファイルを格納するディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
   - 監査証跡の出力先ディレクトリ（監査証跡機能を使用する場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするための mount コマンドのオプションを指定します。
6. コールドスタンバイ構成で使用するサーバの起動コマンドの絶対パスを指定します。
7. コールドスタンバイ構成で使用するサーバの停止コマンドの絶対パスを指定します。
8. コールドスタンバイ構成で使用するサーバの監視コマンドの絶対パスを指定します。
9. 次のように指定します。
   - DB ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
   - 作業用一時ファイルを格納するディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 監査証跡の出力先ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 次のように指定します。

- DB ディレクトリを作成するファイルシステムに対しては，use を指定します。
- 作業用一時ファイルを作成するファイルシステムに対しては，nouse を指定します。
- 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対しては，nouse を指定します。
- 監査証跡の出力先ディレクトリを作成するファイルシステムに対しては，use を指定します。

11. 実行系とするため，online を指定します。

■サーバマシンhadb02（待機系）のservers ファイルの指定例

```
server name  /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb01:                         ...2
            /dev/vg_hadb02:
            /dev/vg_hadb03:
            /dev/vg_hadb04,
       lan_updown use,
       fs_name /dev/vg_hadb01/hadb_db:              ...3
               /dev/vg_hadb02/hadb_workarea:
               /dev/vg_hadb03/hadb_syndict:
               /dev/vg_hadb04/hadb_audit,
       fs_mount_dir /HADB/db:                       ...4
                    /mnt/workarea:
                    /mnt/syndict:
                    /mnt/audit,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":    ...5
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/coldstandby_act.sh",         ...6
       termcommand "/HADB/scripts/coldstandby_term.sh",       ...7
       patrolcommand "/HADB/scripts/coldstandby_patrol.sh",   ...8
       servexec_retry 2,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:nouse:nouse:use,   ...9
       fs_neck use:nouse:nouse:use,   ...10
       initial standby;               ...11
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - DB ディレクトリ
   - 作業用一時ファイルを格納するディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
   - 監査証跡の出力先ディレクトリ（監査証跡機能を使用する場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。

4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. コールドスタンバイ構成で使用するサーバの起動コマンドの絶対パスを指定します。
7. コールドスタンバイ構成で使用するサーバの停止コマンドの絶対パスを指定します。
8. コールドスタンバイ構成で使用するサーバの監視コマンドの絶対パスを指定します。
9. 次のように指定します。
   - DB ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
   - 作業用一時ファイルを格納するディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 監査証跡の出力先ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 次のように指定します。
    - DB ディレクトリを作成するファイルシステムに対しては，use を指定します。
    - 作業用一時ファイルを作成するファイルシステムに対しては，nouse を指定します。
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対しては，nouse を指定します。
    - 監査証跡の出力先ディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 待機系とするため，standby を指定します。

(c) コマンド用環境変数定義ファイルの指定例

この例では，コマンド用環境変数定義を/HADB/scripts/coldstandby.env に格納しています。指定内容は，実行系と待機系で同じにしてください。

■コマンド用環境変数定義ファイルの指定例

```
#!/bin/sh

# The environment variables for HADB
export ADBMGR=adbmanager                    ...1
export ADBDIR=/HADB/server                  ...2
export ADBINFODIR=/HADB/adbinfo             ...3
```

[説明]

1. HADB 管理者（OS ユーザ）のユーザ名adbmanager を指定します。
2. サーバディレクトリの絶対パスを指定します。

3. トラブルシュート情報ファイルを格納するディレクトリの絶対パスを指定します。

(d) サーバの起動コマンドファイルの指定例

この例では，サーバの起動コマンドを/HADB/scripts/coldstandby_act.sh に格納しています。指定内容は，実行系と待機系で同じにしてください。

■サーバの起動コマンドファイルの指定例

```
#!/bin/sh

# Sample of the actcommand for HADB

SU=/bin/su
ECHO=/bin/echo

# Setting environment variables for HADB
source /HADB/scripts/coldstandby.env       ...1

# Execute adbstart command
$SU - $ADBMGR -c "$ECHO y|$ADBDIR/bin/adbstart"

# always 0 return
exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/coldstandby.env を指定します。

(e) サーバの停止コマンドファイルの指定例

この例では，サーバの停止コマンドを/HADB/scripts/coldstandby_term.sh に格納しています。指定内容は，実行系と待機系で同じにしてください。

■サーバの停止コマンドファイルの指定例

```
#!/bin/sh

# Sample of the termcommand for HADB

STS_ACTIVE="ACTIVE"
STS_STOP="STOP"
STS_STARTING="STARTING"
STS_STOPPING="STOPPING"
STS_ABORT="ABORT"
STS_QUIESCE="QUIESCE"
STS_OFFLINE="OFFLINE"
STS_CHGMODE="CHGMODE"
STS_STOPWAIT="STOPWAIT"
STS_COREDUMP="COREDUMP"
STS_FORCE="FORCE"
STS_MAINTNCE="MAINTNCE"

AWK=/bin/awk
```

```
ECHO=/bin/echo
PS=/bin/ps
KILL=/bin/kill
GREP=/bin/grep
SU=/bin/su
READLINK=/bin/readlink

# Setting environment variables for HADB
source /HADB/scripts/coldstandby.env       ...1

# Resolve the path for adbsrvd
ADBSRVD=`$READLINK -e $ADBDIR/bin/adbsrvd`
if [ $? -ne 0 ]
then
  exit 0
fi

# Execute adbstop command
GET_STS=`$SU - $ADBMGR -c "$ADBDIR/bin/adbls -d srv 2>/dev/null" | $GREP -v SVID | $AWK '
{ if ($1 ~ /[0-9]+/) { print $2 } else { print $1 } }'`

case "$1" in
"-e" )
  # Normal stop operations of active system.
  # (when the monend command is executed.)
  if [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
       "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
       "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
       "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_MAINTNCE" ]
  then
    $SU - $ADBMGR -c "$ECHO y|$ADBDIR/bin/adbstop --force"
    STOPRES=$?
    if [ $STOPRES -ne 0 -a $STOPRES -ne 4 ]
    then
      ADB_ID=`$SU - $ADBMGR -c "$PS x" | $GREP $ADBSRVD | $GREP -v $GREP | $AWK '{print $
1}'`
      if [ "$ADB_ID" != "" ]
      then
        $SU - $ADBMGR -c "$KILL $ADB_ID"
      fi
    fi
  fi

  ;;
"-w" )
  # Plan stop operations of active system.
  # (when the monswap command is executed.)
  if [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
       "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
       "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
       "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_MAINTNCE" ]
  then
    $SU - $ADBMGR -c "$ECHO y|$ADBDIR/bin/adbstop --force"
    STOPRES=$?
    if [ $STOPRES -ne 0 -a $STOPRES -ne 4 ]
    then
      ADB_ID=`$SU - $ADBMGR -c "$PS x" | $GREP $ADBSRVD | $GREP -v $GREP | $AWK '{print $
1}'`
```

```
      if [ "$ADB_ID" != "" ]
      then
        $SU - $ADBMGR -c "$KILL $ADB_ID"
      fi
    fi
  fi

  ;;
"-c" )
  # Retry operations of active system.
  # (Before the act_command is resterted.)
  $SU - $ADBMGR -c "adbinfoget -l -o $ADBINFODIR"

  ;;
esac

# stop HADB
if [ "$GET_STS" = "$STS_STARTING" ]
then
  ADB_ID=`$SU - $ADBMGR -c "$PS x" | $GREP $ADBSRVD | $GREP -v $GREP | $AWK '{print $1}'`
  $SU - $ADBMGR -c "$KILL $ADB_ID"

  exit 0
fi

# Wait for end of HADB
while [ "$GET_STS" = "$STS_ACTIVE"   -o "$GET_STS" = "$STS_STARTING" -o ¥
        "$GET_STS" = "$STS_STOPPING" -o "$GET_STS" = "$STS_QUIESCE"  -o ¥
        "$GET_STS" = "$STS_OFFLINE"  -o "$GET_STS" = "$STS_CHGMODE"  -o ¥
        "$GET_STS" = "$STS_STOPWAIT" -o "$GET_STS" = "$STS_COREDUMP" -o ¥
        "$GET_STS" = "$STS_MAINTNCE" ]
do
  GET_STS=`$SU - $ADBMGR -c "$ADBDIR/bin/adbls -d srv 2>/dev/null" | $GREP -v SVID | $AWK
 '{ if ($1 ~ /[0-9]+/) { print $2 } else { print $1 } }'`
  sleep 1
done

exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/coldstandby.env を指定します。

(f)　サーバの監視コマンドファイルの指定例

この例では，サーバの監視コマンドを/HADB/scripts/coldstandby_patrol.sh に格納しています。指定内容は，実行系と待機系で同じにしてください。

■サーバの監視コマンドファイルの指定例

```
#!/bin/sh

# Sample of the patrolcommand for HADB

AWK=/bin/awk
```

```
GREP=/bin/grep
PS=/bin/ps
PGREP=/usr/bin/pgrep
PKILL=/usr/bin/pkill
SU=/bin/su
READLINK=/bin/readlink

# Setting environment variables for HADB
source /HADB/scripts/coldstandby.env       ...1

# Resolve the path for adbsrvd
ADBSRVD=`$READLINK -e $ADBDIR/bin/adbsrvd`
if [ $? -ne 0 ]
then
  exit 0
fi

# Get adbsrvd process id
CHKPID=`$SU - $ADBMGR -c "$PS x" | $GREP $ADBSRVD | $GREP -v $GREP | $AWK '{print $1}'`
if [ "$CHKPID" = "" ]
then
  exit 0
fi

# Execute adbmonitor command for active system
$SU - $ADBMGR -c "$ADBDIR/bin/adbmonitor -n" &
PID=$!

# Trap SIGTERM and terminate adbmonitor
trap "$PKILL -P $PID" 15

## Wait process terminated
CPID=""
CHKPID=`$PS aux | $AWK -v PID=$PID '{ if ($2 == PID) { print $2 } }'`
while [ "$CHKPID" != "" ]
do

  ## Get adbmonitor process id
  if [ "$CPID" = "" ]
  then
    CPID=`$PGREP -P $PID`
  fi

  ## Wait
  if [ "$CPID" != "" ]
  then
    CHKCPID=`$PS aux | $AWK -v CPID=$CPID '{ if ($2 == CPID) { print $2 } }'`
    while [ "$CHKCPID" != "" ]
    do
      sleep 1
      CHKCPID=`$PS aux | $AWK -v CPID=$CPID '{ if ($2 == CPID) { print $2 } }'`
    done
  fi

  sleep 1
  CHKPID=`$PS aux | $AWK -v PID=$PID '{ if ($2 == PID) { print $2 } }'`
done
```

```
exit 0
```

[説明]

1. source 文には，コマンド用環境変数定義ファイルの絶対パス/HADB/scripts/coldstandby.env を指定します。

(6) 各ファイルの指定例（共有ディスクの SCSI リザーブを使用する場合）

次のファイルの指定例を示します。共有ディスクの SCSI リザーブを使用する場合の指定例です。

- sysdef ファイル
- servers ファイル
- コマンド用環境変数定義ファイル（この例では/HADB/scripts/coldstandby.env）
- サーバの起動コマンドファイル（この例では/HADB/scripts/coldstandby_act.sh）
- サーバの停止コマンドファイル（この例では/HADB/scripts/coldstandby_term.sh）
- サーバの監視コマンドファイル（この例では/HADB/scripts/coldstandby_patrol.sh）

ここで説明する指定例は，「図 17-1　コールドスタンバイ構成でのシステム構成例」のシステム構成を前提としています。

(a) sysdef ファイルの指定例

■サーバマシンhadb01（実行系）のsysdef ファイルの指定例

```
environment  name hadb01,
             address 1,                  ...1
             patrol 60,
             lan path11,                 ...2
             lanport HAmon1;             ...3
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             termcmd_at_abort nouse,
             fence_reset nouse,          ...4
             fence_scsi use,             ...5
             fence_lan nouse;
```

[説明]

1. この系のリセット優先度を指定します。この系は，実行系のため，リセット優先度をいちばん高くする必要があるので1 を指定します。
2. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.11 に対応するホスト名path11 を指定します。
3. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。

4. 系のリセットを使用しないため，nouse を指定します。

5. 共有ディスクの SCSI リザーブを使用するため，use を指定します。

■サーバマシンhadb02（待機系）のsysdef ファイルの指定例

```
environment  name hadb02,
             address 2,                  ...1
             patrol 60,
             lan path21,                 ...2
             lanport HAmon1;             ...3
function     pathpatrol 240,
             connect_retry 5:200,
             monbegin_restart nouse,
             termcmd_at_abort nouse,
             fence_reset nouse,          ...4
             fence_scsi use,             ...5
             fence_lan nouse;
```

[説明]

1. この系のリセット優先度を指定します。この系は，待機系のため，2 を指定します。

2. HA モニタの監視パスとして使用する LAN のホスト名を指定します。IP アドレス172.16.0.12 に対応するホスト名path21 を指定します。

3. HA モニタの監視パスとして使用する LAN のサービス名を指定します。ポート番号7777 に対応するサービス名HAmon1 を指定します。

4. 系のリセットを使用しないため，nouse を指定します。

5. 共有ディスクの SCSI リザーブを使用するため，use を指定します。

(b) servers ファイルの指定例

次に示す場合のservers ファイルの指定例を説明します。

- シングルパス構成の場合
- マルチパスソフトウェア（DMMP）による冗長化構成の場合

なお，ここで説明している例は，ファイルシステムに ext4 を使用していることを前提としています。ext4 以外のファイルシステムを使用する場合は，OS のmount コマンドのオプションがここで説明している例とは異なります。

■シングルパス構成の場合のservers ファイルの指定例

・サーバマシンhadb01（実行系）のservers ファイルの指定例

```
server name  /HADB/server,                  ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb01:                 ...2
            /dev/vg_hadb02:
            /dev/vg_hadb03:
            /dev/vg_hadb04,
       lan_updown use,
```

```
fs_name /dev/vg_hadb01/hadb_db:         ...3
        /dev/vg_hadb02/hadb_workarea:
        /dev/vg_hadb03/hadb_syndict:
        /dev/vg_hadb04/hadb_audit,
fs_mount_dir /HADB/db:                  ...4
             /mnt/workarea:
             /mnt/syndict:
             /mnt/audit,
fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":    ...5
             "-t ext4 -o defaults,noatime,_netdev":
             "-t ext4 -o defaults,noatime,_netdev":
             "-t ext4 -o defaults,noatime,_netdev",
actcommand "/HADB/scripts/coldstandby_act.sh",         ...6
termcommand "/HADB/scripts/coldstandby_term.sh",       ...7
patrolcommand "/HADB/scripts/coldstandby_patrol.sh",   ...8
servexec_retry 2,
waitserv_exec yes,
ip_neck use,
uoc_neck nouse,
vg_neck use:nouse:nouse:use,                                        ...9
fs_neck use:nouse:nouse:use,                                        ...10
scsi_device /dev/disk/by-id/scsi-360060e8010205850051104c500000005:   ...11
            /dev/disk/by-id/scsi-360060e8010205850051104c500000006:
            /dev/disk/by-id/scsi-360060e8010205850051104c500000007:
            /dev/disk/by-id/scsi-360060e8010205850051104c500000008:
            /dev/disk/by-id/scsi-360060e8010205850051104c50000000f:
            /dev/disk/by-id/scsi-360060e8010205850051104c50000000e:
            /dev/disk/by-id/scsi-360060e8010205850051104c500000010:
            /dev/disk/by-id/scsi-360060e8010205850051104c500000011:
            /dev/disk/by-id/scsi-360060e8010205850051104c500000013:
            /dev/disk/by-id/scsi-360060e8010205850051104c50000000d,
initial online;                                                     ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - DB ディレクトリ
   - 作業用一時ファイルを格納するディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
   - 監査証跡の出力先ディレクトリ（監査証跡機能を使用する場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするための mount コマンドのオプションを指定します。
6. コールドスタンバイ構成で使用するサーバの起動コマンドの絶対パスを指定します。
7. コールドスタンバイ構成で使用するサーバの停止コマンドの絶対パスを指定します。

8. コールドスタンバイ構成で使用するサーバの監視コマンドの絶対パスを指定します。

9. 次のように指定します。

   - DB ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
   - 作業用一時ファイルを格納するディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 監査証跡の出力先ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。

10. 次のように指定します。

    - DB ディレクトリを作成するファイルシステムに対しては，use を指定します。
    - 作業用一時ファイルを作成するファイルシステムに対しては，nouse を指定します。
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対しては，nouse を指定します。
    - 監査証跡の出力先ディレクトリを作成するファイルシステムに対しては，use を指定します。

11. 系切り替え対象のファイルシステムがあるディスクと，DB エリアがあるディスクの絶対パスを指定します。実行系と待機系で，絶対パスの指定順序を同じにしてください。

12. 実行系とするため，online を指定します。

・サーバマシンhadb02（待機系）のservers ファイルの指定例

```
server name  /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb01:                         ...2
            /dev/vg_hadb02:
            /dev/vg_hadb03:
            /dev/vg_hadb04,
       lan_updown use,
       fs_name /dev/vg_hadb01/hadb_db:              ...3
               /dev/vg_hadb02/hadb_workarea:
               /dev/vg_hadb03/hadb_syndict:
               /dev/vg_hadb04/hadb_audit,
       fs_mount_dir /HADB/db:                       ...4
                    /mnt/workarea:
                    /mnt/syndict:
                    /mnt/audit,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":    ...5
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/coldstandby_act.sh",         ...6
       termcommand "/HADB/scripts/coldstandby_term.sh",       ...7
       patrolcommand "/HADB/scripts/coldstandby_patrol.sh",   ...8
       servexec_retry 2,
       waitserv_exec yes,
```

```
        ip_neck use,
        uoc_neck nouse,
        vg_neck use:nouse:nouse:use,                                          ...9
        fs_neck use:nouse:nouse:use,                                          ...10
        scsi_device /dev/disk/by-id/scsi-360060e8010205850051104c500000005:   ...11
                    /dev/disk/by-id/scsi-360060e8010205850051104c500000006:
                    /dev/disk/by-id/scsi-360060e8010205850051104c500000007:
                    /dev/disk/by-id/scsi-360060e8010205850051104c500000008:
                    /dev/disk/by-id/scsi-360060e8010205850051104c50000000f:
                    /dev/disk/by-id/scsi-360060e8010205850051104c50000000e:
                    /dev/disk/by-id/scsi-360060e8010205850051104c500000010:
                    /dev/disk/by-id/scsi-360060e8010205850051104c500000011:
                    /dev/disk/by-id/scsi-360060e8010205850051104c500000013:
                    /dev/disk/by-id/scsi-360060e8010205850051104c50000000d,
        initial standby;                                                      ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - DB ディレクトリ
   - 作業用一時ファイルを格納するディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
   - 監査証跡の出力先ディレクトリ（監査証跡機能を使用する場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするための mount コマンドのオプションを指定します。
6. コールドスタンバイ構成で使用するサーバの起動コマンドの絶対パスを指定します。
7. コールドスタンバイ構成で使用するサーバの停止コマンドの絶対パスを指定します。
8. コールドスタンバイ構成で使用するサーバの監視コマンドの絶対パスを指定します。
9. 次のように指定します。
   - DB ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
   - 作業用一時ファイルを格納するディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 監査証跡の出力先ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 次のように指定します。

- DB ディレクトリを作成するファイルシステムに対しては，use を指定します。
- 作業用一時ファイルを作成するファイルシステムに対しては，nouse を指定します。
- 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対しては，nouse を指定します。
- 監査証跡の出力先ディレクトリを作成するファイルシステムに対しては，use を指定します。

11. 系切り替え対象のファイルシステムがあるディスクの絶対パスと，DB エリアがあるディスクの絶対パスを指定します。実行系と待機系で，絶対パスの指定順序を同じにしてください。

12. 待機系とするため，standby を指定します。

■マルチパスソフトウェア（DMMP）による冗長化構成の場合のservers ファイルの指定例

・サーバマシンhadb01（実行系）のservers ファイルの指定例

```
server name  /HADB/server,                          ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb01:                         ...2
            /dev/vg_hadb02:
            /dev/vg_hadb03:
            /dev/vg_hadb04,
       lan_updown use,
       fs_name /dev/vg_hadb01/hadb_db:              ...3
               /dev/vg_hadb02/hadb_workarea:
               /dev/vg_hadb03/hadb_syndict:
               /dev/vg_hadb04/hadb_audit,
       fs_mount_dir /HADB/db:                       ...4
                    /mnt/workarea:
                    /mnt/syndict:
                    /mnt/audit,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":    ...5
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/coldstandby_act.sh",         ...6
       termcommand "/HADB/scripts/coldstandby_term.sh",       ...7
       patrolcommand "/HADB/scripts/coldstandby_patrol.sh",   ...8
       servexec_retry 2,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:nouse:nouse:use,       ...9
       fs_neck use:nouse:nouse:use,       ...10
       dmmp_device /dev/mapper/mpath1:    ...11
                   /dev/mapper/mpath2:
                   /dev/mapper/mpath3:
                   /dev/mapper/mpath4:
                   /dev/mapper/mpath11:
                   /dev/mapper/mpath12:
                   /dev/mapper/mpath13:
                   /dev/mapper/mpath14:
                   /dev/mapper/mpath15:
                   /dev/mapper/mpath16,
       initial online;                    ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - DB ディレクトリ
   - 作業用一時ファイルを格納するディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
   - 監査証跡の出力先ディレクトリ（監査証跡機能を使用する場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。
4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするための mount コマンドのオプションを指定します。
6. コールドスタンバイ構成で使用するサーバの起動コマンドの絶対パスを指定します。
7. コールドスタンバイ構成で使用するサーバの停止コマンドの絶対パスを指定します。
8. コールドスタンバイ構成で使用するサーバの監視コマンドの絶対パスを指定します。
9. 次のように指定します。
   - DB ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
   - 作業用一時ファイルを格納するディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 監査証跡の出力先ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 次のように指定します。
    - DB ディレクトリを作成するファイルシステムに対しては，use を指定します。
    - 作業用一時ファイルを作成するファイルシステムに対しては，nouse を指定します。
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対しては，nouse を指定します。
    - 監査証跡の出力先ディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 系切り替え対象のファイルシステムがあるディスクの絶対パスと，DB エリアがあるディスクの絶対パスを指定します。実行系と待機系で，絶対パスの指定順序を同じにしてください。
12. 実行系とするため，online を指定します。

・サーバマシンhadb02（待機系）のservers ファイルの指定例

```
server name  /HADB/server,                        ...1
       alias HADB,
       acttype monitor,
       disk /dev/vg_hadb01:                      ...2
            /dev/vg_hadb02:
            /dev/vg_hadb03:
            /dev/vg_hadb04,
       lan_updown use,
       fs_name /dev/vg_hadb01/hadb_db:           ...3
               /dev/vg_hadb02/hadb_workarea:
               /dev/vg_hadb03/hadb_syndict:
               /dev/vg_hadb04/hadb_audit,
       fs_mount_dir /HADB/db:                    ...4
                    /mnt/workarea:
                    /mnt/syndict:
                    /mnt/audit,
       fs_mount_opt "-t ext4 -o defaults,noatime,_netdev":    ...5
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev":
                    "-t ext4 -o defaults,noatime,_netdev",
       actcommand "/HADB/scripts/coldstandby_act.sh",         ...6
       termcommand "/HADB/scripts/coldstandby_term.sh",       ...7
       patrolcommand "/HADB/scripts/coldstandby_patrol.sh",   ...8
       servexec_retry 2,
       waitserv_exec yes,
       ip_neck use,
       uoc_neck nouse,
       vg_neck use:nouse:nouse:use,       ...9
       fs_neck use:nouse:nouse:use,       ...10
       dmmp_device /dev/mapper/mpath1:    ...11
                   /dev/mapper/mpath2:
                   /dev/mapper/mpath3:
                   /dev/mapper/mpath4:
                   /dev/mapper/mpath11:
                   /dev/mapper/mpath12:
                   /dev/mapper/mpath13:
                   /dev/mapper/mpath14:
                   /dev/mapper/mpath15:
                   /dev/mapper/mpath16,
       initial standby;                   ...12
```

[説明]

1. サーバディレクトリの絶対パスを指定します。
2. 次のディレクトリを作成するファイルシステムがある VG の絶対パスを指定します。
   - DB ディレクトリ
   - 作業用一時ファイルを格納するディレクトリ
   - 同義語辞書ファイルの格納ディレクトリ（同義語検索を行う場合に作成するディレクトリです）
   - 監査証跡の出力先ディレクトリ（監査証跡機能を使用する場合に作成するディレクトリです）
3. 2.で示したディレクトリを作成するファイルシステムを構築した LV の絶対パスを指定します。

4. 2.で示したディレクトリを作成するファイルシステムをマウントするマウントポイントの絶対パスを指定します。
5. 2.で示したディレクトリを作成するファイルシステムをマウントするためのmount コマンドのオプションを指定します。
6. コールドスタンバイ構成で使用するサーバの起動コマンドの絶対パスを指定します。
7. コールドスタンバイ構成で使用するサーバの停止コマンドの絶対パスを指定します。
8. コールドスタンバイ構成で使用するサーバの監視コマンドの絶対パスを指定します。
9. 次のように指定します。
   - DB ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
   - 作業用一時ファイルを格納するディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムがある VG に対しては，nouse を指定します。
   - 監査証跡の出力先ディレクトリを作成するファイルシステムがある VG に対しては，use を指定します。
10. 次のように指定します。
    - DB ディレクトリを作成するファイルシステムに対しては，use を指定します。
    - 作業用一時ファイルを作成するファイルシステムに対しては，nouse を指定します。
    - 同義語辞書ファイルの格納ディレクトリを作成するファイルシステムに対しては，nouse を指定します。
    - 監査証跡の出力先ディレクトリを作成するファイルシステムに対しては，use を指定します。
11. 系切り替え対象のファイルシステムがあるディスクの絶対パスと，DB エリアがあるディスクの絶対パスを指定します。実行系と待機系で，絶対パスの指定順序を同じにしてください。
12. 待機系とするため，standby を指定します。

(c)　コマンド用環境変数定義ファイルの指定例

コマンド用環境変数定義ファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合)」の「(c)　コマンド用環境変数定義ファイルの指定例」を参照してください。

(d)　サーバの起動コマンドファイルの指定例

サーバの起動コマンドファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合)」の「(d)　サーバの起動コマンドファイルの指定例」を参照してください。

(e)　サーバの停止コマンドファイルの指定例

サーバの停止コマンドファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合)」の「(e)　サーバの停止コマンドファイルの指定例」を参照してください。

(f)　サーバの監視コマンドファイルの指定例

サーバの監視コマンドファイルの指定例については，「(5)　各ファイルの指定例（系のリセットを使用する場合）」の「(f)　サーバの監視コマンドファイルの指定例」を参照してください。

(7)　HA モニタの起動設定

OS の起動時に，自動的に HA モニタが起動するように設定してください。設定方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの起動からサーバの起動までを自動化する』を参照してください。

17.3.5　各 HADB サーバでのサーバ定義の作成

コールドスタンバイ構成の場合は，実行系と待機系の両方の HADB サーバでサーバ定義を作成する必要があります。

重要

- サーバ定義の指定は，実行系と待機系の両方の HADB サーバで必ず同じにしてください。
- adb_sys_multi_node_info オペランドは指定できません。

サーバ定義を作成する際，adb_blk_path_wrk オペランドを必ず指定してください。adb_blk_path_wrk オペランドには，adbinit コマンドの初期設定オプションのadb_init_wrk_blk_path に指定したパスを指定します。詳細については，「17.3.7　データベースの作成」を参照してください。

17.3.6　クライアント定義の作成

クライアント定義の指定例を次に示します。

クライアント定義の作成については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合）』の『クライアント定義の作成』を参照してください。クライアント定義のオペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』の『クライアント定義のオペランドの内容』を参照してください。

■クライアント定義の指定例

```
set adb_clt_rpc_srv_host = 10.196.108.143      ...1
set adb_clt_rpc_srv_port = 23650
set adb_clt_ap_name = ”AP01”
```

[説明]

1. 接続先のホスト名として，クライアントーサーバ間の通信で使用するエイリアス IP アドレス 10.196.108.143 を指定します。

17.3.7　データベースの作成

コールドスタンバイ構成の場合のデータベースの作成手順を説明します。

(1)　データベース作成の手順

データベースの作成手順の流れを次に示します。

1. 〔実行系での作業〕DB ディレクトリ用のファイルシステムを構築する
2. 〔実行系での作業〕DB ディレクトリ用のファイルシステムをマウントする
3. 〔実行系での作業〕データベースを作成する
4. 〔実行系での作業〕DB ディレクトリ用のファイルシステムをアンマウントする
5. 〔待機系での作業〕DB ディレクトリ用のファイルシステムをマウントする
6. 〔待機系での作業〕DB ディレクトリ用のファイルシステムをアンマウントする
7. 〔実行系での作業〕作業用一時ファイルを格納するファイルシステムを構築する
8. 〔実行系・待機系での作業〕作業用一時ファイルを格納するファイルシステムをマウントおよびアンマウントする
9. 〔実行系での作業〕同義語辞書ファイルを格納するファイルシステムを構築する※1
10. 〔実行系・待機系での作業〕同義語辞書ファイルを格納するファイルシステムをマウントおよびアンマウントする※1
11. 〔実行系での作業〕監査証跡の出力先ディレクトリを作成するファイルシステムを構築する※2
12. 〔実行系・待機系での作業〕監査証跡の出力先ディレクトリを作成するファイルシステムをマウントおよびアンマウントする※2
13. 〔実行系での作業〕共有するファイルシステムを構築する
14. 〔実行系・待機系での作業〕ファイルシステムを共有化する

注※1

同義語検索をする場合に必要な作業です。

注※2

監査証跡機能を使用する場合に必要な作業です。

(2)　〔実行系での作業〕DB ディレクトリ用のファイルシステムの構築

この手順は，スーパユーザで実施してください。

OS のコマンドを実行系で実行し，DB ディレクトリ用のファイルシステムを構築します。

DB ディレクトリ用の LV として作成した/dev/vg_hadb01/hadb_db を，ext4 のファイルシステムで構築する場合の実行例を次に示します。

■コマンドの実行例

```
mkfs -t ext4 /dev/vg_hadb01/hadb_db
```

(3) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのマウント

この手順は，スーパユーザで実施してください。

次に示す手順に従って，実行系で DB ディレクトリ用のファイルシステムをマウントします。

手順

1. DB ディレクトリ用のマウントポイントを作成する
   OS のコマンドを実行し，DB ディレクトリ用のマウントポイントを作成します。
   ■コマンドの実行例

```
mkdir -p /HADB/db
```

2. DB ディレクトリ用のファイルシステムがある VG を活性化する
   OS のコマンドを実行し，DB ディレクトリ用のファイルシステムがある VG を活性化します。
   ■コマンドの実行例

```
vgchange -a y /dev/vg_hadb01
```

3. DB ディレクトリ用ファイルシステムをマウントする
   OS のコマンドを実行し，DB ディレクトリ用ファイルシステムをマウントします。
   ■コマンドの実行例

```
mount /dev/vg_hadb01/hadb_db /HADB/db -t ext4 -o defaults,noatime,_netdev
```

4. DB ディレクトリのオーナーを変更する
   OS のコマンドを実行し，DB ディレクトリのオーナーを変更します。DB ディレクトリを使用する HADB 管理者（OS ユーザ）と HADB 管理グループを指定してください。
   HADB 管理者（OS ユーザ）のユーザ名がadbmanager，HADB 管理グループがadbgroup の場合の実行例を次に示します。
   ■コマンドの実行例

```
chown adbmanager.adbgroup /HADB/db
```

(4) 〔実行系での作業〕データベースの作成

この手順は，HADB 管理者で実施してください。

次に示す手順に従って，実行系でデータベースを作成します。

手順

1. DB ディレクトリ用のマウントポイントに，DB ディレクトリ用のファイルシステムがマウントされていることを確認する
2. adbinit コマンドを実行し，データベースを作成する

   adbinit コマンドの初期設定オプションの指定例を次に示します。

   ■初期設定オプションの指定例（シングルパス構成の場合）

```
set adb_init_dbarea_initialize = Y
set adb_init_wrk_blk_path = /dev/disk/by-id/wwn-0x60060e8010205850051104c50000000d
set adb_init_mst_blk_path = /dev/disk/by-id/wwn-0x60060e8010205850051104c50000000e
set adb_init_dic_blk_path = /dev/disk/by-id/wwn-0x60060e8010205850051104c50000000f
set adb_init_stbl_blk_path = /dev/disk/by-id/wwn-0x60060e8010205850051104c500000010
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x60060e8010205850051104c500000011
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/disk/by-id/wwn-0x60060e8010205850051104c500000013
```

   ■初期設定オプションの指定例（マルチパスソフトウェア（DMMP）による冗長化構成の場合）

```
set adb_init_dbarea_initialize = Y
set adb_init_wrk_blk_path = /dev/mapper/mpath16
set adb_init_mst_blk_path = /dev/mapper/mpath12
set adb_init_dic_blk_path = /dev/mapper/mpath11
set adb_init_stbl_blk_path = /dev/mapper/mpath13
adbinitdbarea -n ADBUTBL01 -i 2G ¥
              -v /dev/mapper/mpath14
adbinitdbarea -n ADBUIDX01 -i 2G ¥
              -v /dev/mapper/mpath15
```

(5) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのアンマウント

この手順は，スーパユーザで実施してください。

次に示す手順に従って，実行系で DB ディレクトリ用のファイルシステムをアンマウントします。

手順

1. DB ディレクトリ用ファイルシステムをアンマウントする

   OS のコマンドを実行し，DB ディレクトリ用ファイルシステムをアンマウントします。

   ■コマンドの実行例

```
umount /dev/vg_hadb01/hadb_db
```

2. DB ディレクトリ用のファイルシステムがある VG を非活性化する

   OS のコマンドを実行系で実行し，DB ディレクトリ用のファイルシステムがある VG を非活性化します。

   ■コマンドの実行例

```
vgchange -a n /dev/vg_hadb01
```

(6) 〔待機系での作業〕DB ディレクトリ用のファイルシステムのマウント

この手順は，スーパユーザで実施してください。

実行系と同様に，待機系で DB ディレクトリ用のファイルシステムをマウントします。実行系と同じパスになるようにマウントポイントを作成してください。

ファイルシステムをマウントする方法については，「(3) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのマウント」を参照してください。

> **重要**
>
> 待機系では，DB ディレクトリ用のファイルシステムのマウントポイントを作成するだけで，adbinit コマンドは実行しません。

(7) 〔待機系での作業〕DB ディレクトリ用のファイルシステムのアンマウント

この手順は，スーパユーザで実施してください。

実行系と同様に，待機系で DB ディレクトリ用のファイルシステムをアンマウントします。

ファイルシステムをアンマウントする方法については，「(5) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのアンマウント」を参照してください。

(8) 〔実行系での作業〕作業用一時ファイルを格納するファイルシステムの構築

この手順は，スーパユーザで実施してください。

OS のコマンドを実行系で実行し，作業用一時ファイルを格納するファイルシステムを構築します。

作業用一時ファイルを格納する LV として作成した/dev/vg_hadb02/hadb_workarea を，ext4 のファイルシステムで構築する場合の実行例を次に示します。

■コマンドの実行例

```
mkfs -t ext4 /dev/vg_hadb02/hadb_workarea
```

(9) 〔実行系・待機系での作業〕作業用一時ファイルを格納するファイルシステムのマウントおよびアンマウント

この手順は，スーパユーザで実施してください。

実行系で作業用一時ファイルを格納するファイルシステムをマウントしたあと，アンマウントします。

待機系でも同様に，作業用一時ファイルを格納するファイルシステムをマウントしたあと，アンマウントします。

実行系と待機系で同じパスになるように，マウントポイントを作成してください。

ファイルシステムをマウントする方法については，「(3) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのマウント」を参照してください。ファイルシステムをアンマウントする方法については，「(5) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのアンマウント」を参照してください。

(10) 〔実行系での作業〕同義語辞書ファイルを格納するファイルシステムの構築

同義語検索をする場合は，この手順を実施する必要があります。

この手順は，スーパユーザで実施してください。

OS のコマンドを実行系で実行し，同義語辞書ファイルを格納するファイルシステムを構築します。

同義語辞書ファイルを格納する LV として作成した/dev/vg_hadb03/hadb_syndict を，ext4 のファイルシステムで構築する場合の実行例を次に示します。

■コマンドの実行例

```
mkfs -t ext4 /dev/vg_hadb03/hadb_syndict
```

(11) 〔実行系・待機系での作業〕同義語辞書ファイルを格納するファイルシステムのマウントおよびアンマウント

同義語検索をする場合は，この手順を実施する必要があります。

この手順は，スーパユーザで実施してください。

実行系で同義語辞書ファイルを格納するファイルシステムをマウントしたあと，アンマウントします。

待機系でも同様に，同義語辞書ファイルを格納するファイルシステムをマウントしたあと，アンマウントします。

実行系と待機系で同じパスになるように，マウントポイントを作成してください。

ファイルシステムをマウントする方法については，「(3) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのマウント」を参照してください。ファイルシステムをアンマウントする方法については，「(5) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのアンマウント」を参照してください。

(12) 〔実行系での作業〕監査証跡の出力先ディレクトリを作成するファイルシステムの構築

監査証跡機能を使用する場合は，この手順を実施する必要があります。

この手順は，スーパユーザで実施してください。

OS のコマンドを実行系で実行し，監査証跡の出力先ディレクトリを作成するファイルシステムを構築します。

監査証跡ファイルを出力する LV として作成した/dev/vg_hadb04/hadb_audit を，ext4 のファイルシステムで構築する場合の実行例を次に示します。

■コマンドの実行例

```
mkfs -t ext4 /dev/vg_hadb04/hadb_audit
```

(13) 〔実行系・待機系での作業〕監査証跡の出力先ディレクトリを作成するファイルシステムのマウントおよびアンマウント

監査証跡機能を使用する場合は，この手順を実施する必要があります。

この手順は，スーパユーザで実施してください。

実行系で，監査証跡の出力先ディレクトリを作成するファイルシステムをマウントしたあとに，アンマウントします。

待機系でも同様に，監査証跡の出力先ディレクトリを作成するファイルシステムをマウントしたあとに，アンマウントします。

実行系と待機系で同じパスになるように，マウントポイントを作成してください。

ファイルシステムをマウントする方法については，「(3) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのマウント」を参照してください。ファイルシステムをアンマウントする方法については，「(5) 〔実行系での作業〕DB ディレクトリ用のファイルシステムのアンマウント」を参照してください。

(14) 〔実行系での作業〕共有するファイルシステムの構築

この手順は，スーパユーザで実施してください。

実行系と待機系で，次に示すファイルシステムを共有します。

- データインポートで使用する入力データファイルを格納するファイルシステム
- ADB_CSVREAD 関数で使用する CSV ファイルを格納するファイルシステム
- アーカイブディレクトリ用のファイルシステム
- 監査証跡の保存先ディレクトリを作成するファイルシステム

これらのファイルシステムを構築します。

(15) 〔実行系・待機系での作業〕ファイルシステムの共有化

「(14) 〔実行系での作業〕共有するファイルシステムの構築」で構築したファイルシステムを，実行系と待機系で共有します。

NFS を用いて共有する手順の例を示します。この手順は，スーパユーザで実施してください。

手順

1. NFS サーバを用意し，実行系と待機系で共有したいディレクトリをエクスポートする
2. 実行系と待機系で同じパスになるように，手順 1.でエクスポートしたディレクトリをマウントする

17.4　コールドスタンバイ構成の開始と終了

ここでは，コールドスタンバイ構成の開始および終了方法について説明します。

運用する際には，あわせて「10.2　HADB サーバの開始・終了と稼働モード」も参照してください。

17.4.1　コールドスタンバイ構成の開始方法

(1)　コールドスタンバイ構成の開始手順

コールドスタンバイ構成の開始手順を次に示します。

手順

1. 実行系と待機系で HA モニタが起動していることを確認する
   確認方法については，マニュアル『HA モニタ Linux(R)(x86)編』の『システムの運用』を参照してください。
2. 実行系と待機系で HA モニタのmonbegin コマンドを実行する

> **メモ**
>
> HA モニタのmonbegin コマンドを実行すると，その処理の延長で，実行系でadbstart コマンドが自動的に実行されます。

(2)　コールドスタンバイ構成の開始モード

コールドスタンバイ構成には，次の開始モードがあります。

表 17-4　コールドスタンバイ構成の開始モード

項番	開始モード	実行するコマンド	説明	前回の終了モード
1	正常開始	monbegin	通常の開始モードです。 正常開始では，データベースの回復処理は行いません。	コールドスタンバイ構成が正常終了した場合
2	再開始		前回の終了モードが右記の場合，自動的に再開始となります。 再開始では，実行系の HADB サーバがシステムログを使用してデータベースの回復処理を行います。	コールドスタンバイ構成が異常終了，または強制終了した場合

(3)　コールドスタンバイ構成の開始処理完了の確認

次に示す内容を確認してください。

- 実行系で HA モニタのmonshow コマンドを実行します。実行系の状態が実行処理中（ONL）に，待機系の状態が待機処理中（SBY）になっていることを確認してください。

  monshow コマンドについては，マニュアル『HA モニタ Linux(R)(x86)編』の『コマンド』を参照してください。
- 実行系の HADB サーバのメッセージログファイルを参照します。adbstart コマンドのリターンコードが，0 または4 となっていることを確認してください。

17.4.2　コールドスタンバイ構成の終了方法

(1)　コールドスタンバイ構成の終了手順

(a)　コールドスタンバイ構成を正常終了させる場合

コールドスタンバイ構成を正常終了させる手順を次に示します。

手順

1. 実行系でadbstop コマンドを実行する
2. 実行系で HA モニタのmonend コマンドを実行する

(b)　コールドスタンバイ構成を強制終了させる場合

実行系で HA モニタのmonend コマンドを実行してください。

(2)　コールドスタンバイ構成の終了モード

コールドスタンバイ構成には，次の終了モードがあります。

表 17-5　コールドスタンバイ構成の終了モード

項番	終了形態		実行するコマンドおよびオプション	説明	コールドスタンバイ構成の終了モード
1	コールドスタンバイ構成の終了	コールドスタンバイ構成の正常終了	adbstop および monend	通常の終了モードです。 実行系でadbstop コマンドが正常終了したあと，HA モニタのmonend コマンドを実行すると，HADB サーバが正常終了し，コールドスタンバイ構成が正常終了します。	正常終了

項番	終了形態		実行するコマンドおよびオプション	説明	コールドスタンバイ構成の終了モード
2		コールドスタンバイ構成の強制終了	monend	実行系で HA モニタのmonend コマンドだけを実行すると，HADB サーバが強制終了し，コールドスタンバイ構成が強制終了します。	強制終了
3		コールドスタンバイ構成の異常終了	－	運用が続行できない障害が発生した場合，処理中のトランザクションの完了を待たないで，HADB サーバをすぐに終了します。 待機系が存在しない場合は，コールドスタンバイ構成が異常終了します。	異常終了
4	系切り替えの発生	実行系の HADB サーバの正常終了	adbstop および monswap	待機系が存在する場合，実行系でadbstop コマンドが正常終了したあとに，HA モニタのmonswap コマンドを実行すると，実行系の HADB サーバが正常終了し，系が切り替わります。	－
5		実行系の HADB サーバの強制終了	monswap	待機系が存在する場合，実行系で HA モニタのmonswap コマンドだけを実行すると，実行系の HADB サーバが強制終了し，系が切り替わります。系の切り替え時に，トランザクションの回復処理が実行されます。	－
6		実行系の HADB サーバの異常終了	－	運用が続行できない障害が発生した場合，処理中のトランザクションの完了を待たないで，HADB サーバをすぐに終了します。 待機系が存在する場合，実行系が異常終了すると，系が切り替わります。系の切り替え時に，トランザクションの回復処理が実行されます。	－

（凡例）

－：該当しません。

17.5　AP の運用（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成での，AP の運用に関する注意点について説明します。AP の作成方法などについては，マニュアル『HADB　AP 開発ガイド』を参照してください。

17.5.1　HADB サーバへの接続

コールドスタンバイ構成の場合，AP はエイリアス IP アドレスを使用して，実行系の HADB サーバに接続するようにしてください。エイリアス IP アドレスの設定方法については，マニュアル『HA モニタ Linux(R)(x86)編』を参照してください。

17.5.2　実行系で障害が発生した場合

実行系で障害が発生した場合，実行中だったトランザクションはロールバックします。

また，HADB サーバに接続中のすべての AP が HADB サーバから切断されます。トランザクションを再実行する際は，改めて HADB サーバに接続してください。

17.6　データベースのバックアップ（コールドスタンバイ構成の場合）

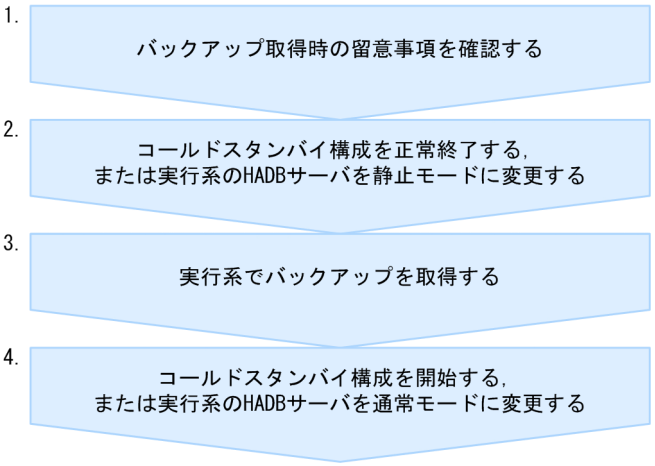
ここでは，コールドスタンバイ構成の場合のバックアップの取得方法，およびバックアップからのデータベースの回復方法について説明します。

コールドスタンバイ構成ではない場合のバックアップの取得方法，およびバックアップからのデータベースの回復方法については，「10.3　データベースのバックアップ」を参照してください。

17.6.1　バックアップの取得方法

データベースのバックアップを取得する手順を次に示します。

図 17-2　バックアップを取得する手順（コールドスタンバイ構成を使用している場合）



各手順の詳細を以降で説明します。

(1)　バックアップ取得時の留意事項を確認する

「10.3.1　バックアップの取得方法」を参照して，バックアップ取得時の留意事項を確認してください。

(2)　コールドスタンバイ構成を正常終了する，または実行系の HADB サーバを静止モードに変更する

次のどちらかの操作を実行してください。

- adbstop コマンドおよび HA モニタのmonend コマンドを実行して，コールドスタンバイ構成を正常終了する
- 実行系でadbchgsrvmode コマンドを実行して，実行系の HADB サーバを静止モードに変更する

上記のどちらかの操作を実行しないと，データベースを回復する際にデータベースの整合性が取れなくなるおそれがあります。

HADB サーバの状態を確認する場合は，実行系の HADB サーバでadbls -d srv コマンドを実行してください。

(3)　実行系でバックアップを取得する

実行系でバックアップを取得します。バックアップを取得するファイルの一覧については，「10.3.1　バックアップの取得方法」の「(2)　バックアップを取得する手順」を参照してください。

ファイルがシンボリックリンクの場合は，リンク先のファイル，またはブロックスペシャルファイルのコピーが必要です。また，バックアップファイルはスパースファイルにしないでください。

なお，コールドスタンバイ構成を正常終了している場合は，コピーを実行する前に，DB ディレクトリが$DBDIR にマウントされていることを確認してください。マウントされていない場合は，OS のmount コマンドを実行して，DB ディレクトリを$DBDIR にマウントしてください。

(4)　コールドスタンバイ構成を開始する，または実行系の HADB サーバを通常モードに変更する

次のどちらかの操作を実行してください。

- バックアップの取得前に，adbstop コマンドおよび HA モニタのmonend コマンドでコールドスタンバイ構成を正常終了した場合

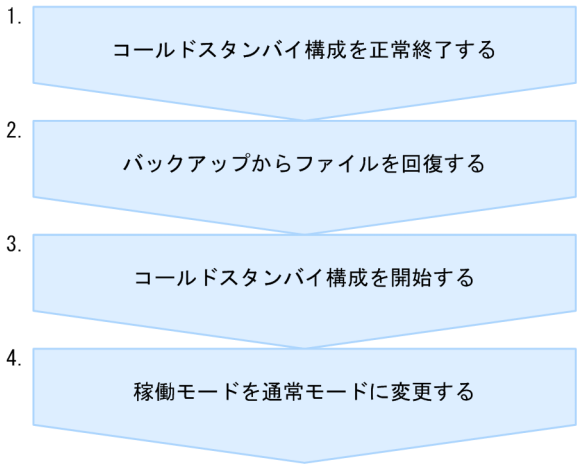
  HA モニタのmonbegin コマンドを実行して，コールドスタンバイ構成を開始してください。
- バックアップの取得前に，adbchgsrvmode コマンドで実行系の HADB サーバを静止モードにした場合

  実行系でadbchgsrvmode コマンドを実行して，実行系の HADB サーバを通常モードに変更してください。

17.6.2　バックアップからの回復方法

バックアップからの回復手順を次に示します。

図 17-3　バックアップからの回復手順（コールドスタンバイ構成の場合）



なお，バックアップからの回復を行う際は，「10.3.2　バックアップからの回復方法」の内容も参照してください。

各手順の詳細を以降で説明します。

(1)　コールドスタンバイ構成を正常終了する

adbstop コマンドおよび HA モニタのmonend コマンドを実行して，コールドスタンバイ構成を正常終了してください。

コールドスタンバイ構成を正常終了しないで，データベースの回復を実行した場合，データベースの内容が不整合となるおそれがあります。

(2)　バックアップからファイルを回復する

「17.6.1　バックアップの取得方法」の「(3)　実行系でバックアップを取得する」で取得したバックアップからファイルを回復してください。

なお，バックアップからファイルを回復する前に，DB ディレクトリが$DBDIR にマウントされていることを確認してください。マウントされていない場合は，OS のmount コマンドを実行して，DB ディレクトリを$DBDIR にマウントしてください。

(3)　コールドスタンバイ構成を開始する

HA モニタのmonbegin コマンドを実行して，コールドスタンバイ構成を開始してください。

(4)　稼働モードを通常モードに変更する

静止モードでバックアップを取得した場合，データベースを回復したあとに HADB サーバを開始すると，HADB サーバが静止モードのままとなっています。この場合，実行系でadbchgsrvmode コマンドを実行して，実行系の HADB サーバの稼働モードを通常モードに変更してください。

17.6.3　バックアップの運用例（OS コマンドを使用）

ここでは，OS のコマンドを使用したバックアップの運用例（フルバックアップを取得する運用例）について説明します。

(1)　システムの構成例

「図 17-1　コールドスタンバイ構成でのシステム構成例」で示すコールドスタンバイ構成で，実行系はhadb01 とします。DB ディレクトリ，アーカイブディレクトリ，同義語辞書ファイルの格納ディレクトリ，および監査証跡の出力先ディレクトリは，次の構成となっているとします。

■hadb01 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
 ├ADBDIC：ブロックスペシャルファイル（サイズは10MB）
 ├ADBMST：ブロックスペシャルファイル（サイズは512MB）
 ├ADBSTBL：ブロックスペシャルファイル（サイズは512MB）
 ├ADBWORK：ファイルシステム上のディレクトリ
 ├ADBWRK：ブロックスペシャルファイル（サイズは2GB）
 ├ADBUTBL01：ブロックスペシャルファイル（サイズは4GB）
 ├ADBUIDX01：ブロックスペシャルファイル（サイズは2GB）
 ├SPOOL：ファイルシステム上のディレクトリ
 └ADBSYS：ファイルシステム上のディレクトリ
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリの構成

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索をする場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/mnt/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用する場合に作成するディレクトリです。

■hadb02 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリのマウントポイント

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索をする場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリのマウントポイント

```
/mnt/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用する場合に作成するディレクトリです。

(2) バックアップの取得

OS のコマンドを使用して，バックアップを取得する手順を説明します。

手順

1. コールドスタンバイ構成を正常終了する※1

   詳細については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

2. OS のコマンドでバックアップを取得する

   OS のcp コマンドおよびdd コマンドをhadb01（実行系）で実行し，DB ディレクトリとアーカイブディレクトリのバックアップを取得します。

   - DB ディレクトリのバックアップの取得

     ```
     cp -r /HADB/db /HADB_bkup/db/dbdir
     dd if=/HADB/db/ADBMST of=/HADB_bkup/db/ADBMST bs=524288
     dd if=/HADB/db/ADBDIC of=/HADB_bkup/db/ADBDIC bs=524288
     dd if=/HADB/db/ADBSTBL of=/HADB_bkup/db/ADBSTBL bs=524288
     dd if=/HADB/db/ADBUTBL01 of=/HADB_bkup/db/ADBUTBL01 bs=524288
     dd if=/HADB/db/ADBUIDX01 of=/HADB_bkup/db/ADBUIDX01 bs=524288
     ```

     バックアップの取得先ディレクトリは/HADB_bkup/db です。

     DB ディレクトリのバックアップにある DB エリアファイル（シンボリックリンク）と，このあとで取得する DB エリアファイルのバックアップが同じパスにならないようにしてください。

     DB ディレクトリが$DBDIR にマウントされていない場合は，OS のmount コマンドを実行してマウントしてください。

     ブロックスペシャルファイルのデータをバックアップする場合，ボリューム全体をコピーするため，実際の使用量よりもバックアップ容量が増えます。

   - アーカイブディレクトリのバックアップの取得

     ```
     cp -r /HADB/archive /HADB_bkup/archive
     ```

     アーカイブマルチチャンク表を定義している場合は，アーカイブディレクトリのバックアップを取得する必要があります。

     アーカイブディレクトリを NFS サーバ上に作成している場合は，アーカイブディレクトリのバックアップをhadb02（待機系）で取得する必要はありません。

   - 同義語辞書ファイルの格納ディレクトリのバックアップの取得

     ```
     cp -r /mnt/syndict /HADB_bkup/syndict
     ```

     同義語検索をしている場合は，同義語辞書ファイルの格納ディレクトリのバックアップを取得する必要があります。

> **メモ**
>
> 監査証跡の出力先ディレクトリ下の監査証跡ファイルは，監査証跡の保存先ディレクトリに移動する運用のため，監査証跡の出力先ディレクトリのバックアップを取得する必要はありません。

3. コールドスタンバイ構成を開始する※2

   詳細については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

注※1

手順 1.は，次の方法で代替することもできます。

hadb01（実行系）でadbchgsrvmode コマンドを実行し，実行系の HADB サーバの稼働モードを静止モードに変更します。

```
adbchgsrvmode --quiescence
```

注※2

手順 1.で注※1 の方法を実施した場合は，hadb01（実行系）でadbchgsrvmode コマンドを実行し，実行系の HADB サーバの稼働モードを通常モードに変更してください。

```
adbchgsrvmode --normal
```

(3)　バックアップからの回復

バックアップからの回復手順を説明します。

手順

1. コールドスタンバイ構成を正常終了する

   詳細については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

2. OS のコマンドでデータベースを回復する

   OS のcp コマンドおよびdd コマンドをhadb01（実行系）で実行し，バックアップからデータベースを回復します。

   - DB ディレクトリの回復

```
cp -r /HADB_bkup/db/dbdir/* /HADB/db
dd if=/HADB_bkup/db/ADBMST of=/HADB/db/ADBMST bs=524288
dd if=/HADB_bkup/db/ADBDIC of=/HADB/db/ADBDIC bs=524288
dd if=/HADB_bkup/db/ADBSTBL of=/HADB/db/ADBSTBL bs=524288
dd if=/HADB_bkup/db/ADBUTBL01 of=/HADB/db/ADBUTBL01 bs=524288
dd if=/HADB_bkup/db/ADBUIDX01 of=/HADB/db/ADBUIDX01 bs=524288
```

   バックアップファイルの格納ディレクトリは，/HADB_bkup/db です。

   DB ディレクトリが$DBDIR にマウントされていない場合は，OS のmount コマンドを実行してマウントしてください。

- アーカイブディレクトリの回復

```
rm -r /HADB/archive/*
cp -r /HADB_bkup/archive/* /HADB/archive
```

アーカイブマルチチャンク表を定義している場合は，アーカイブディレクトリを回復する必要があります。

アーカイブディレクトリを NFS サーバ上に作成している場合は，hadb02（待機系）でアーカイブディレクトリを回復する必要はありません。

- 同義語辞書ファイルの格納ディレクトリの回復

```
rm -r /mnt/syndict/*
cp -r /HADB_bkup/syndict/* /mnt/syndict
```

同義語検索をしている場合は，同義語辞書ファイルの格納ディレクトリを回復する必要があります。

3. コールドスタンバイ構成を開始する

詳細については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

コールドスタンバイ構成の開始後，実行系の HADB サーバの稼働モードが静止モードになっている場合は，hadb01（実行系）でadbchgsrvmode コマンドを実行し，実行系の HADB サーバの稼働モードを通常モードに変更してください。

```
adbchgsrvmode --normal
```

17.6.4　バックアップの運用例（ShadowImage を使用）

ここでは，ShadowImage を使用したバックアップの運用例について説明します。

(1)　システムの構成例

「図 17-1　コールドスタンバイ構成でのシステム構成例」で示すコールドスタンバイ構成で，実行系は hadb01 とします。DB ディレクトリ，アーカイブディレクトリ，および同義語辞書ファイルの格納ディレクトリは，次の構成となっているとします。

■hadb01 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db（FS001上に構築）
 ├ADBDIC：ブロックスペシャルファイル（LU002を使用）
 ├ADBMST：ブロックスペシャルファイル（LU001を使用）
 ├ADBSTBL：ブロックスペシャルファイル（LU003を使用）
 ├ADBWORK：ファイルシステム上のディレクトリ
 ├ADBWRK：ブロックスペシャルファイル（WRK003を使用）
 ├ADBUTBL01：ブロックスペシャルファイル（LU004を使用）
 ├ADBUIDX01：ブロックスペシャルファイル（LU005を使用）
 ├SPOOL：ファイルシステム上のディレクトリ
 └ADBSYS：ファイルシステム上のディレクトリ
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリの構成

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索をする場合に作成するディレクトリです。

- 監査証跡の出力先ディレクトリの構成

```
/mnt/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用する場合に作成するディレクトリです。

■hadb02 のディレクトリ構成

- DB ディレクトリの構成

```
/HADB/db
```

- アーカイブディレクトリの構成

```
/HADB/archive
```

アーカイブディレクトリは，アーカイブマルチチャンク表を定義した場合に作成するディレクトリです。

- 同義語辞書ファイルの格納ディレクトリのマウントポイント

```
/mnt/syndict
```

同義語辞書ファイルの格納ディレクトリは，同義語検索をする場合に作成するディレクトリです。

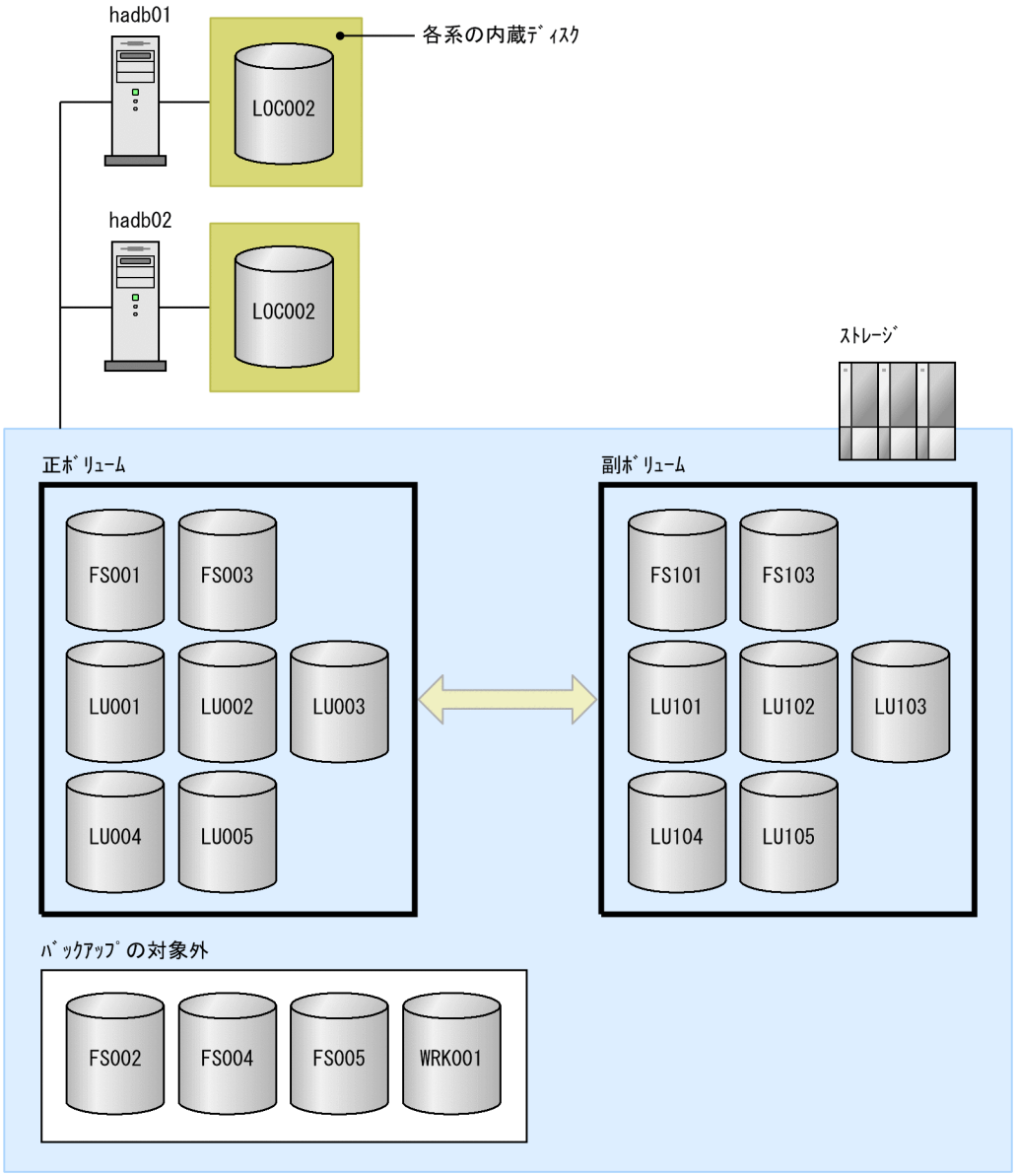
- 監査証跡の出力先ディレクトリのマウントポイント

```
/mnt/audit
```

監査証跡の出力先ディレクトリは，監査証跡機能を使用する場合に作成するディレクトリです。

■Logical Unit の構成

コールドスタンバイ構成のときのストレージの構成例を次の図に示します。



[説明]

- FS001 は DB ディレクトリ用のファイルシステムです。
- FS002 は作業用一時ファイルを格納しているファイルシステムです。
- FS003 は同義語辞書ファイルを格納しているファイルシステムです。
- FS004 は監査証跡ファイルの出力先のファイルシステムです。
- FS005 は次のファイルシステムです。
  - データインポートで使用する入力データファイルを格納するファイルシステム
  - ADB_CSVREAD 関数で使用する CSV ファイルを格納するファイルシステム
  - 監査証跡の保存先ディレクトリを作成するファイルシステム
- LU001～LU005 は，DB エリアファイルに使用しています。
- WRK001 は作業用表 DB エリアファイルに使用しています。

- 正ボリュームFS00X，LU00X に対応する副ボリュームはFS10X，LU10X です。

(2)　バックアップの取得

ShadowImage を使用して，バックアップを取得する手順を説明します。

手順

1. コールドスタンバイ構成を正常終了する

   詳細については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

2. バックアップ対象のファイルシステムをアンマウントする

3. バックアップを取得する

   FS001，FS003，およびLU001～LU005 のすべての Logical Unit に対して，「10.3.4　バックアップの運用例（ShadowImage を使用）」の「(3)　バックアップの取得」の手順に従ってバックアップを取得してください。

4. コールドスタンバイ構成を開始する

   詳細については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

(3)　バックアップからの回復

バックアップからの回復手順を説明します。

手順

1. コールドスタンバイ構成を終了する

   ```
   adbstop
   monend
   ```

   hadb01（実行系）でadbstop コマンド，および HA モニタのmonend コマンドを実行して，コールドスタンバイ構成を終了します。

   なお，このあとの手順でファイルシステムをアンマウントするため，静止モードではバックアップを取得できません。

2. バックアップ対象のファイルシステムをアンマウントする

3. データベースを回復する

   FS001，FS003，およびLU001～LU005 のすべての Logical Unit を，「10.3.4　バックアップの運用例（ShadowImage を使用）」の「(4)　バックアップからの回復」の手順に従って回復してください。

4. コールドスタンバイ構成を開始する

   詳細については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

17.7　状態監視（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成特有のシステムの監視方法を説明します。

17.7.1　メモリの使用状況を確認する場合

HADB サーバが使用しているメモリの使用状況を確認したい場合は，adbstat コマンドを実行してください。確認方法については，「10.6.1　全メモリの使用状況を確認したい場合」を参照してください。

adbstat コマンドを実行すると，実行系の HADB サーバの全メモリの最大使用量（Total_memory_max_size）を確認できます。

17.8　DB エリアの運用（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成の場合の，DB エリアの運用について説明します。

17.8.1　データ用 DB エリアの追加・削除・拡張（コールドスタンバイ構成の場合）

実行系と待機系で同じ Logical Unit を指定する必要があるため，WWN を使用してください。

■共有ディスクの SCSI リザーブを使用している場合

共有ディスクの SCSI リザーブを使用している場合は，HA モニタのservers ファイル中のオペランドの指定を変更する必要があります。手順を次に示します。

- **手順（データ用 DB エリアを追加または拡張する場合）**

1. コールドスタンバイ構成を正常終了する

   終了方法については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

2. HA モニタのservers ファイルの指定を変更する

   scsi_device オペランドまたはdmmp_device オペランドに，追加するディスクのデバイス名を追加してください。実行系と待機系の両方のservers ファイルの指定を変更してください。

3. コールドスタンバイ構成を正常開始する

   開始方法については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

4. adbmodarea コマンドを実行して，データ用 DB エリアを追加または拡張する

   データ用 DB エリアを追加する方法については，「11.10.1　データ用 DB エリアを追加する方法」を参照してください。

   データ用 DB エリアを拡張する方法については，「11.10.3　データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加）」を参照してください。

- **手順（データ用 DB エリアを削除する場合）**

1. adbmodarea コマンドを実行して，データ用 DB エリアを削除する

   データ用 DB エリアを削除する方法については，「11.10.2　データ用 DB エリアを削除する方法」を参照してください。

2. コールドスタンバイ構成を正常終了する

   終了方法については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

3. HA モニタのservers ファイルの指定を変更する

   scsi_device オペランドまたはdmmp_device オペランドから，削除したディスクのデバイス名を削除してください。実行系と待機系の両方のservers ファイルの指定を変更してください。

4. コールドスタンバイ構成を正常開始する

   開始方法については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

17.8.2　作業表用 DB エリアの運用（コールドスタンバイ構成の場合）

実行系と待機系で同じ Logical Unit を指定しても問題ありません。

17.9　表およびインデクスの運用（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成の場合の，表およびインデクスの運用について説明します。

17.9.1　実表の運用（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成の場合の，実表の運用について説明します。

コールドスタンバイ構成の場合，実表の運用のうち，次の運用項目ではコールドスタンバイ構成特有の運用が必要となります。

- 実表の更新不可状態を解除する方法
- アーカイブマルチチャンク表を使用する際の運用

上記以外については，コールドスタンバイ構成ではない場合と同じです。「11.1　実表の運用」を参照してください。

(1)　実表の更新不可状態を解除する方法（コールドスタンバイ構成の場合）

adbimport コマンド，adbidxrebuild コマンド，またはadbunarchivechunk コマンドの実行中に系切り替えが発生した場合，処理対象の実表は更新不可状態となります。この場合，新しく実行系となった系でコマンドを再実行してください。

なお，次に示すファイルシステムが，両方の系から同一のパスで参照できる場合は，コマンドを再実行する際に，中断した処理の続きから実行できます。

- DB ディレクトリを格納したファイルシステム
- データインポートで使用する入力データファイルを格納するファイルシステム（adbimport コマンドの場合）
- 作業用一時ファイルを格納するファイルシステム

実表の更新不可状態については，「15.8.1　実表が更新不可状態になった場合」を参照してください。

(2)　アーカイブマルチチャンク表を使用する際の運用（コールドスタンバイ構成の場合）

実行系と待機系の両方から同一の絶対パスでアーカイブディレクトリを参照できるようにしてください。

例えば，分散ファイルシステムの一つである NFS を使用する場合は，次の手順でアーカイブディレクトリを準備します。

手順

1. NFS サーバを用意し，NFS サーバ上でアーカイブディレクトリを作成してください。

2. 手順の 1.で作成したアーカイブディレクトリをエクスポートしてください。

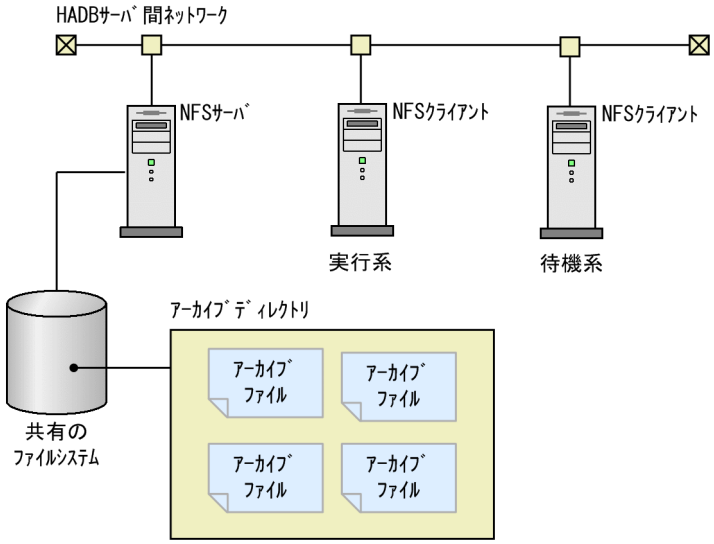
   この操作は，NFS サーバで実行します。

3. 手順の 2.でエクスポートしたディレクトリを，実行系と待機系で同じパスとなるようにマウントしてください。

   この操作は，実行系と待機系の両方で実行します。

NFS を使用したシステム構成例を次の図に示します。

図 17-4　NFS を使用したシステム構成例



17.9.2　インデクスの運用（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成の場合の，B-tree インデクス，テキストインデクスおよびレンジインデクスの運用について説明します。

コールドスタンバイ構成の場合，B-tree インデクス，テキストインデクスおよびレンジインデクスの運用のうち，次の運用項目ではコールドスタンバイ構成特有の運用が必要となります。

- B-tree インデクスの未完状態を解除する方法
- テキストインデクスの未完状態を解除する方法
- レンジインデクスの未完状態を解除する方法

上記以外については，コールドスタンバイ構成ではない場合と同じです。「11.3　インデクスの運用」を参照してください。

(1) B-tree インデクスの未完状態を解除する方法（コールドスタンバイ構成の場合）

adbimport コマンドまたはadbidxrebuild コマンドの実行中に系切り替えが発生した場合，B-tree インデクスは未完状態となります。この場合，新しく実行系となった系でコマンドを再実行してください。

なお，次に示すファイルシステムが，両方の系から同一のパスで参照できる場合は，コマンドを再実行する際に，中断した処理の続きから実行できます。

- DB ディレクトリを格納したファイルシステム
- データインポートで使用する入力データファイルを格納するファイルシステム（adbimport コマンドの場合）
- 作業用一時ファイルを格納するファイルシステム

B-tree インデクスの未完状態については，「15.9.1　B-tree インデクスが未完状態になったときの対処方法」を参照してください。

(2) テキストインデクスの未完状態を解除する方法（コールドスタンバイ構成の場合）

adbimport コマンドまたはadbidxrebuild コマンドの実行中に系切り替えが発生した場合，テキストインデクスは未完状態となります。この場合，新しく実行系となった系でコマンドを再実行してください。

なお，次に示すファイルシステムが，両方の系から同一のパスで参照できる場合は，コマンドを再実行する際に，中断した処理の続きから実行できます。

- DB ディレクトリを格納したファイルシステム
- データインポートで使用する入力データファイルを格納するファイルシステム（adbimport コマンドの場合）
- 作業用一時ファイルを格納するファイルシステム

テキストインデクスの未完状態については，「15.10.1　テキストインデクスが未完状態になったときの対処方法」を参照してください。

(3) レンジインデクスの未完状態を解除する方法（コールドスタンバイ構成の場合）

adbidxrebuild コマンドの実行中に系切り替えが発生した場合，レンジインデクスは未完状態となります。この場合，新しく実行系となった系でコマンドを再実行してください。

なお，次に示すファイルシステムが，両方の系から同一のパスで参照できる場合は，コマンドを再実行する際に，中断した処理の続きから実行できます。

- DB ディレクトリを格納したファイルシステム
- 作業用一時ファイルを格納するファイルシステム

レンジインデクスの未完状態については，「15.11.1　レンジインデクスが未完状態になったときの対処方法」を参照してください。

17.10　DB ディレクトリの運用（コールドスタンバイ構成の場合）

DB ディレクトリは，コールドスタンバイ構成を開始するときに，HA モニタがマウントポイントにマウントします。

adbinit コマンドを実行するときは，DB ディレクトリを手動でマウントしてください。

17.11　統計情報の運用（コールドスタンバイ構成の場合）

コールドスタンバイ構成の場合，待機系の HADB サーバの統計情報は参照できません。

統計情報の確認方法については，「10.10　統計解析の運用（HADB サーバの稼働情報の確認）」を参照してください。

17.12　障害が発生した場合の運用

ここでは，障害が発生した場合の運用について説明します。

17.12.1　実行系で障害が発生した場合

実行系で障害が発生して HADB サーバが異常終了した場合，HA モニタのservers ファイルの servexec_retry オペランドに指定した回数分，HADB サーバを再開始しようとします。再開始に失敗した場合，系が切り替わります。

障害の原因を特定するために，「14.1　障害発生時の対処の流れ」を参照して対処してください。そこで説明している手順は，障害が発生した系で実施してください。

また，障害が発生した系で，次のトラブルシュート情報を取得してください。

- HADB サーバのトラブルシュート情報をadbinfoget コマンドで取得してください。
- HA モニタのトラブルシュート情報を HA モニタのmonts コマンドで取得してください。

なお，HADB サーバを再開始した場合，「17.3.2　HADB サーバのインストールおよび環境設定」で作成したトラブルシュート情報ファイルを格納するディレクトリに，トラブルシュート情報ファイルが格納されます。格納されたトラブルシュート情報ファイルは自動では削除されないため，使用しないトラブルシュート情報ファイルを定期的に削除してください。

17.12.2　待機系で障害が発生した場合

待機系で障害が発生した場合，実行系は運用を継続します。

障害の原因を特定するために，「14.1　障害発生時の対処の流れ」を参照して対処してください。そこで説明している手順は，障害が発生した系で実施してください。

また，障害が発生した系で，次のトラブルシュート情報を取得してください。

- HADB サーバのトラブルシュート情報をadbinfoget コマンドで取得してください。
- HA モニタのトラブルシュート情報を HA モニタのmonts コマンドで取得してください。

17.12.3　実行系と待機系の間で通信障害が発生した場合

実行系と待機系の間で通信障害が発生した場合，通信障害の発生元のサーバマシンは，HA モニタによって電源断の停止状態になります。

停止したサーバマシンを再起動したあと，通信障害の原因を取り除いてください。そのあと，該当する系をコールドスタンバイ構成に復帰させてください。復帰の方法については，「17.12.4　コールドスタンバイ構成への系復帰」を参照してください。

17.12.4　コールドスタンバイ構成への系復帰

障害が発生して系切り替えが発生した場合，障害が発生した系は停止状態になります。障害の原因を取り除いたあとに，HA モニタのmonbegin コマンドを実行すると，待機系としてコールドスタンバイ構成に復帰できます。

17.13　コールドスタンバイ構成での計画系切り替え

計画系切り替えをする場合は，実行系でadbstop コマンドを実行したあとに HA モニタのmonswap コマンドを実行してください。この場合，実行系の HADB サーバが正常終了したあとに，系が切り替わります。

なお，実行系で HA モニタのmonswap コマンドだけを実行した場合は，実行系の HADB サーバを強制終了したあとに系が切り替わります。

17.14　トラブルシュート（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成でトラブルが発生した場合の対処方法について説明します。

17.14.1　コールドスタンバイ構成の開始または終了に関するトラブル

コールドスタンバイ構成の開始または終了に関するトラブルが発生したときの対処方法について説明します。

(1)　コールドスタンバイ構成を正常開始できない場合

コールドスタンバイ構成が正常開始できない場合の考えられる原因と対処方法を次の表に示します。

表 17-6　コールドスタンバイ構成を正常開始できない場合の考えられる原因と対処方法

項番	考えられる原因	対処方法
1	実行系または待機系で障害が発生しています。	HA モニタのmonbegin コマンドではなく，adbstart コマンドで HADB サーバを開始してください。ただし，この場合，コールドスタンバイ構成にはなりません。 または，代替マシンがある場合は，代替マシンの環境を，障害が発生したサーバマシンとまったく同じ環境に設定したあとに，「17.4.1　コールドスタンバイ構成の開始方法」の手順に従って，コールドスタンバイ構成を開始してください。
2	DB ディレクトリがマウントされていません。	HA モニタのservers ファイルに指定した，次に示すオペランドの指定を見直してください。 • disk オペランド • fs_name オペランド • fs_mount_dir オペランド • fs_mount_opt オペランド
3	DB ディレクトリが正しく初期化されていません。	adbinit コマンドを実行するときに，DB ディレクトリをマウントしていたかどうかを確認してください。

上記以外の場合は，出力されたメッセージの対処方法に従って原因を取り除いてください。そのあとに，「17.4.1　コールドスタンバイ構成の開始方法」の手順に従って，コールドスタンバイ構成を開始してください。

(2)　コールドスタンバイ構成を再開始できない場合

実行系または待機系で障害が発生している場合は，障害が発生した系をコールドスタンバイ構成から除いてください。そのあとで，「17.4.1　コールドスタンバイ構成の開始方法」の手順に従って，コールドスタンバイ構成を再開始してください。

それ以外の場合は，出力されたメッセージの対処方法に従って原因を取り除いてください。そのあとに，「17.4.1　コールドスタンバイ構成の開始方法」の手順に従って，コールドスタンバイ構成を再開始してください。

(3)　コールドスタンバイ構成を終了できない場合

出力されたメッセージの対処方法に従って原因を取り除いてください。そのあとに，「17.4.2　コールドスタンバイ構成の終了方法」の手順に従って，コールドスタンバイ構成を終了してください。

(4)　系切り替え後に HADB サーバが開始できない場合

系切り替え後に HADB サーバが開始できない場合の考えられる原因と対処方法を次の表に示します。

表 17-7　系切り替え後に HADB サーバが開始できない場合の考えられる原因と対処方法

項番	考えられる原因	対処方法
1	DB ディレクトリがマウントされていません。	HA モニタのservers ファイルに指定した，次に示すオペランドの指定を見直してください。 ・disk オペランド ・fs_name オペランド ・fs_mount_dir オペランド ・fs_mount_opt オペランド
2	HADB サーバのサーバ定義が，実行系と待機系で異なっています。	実行系と待機系の両方の HADB サーバで，サーバ定義を同じにしてください。 実行系の HADB サーバが強制終了または異常終了した場合，系切り替え時の HADB サーバの開始モードは，再開始になります。このとき，開始する HADB サーバのスレッド数やメモリが，系切り替え前の実行系の HADB サーバよりも少なく設定されていると，再開始に失敗します。
3	DB エリアの拡張時に，待機系では参照できないブロックスペシャルファイルを指定しています。	次に示す手順に従って対処してください。 1. OS のdd コマンドなどを使用して，実行系と待機系から参照できるブロックスペシャルファイルに，DB エリアファイルの内容をコピーしてください。 2. DB ディレクトリ内の DB エリアファイル（ブロックスペシャルファイル）へのシンボリックリンクを，手順 1. で使用したブロックスペシャルファイルへのリンクに変更してください。
4	DB エリアの拡張時に，待機系とパスが異なるブロックスペシャルファイルを指定しています。	実行系と待機系で同じパスになるように，WWN を使用することを推奨します。 DB ディレクトリ内の DB エリアファイルはブロックスペシャルファイルへのシンボリックリンクになります。WWN を使用したパスになるように，手動でリンクを変更してください。

上記以外の場合は，出力されたメッセージの対処方法に従って原因を取り除いてください。そのあとに，「17.4.1　コールドスタンバイ構成の開始方法」の手順に従って，コールドスタンバイ構成を再開始してください。

17.14.2　AP に関するトラブル（コールドスタンバイ構成の場合）

AP に関するトラブルが発生した場合，最初に「15.1　AP に関するトラブル」を参照して対処してください。対処できない場合に，ここの説明をお読みください。

(1)　AP が実行できない場合の対処方法

AP が実行できない場合の考えられる原因と対処方法を次の表に示します。

表 17-8　AP が実行できない場合の考えられる原因と対処方法

項番	考えられる原因	対処方法
1	接続先のホスト名が誤っている可能性があります。	HADB クライアントと HADB サーバ間の通信で使用しているエイリアス IP アドレスに対応するホスト名を指定しているか，確認してください。 クライアント定義など，AP の接続先のホスト名を指定している個所を確認して修正してください。

17.14.3　コマンドに関するトラブル（コールドスタンバイ構成の場合）

コマンドに関するトラブルが発生した場合，最初に「15.2　コマンドに関するトラブル」を参照して対処してください。対処できない場合に，ここの説明をお読みください。

(1)　コマンドが実行できない場合の対処方法

次に示すコマンドが実行できない場合の，考えられる原因と対処方法を次の表に示します。

- adbimport
- adbidxrebuild
- adbmergechunk
- adbunarchivechunk
- adbreorgsystemdata
- adbsyndict

表 17-9　コマンドが実行できない場合の考えられる原因と対処方法

項番	考えられる原因	対処方法
1	データインポートで使用する入力データファイルを格納するファイルシステムがマウントされていません。	共有するファイルシステムの設定を見直してください。 実行系と待機系で同じパスになるように，ファイルシステムを共有してください。
2	次のファイルシステムがマウントされていません。 • 作業用一時ファイルを格納するファイルシステム • 同義語辞書ファイルを格納するファイルシステム	HA モニタのservers ファイルに指定した，次に示すオペランドの指定を見直してください。 • disk オペランド • fs_name オペランド • fs_mount_dir オペランド • fs_mount_opt オペランド 指定に誤りがあった場合は，手動で正しくマウントしたあと，コマンドを再実行してください。
3	実行系と待機系でファイルシステムのマウントポイントのパスが異なっています。	HA モニタのservers ファイルに指定した，次に示すオペランドの指定を見直してください。 • disk オペランド • fs_name オペランド • fs_mount_dir オペランド • fs_mount_opt オペランド 指定に誤りがあった場合は，手動で正しくマウントしたあと，コマンドを再実行してください。 なお，コマンド実行中に系切り替えが発生した場合，次に示すファイルシステムを実行系と待機系で同じパスで参照できれば，中断した処理の続きからコマンドを再実行できます。 • DB ディレクトリを格納するファイルシステム • データインポートで使用する入力データファイルを格納するファイルシステム • 作業用一時ファイルを格納するファイルシステム • 同義語辞書ファイルを格納するファイルシステム

17.14.4　DB ディレクトリのファイルに関するトラブル（コールドスタンバイ構成の場合）

(1)　システムログファイルに障害が発生した場合の対処方法

出力されたメッセージの対処に従ってください。

また，DB ディレクトリのファイルシステムが$DBDIR にマウントされているかどうかを確かめてください。

「15.5.2　システムログファイルでトラブルが発生した場合の対処」もあわせて参照してください。

17.15　サーバマシンの OS のホスト名・IP アドレスの変更（コールドスタンバイ構成の場合）

ここでは，コールドスタンバイ構成の場合に，HADB サーバがインストールされているサーバマシンの OS のホスト名または IP アドレスを変更する場合の手順について説明します。

なお，実行系および待機系の IP アドレスだけを変更する場合は，HADB サーバと HA モニタに対して実施する作業はありません。

■コールドスタンバイ構成の HADB サーバで実施する作業

サーバマシンの OS のホスト名または IP アドレスを変更する場合に，コールドスタンバイ構成の HADB サーバで実施する作業を次に示します。

1. コールドスタンバイ構成を正常終了する

   コールドスタンバイ構成を正常終了してください。終了方法については，「17.4.2　コールドスタンバイ構成の終了方法」の「(1)　コールドスタンバイ構成の終了手順」の「(a)　コールドスタンバイ構成を正常終了させる場合」を参照してください。

   また，すべての系の HA モニタについても，monstop コマンドで正常停止してください。

2. すべての系の HA モニタについて設定を変更する

   「17.3.4　HA モニタの環境設定」を参照して，すべての系の HA モニタについて設定を変更してください。

   - HA モニタの監視パスとして使用する LAN のホスト名を変更する場合

     HA モニタの環境設定ファイル（sysdef ファイル）の設定を変更してください。

   - エイリアス IP アドレスを変更する場合

     LAN 接続時の状態設定ファイル（*サーバ識別名*.up ファイル），および LAN 切り離し時の状態設定ファイル（*サーバ識別名*.down ファイル）の設定を変更してください。

3. コールドスタンバイ構成を開始する

   すべての系の HA モニタをmonstart コマンドで起動してください。そのあと，コールドスタンバイ構成を開始してください。開始方法については，「17.4.1　コールドスタンバイ構成の開始方法」の「(1)　コールドスタンバイ構成の開始手順」を参照してください。

■HADB クライアントで実施する作業

サーバマシンの OS のホスト名または IP アドレスを変更する場合に，HADB クライアントで実施する作業を次に示します。HADB クライアントと HADB サーバ間の通信で使用するエイリアス IP アドレスを変更する場合に実施してください。

1. AP を停止する

   コールドスタンバイ構成の HADB サーバに接続する AP をすべて停止してください。

2. クライアント定義に指定しているエイリアス IP アドレスを変更する

HADB クライアントと HADB サーバ間の通信で使用するエイリアス IP アドレスを変更する場合は，「17.3.6　クライアント定義の作成」を参照して，クライアント定義adb_clt_rpc_srv_host オペランドの値を変更してください。

クライアント定義adb_clt_rpc_srv_host オペランドについては，マニュアル『HADB　AP 開発ガイド』の『クライアント定義の設計』の『クライアント定義のオペランドの内容』を参照してください。

- JDBC ドライバを使用している場合

  クライアント定義adb_clt_rpc_srv_host オペランドに対応するプロパティで指定している HADB サーバのホスト名または IP アドレスを変更してください。JDBC ドライバで使用するプロパティは，システムプロパティ，ユーザプロパティ，または接続用の URL のプロパティがあります。詳細については，マニュアル『HADB　AP 開発ガイド』の『JDBC ドライバの環境設定』を参照してください。

- ODBC ドライバまたは CLI 関数を使用している場合

  クライアント定義adb_clt_rpc_srv_host オペランドに指定している HADB サーバのホスト名または IP アドレスを変更してください。詳細については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』を参照してください。

3. AP を開始する

コールドスタンバイ構成の HADB サーバに接続する AP を開始してください。

重要

コールドスタンバイ構成ではない場合は，「8.11　サーバマシンの OS のホスト名・IP アドレスの変更」を参照してください。

17.16　HADB サーバのバージョンアップ（コールドスタンバイ構成の場合）

コールドスタンバイ構成の HADB サーバをバージョンアップする場合の手順を次に示します。

手順

1. コールドスタンバイ構成を正常終了する
   コールドスタンバイ構成を正常終了する手順については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。
2. 実行系および待機系の HADB サーバをバージョンアップする
   バージョンアップの手順については，「8.6　HADB サーバのバージョンアップ」を参照してください。
3. コールドスタンバイ構成を開始する
   コールドスタンバイ構成を開始する手順については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

コールドスタンバイ構成を正常終了したあとに，HADB サーバをバージョンアップしてください。なお，系切り替えが発生したあとにコールドスタンバイ構成を正常終了した場合は，いったんコールドスタンバイ構成を正常開始してから正常終了してください。この操作をしないと，バージョンアップ後の HADB サーバの開始に失敗することがあります（KFAA40002-E メッセージが出力されます）。

コールドスタンバイ構成の HADB サーバをバージョンアップする場合は，実行系と待機系の両方の HADB サーバが正常終了している状態でバージョンアップしてください。

17.17　修正版 HADB サーバとの入れ替え（コールドスタンバイ構成の場合）

次に示すどちらかの方法で，修正版 HADB サーバに入れ替えることができます。

- コールドスタンバイ構成を正常終了して入れ替えを実施する方法
- 系ごとに入れ替えを実施する方法

17.17.1　コールドスタンバイ構成を正常終了して入れ替えを実施する方法

コールドスタンバイ構成を正常終了してから，修正版 HADB サーバに入れ替える手順を次に示します。

手順

1. コールドスタンバイ構成を正常終了する
   コールドスタンバイ構成を正常終了する手順については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。
2. 修正版 HADB サーバとの入れ替えを実行系および待機系で実施する
   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。
3. コールドスタンバイ構成を正常開始する
   コールドスタンバイ構成を正常開始する手順については，「17.4.1　コールドスタンバイ構成の開始方法」を参照してください。

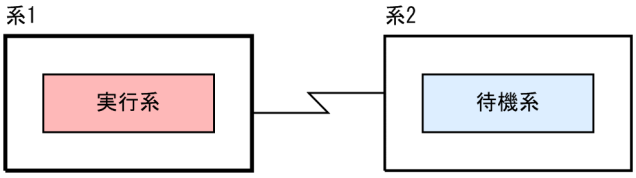
重要

- コールドスタンバイ構成を正常終了したあとに，修正版 HADB サーバとの入れ替えを実施してください。コールドスタンバイ構成で修正版 HADB サーバとの入れ替えを実施する場合は，実行系と待機系の両方の HADB サーバが正常終了している必要があります。
- 系切り替えが発生したあとにコールドスタンバイ構成を正常終了した場合は，いったんコールドスタンバイ構成を正常開始してから正常終了してください。この操作をしないと，修正版 HADB サーバとの入れ替えを実施したあとの HADB サーバの開始に失敗することがあります（KFAA40002-E メッセージが出力されます）。

17.17.2　系ごとに入れ替えを実施する方法

片方の系を停止して，系ごとに修正版 HADB サーバに入れ替える手順を次に示します。この方法の場合，実行系の HADB サーバの稼働中に，もう片方の HADB サーバを修正版に入れ替えることができるため，業務を停止する必要がありません。

なお，入れ替えを実施するコールドスタンバイ構成は，次のとおりとします。



手順

1. 系 2（待機系）を停止する

   系 2 で，HA モニタのmonsbystp コマンドを実行し，系 2 を停止してください。

2. 修正版 HADB サーバとの入れ替えを系 2 で実施する

   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。

3. 系 2 をコールドスタンバイ構成に復帰させる

   コールドスタンバイ構成への復帰方法については，「17.12.4　コールドスタンバイ構成への系復帰」を参照してください。

   系 2 は，待機系としてコールドスタンバイ構成に復帰します。

4. 計画系切り替えを実施する

   系 1 でadbstop コマンドを実行したあとに HA モニタのmonswap コマンドを実行し，計画系切り替えを実施してください。計画系切り替えが完了すると，系 2 が実行系になり，系 1 の HADB サーバが正常終了します。

   計画系切り替えについては，「17.13　コールドスタンバイ構成での計画系切り替え」を参照してください。

5. 修正版 HADB サーバとの入れ替えを系 1 で実施する

   修正版 HADB サーバとの入れ替え方法については，「8.8　修正版 HADB サーバとの入れ替え」を参照してください。

6. 系 1 をコールドスタンバイ構成に復帰させる

   コールドスタンバイ構成への復帰方法については，「17.12.4　コールドスタンバイ構成への系復帰」を参照してください。

   系 1 は，待機系としてコールドスタンバイ構成に復帰します。

7. 計画系切り替えを実施する

   系 2 でadbstop コマンドを実行したあとに HA モニタのmonswap コマンドを実行し，計画系切り替えを実施してください。計画系切り替えが完了すると，系 1 が実行系になり，系 2 の HADB サーバが正常終了します。

   計画系切り替えについては，「17.13　コールドスタンバイ構成での計画系切り替え」を参照してください。

8. 系 2 をコールドスタンバイ構成に復帰させる

コールドスタンバイ構成への復帰方法については，「17.12.4　コールドスタンバイ構成への系復帰」を参照してください。

系 2 は，待機系としてコールドスタンバイ構成に復帰します。

以上の手順で，修正版 HADB サーバとの入れ替えが完了します。なお，実行系と待機系の関係は，入れ替えを実施する前と同じになっています。

17.18　コールドスタンバイ構成で同義語検索を行う場合の運用

コールドスタンバイ構成の構築後に同義語検索を行うことになった場合，同義語辞書ファイルを格納するファイルシステムを準備し，servers ファイルの指定を変更する必要があります。各作業の内容について説明します。

17.18.1　同義語辞書ファイルを格納するファイルシステムの準備

同義語辞書ファイルを格納するファイルシステムは系切り替えの対象とします。系切り替え対象のファイルシステムは，実行系と待機系から参照できるディスク（共有ディスク）から構成される VG 内の LV 上に作成してください。共有ディスクについては，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの構成』を参照してください。

17.18.2　servers ファイルの指定の変更

HA モニタのservers ファイルに指定するオペランドの指定値を変更します。

「図 17-1　コールドスタンバイ構成でのシステム構成例」のFS003 を，同義語辞書ファイルを格納するファイルシステムとします。VG の名称はvg_hadb03 とします。LV の名称はhadb_syndict とします。

- disk オペランド

  同義語辞書ファイルを格納するファイルシステムを含む VG の絶対パスの指定を追加してください。

  disk オペランドには，/dev/vg_hadb03 を追加します。

- fs_name オペランド

  同義語辞書ファイルを格納するファイルシステムに対応する論理ボリュームの絶対パスの指定を追加してください。fs_name オペランドには，/dev/vg_hadb03/hadb_syndict を追加します。

- fs_mount_dir オペランド

  同義語辞書ファイルを格納するファイルシステムをマウントするディレクトリのパスの指定を追加してください。実行系と待機系で同じパスになるように指定してください。

- fs_mount_opt オペランド

  同義語辞書ファイルを格納するファイルシステムをマウントするためのマウントオプションを追加してください。

- vg_neck

  nouse を指定してください。

- fs_neck

  nouse を指定してください。

servers ファイルの指定例については，次の個所を参照してください。

- 系のリセットを使用している場合
  「17.3.4　HA モニタの環境設定」の「(5)　各ファイルの指定例（系のリセットを使用する場合）」の「(b)　servers ファイルの指定例」を参照してください。
- 共有ディスクの SCSI リザーブを使用している場合
  「17.3.4　HA モニタの環境設定」の「(6)　各ファイルの指定例（共有ディスクの SCSI リザーブを使用する場合）」の「(b)　servers ファイルの指定例」を参照してください。

17.18.3　同義語辞書ファイルを格納するファイルシステムの構築

同義語辞書ファイルを格納するファイルシステムを実行系で構築します。

同義語辞書ファイルを格納する LV として作成した/dev/vg_hadb03/hadb_syndict を，ext4 のファイルシステムで初期化する場合の実行例を次に示します。

```
mkfs -t ext4 /dev/vg_hadb03/hadb_syndict
```

上記のコマンドは，スーパユーザで実行してください。

17.18.4　同義語辞書ファイルを格納するファイルシステムのマウント

実行系で，同義語辞書ファイルを格納するファイルシステムをマウントしてください。そのあと，アンマウントしてください。

待機系でも同様に，同義語辞書ファイルを格納するファイルシステムをマウントし，そのあとにアンマウントしてください。

ファイルシステムをマウントする方法については，「17.3.7　データベースの作成」の「(3)　［実行系での作業］DB ディレクトリ用のファイルシステムのマウント」を参照してください。ファイルシステムをアンマウントする方法については，「17.3.7　データベースの作成」の「(5)　［実行系での作業］DB ディレクトリ用のファイルシステムのアンマウント」を参照してください。

> **重要**
>
> 実行系と待機系で同じパスとなるようにマウントポイントを作成してください。

このあとの作業については，「11.16.1　同義語検索を行うための準備をする」で説明している手順に従ってください。

重要

- サーバ定義のadb_syndict_storage_path オペランドには，マウントした同義語辞書ファイルを格納するファイルシステムのディレクトリを指定してください。
- 同義語一覧定義ファイル，辞書情報登録ファイル，および辞書情報削除ファイルの格納先は，共有のファイルシステムにしてください。共有のファイルシステムに作成しないと，両方の系からadbsyndict コマンドが実行できません。

17.19　コールドスタンバイ構成で監査証跡機能を使用する場合の運用

コールドスタンバイ構成の構築後に監査証跡機能を使用することになった場合，次のことをする必要があります。

- 監査証跡の出力先ディレクトリを作成するファイルシステムの準備
- 監査証跡の保存先ディレクトリの準備
- servers ファイルの指定の変更

上記の各作業について説明します。

なお，JP1/Audit と連携して，監査証跡を収集および一元管理する場合は，「17.19.4　JP1/Audit Management - Manager との連携（コールドスタンバイ構成の場合）」を参照してください。

17.19.1　監査証跡の出力先ディレクトリを作成するファイルシステムの準備

監査証跡の出力先ディレクトリを作成するファイルシステムは系切り替えの対象とします。系切り替え対象のファイルシステムは，実行系と待機系から参照できるディスク（共有ディスク）から構成される VG 内の LV 上に作成してください。共有ディスクについては，マニュアル『HA モニタ Linux(R)(x86)編』の『共有ディスクの構成』を参照してください。

■監査証跡の出力先ディレクトリを作成するファイルシステムの構築

監査証跡の出力先ディレクトリを作成するファイルシステムを実行系で構築します。監査証跡ファイルを格納する LV として作成した/dev/vg_hadb04/hadb_audit を，ext4 のファイルシステムで初期化する場合の実行例を次に示します。

```
mkfs -t ext4 /dev/vg_hadb04/hadb_audit
```

上記のコマンドは，スーパユーザで実行してください。

■監査証跡の出力先ディレクトリを作成するファイルシステムのマウント

実行系で，監査証跡の出力先ディレクトリを作成するファイルシステムをマウントしてください。そのあとでアンマウントしてください。待機系でも同様に，監査証跡の出力先ディレクトリを作成するファイルシステムをマウントし，そのあとでアンマウントしてください。

ファイルシステムをマウントする方法については，「17.3.7　データベースの作成」の「(3)　［実行系での作業］DB ディレクトリ用のファイルシステムのマウント」を参照してください。ファイルシステムをアンマウントする方法については，「17.3.7　データベースの作成」の「(5)　［実行系での作業］DB ディレクトリ用のファイルシステムのアンマウント」を参照してください。

重要

- 実行系と待機系で同じパスとなるようにマウントポイントを作成してください。

- サーバ定義のadb_audit_log_path オペランドには，マウントしたファイルシステムに作成したディレクトリを指定してください。

17.19.2　監査証跡の保存先ディレクトリを作成するファイルシステムの準備

監査証跡を検索する際に使用するADB_AUDITREAD 関数中に指定する監査証跡ファイルのパスが，実行系と待機系の両方から同じ絶対パスで参照できるように環境を構築してください。そのため，共有のファイルシステムに監査証跡の保存先ディレクトリを作成してください。

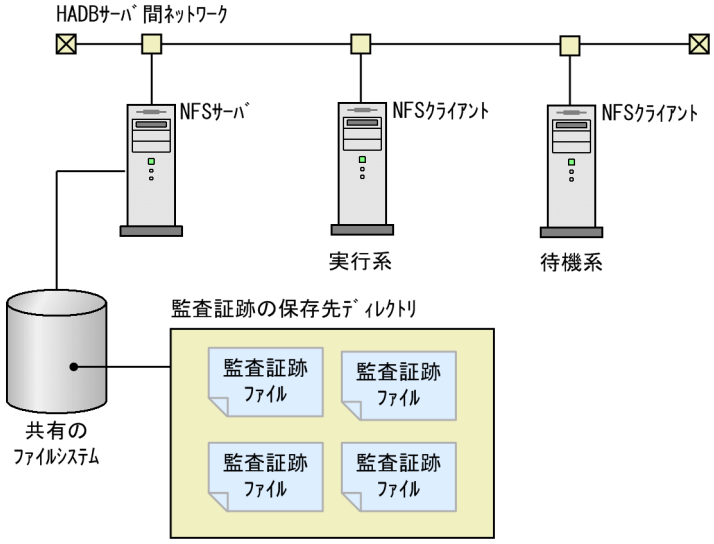
例として，分散ファイルシステムの一つである NFS を使用したときの環境設定方法を説明します。

手順

1. NFS サーバを用意し，NFS サーバ上で監査証跡の保存先ディレクトリを作成してください。
2. 手順の 1.で作成した監査証跡の保存先ディレクトリをエクスポートしてください。
   この操作は，NFS サーバで実行します。
3. 手順の 2.でエクスポートしたディレクトリを，実行系と待機系で同じパスとなるようにマウントしてください。
   この操作は，実行系と待機系の両方で実行します。

NFS を使用したシステム構成例を次の図に示します。

図 17-5　NFS を使用したシステム構成例



17.19.3　servers ファイルの指定の変更

HA モニタのservers ファイルに指定するオペランドの指定値を変更します。

「図 17-1　コールドスタンバイ構成でのシステム構成例」のFS004 を，監査証跡の出力先ディレクトリを作成するファイルシステムとします。VG の名称はvg_hadb04 とします。LV の名称はhadb_audit とします。

- disk

  監査証跡の出力先ディレクトリを作成するファイルシステムを含む VG の絶対パスの指定を追加してください。「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，disk オペランドには，/dev/vg_hadb04 を追加します。

- fs_name

  監査証跡の出力先ディレクトリを作成するファイルシステムに対応する論理ボリュームの絶対パスの指定を追加してください。「図 17-1　コールドスタンバイ構成でのシステム構成例」の場合，fs_name オペランドには，/dev/vg_hadb04/hadb_audit を追加します。

- fs_mount_dir

  監査証跡の出力先ディレクトリを作成するファイルシステムをマウントするディレクトリのパスの指定を追加してください。実行系と待機系で同じパスになるように指定してください。

- fs_mount_opt

  監査証跡の出力先ディレクトリを作成するファイルシステムをマウントするためのマウントオプションを追加してください。

- vg_neck

  use の指定を追加してください。

- fs_neck

  use の指定を追加してください。

- scsi_device

  このオペランドには，SCSI リザーブの対象となるデバイス名を絶対パスで指定します。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクが次のどれかの条件を満たすときに，監査証跡ファイルを格納するファイルシステムのデバイス名を，このオペランドに絶対パスで追加してください。

  - シングルパス構成の場合
  - VMware ESXi での仮想化環境の場合（DMMP を使用する場合を除く）
  - マルチパスソフトウェア（HDLM）による冗長化構成の場合

- dmmp_device

  このオペランドには，SCSI リザーブの対象となるデバイスを DMMP の論理デバイスの絶対パスで指定します。

  なお，系のリセットを使用する場合は，このオペランドを指定する必要はありません。

  共有ディスクの SCSI リザーブを使用する場合で，かつ共有ディスクがマルチパスソフトウェア（DMMP）による冗長化構成のときに，監査証跡ファイルを格納するファイルシステムのデバイス名を，このオペランドに絶対パスで追加してください。

servers ファイルの指定例については，「17.3.4　HA モニタの環境設定」の「(5)　各ファイルの指定例（系のリセットを使用する場合）」の「(b)　servers ファイルの指定例」を参照してください。

> **メモ**
>
> コールドスタンバイ構成を終了してから，servers ファイルのオペランドの指定値を変更してください。コールドスタンバイ構成の終了方法については，「17.4.2　コールドスタンバイ構成の終了方法」を参照してください。

17.19.4　JP1/Audit Management - Manager との連携（コールドスタンバイ構成の場合）

JP1/Audit と連携して，監査証跡を収集および一元管理する場合は，adbconvertaudittrailfile コマンドで監査証跡を変換し，統一フォーマット用監査証跡ファイルに出力してください。監査証跡の変換については，「2.18.9　監査証跡の変換（JP1/Audit Management - Manager との連携）」を参照してください。

■統一フォーマット用監査証跡の出力先ディレクトリを作成するファイルシステムの構築

統一フォーマット用監査証跡の出力先ディレクトリを作成するファイルシステムは，系ローカルのファイルシステムとして構築してください。

■JP1/Audit の環境設定

adbconvertaudittrailfile コマンドを実行するすべての系に対して，JP1/Audit の環境設定を行ってください。

JP1/Audit の環境設定については，「12.8.3　監査証跡機能と JP1/Audit を連携する際の環境設定」の「(2)　JP1/Audit の環境設定」を参照してください。また，adbconvertaudittrailfile コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbconvertaudittrailfile（監査証跡ファイルの変換）』を参照してください。

なお，JP1/Base をクラスタ環境として構築する必要はありません。

17.20　コールドスタンバイ構成で更新行のカラム化機能を使用する場合の運用

更新行のカラム化機能については，「11.18　更新行のカラム化機能の運用（カラムストア表の検索性能の維持）」を参照してください。ここでは，コールドスタンバイ構成で更新行のカラム化機能を使用する場合の留意事項について説明します。

- 系が切り替わったあとも，更新行のカラム化機能の状態（有効または無効）は引き継がれます。
- 更新行のカラム化機能の有効時に系切り替えが発生した場合，系切り替えが完了したあとにメンテナンス処理が実行されます。
- 更新行のカラム化機能に関するエラーの対処をしないで，更新行のカラム化機能を有効な状態のままにしておくと，系切り替えが完了したあとに実行されるメンテナンス処理がエラーとなり，エラーメッセージが出力されます。
- メンテナンス処理が 24 時間実行されない状態が続くと，KFAA51277-I メッセージが出力されますが，系切り替えが発生すると，時間経過の監視がリセットされます。そのため，系切り替えの完了後，メンテナンス処理が実行されない状態がさらに 24 時間続いたときにKFAA51277-I メッセージが出力されます。
- 系の切り替え時，新たに実行系になる系で更新行のカラム化機能で使用するファイルの入出力制御の開始処理が行われます。この処理に失敗した場合，系切り替えが完了したあとに実行されるメンテナンス処理がエラーになります。

18 Docker および Kubernetes を使用した HADB の運用

この章では，Docker および Kubernetes を使用した HADB の運用方法について説明します。

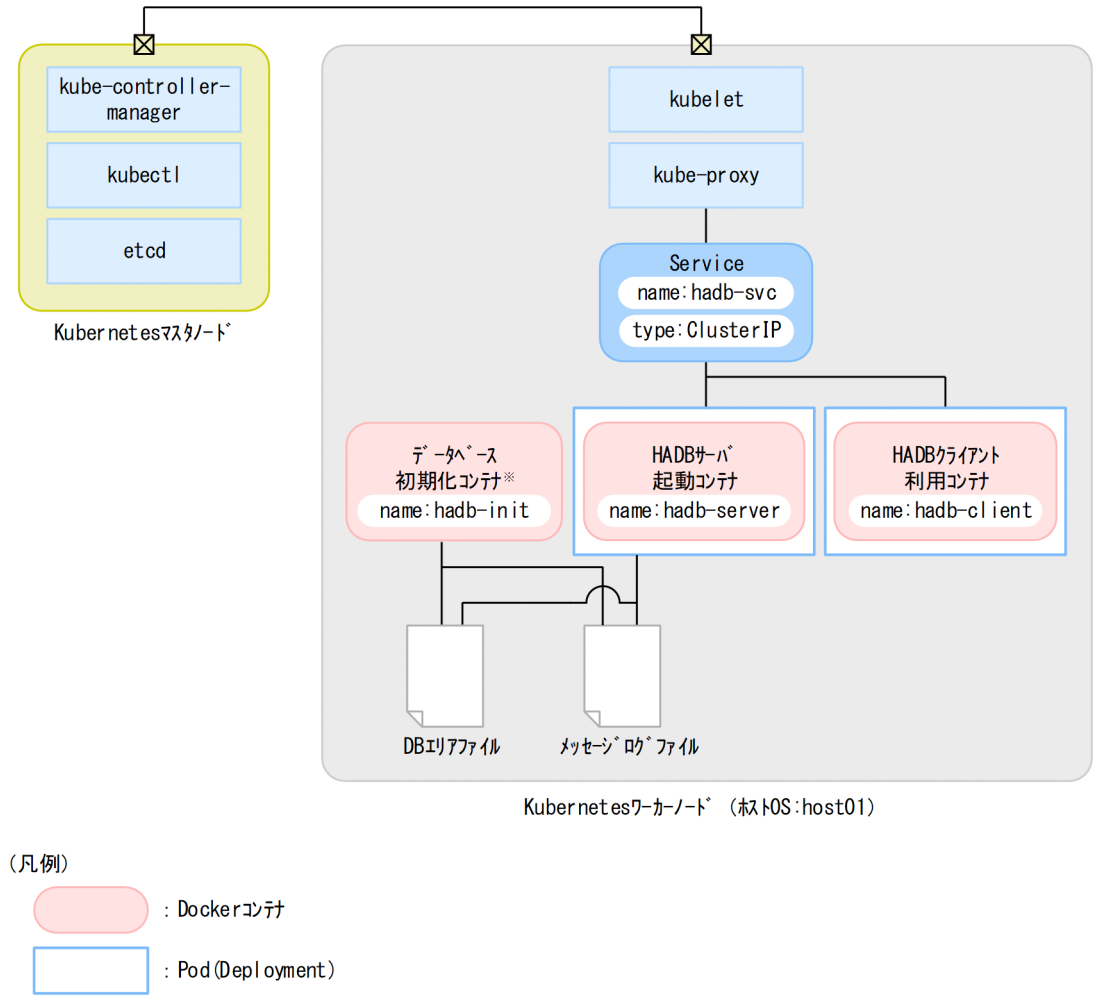
18.1　この章を読む際の留意事項

Docker および Kubernetes を使用した HADB の運用方法の説明は，Docker および Kubernetes についての知識があることを前提としています。Docker および Kubernetes については，それぞれの公式ドキュメントを参照してください。

18.2　Docker および Kubernetes を使用したシステム構成例

Docker および Kubernetes を使用したシステム構成例を次の図に示します。

図 18-1　Docker および Kubernetes を使用したシステム構成例



注※

データベース初期化コンテナは，必要に応じて起動してください。

18.2.1　前提 OS および前提ソフトウェア

Docker および Kubernetes 上で HADB を使用する場合の前提 OS と，前提ソフトウェアを次に示します。

前提 OS

- ホスト OS：RHEL 7.1 以降
- コンテナ内 OS：RHEL 7.1 以降

前提ソフトウェア

- Docker：18.06.1 以降
- Kubernetes：1.13.1 以降

18.2.2　ノード構成

「図 18-1　Docker および Kubernetes を使用したシステム構成例」で示す構成例の場合，Kubernetes マスタノードに対して，1 つの Kubernetes ワーカーノードで構成されています。

HADB 用のコンテナが起動するホスト OS のホスト名は`host01` となります。

18.2.3　コンテナ構成

「図 18-1　Docker および Kubernetes を使用したシステム構成例」で示す構成例の場合，1 つの Kubernetes ワーカーノードに 2 つの Pod（Deployment）が配置され，Pod 内にはそれぞれ次の HADB 用のコンテナが配置されています。

- HADB サーバ起動コンテナ
  HADB サーバを起動するためのコンテナです。
- HADB クライアント利用コンテナ
  HADB クライアントを使用するためのコンテナです。

また，初めて HADB を使用する場合は，上記に加えて，データベース初期化コンテナを起動する必要があります。データベース初期化コンテナは，HADB 用のデータベースを初期化する（DB エリアファイルを作成する）ためのコンテナです。

18.2.4　ネットワーク構成

Pod 間の通信は，Kubernetes のオーバーレイネットワークを使用します。そのため，Kubernetes の Service オブジェクトを ClusterIP 指定で作成します。ポート番号はホスト OS およびコンテナ内 OS ともにデフォルトの`23650` を使用します。

18.2.5　ストレージ構成

Docker および Kubernetes 上で HADB を使用する場合，ホスト OS 上のデバイスに次のファイルが格納されます。

- DB エリアファイル

- $ADBDIR/spool 下のファイル（メッセージログファイルなど）

そのため，ホスト OS 上に上記のファイルを格納する領域を用意しておいてください。「図 18-1　Docker および Kubernetes を使用したシステム構成例」で示す構成例の場合，ホスト OS 上に次のディレクトリ領域を作成し，コンテナ内のディレクトリにマウントします。

表 18-1　ホスト OS 上に用意する領域情報

項番	格納されるファイル	ホスト OS 上のパス	コンテナ内のマウント先
1	DB エリアファイル	/exports/hadb-server/dbarea	/HADB/db
2	$ADBDIR/spool 下のファイル	/exports/hadb-server/spool	/HADB/server/spool

ホスト OS 上のディレクトリ領域にパーミッションを設定して，コンテナ内の HADB 管理者に対して書き込み権限および読み取り権限を設定してください。

重要

「図 18-1　Docker および Kubernetes を使用したシステム構成例」で示す構成例の場合，構成を簡素化するために，永続ストレージとしてホスト OS 上の領域を使用しています。ただし，このような特定の Kubernetes ノードに依存する運用は，Kubernetes による分散環境の利点を十分に生かすことができません。そのため，実際に運用する際は PV（Persistent Volume）および PVC（Persistent Volume Claim）を作成するなどして，特定のノードに依存しない永続ストレージ環境を構築することを推奨します。

18.2.6　カーネルパラメタ

コンテナ環境からのカーネルパラメタの指定は，限定された一部のものしか指定できません。そのため，HADB を動作させるのに必要なカーネルパラメタの指定は，コンテナを起動するホスト OS 側で行ってください。その際，「18.3.1　Docker 環境のインストール」の「(1)　dockerd に対するカーネルパラメタの設定」もあわせて参照してください。

カーネルパラメタについては，「6.2　カーネルパラメタの見積もり」を参照してください。

18.3　前提ソフトウェアのセットアップ

ここでは，Docker および Kubernetes を使用して HADB を運用する際の前提ソフトウェアのセットアップ手順を説明します。

18.3.1　Docker 環境のインストール

Kubernetes クラスタを構成するすべてのノードに対して，Docker をインストールしてください。Docker のインストール方法については，Docker の公式ドキュメントを参照してください。

(1)　dockerd に対するカーネルパラメタの設定

dockerd はデフォルトで許可されているメモリロックの量（memlock）が少ないため，次の表に示す設定ファイルの内容を変更し，ロックできるメモリの量を増やしてください。

表 18-2　変更が必要な設定ファイル（daemon.json ファイル）

項番	変更が必要な設定ファイル	所有者	アクセス権限
1	/etc/docker/daemon.json	スーパユーザ	644（デフォルト）

上記のdaemon.json ファイルに，JSON 形式で次の内容を追記してください。

```
"default-ulimits": {
  "memlock": {
    "Name": "memlock",
    "Hard": -1,
    "Soft": -1
  }
}
```

メモ

daemon.json ファイルは，JSON 形式で記述する必要があります。JSON 形式に沿う形で追記してください。daemon.json ファイルの内容の詳細については，Docker の公式ドキュメントを参照してください。

変更後の設定を有効にするために，次のコマンドを実行してください。

```
systemctl restart docker
```

18.3.2　Kubernetes 環境のインストールとノードの構築

Kubernetes クラスタを構成するすべてのノードに対して，Kubernetes をインストールしてください。Kubernetes のインストール方法については，Kubernetes の公式ドキュメントを参照してください。

Kubernetes のインストール後，`kubeadm` コマンドを実行して Kubernetes マスタノードと Kubernetes ワーカーノードを構築してください。「図 18-1　Docker および Kubernetes を使用したシステム構成例」で示す構成例の場合，1 つの Kubernetes マスタノードと 1 つの Kubernetes ワーカーノードが存在していることを前提としています。

18.4　Docker での HADB の環境構築

ここでは，まず Docker で HADB の環境を構築する手順を説明します。Docker での HADB の環境構築が完了したら，「18.5　Kubernetes を使用した HADB の環境構築」を参照してください。

18.4.1　Dockerfile による Docker イメージの生成

HADB 用の各コンテナの Docker イメージを生成する手順を説明します。なお，以降で説明する手順は HADB 用の各コンテナを起動する Kubernetes ワーカーノード（ホスト OS）上で実施してください。

(1)　データベース初期化コンテナの Docker イメージを生成する

データベース初期化コンテナの Docker イメージを生成してください。なお，初期化済みの DB エリアファイルが存在する場合は，この Docker イメージを生成する必要はありません。

手順

1. コンテナの Docker イメージを生成するDockerfile を準備する

   Docker イメージを生成するためのDockerfile を準備してください。Dockerfile は，データベース初期化コンテナを起動する Kubernetes ワーカーノード上に保存してください。保存先を次に示します。

   ■Dockerfile の保存先

   ```
   /home/docker/hadb_init/Dockerfile
   ```

   Dockerfile の例を次に示します。

   ■Dockerfile の例

   ```
   FROM registry.access.redhat.com/rhel7:latest

   MAINTAINER Hitachi, Ltd.

   ARG HADB_VERSION

   RUN set -ex; ¥
           yum -y install ¥
               file ¥
               gdb ¥
               hostname ¥
               iproute ¥
               libaio ¥
               lvm2 ¥
               sysstat ¥
        && rm -rf /var/cache/yum/* ¥
        && yum clean all ¥
        && ln -sf /usr/share/zoneinfo/Asia/Tokyo /etc/localtime ¥
        && localedef -i ja_JP -f UTF-8 ja_JP.UTF-8 ¥
        && groupadd --gid=990 adbgroup && useradd --uid=990 -g adbgroup -m adbmanager ¥
        && mkdir -p /HADB/db && chown -R adbmanager:adbgroup /HADB
   ```

```
COPY ./hitachi_advanced_data_binder_server-${HADB_VERSION}.tar.gz ¥
     ./adbinstall ¥
     ./adbinit.opt ¥
     /home/adbmanager/install/
RUN chown -R adbmanager:adbgroup /home/adbmanager/install ¥
 && chmod 500 /home/adbmanager/install/adbinstall

USER adbmanager

ENV ADBDIR=/HADB/server ¥
    ADBLANG=UTF8 ¥
    LANG=ja_JP.UTF-8
ENV DBDIR=/HADB/db ¥
    ADBCLTDIR=$ADBDIR ¥
    ADBCLTLANG=$ADBLANG ¥
    LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ADBDIR/lib:$ADBDIR/client/lib ¥
    PATH=$PATH:$ADBDIR/bin:$ADBDIR/client/bin

RUN set -eux; ¥
        /home/adbmanager/install/adbinstall -s $ADBDIR ¥
     && cp -rf $ADBDIR/sample/conf $ADBDIR ¥
     && cp -f /home/adbmanager/install/adbinit.opt $ADBDIR/conf/ ¥
     && rm -rf /home/adbmanager/install

USER root

VOLUME ["$DBDIR", "$ADBDIR/spool"]

ENTRYPOINT chown -R adbmanager:adbgroup "$DBDIR" "$ADBDIR"/spool ¥
        && runuser adbmanager -c '"$ADBDIR"/bin/adbinit -u "$HADB_ADMIN_NAME" -p "$HADB_A
DMIN_PASSWORD" "$ADBDIR"/conf/adbinit.opt "$DBDIR"'
```

[説明]

上記のDockerfile の例は，次のすべての設定が完了していることを前提としています。

- コンテナ内のインターネット接続
- プロキシ設定
- リポジトリ設定

上記の設定が完了している場合は，コンテナ内でyum コマンドによるパッケージのインストールが可能です。上記の設定が完了していない場合は，先に設定を完了してください。または，Dockerfile を編集して，yum コマンドによるインストールの設定を組み込んでください。

メモ

上記のDockerfile の例では，コンテナ起動時（docker run コマンド実行時）に--env オプションを使用して，HADB のデータベース作成時に登録するユーザの認可識別子やパスワードなどの秘密情報を受け渡します。新しいバージョンの Docker（18.09 以降）を使用する場合は，BuildKit が提供しているdocker build の--secret フラグを使用することを推奨します。

2. HADB サーバのインストールに必要なファイルおよび初期設定オプションファイルを準備する

手順 1.のDockerfile を格納したディレクトリに，次のファイルを格納してください。

- HADB サーバのインストールに必要な次のファイル
  - HADB サーバのプログラムの圧縮ファイル（hitachi_advanced_data_binder_server-*$VER*.tar.gz）
  - adbinstall コマンドの実行形式ファイル（adbinstall）
- adbinit コマンドの初期設定オプションファイル（adbinit.opt）

$VER は，HADB サーバのバージョンとリリース番号です。

メモ

- 各定義ファイルやオプションファイルのひな形は，$ADBDIR/sample/conf/ディレクトリに格納されています。
- HADB サーバのインストール時に必要なファイルを格納する際，権限の付与や，コピーしたファイルの確認作業などが必要になります。詳細な方法については，「8.2.2　インストール手順」を参照してください。その際，次の点に注意してください。
  - インストールデータを格納するディレクトリは，Dockerfile を格納したディレクトリ（/home/docker/hadb_init）と読み替えてください。
  - !8A9_HADB ファイルのコピーは必要ありません。HADB サーバのプログラムの圧縮ファイルとadbinstall コマンドの実行形式ファイルをコピーしてください。
- 初期設定オプションファイルの作成方法については，マニュアル『HADB　コマンドリファレンス』の『adbinit（データベースの初期設定)』の『adbinit コマンドの指定形式』の『初期設定オプションの形式』を参照してください。

次のコマンドを実行すると，すべてのファイルがそろっているかどうかを確認できます。

■コマンドの実行例

```
ls /home/docker/hadb_init
```

すべてのファイルがそろっている場合，次のような実行結果が表示されます。

■コマンドの実行結果の例

```
Dockerfile   adbinstall
adbinit.opt  hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，HADB サーバのバージョンとリリース番号です。

3. Docker イメージを生成する

次のコマンドを実行し，Docker イメージを生成してください。

■コマンドの実行例

```
cd /home/docker/hadb_init
docker build -f ./Dockerfile -t hadb-init:$VER ¥
--build-arg HADB_VERSION=$VER ¥
.
```

$VER は，HADB サーバのバージョンとリリース番号です。

上記のコマンドが正常終了したら，次のコマンドを実行してください。次のコマンドを実行すると，Docker イメージが正しく生成されているかどうかを確認できます。

■コマンドの実行例

```
docker images hadb-init:$VER
```

$VER は，HADB サーバのバージョンとリリース番号です。

Docker イメージが正しく生成されている場合，次のような結果が表示されます。

■コマンドの実行結果の例

```
REPOSITORY   TAG      IMAGE ID       CREATED          SIZE
hadb-init    $VER     6210a7de50ea   3 seconds ago    204 MB
```

$VER は，HADB サーバのバージョンとリリース番号です。

なお，IMAGE ID 列やCREATED 列，SIZE 列に表示される値は実行環境によって異なります。

(2) HADB サーバ起動コンテナの Docker イメージを生成する

HADB サーバ起動コンテナの Docker イメージを生成してください。

手順

1. コンテナの Docker イメージを生成するDockerfile を準備する

   Docker イメージを生成するためのDockerfile を準備してください。Dockerfile は，HADB サーバ起動コンテナを起動する Kubernetes ワーカーノード上に保存してください。保存先を次に示します。

   ■Dockerfile の保存先

   ```
   /home/docker/hadb_server/Dockerfile
   ```

   Dockerfile の例を次に示します。

   ■Dockerfile の例

   ```
   FROM registry.access.redhat.com/rhel7:latest

   MAINTAINER Hitachi, Ltd.

   ARG HADB_VERSION

   RUN set -ex; ¥
           yum -y install ¥
               file ¥
               gdb ¥
               hostname ¥
               iproute ¥
               libaio ¥
               lvm2 ¥
               sysstat ¥
        && rm -rf /var/cache/yum/* ¥
        && yum clean all ¥
        && ln -sf /usr/share/zoneinfo/Asia/Tokyo /etc/localtime ¥
   ```

```
     && localedef -i ja_JP -f UTF-8 ja_JP.UTF-8 ¥
     && groupadd --gid=990 adbgroup && useradd --uid=990 -g adbgroup -m adbmanager ¥
     && mkdir -p /HADB/db && chown -R adbmanager:adbgroup /HADB

COPY ./hitachi_advanced_data_binder_server-${HADB_VERSION}.tar.gz ¥
     ./adbinstall ¥
     ./server.def ¥
     /home/adbmanager/install/
RUN chown -R adbmanager:adbgroup /home/adbmanager/install ¥
 && chmod 500 /home/adbmanager/install/adbinstall

USER adbmanager

ENV ADBDIR=/HADB/server ¥
    ADBLANG=UTF8 ¥
    LANG=ja_JP.UTF-8
ENV DBDIR=/HADB/db ¥
    ADBCLTDIR=$ADBDIR ¥
    ADBCLTLANG=$ADBLANG ¥
    LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ADBDIR/lib:$ADBDIR/client/lib ¥
    PATH=$PATH:$ADBDIR/bin:$ADBDIR/client/bin

RUN set -eux; ¥
        /home/adbmanager/install/adbinstall -s $ADBDIR ¥
     && cp -rf $ADBDIR/sample/conf $ADBDIR ¥
     && cp -f /home/adbmanager/install/server.def $ADBDIR/conf/ ¥
     && rm -rf /home/adbmanager/install

# Change the execution user back to the root user.
# This is needed to create /dev/HADB/pth by executing the shell script specified in the E
NTRYPOINT instruction.
USER root

VOLUME [”$DBDIR”, ”$ADBDIR/spool”]

EXPOSE 23650

COPY ./docker-entrypoint.sh /usr/local/bin/
ENTRYPOINT [”docker-entrypoint.sh”]
```

[説明]

上記のDockerfile の例は，次のすべての設定が完了していることを前提としています。

- コンテナ内のインターネット接続
- プロキシ設定
- リポジトリ設定

上記の設定が完了している場合は，コンテナ内でyum コマンドによるパッケージのインストールが可能です。上記の設定が完了していない場合は，先に設定を完了してください。または，Dockerfile を編集して，yum コマンドによるインストールの設定を組み込んでください。

2. HADB サーバ起動用のシェルスクリプトを準備する

手順 1.のDockerfile と同じ場所に，HADB サーバ起動用のシェルスクリプトを準備してください。

HADB サーバ起動コンテナの Docker イメージを生成するDockerfile 内では，HADB サーバを起動し，HADB サーバが起動している間待機するシェルスクリプトをENTRYPOINT で呼び出しています。そのため，HADB サーバ起動用のシェルスクリプトが必要になります。シェルスクリプト（docker-entrypoint.sh）の例を次に示します。

■シェルスクリプトの例

```
#!/bin/bash

# Create directories for storing communication-information files.
mkdir -p -m 777 /dev/HADB/pth

# Start the HADB server as an HADB administrator.
runuser adbmanager -c "yes | adbstart"

# When the container stops, terminate HADB normally by using the adbstop command with the
 --cancel option.
trap 'runuser adbmanager -c "adbstop --cancel"' SIGTERM

# Check whether adbsrvd exists.
while [ $? -eq 0 ];
do
  # The wait interval is 1 second.
  sleep 1
  ps -C adbsrvd -o command= | grep -E "^${ADBDIR}/bin/adbsrvd¥$" 1>/dev/null 2>&1
done

exit
```

重要

任意のユーザが実行できるように，このシェルスクリプトには必ず実行権限を付与してください。

3. HADB サーバのインストールに必要なファイルおよびサーバ定義ファイルを準備する

手順 1.のDockerfile を格納したディレクトリに，次のファイルを格納してください。

- HADB サーバのインストールに必要な次のファイル
  - HADB サーバのプログラムの圧縮ファイル（hitachi_advanced_data_binder_server-*$VER*.tar.gz）
  - adbinstall コマンドの実行形式ファイル（adbinstall）
- サーバ定義ファイル（server.def）

$VER は，HADB サーバのバージョンとリリース番号です。

メモ

- 各定義ファイルやオプションファイルのひな形は，$ADBDIR/sample/conf/ディレクトリに格納されています。

- HADB サーバのインストール時に必要なファイルを格納する際，権限の付与や，コピーしたファイルの確認作業などが必要になります。詳細な方法については，「8.2.2　インストール手順」を参照してください。その際，次の点に注意してください。
  - インストールデータを格納するディレクトリは，Dockerfile を格納したディレクトリ（/home/docker/hadb_server）と読み替えてください。
  - !8A9_HADB ファイルのコピーは必要ありません。HADB サーバのプログラムの圧縮ファイルとadbinstall コマンドの実行形式ファイルをコピーしてください。
- サーバ定義ファイルの作成方法については，「8.5.1　サーバ定義の作成方法および格納先」を参照してください。
- サーバ定義ファイルを作成する際，DB ディレクトリのパスを指定するadb_db_path オペランドには，「/HADB/db（Dockerfile 内の$DBDIR の値）」を指定してください。その際，「18.2.5　ストレージ構成」も参照してください。

次のコマンドを実行すると，すべてのファイルがそろっているかどうかを確認できます。

■コマンドの実行例

```
ls /home/docker/hadb_server
```

すべてのファイルがそろっている場合，次のような実行結果が表示されます。

■コマンドの実行結果の例

```
Dockerfile  docker-entrypoint.sh                                server.def
adbinstall  hitachi_advanced_data_binder_server-$VER.tar.gz
```

$VER は，HADB サーバのバージョンとリリース番号です。

4. Docker イメージを生成する

次のコマンドを実行し，Docker イメージを生成してください。

■コマンドの実行例

```
cd /home/docker/hadb_server
docker build -f ./Dockerfile -t hadb-server:$VER ¥
--build-arg HADB_VERSION=$VER ¥
.
```

$VER は，HADB サーバのバージョンとリリース番号です。

上記のコマンドが正常終了したら，次のコマンドを実行してください。次のコマンドを実行すると，Docker イメージが正しく生成されているかどうかを確認できます。

■コマンドの実行例

```
docker images hadb-server:$VER
```

$VER は，HADB サーバのバージョンとリリース番号です。

Docker イメージが正しく生成されている場合，次のような結果が表示されます。

■コマンドの実行結果の例

```
REPOSITORY    TAG     IMAGE ID       CREATED          SIZE
hadb-server   $VER    2eeb8964de7a   3 seconds ago    204 MB
```

$VER は，HADB サーバのバージョンとリリース番号です。

なお，IMAGE ID 列やCREATED 列，SIZE 列に表示される値は実行環境によって異なります。

(3) HADB クライアント利用コンテナの Docker イメージを生成する

HADB クライアント利用コンテナの Docker イメージを生成してください。

手順

1. コンテナの Docker イメージを生成するDockerfile を準備する

   Docker イメージを生成するためのDockerfile を準備してください。Dockerfile は，HADB クライアント利用コンテナを起動する Kubernetes ワーカーノード上に保存してください。保存先を次に示します。

   ■Dockerfile の保存先

   ```
   /home/docker/hadb_client/Dockerfile
   ```

   Dockerfile の例を次に示します。

   ■Dockerfile の例

   ```
   FROM registry.access.redhat.com/rhel7:latest

   MAINTAINER Hitachi, Ltd.

   ARG HADB_VERSION

   RUN set -ex; ¥
           yum -y install ¥
               libaio ¥
        && rm -rf /var/cache/yum/* ¥
        && yum clean all ¥
        && ln -sf /usr/share/zoneinfo/Asia/Tokyo /etc/localtime ¥
        && localedef -i ja_JP -f UTF-8 ja_JP.UTF-8 ¥
        && useradd -m adbuser

   COPY ./hitachi_advanced_data_binder_client-${HADB_VERSION}.tar.gz ¥
        ./adbinstall ¥
        ./client.def ¥
        /home/adbuser/install/
   RUN chown -R adbuser:adbuser /home/adbuser/install ¥
    && chmod 555 /home/adbuser/install/adbinstall

   USER adbuser

   ENV ADBCLTDIR=/home/adbuser/client ¥
       ADBCLTLANG=UTF8 ¥
       LANG=ja_JP.UTF-8
   ENV LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ADBCLTDIR/client/lib ¥
       PATH=$PATH:$ADBCLTDIR/client/bin
   ```

```
RUN set -eux; ¥
        /home/adbuser/install/adbinstall -c $ADBCLTDIR ¥
     && cp -rf $ADBCLTDIR/sample/conf $ADBCLTDIR ¥
     && cp -f /home/adbuser/install/client.def $ADBCLTDIR/conf/ ¥
     && rm -rf /home/adbuser/install

# Keep the container running.
CMD ["tail","-f","/dev/null"]
```

[説明]

上記のDockerfile の例は，次のすべての設定が完了していることを前提としています。

- コンテナ内のインターネット接続
- プロキシ設定
- リポジトリ設定

上記の設定が完了している場合は，コンテナ内でyum コマンドによるパッケージのインストールが可能です。上記の設定が完了していない場合は，先に設定を完了してください。または，Dockerfile を編集して，yum コマンドによるインストールの設定を組み込んでください。

2. HADB クライアントのインストールに必要なファイルおよびクライアント定義ファイルの準備をする

手順 1.のDockerfile を格納したディレクトリに，次のファイルを格納してください。

- HADB クライアントのインストールに必要な次のファイル
  - HADB クライアントのプログラムの圧縮ファイル（hitachi_advanced_data_binder_client-*$VER*.tar.gz）
  - adbinstall コマンドの実行形式ファイル（adbinstall）
- クライアント定義ファイル（client.def）

$VER は，HADB クライアントのバージョンとリリース番号です。

メモ

- 各定義ファイルやオプションファイルのひな形は，$ADBDIR/sample/conf/ディレクトリに格納されています。
- ファイルの格納方法については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントのインストールおよびアンインストール』の『Linux 版の HADB クライアントの場合』の『HADB クライアントのインストール』を参照してください。その際，インストールデータを格納するディレクトリは，Dockerfile を格納したディレクトリ（/home/docker/hadb_client）と読み替えてください。
- クライアント定義ファイルの作成方法については，マニュアル『HADB　AP 開発ガイド』の『HADB クライアントの環境設定（ODBC ドライバおよび CLI 関数を使用する場合)』の『クライアント定義の作成』を参照してください。
- クライアント定義ファイルを作成する際，HADB サーバのホスト名を指定する adb_clt_rpc_srv_host オペランドには，後述する Kubernetes の Service オブジェクト

の名前である「hadb-svc」を指定してください。「hadb-svc」は，「図 18-1　Docker および Kubernetes を使用したシステム構成例」の図中に示している Service オブジェクトです。

次のコマンドを実行すると，すべてのファイルがそろっているかどうかを確認できます。

■コマンドの実行例

```
ls /home/docker/hadb_client
```

すべてのファイルがそろっている場合，次のような実行結果が表示されます。

■コマンドの実行結果の例

```
Dockerfile  client.def
adbinstall  hitachi_advanced_data_binder_client-$VER.tar.gz
```

$VER は，HADB クライアントのバージョンとリリース番号です。

3. Docker イメージを生成する

次のコマンドを実行し，Docker イメージを生成してください。

■コマンドの実行例

```
cd /home/docker/hadb_client
docker build -f ./Dockerfile -t hadb-client:$VER ¥
--build-arg HADB_VERSION=$VER ¥
.
```

$VER は，HADB クライアントのバージョンとリリース番号です。

上記のコマンドが正常終了したら，次のコマンドを実行してください。次のコマンドを実行すると，Docker イメージが正しく生成されているかどうかを確認できます。

■コマンドの実行例

```
docker images hadb-client:$VER
```

$VER は，HADB クライアントのバージョンとリリース番号です。

Docker イメージが正しく生成されている場合，次のような結果が表示されます。

■コマンドの実行結果の例

```
REPOSITORY    TAG     IMAGE ID       CREATED          SIZE
hadb-client   $VER    9f266d35e02c   3 seconds ago    204 MB
```

$VER は，HADB クライアントのバージョンとリリース番号です。

なお，IMAGE ID 列やCREATED 列，SIZE 列に表示される値は実行環境によって異なります。

18.4.2　コンテナ内の HADB が使用する DB ディレクトリの初期化

HADB 用のデータベースを初期化する方法を説明します。なお，初期化済みの DB エリアファイルが存在する場合は，この方法を実施する必要はありません。

次のコマンドを実行して，HADB 用のデータベースを初期化してください。

```
docker run ¥
-it ¥
--rm ¥
--mount type=bind,src=/exports/hadb-server/dbarea,dst=/HADB/db ¥
--mount type=bind,src=/exports/hadb-server/spool,dst=/HADB/server/spool ¥
--env HADB_ADMIN_NAME="ADBADMIN" ¥                    …1
--env HADB_ADMIN_PASSWORD="#HelloDockerHADB_ZA" ¥    …2
hadb-init:$VER
```

$VER は，HADB サーバのバージョンとリリース番号です。

[説明]

1. データベースの作成時に登録するユーザの認可識別子（ADBADMIN）を指定してください。
2. 1.で作成するユーザのパスワード（#HelloDockerHADB_ZA）を指定してください。

18.5　Kubernetes を使用した HADB の環境構築

ここでは，Kubernetes を使用して HADB を運用する場合の環境構築の手順を説明します。以降の手順は，Docker での HADB の環境構築が完了したあとに実施してください。Docker での HADB の環境構築については，「18.4　Docker での HADB の環境構築」を参照してください。

18.5.1　Kubernetes の各リソースオブジェクトの生成

Kubernetes の各リソースオブジェクトを生成する手順を説明します。

生成するリソースオブジェクトは次のとおりです。

- HADB サーバ用 Deployment
- HADB クライアント用 Deployment
- コンテナ間通信用 Service

各リソースオブジェクトの生成は，Kubernetes マスタノード上から`kubectl` コマンドを使用して実施してください。

`kubectl` コマンドには次の実行方法がありますが，ここではマニフェストファイルを指定する方法を説明します。

- リソースオブジェクトの生成方法を定義したマニフェストファイル（`yaml` ファイル）を指定する方法
- `run` サブコマンドに直接 Docker イメージを指定する方法

> **メモ**
>
> `kubectl` コマンドを実行できるユーザで各リソースオブジェクトを生成してください。なお，スーパユーザで実行することを想定しています。異なるユーザで実行する場合は，`yaml` ファイルの格納場所などを適宜変更してください。

(1)　マニフェストファイル（yaml ファイル）を準備する

リソースオブジェクトの生成方法を定義したマニフェストファイルを準備します。マニフェストファイルは，Kubernetes マスタノードに保存してください。保存先を次に示します。

■マニフェストファイルの保存先

```
/root/k8s/hadb-k8s-manifest.yaml
```

マニフェストファイルの例を次に示します。

メモ

マニフェストファイルは，行頭のインデント（文字の開始位置）が厳密に評価されます。注意して作成してください。

■マニフェストファイルの例

```
apiVersion: v1
kind: Service
metadata:
  name: hadb-svc
spec:
  type: ClusterIP
  selector:
    app: hadb-server
  ports:
    - name: hadb-port
      protocol: TCP
      port: 23650
      targetPort: 23650
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hadb-server
  labels:
    name: hadb-server
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hadb-server
  template:
    metadata:
      labels:
        app: hadb-server
      annotations:
        security.alpha.kubernetes.io/sysctls: net.ipv4.ip_local_port_range=32767 61001
        security.alpha.kubernetes.io/unsafe-sysctls: kernel.msgmnb=16386,
          kernel.msgmni=736,
          kernel.sem=250 32001 32 129,
          kernel.shmmax=18446744073692774400,
          kernel.shmmni=4097,
          kernel.shmall=18446744073692774400,
          net.core.somaxconn=129
    spec:
      # This container starts the HADB server.
      containers:
      - name: hadb-server
        image: hadb-server:$VER
        ports:
        - containerPort: 23650
          protocol: TCP
        resources:
          requests:
```

```
            cpu: 2
            memory: 8Gi
          limits:
            cpu: 45
            memory: 160Gi
        volumeMounts:
        - name: dbarea-vol
          mountPath: /HADB/db
        - name: spool-vol
          mountPath: /HADB/server/spool
        lifecycle:
          preStop:
            exec:
              command: ["/bin/sh", "-c", "echo Terminating HADB...; sleep 3; runuser adbmana
ger -c 'adbstop --cancel'; sleep 10"]
      volumes:
      - name: dbarea-vol
        hostPath:
          path: /exports/hadb-server/dbarea
          type: DirectoryOrCreate
      - name: spool-vol
        hostPath:
          path: /exports/hadb-server/spool
          type: DirectoryOrCreate
      terminationGracePeriodSeconds: 30
      nodeSelector:
        kubernetes.io/hostname: host01
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hadb-client
  labels:
    name: hadb-client
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hadb-client
  template:
    metadata:
      labels:
        app: hadb-client
      annotations:
        security.alpha.kubernetes.io/sysctls: net.ipv4.ip_local_port_range=32767 61001
        security.alpha.kubernetes.io/unsafe-sysctls: kernel.msgmnb=16386,
          kernel.msgmni=736,
          kernel.sem=250 32001 32 129,
          kernel.shmmax=18446744073692774400,
          kernel.shmmni=4097,
          kernel.shmall=18446744073692774400,
          net.core.somaxconn=129
    spec:
      containers:
      - name: hadb-client
        image: hadb-client:$VER
        resources:
          requests:
```

```
            cpu: 2
            memory: 8Gi
          limits:
            cpu: 4
            memory: 16Gi
```

$VER は，HADB サーバのバージョンとリリース番号です。

［説明］

上記のマニフェストファイルは，HADB 用に作成する Kubernetes の各リソースオブジェクトの定義を，すべて 1 つのファイルに記述しています。

Kubernetes の各リソースオブジェクト

- HADB サーバ用 Deployment
- HADB クライアント用 Deployment
- コンテナ間通信用 Service

(2) kubectl コマンドでリソースオブジェクトを生成する

次のコマンドを実行して，HADB 用に Kubernetes の各リソースオブジェクトを生成してください。

■コマンドの実行例

```
cd /root/k8s
kubectl apply -f ./hadb-k8s-manifest.yaml --record
```

リソースオブジェクトが正常に生成されている場合，次のような結果が表示されます。

```
service/hadb-svc created
deployment.apps/hadb-server created
deployment.apps/hadb-client created
```

また，次のコマンドを実行すると，生成されたリソースオブジェクトが正しく動作しているかどうかを確認できます。

■コマンドの実行例

```
kubectl get deployments
kubectl get pods
kubectl get services hadb-svc
```

リソースオブジェクトが正しく動作している場合，次のような結果が表示されます。

■コマンドの実行結果の例

```
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hadb-client   1/1     1            1           14s
hadb-server   1/1     1            1           14s

NAME                             READY   STATUS    RESTARTS   AGE
hadb-client-6fbb9fbb97-4xvkr     1/1     Running   0          19s
hadb-server-5fd5cc7fb8-8jv58     1/1     Running   0          20s

NAME       TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
hadb-svc   ClusterIP   10.233.47.129   <none>        23650/TCP   28s
```

なお，各列に表示される値は実行環境によって異なります。また，kubectl のdescribe サブコマンドを使用すると，各リソースオブジェクトのさらに詳細な情報を参照できます。

18.6　Docker および Kubernetes を使用した HADB の運用

ここでは，Docker および Kubernetes を使用した HADB の運用例を説明します。

18.6.1　HADB のコマンドの実行方法

Docker および Kubernetes を使用して HADB のコマンドを実行する手順を説明します。

手順

1. HADB サーバ起動コンテナのシェルを取得する

   次のコマンドを実行してください。

   ■コマンドの実行例

   ```
   kubectl exec -it hadb-server-5fd5cc7fb8-8jv58 /bin/bash
   ```

   [説明]

   次のとおりに指定してください。

   - Pod 名（ここでは`hadb-server-5fd5cc7fb8-8jv58`）

     `kubectl get pods` コマンドで取得した，HADB サーバ起動コンテナが配置されている Pod の名前を指定してください。

   - シェル（ここでは`/bin/bash`）

     引数を指定する場合は，指定するシェルの前にダブルダッシュ（`--`）を指定してください。

     ■コマンドの実行例

     ```
     kubectl exec -it hadb-server-5fd5cc7fb8-8jv58 -- /bin/bash -l
     ```

2. ユーザを HADB 管理者に切り替える

   手順 1.のコマンドを実行すると，HADB サーバ起動コンテナのシェルプロンプトが表示されます。この時点では，HADB サーバ起動コンテナ内の`root` ユーザでログインしている状態になるため，次の `runuser` コマンドを実行して，ユーザを HADB 管理者に切り替えてください。

   ■コマンドの実行例

   ```
   [root@hadb-server-5fd5cc7fb8-8jv58 /]# runuser adbmanager
   [adbmanager@hadb-server-5fd5cc7fb8-8jv58 /]$ cd
   [adbmanager@hadb-server-5fd5cc7fb8-8jv58 ~]$
   ```

   [説明]

   下線部分は，コンテナ内のシェルプロンプトです。表示内容は実行環境によって異なります。また，`adbmanager` は，HADB サーバ起動コンテナ内の HADB 管理者のユーザ名です。

3. HADB のコマンドを実行する

通常の Linux と同じように HADB 管理者で各種コマンドを実行できます。各種コマンドの詳しい使い方については，『HADB　コマンドリファレンス』の各章を参照してください。ただし，adbstop コマンドは実行しないでください。

なお，adbimport コマンドを実行する場合に，必要な入力データファイルがコンテナの外部にあるときは，次のどちらかを実施してください。

- コンテナ内からアクセス可能な場所に，入力データファイルを格納する
- コンテナ内に入力データファイルをコピーする

コンテナ内に入力データファイルをコピーする場合は，次のコマンドを実行してください。ホスト OS 上の入力データファイル（/root/import/data.csv）を，HADB サーバ起動コンテナ内のディレクトリ（/home/adbmanager/）にコピーできます。

■コマンドの実行例

```
kubectl cp /root/import/data.csv hadb-server-5fd5cc7fb8-8jv58:/home/adbmanager/
```

[説明]

下線部分は Pod 名です。kubectl get pods コマンドで取得した，HADB サーバ起動コンテナが配置されている Pod 名を指定してください。

> **メモ**
>
> 上記のコマンドでコピーしたファイルは，ファイルの所有者がコンテナ内の HADB 管理者になっていません。HADB サーバ起動コンテナのシェルプロンプトを取得したあとに，chown コマンドを使用して，HADB 管理者に所有者を変更してください。

4. HADB サーバ起動コンテナのシェルを終了する

コンテナ内のシェルプロンプトでexit を入力してください。runuser コマンドでユーザを切り替えている場合，exit を複数回実行してください。

> **重要**
>
> コンテナ内でadbstop コマンドを実行しないでください。adbstop コマンドを実行して HADB サーバを終了させると，HADB サーバ起動コンテナ自体も同時に停止してしまいます。なお，コンテナ内でadbstop コマンドを実行してしまった場合は，Deployment オブジェクトの定義のreplicas 数に応じて，Kubernetes が HADB サーバ起動コンテナを含む Pod オブジェクトを再作成します。そのため，HADB サーバ起動コンテナを再度手動で作成したり，HADB サーバ起動コンテナを含む Deployment オブジェクトを再作成したりする必要はありません。

18.6.2　adbsql コマンドの実行方法

Docker および Kubernetes を使用してadbsql コマンドを実行する手順を示します。adbsql コマンドを使用して HADB サーバに接続するには，HADB クライアント利用コンテナのシェルを取得します。

手順

1. HADB クライアント利用コンテナのシェルを取得する

   次のコマンドを実行してください。

   ■コマンドの実行例

   ```
   kubectl exec -it hadb-client-6fbb9fbb97-4xvkr /bin/bash
   ```

   [説明]

   次のとおりに指定してください。

   - Pod 名（ここではhadb-client-6fbb9fbb97-4xvkr）

     kubectl get pods コマンドで取得した，HADB クライアント利用コンテナが配置されている Pod の名前を指定してください。

   - シェル（ここでは/bin/bash）

     引数を指定する場合は，指定するシェルの前にダブルダッシュ（--）を指定してください。

     ■コマンドの実行例

     ```
     kubectl exec -it hadb-client-6fbb9fbb97-4xvkr -- /bin/bash -l
     ```

   > **メモ**
   >
   > 手順 1.のコマンドを実行すると，HADB クライアント利用コンテナのシェルプロンプトが表示されます。これ以降は，通常の Linux と同じように HADB クライアントを使用できます。

2. コマンドを実行する

   adbsql コマンドを使用する例を次に示します。

   ■コマンドの実行例

   ```
   [adbuser@hadb-client-6fbb9fbb97-4xvkr ~]$ adbsql
     USER-ID ?
   ADBUSER
     PASSWORD ?
   COMMAND ?     +----2----+----3----+----4----+----5----+----6----+----7----+
   SELECT * FROM MASTER.SQL_TABLES;
   .
   .
   .
   ```

   [説明]

   下線部分は，コンテナ内のシェルプロンプトです。表示内容は実行環境によって異なります。3 行目の「ADBUSER」は，HADB サーバに接続する際のユーザの認可識別子です。使用するデータベースによって異なります。

   adbsql コマンドの詳しい使い方については，マニュアル『HADB　コマンドリファレンス』の『adbsql (SQL の実行)』を参照してください。

3. HADB クライアント利用コンテナのシェルを終了する

コンテナ内のシェルプロンプトで`exit` を入力してください。

18.6.3 AP の運用

AP から Docker および Kubernetes を使用した HADB の環境を利用する場合は，AP を実行する Pod を同一の Kubernetes クラスタ上に作成してください。HADB クライアント用の Pod と同様に，HADB サーバのホスト名およびポート番号を使用できます。Pod の作成方法については，Docker および Kubernetes の公式ドキュメントを参照してください。

18.7　Docker および Kubernetes の使用停止

ここでは，Docker および Kubernetes の使用をやめる手順を説明します。

Kubernetes のリソースオブジェクトを削除し，そのあとに Docker イメージを削除することで，Docker および Kubernetes の使用をやめることができます。

18.7.1　Kubernetes の使用停止

次のコマンドを実行し，作成した Kubernetes のリソースオブジェクトを削除してください。削除完了のメッセージが出力されるまで，しばらく時間が掛かる場合があります。

■コマンドの実行例

```
cd /root/k8s
kubectl delete -f ./hadb-k8s-manifest.yaml
```

> **重要**
>
> リソースオブジェクトを直ちに削除する「--grace-period=0 --force」オプションの指定は推奨しません。このオプションを指定してリソースオブジェクトを削除した場合，HADB のデータベースが破壊されるおそれがあります。

リソースオブジェクトが正常に削除されている場合，次のような実行結果が表示されます。

■コマンドの実行結果の例

```
service ”hadb-svc” deleted
deployment.apps ”hadb-server” deleted
deployment.apps ”hadb-client” deleted
```

18.7.2　Docker の使用停止

Docker イメージを今後使用する予定がない場合は，Docker イメージを作成したノードで次のコマンドを実行して，Docker イメージを削除してください。なお，先に Kubernetes のリソースオブジェクトを削除していない場合，Docker イメージは削除できません（エラーとなります）。Kubernetes のリソースオブジェクトを削除したあとに，Docker イメージを削除してください。

■コマンドの実行例

```
docker rmi -f hadb-init:$VER
docker rmi -f hadb-server:$VER
docker rmi -f hadb-client:$VER
```

$VER は，HADB サーバのバージョンとリリース番号です。

付録

付録 A　HADB サーバのディレクトリの構成

ここでは，HADB サーバのサーバディレクトリ（インストール時），サーバディレクトリ（運用時），および DB ディレクトリの構成について説明します。

付録 A.1　サーバディレクトリ（インストール時）の構成

HADB サーバをインストールしたときのサーバディレクトリの構成を次の表に示します。

表に示すディレクトリおよびファイルが作成されるタイミングは，HADB サーバのインストール時です。また，表に示すディレクトリおよびファイルが削除されるタイミングは，HADB サーバのアンインストール時です。

表 A-1　サーバディレクトリ（インストール時）の構成

項番	ディレクトリ名またはファイル名	説明
1	$INSTDIR/bin	各種コマンドを格納するディレクトリ
2	$INSTDIR/bin/adbarchivechunk	アーカイブチャンクコマンド
3	$INSTDIR/bin/adbaudittrail	監査証跡機能の管理コマンド
4	$INSTDIR/bin/adbcancel	キャンセルコマンド
5	$INSTDIR/bin/adbchgchunkcomment	チャンクのコメントの設定・変更・削除コマンド
6	$INSTDIR/bin/adbchgchunkstatus	チャンクの状態変更コマンド
7	$INSTDIR/bin/adbchgnodetype	HADB サーバのノード種別変更コマンド
8	$INSTDIR/bin/adbchgsqltrc	SQL トレース情報の出力開始・停止コマンド
9	$INSTDIR/bin/adbchgsrvmode	HADB サーバの稼働モード変更コマンド
10	$INSTDIR/bin/adbclientdefmang	クライアント定義の集中管理コマンド
11	$INSTDIR/bin/adbcolumnize	更新行のカラム化機能の管理コマンド
12	$INSTDIR/bin/adbconvertaudittrailfile	監査証跡ファイルの変換コマンド
13	$INSTDIR/bin/adbdbstatus	データベース状態解析コマンド
14	$INSTDIR/bin/adbexport	データエクスポートコマンド
15	$INSTDIR/bin/adbgetcst	コスト情報収集コマンド
16	$INSTDIR/bin/adbidxrebuild	インデクス再作成コマンド
17	$INSTDIR/bin/adbimport	データインポートコマンド
18	$INSTDIR/bin/adbinfoget	トラブルシュート情報収集コマンド
19	$INSTDIR/bin/adbinfosweep	トラブルシュート情報削除コマンド

項番	ディレクトリ名またはファイル名	説明
20	$INSTDIR/bin/adbinit	データベース初期設定コマンド
21	$INSTDIR/bin/adbls	HADB サーバの状態表示コマンド
22	$INSTDIR/bin/adbmergechunk	マージチャンクコマンド
23	$INSTDIR/bin/adbmodarea	DB エリア追加・変更コマンド
24	$INSTDIR/bin/adbmodbuff	バッファの変更コマンド
25	$INSTDIR/bin/adbmonitor	サーバ監視コマンド
26	$INSTDIR/bin/adbreorgsystemdata	システム表の再編成コマンド
27	$INSTDIR/bin/adbshmdump	共有メモリダンプコマンド
28	$INSTDIR/bin/adbsrvd	サーバデーモン
29	$INSTDIR/bin/adbstart	HADB サーバの起動コマンド
30	$INSTDIR/bin/adbstat	統計解析コマンド
31	$INSTDIR/bin/adbstop	HADB サーバの停止コマンド
32	$INSTDIR/bin/adbsyndict	同義語辞書管理コマンド
33	$INSTDIR/bin/adbsystoru	ITRU 用トラブルシュート情報の取得コマンド
34	$INSTDIR/bin/adbtdmpget	システムで使用
35	$INSTDIR/bin/adbtdmpview	
36	$INSTDIR/bin/adbterrno	
37	$INSTDIR/bin/adbtgetdef	
38	$INSTDIR/bin/adbtgetsqltrc	
39	$INSTDIR/bin/adbthdlinfo	
40	$INSTDIR/bin/adbtmsglog	
41	$INSTDIR/bin/adbtstat	
42	$INSTDIR/bin/adbtsyslog	
43	$INSTDIR/bin/adbtmk2	
44	$INSTDIR/bin/adbunarchivechunk	アンアーカイブチャンクコマンド
45	$INSTDIR/bin/linaphx64	システムで使用
46	$INSTDIR/bin/patrol_sby_exe	HA モニタ用セカンダリ監視コマンド（開始）
47	$INSTDIR/bin/patrol_sby_term	HA モニタ用セカンダリ監視コマンド（停止）
48	$INSTDIR/bin/stop_sby	HA モニタ用セカンダリ停止コマンド
49	$INSTDIR/client	HADB クライアントで使用するコマンドおよびライブラリを格納するディレクトリ

項番	ディレクトリ名またはファイル名	説明
50	$INSTDIR/client/bin	HADB クライアントで使用する各種コマンドを格納するディレクトリ
51	$INSTDIR/client/bin/adbsql	SQL 実行コマンド
52	$INSTDIR/client/lib	HADB クライアントで使用する各種ライブラリを格納するディレクトリ
53	$INSTDIR/client/lib/libadbclt.so	クライアントライブラリ
54	$INSTDIR/client/lib/adbjdbc8.jar	JDBC ドライバ（JRE8 版）
55	$INSTDIR/conf	定義ファイルを格納するディレクトリ
56	$INSTDIR/include	ユーザ提供ヘッダファイルを格納するディレクトリ
57	$INSTDIR/include/adbcli.h	ユーザ提供ヘッダファイル
58	$INSTDIR/include/adbtypes.h	
59	$INSTDIR/include/adbcnv.h	
60	$INSTDIR/include/adbodb.h	
61	$INSTDIR/lib	各種ライブラリを格納するディレクトリ
62	$INSTDIR/lib/libadb.so	データベースアクセス制御
63	$INSTDIR/lib/libadbrpcs.so	通信ライブラリ
64	$INSTDIR/lib/libadbsort.so	ソートライブラリ
65	$INSTDIR/lib/libtapa.so	システムで使用
66	$INSTDIR/lib/adbmsg.cat	メッセージカタログファイル
67	$INSTDIR/lib/sysdef	定義解析情報ファイルを格納するディレクトリ
68	$INSTDIR/lib/sysdef/adb.def	定義解析情報ファイル
69	$INSTDIR/lib/sysdef/adbarchivechunk.def	
70	$INSTDIR/lib/sysdef/adbclientdefmang.def	
71	$INSTDIR/lib/sysdef/adbclt.def	
72	$INSTDIR/lib/sysdef/adbexport.def	
73	$INSTDIR/lib/sysdef/adbgetcst.def	
74	$INSTDIR/lib/sysdef/adbidxrebuild.def	
75	$INSTDIR/lib/sysdef/adbimport.def	
76	$INSTDIR/lib/sysdef/adbimport_clm.def	
77	$INSTDIR/lib/sysdef/adbimport_clm_csv.def	
78	$INSTDIR/lib/sysdef/adbinit.def	

項番	ディレクトリ名またはファイル名	説明
79	$INSTDIR/lib/sysdef/adbmergechunk.def	
80	$INSTDIR/lib/sysdef/adbmodarea.def	
81	$INSTDIR/lib/sysdef/adbmodbuff.def	
82	$INSTDIR/lib/sysdef/adbreorgsystemdata.def	
83	$INSTDIR/lib/sysdef/adbunarchivechunk.def	
84	$INSTDIR/sample	サンプル AP および定義ファイルのひな形を格納するディレクトリ
85	$INSTDIR/sample/cli_sample1.c	サンプル AP（CLI 関数用）
86	$INSTDIR/sample/create_sampledb.sh	SAMPLE 表を作成するシェルスクリプト
87	$INSTDIR/sample/reorg_column_sample.sh	カラムストア表を再編成するシェルスクリプト
88	$INSTDIR/sample/SAMPLE.txt	SAMPLE 表に格納するデータ
89	$INSTDIR/sample/SAMPLE_table.sql	SAMPLE 表を定義する定義系 SQL を記述したファイル
90	$INSTDIR/sample/Sample1.java	サンプル AP（JDBC 用）
91	$INSTDIR/sample/odbc_sample1.c	サンプル AP（ODBC 用）
92	$INSTDIR/sample/conf	定義ファイルのひな形を格納するディレクトリ
93	$INSTDIR/sample/conf/adbarchivechunk.opt	アーカイブチャンクオプションのひな形ファイル
94	$INSTDIR/sample/conf/adbclientdefmang.def	クライアント管理定義のひな形ファイル
95	$INSTDIR/sample/conf/adbexport.opt	エクスポートオプションのひな形ファイル
96	$INSTDIR/sample/conf/adbgetcst.opt	コスト情報収集オプションのひな形ファイル
97	$INSTDIR/sample/conf/adbidxrebuild.opt	インデクス再作成オプションのひな形ファイル
98	$INSTDIR/sample/conf/adbimport.opt	インポートオプションのひな形ファイル
99	$INSTDIR/sample/conf/adbinit.opt	初期設定オプションのひな形ファイル
100	$INSTDIR/sample/conf/adbmergechunk.opt	マージチャンクオプションのひな形ファイル
101	$INSTDIR/sample/conf/adbmodarea.opt	DB エリア追加・変更オプションのひな形ファイル
102	$INSTDIR/sample/conf/adbmodbuff.opt	バッファ変更オプションのひな形ファイル
103	$INSTDIR/sample/conf/adbunarchivechunk.opt	アンアーカイブチャンクオプションのひな形ファイル
104	$INSTDIR/sample/conf/client.def	クライアント定義のひな形ファイル
105	$INSTDIR/sample/conf/server.def	サーバ定義のひな形ファイル
106	$INSTDIR/sample/docker	Dockerfile と HADB サーバ起動スクリプトを格納するディレクトリ
107	$INSTDIR/sample/docker/docker-entrypoint.sh	Docker コンテナ用 HADB サーバ起動スクリプト

項番	ディレクトリ名またはファイル名	説明
108	$INSTDIR/sample/docker/Dockerfile_hadb-client	HADB クライアント利用コンテナのDockerfile
109	$INSTDIR/sample/docker/Dockerfile_hadb-init	データベース初期化コンテナのDockerfile
110	$INSTDIR/sample/docker/Dockerfile_hadb-server	HADB サーバ起動コンテナのDockerfile
111	$INSTDIR/sample/k8s	Kubernetes 用のマニフェストファイルを格納するディレクトリ
112	$INSTDIR/sample/k8s/hadb-k8s-manifest.yaml	Kubernetes 用のマニフェストファイル（yaml ファイル）
113	$INSTDIR/sample/scripts	HA モニタ用コマンドのサンプルファイルを格納するディレクトリ
114	$INSTDIR/sample/scripts/multinode.env	HA モニタ用コマンドの環境変数定義ファイル（マルチノード機能）
115	$INSTDIR/sample/scripts/multinode_user.sh	HA モニタ用ユーザコマンドのサンプルシェルファイル（マルチノード機能）
116	$INSTDIR/sample/scripts/multinode_act.sh	HA モニタ用起動コマンドのサンプルシェルファイル（マルチノード機能）
117	$INSTDIR/sample/scripts/multinode_term.sh	HA モニタ用停止コマンドのサンプルシェルファイル（マルチノード機能）（系のリセット用）
118	$INSTDIR/sample/scripts/multinode_term_scsi.sh	HA モニタ用停止コマンドのサンプルシェルファイル（マルチノード機能）（共有ディスクの SCSI リザーブ用）
119	$INSTDIR/sample/scripts/multinode_patrol.sh	HA モニタ用監視コマンドのサンプルシェルファイル（マルチノード機能）
120	$INSTDIR/sample/scripts/coldstandby.env	HA モニタ用コマンドの環境変数定義ファイル（コールドスタンバイ構成）
121	$INSTDIR/sample/scripts/coldstandby_act.sh	HA モニタ用起動コマンドのサンプルシェルファイル（コールドスタンバイ構成）
122	$INSTDIR/sample/scripts/coldstandby_term.sh	HA モニタ用停止コマンドのサンプルシェルファイル（コールドスタンバイ構成）
123	$INSTDIR/sample/scripts/coldstandby_patrol.sh	HA モニタ用監視コマンドのサンプルシェルファイル（コールドスタンバイ構成）
124	$INSTDIR/sample/jp1audit	JP1/Audit 用定義ファイルのサンプルファイルを格納するディレクトリ
125	$INSTDIR/sample/jp1audit/admjevlog_HADB.conf	JP1/Audit 用動作定義ファイルのサンプルファイル
126	$INSTDIR/sample/jp1audit/HADB.conf	JP1/Audit 用製品定義ファイルのサンプルファイル
127	$INSTDIR/spool	実行結果ログを格納するディレクトリ
128	$INSTDIR/adbinstsv.log	HADB サーバをインストールしたときの実行結果ログファイル

付録 A.2　サーバディレクトリ（運用時）の構成

HADB サーバの運用で使用するサーバディレクトリの構成を次の表に示します。

表 A-2　サーバディレクトリ（運用時）の構成

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
1	$ADBDIR/bin	各種コマンドを格納するディレクトリ	HADB サーバのインストール時	HADB サーバのアンインストール時
2	$ADBDIR/bin/adbarchivechunk	アーカイブチャンクコマンド		
3	$ADBDIR/bin/adbaudittrail	監査証跡機能の管理コマンド		
4	$ADBDIR/bin/adbcancel	キャンセルコマンド		
5	$ADBDIR/bin/adbchgchunkcomment	チャンクのコメントの設定・変更・削除コマンド		
6	$ADBDIR/bin/adbchgchunkstatus	チャンクの状態変更コマンド		
7	$ADBDIR/bin/adbchgnodetype	HADB サーバのノード種別変更コマンド		
8	$ADBDIR/bin/adbchgsqltrc	SQL トレース情報の出力開始・停止コマンド		
9	$ADBDIR/bin/adbchgsrvmode	HADB サーバの稼働モード変更コマンド		
10	$ADBDIR/bin/adbclientdefmang	クライアント定義の集中管理コマンド		
11	$ADBDIR/bin/adbcolumnize	更新行のカラム化機能の管理コマンド		
12	$ADBDIR/bin/adbconvertaudittrailfile	監査証跡ファイルの変換コマンド		
13	$ADBDIR/bin/adbdbstatus	データベース状態解析コマンド		
14	$ADBDIR/bin/adbexport	データエクスポートコマンド		
15	$ADBDIR/bin/adbgetcst	コスト情報収集コマンド		
16	$ADBDIR/bin/adbidxrebuild	インデクス再作成コマンド		
17	$ADBDIR/bin/adbimport	データインポートコマンド		

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
18	$ADBDIR/bin/adbinfoget	トラブルシュート情報収集コマンド		
19	$ADBDIR/bin/adbinfosweep	トラブルシュート情報削除コマンド		
20	$ADBDIR/bin/adbinit	データベース初期設定コマンド		
21	$ADBDIR/bin/adbls	HADB サーバの状態表示コマンド		
22	$ADBDIR/bin/adbmergechunk	マージチャンクコマンド		
23	$ADBDIR/bin/adbmodarea	DB エリア追加・変更コマンド		
24	$ADBDIR/bin/adbmodbuff	バッファの変更コマンド		
25	$ADBDIR/bin/adbmonitor	サーバ監視コマンド		
26	$ADBDIR/bin/adbreorgsystemdata	システム表の再編成コマンド		
27	$ADBDIR/bin/adbshmdump	共有メモリダンプコマンド		
28	$ADBDIR/bin/adbsrvd	サーバデーモン		
29	$ADBDIR/bin/adbstart	HADB サーバの起動コマンド		
30	$ADBDIR/bin/adbstat	統計解析コマンド		
31	$ADBDIR/bin/adbstop	HADB サーバの停止コマンド		
32	$ADBDIR/bin/adbsyndict	同義語辞書管理コマンド		
33	$ADBDIR/bin/adbsystoru	ITRU 用トラブルシュート情報の取得コマンド		
34	$ADBDIR/bin/adbtdmpget	システムで使用		
35	$ADBDIR/bin/adbtdmpview			
36	$ADBDIR/bin/adbterrno			
37	$ADBDIR/bin/adbtgetdef			
38	$ADBDIR/bin/adbtgetsqltrc			
39	$ADBDIR/bin/adbthdlinfo			
40	$ADBDIR/bin/adbtmsglog			
41	$ADBDIR/bin/adbtstat			
42	$ADBDIR/bin/adbtsyslog			

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
43	$ADBDIR/bin/adbtmk2			
44	$ADBDIR/bin/adbunarchivechunk	アンアーカイブチャンクコマンド		
45	$ADBDIR/bin/linaphx64	システムで使用		
46	$ADBDIR/bin/patrol_sby_exe	HA モニタ用セカンダリ監視コマンド（開始）		
47	$ADBDIR/bin/patrol_sby_term	HA モニタ用セカンダリ監視コマンド（停止）		
48	$ADBDIR/bin/stop_sby	HA モニタ用セカンダリ停止コマンド		
49	$ADBDIR/client	HADB クライアントで使用するコマンドおよびライブラリを格納するディレクトリ		
50	$ADBDIR/client/bin	HADB クライアントで使用する各種コマンドを格納するディレクトリ		
51	$ADBDIR/client/bin/adbsql	SQL 実行コマンド		
52	$ADBDIR/client/lib	HADB クライアントで使用する各種ライブラリを格納するディレクトリ		
53	$ADBDIR/client/lib/libadbclt.so	クライアントライブラリ		
54	$ADBDIR/client/lib/adbjdbc8.jar	JDBC ドライバ（JRE8 版）		
55	$ADBDIR/conf	定義ファイルを格納するディレクトリ		
56	$ADBDIR/include	ユーザ提供ヘッダファイルを格納するディレクトリ		
57	$ADBDIR/include/adbcli.h	ユーザ提供ヘッダファイル		
58	$ADBDIR/include/adbtypes.h			
59	$ADBDIR/include/adbcnv.h			
60	$ADBDIR/include/adbodb.h			
61	$ADBDIR/lib	各種ライブラリを格納するディレクトリ		
62	$ADBDIR/lib/libadb.so	データベースアクセス制御		

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
63	$ADBDIR/lib/libadbrpcs.so	通信ライブラリ		
64	$ADBDIR/lib/libadbsort.so	ソートライブラリ		
65	$ADBDIR/lib/libtapa.so	システムで使用		
66	$ADBDIR/lib/adbmsg.cat	メッセージカタログファイル		
67	$ADBDIR/lib/sysdef	定義解析情報ファイルを格納するディレクトリ		
68	$ADBDIR/lib/sysdef/adb.def	定義解析情報ファイル		
69	$ADBDIR/lib/sysdef/adbarchivechunk.def			
70	$ADBDIR/lib/sysdef/adbclientdefmang.def			
71	$ADBDIR/lib/sysdef/adbclt.def			
72	$ADBDIR/lib/sysdef/adbexport.def			
73	$ADBDIR/lib/sysdef/adbgetcst.def			
74	$ADBDIR/lib/sysdef/adbidxrebuild.def			
75	$ADBDIR/lib/sysdef/adbimport.def			
76	$ADBDIR/lib/sysdef/adbimport_clm.def			
77	$ADBDIR/lib/sysdef/adbimport_clm_csv.def			
78	$ADBDIR/lib/sysdef/adbinit.def			
79	$ADBDIR/lib/sysdef/adbmergechunk.def			
80	$ADBDIR/lib/sysdef/adbmodarea.def			
81	$ADBDIR/lib/sysdef/adbmodbuff.def			
82	$ADBDIR/lib/sysdef/adbreorgsystemdata.def			
83	$ADBDIR/lib/sysdef/adbunarchivechunk.def			
84	$ADBDIR/sample	サンプル AP および定義ファイルのひな形を格納するディレクトリ		

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
85	$ADBDIR/sample/cli_sample1.c	サンプル AP（CLI 関数用）		
86	$ADBDIR/sample/create_sampledb.sh	SAMPLE 表を作成するシェルスクリプト		
87	$ADBDIR/sample/reorg_column_sample.sh	カラムストア表を再編成するシェルスクリプト		
88	$ADBDIR/sample/SAMPLE.txt	SAMPLE 表に格納するデータ		
89	$ADBDIR/sample/SAMPLE_table.sql	SAMPLE 表を定義する定義系 SQL を記述したファイル		
90	$ADBDIR/sample/Sample1.java	サンプル AP（JDBC 用）		
91	$ADBDIR/sample/odbc_sample1.c	サンプル AP（ODBC 用）		
92	$ADBDIR/sample/conf	定義ファイルのひな形を格納するディレクトリ		
93	$ADBDIR/sample/conf/adbarchivechunk.opt	アーカイブチャンクオプションのひな形ファイル		
94	$ADBDIR/sample/conf/adbclientdefmang.def	クライアント管理定義のひな形ファイル		
95	$ADBDIR/sample/conf/adbexport.opt	エクスポートオプションのひな形ファイル		
96	$ADBDIR/sample/conf/adbgetcst.opt	コスト情報収集オプションのひな形ファイル		
97	$ADBDIR/sample/conf/adbidxrebuild.opt	インデクス再作成オプションのひな形ファイル		
98	$ADBDIR/sample/conf/adbimport.opt	インポートオプションのひな形ファイル		
99	$ADBDIR/sample/conf/adbinit.opt	初期設定オプションのひな形ファイル		
100	$ADBDIR/sample/conf/adbmergechunk.opt	マージチャンクオプションのひな形ファイル		
101	$ADBDIR/sample/conf/adbmodarea.opt	DB エリア追加・変更オプションのひな形ファイル		
102	$ADBDIR/sample/conf/adbmodbuff.opt	バッファ変更オプションのひな形ファイル		

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
103	$ADBDIR/sample/conf/adbunarchivechunk.opt	アンアーカイブチャンクオプションのひな形ファイル		
104	$ADBDIR/sample/conf/client.def	クライアント定義のひな形ファイル		
105	$ADBDIR/sample/conf/server.def	サーバ定義のひな形ファイル		
106	$ADBDIR/sample/docker	Dockerfile と HADB サーバ起動スクリプトを格納するディレクトリ		
107	$ADBDIR/sample/docker/docker-entrypoint.sh	Docker コンテナ用 HADB サーバ起動スクリプト		
108	$ADBDIR/sample/docker/Dockerfile_hadb-client	HADB クライアント利用コンテナのDockerfile		
109	$ADBDIR/sample/docker/Dockerfile_hadb-init	データベース初期化コンテナのDockerfile		
110	$ADBDIR/sample/docker/Dockerfile_hadb-server	HADB サーバ起動コンテナのDockerfile		
111	$ADBDIR/sample/k8s	Kubernetes 用のマニフェストファイルを格納するディレクトリ		
112	$ADBDIR/sample/k8s/hadb-k8s-manifest.yaml	Kubernetes 用のマニフェストファイル（yaml ファイル）		
113	$ADBDIR/sample/scripts	HA モニタ用コマンドのサンプルファイルを格納するディレクトリ		
114	$ADBDIR/sample/scripts/multinode.env	HA モニタ用コマンドの環境変数定義ファイル（マルチノード機能）		
115	$ADBDIR/sample/scripts/multinode_user.sh	HA モニタ用ユーザコマンドのサンプルシェルファイル（マルチノード機能）		
116	$ADBDIR/sample/scripts/multinode_act.sh	HA モニタ用起動コマンドのサンプルシェルファイル（マルチノード機能）		
117	$ADBDIR/sample/scripts/multinode_term.sh	HA モニタ用停止コマンドのサンプルシェルファ		

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
		イル（マルチノード機能）（系のリセット用）		
118	$ADBDIR/sample/scripts/multinode_term_scsi.sh	HA モニタ用停止コマンドのサンプルシェルファイル（マルチノード機能）（共有ディスクの SCSI リザーブ用）		
119	$ADBDIR/sample/scripts/multinode_patrol.sh	HA モニタ用監視コマンドのサンプルシェルファイル（マルチノード機能）		
120	$ADBDIR/sample/scripts/coldstandby.env	HA モニタ用コマンドの環境変数定義ファイル（コールドスタンバイ構成）		
121	$ADBDIR/sample/scripts/coldstandby_act.sh	HA モニタ用起動コマンドのサンプルシェルファイル（コールドスタンバイ構成）		
122	$ADBDIR/sample/scripts/coldstandby_term.sh	HA モニタ用停止コマンドのサンプルシェルファイル（コールドスタンバイ構成）		
123	$ADBDIR/sample/scripts/coldstandby_patrol.sh	HA モニタ用監視コマンドのサンプルシェルファイル（コールドスタンバイ構成）		
124	$ADBDIR/sample/jp1audit	JP1/Audit 用定義ファイルのサンプルファイルを格納するディレクトリ		
125	$ADBDIR/sample/jp1audit/admjevlog_HADB.conf	JP1/Audit 用動作定義ファイルのサンプルファイル		
126	$ADBDIR/sample/jp1audit/HADB.conf	JP1/Audit 用製品定義ファイルのサンプルファイル		
127	$ADBDIR/spool	実行結果ログを格納するディレクトリ		
128	$ADBDIR/spool/adbdump*YYYYMMDDhhmmss*.*サーバプロセスのプロセス ID*※1	HADB ダンプファイル	HADB サーバの開始時	• HADB サーバの正常終了時 • adbinfosweep コマンド実行時

項番	ディレクトリ名またはファイル名	説明	作成されるタイミング	削除されるタイミング
129	$ADBDIR/spool/adbdumperror*YYYYMMDDhhmmssSSSSSS_TTTTTTTTTTTTTTTT.サーバプロセスのプロセス ID* ※1, ※2		KFAA60007-E メッセージが出力されるエラー発生時	adbinfosweep コマンド実行時
130	$ADBDIR/spool/adbmessage*XX*.log※3	サーバメッセージログファイル	HADB サーバの開始時	
131	$ADBDIR/spool/adbsqltrc*XX*.log※4	SQL トレースファイル		HADB サーバのアンインストール時
132	$ADBDIR/spool/adbstatlog*XX*※3, ※5	統計ログファイル		
133	$ADBDIR/spool/adbsrvd.pid	HADB サーバプロセスのプロセス ID 格納ファイル		• HADB サーバの正常終了時 • adbinfosweep コマンド実行時
134	$ADBDIR/spool/.defrslt	サーバ定義格納ファイル		adbinfosweep コマンド実行時
135	$ADBDIR/spool/.adbmessage	メッセージログファイル番号管理ファイル		
136	$ADBDIR/spool/.adbshmid	共有メモリ ID 格納ファイル		
137	$ADBDIR/spool/.adbsemid	セマフォ集合 ID 格納ファイル		
138	$ADBDIR/spool/.adbsqltrc	SQL トレースファイルのファイル番号を管理するファイル		HADB サーバのアンインストール時
139	$ADBDIR/spool/.adbstatlog※5	統計ログファイルの管理ファイル		
140	$ADBDIR/spool/adbcolumnize01	更新行のカラム化機能で使用するファイル		
141	$ADBDIR/spool/.adbcolumnize			
142	$ADBDIR/spool/adboptlog*XX*※6	アクセスパス探索情報ログファイル		
143	$ADBDIR/spool/.adboptlog	アクセスパス探索情報ログの管理ファイル		
144	$ADBDIR/spool/.hadb_map	システムで使用		
145	$ADBDIR/spool/.hadb_ulimit			
146	$ADBDIR/adbinstsv.log	HADB サーバをインストールしたときの実行結果ログファイル	HADB サーバのインストール時	

注※1

ファイル名の *YYYYMMDDhhmmss* は，ファイルを生成した時刻になります。

注※2

ファイル名の *SSSSSS* は，HADB ダンプファイルを生成した時刻のマイクロ秒になります。また，*TTTTTTTTTTTTTTTTTTT* は，HADB サーバの内部矛盾エラーを検知したリアルスレッドのスレッド ID になります。

注※3

ファイル名の *XX* は通番であり，01～04 のどれかになります。

注※4

ファイル名の *XX* は通番であり，01～08 のどれかになります。

注※5

サーバ定義adb_sta_log_path オペランドを指定している場合は，サーバ定義adb_sta_log_path オペランドで指定したディレクトリ下に出力されます。

サーバ定義adb_sta_log_path オペランドについては，「7.2.7　統計情報に関するオペランド（set 形式）」のadb_sta_log_path オペランドを参照してください。

注※6

ファイル名の *XX* は通番であり，01～02 のどちらかになります。

付録 A.3　DB ディレクトリの構成

adbinit コマンドを使用してデータベースの初期設定をしたときに作成される DB ディレクトリの構成を次の表に示します。

表 A-3　DB ディレクトリの構成

項番	ディレクトリ名およびファイル名	説明	作成されるタイミング	削除されるタイミング
1	$DBDIR/ADBMST	マスタディレクトリ用 DB エリアファイル	adbinit コマンド実行時	adbinit コマンド実行時（既存のファイルが存在する場合）
2	$DBDIR/ADBDIC	ディクショナリ用 DB エリアファイル		
3	$DBDIR/ADBSTBL	システム表用 DB エリアファイル		
4	$DBDIR/ADBWRK	作業表用 DB エリアファイル		
5	$DBDIR/DBAREA※1	データ用 DB エリアファイル	• adbinit コマンド実行時 • adbmodarea コマンド実行時（データ用 DB エリアを追加する場合）	• adbinit コマンド実行時（既存のファイルが存在する場合）

項番	ディレクトリ名およびファイル名	説明	作成されるタイミング	削除されるタイミング
			• adbmodarea コマンド実行時（データ用 DB エリアを拡張する場合）	• adbmodarea コマンド実行時（データ用 DB エリアを削除する場合）
6	$DBDIR/ADBSYS	システムディレクトリ	adbinit コマンド実行時	adbinit コマンド実行時（既存のディレクトリが存在する場合）
7	$DBDIR/ADBSYS/ADBSLG	システムログファイル用ディレクトリ		
8	$DBDIR/ADBSYS/ADBSTS	ステータスファイル用ディレクトリ		
9	$DBDIR/ADBSYS/ADBUTL	コマンドステータスファイル用ディレクトリ		
10	$DBDIR/ADBWORK	作業用ディレクトリ		
11	$DBDIR/SPOOL	障害情報（コアファイル）の出力先ディレクトリ※2		
12	$DBDIR/SPOOL/core.*サーバプロセスのプロセス ID*	障害情報（コアファイル）	HADB サーバの異常終了時	adbinfosweep コマンド実行時

注※1

adbinit コマンドまたはadbmodarea コマンドで，ユーザが指定した名称になります。

注※2

サーバ定義adb_core_path オペランドを指定した場合は，$DBDIR/SPOOL ディレクトリ下ではなく，adb_core_path オペランドに指定したディレクトリ下に，障害情報（コアファイル）が出力されます。

なお，コアファイルの出力が systemd-coredump によって制御されている場合，コアファイルは/var/lib/systemd/coredump ディレクトリ下に出力されます。このとき，$DBDIR/SPOOL ディレクトリは，トラブルシュートの情報収集時にコアファイル展開用の一時ディレクトリとして使用されます。サーバ定義adb_core_path オペランドが指定されている場合は，adb_core_path オペランドに指定したディレクトリがコアファイル展開用の一時ディレクトリになります。

付録 B　ディクショナリ表

ここでは，ディクショナリ表に格納されている情報，ディクショナリ表に定義されている B-tree インデクス，およびディクショナリ表の検索方法について説明します。

付録 B.1　ディクショナリ表の概要

ディクショナリ表には，表およびインデクスの定義情報や，DB エリアの情報が格納されています。

ディクショナリ表は，実表とビュー表の 2 種類があります。

- ディクショナリ表（実表）

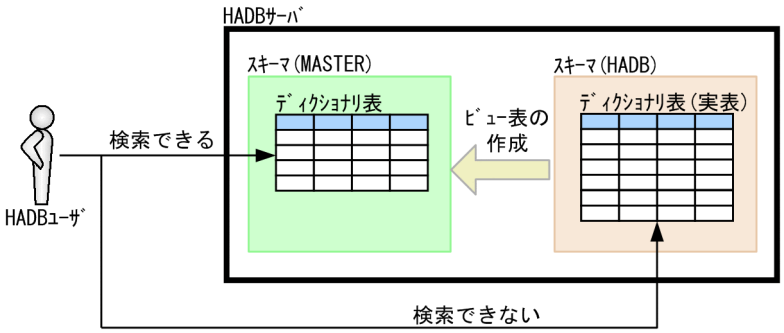
  このマニュアルでは，ディクショナリ表の実表を，**ディクショナリ表（実表）**と表記しています。ディクショナリ表（実表）のスキーマ名は，HADB になります。

- ディクショナリ表（ビュー表）

  このマニュアルでは，ディクショナリ表のビュー表を，**ディクショナリ表**と表記しています。ディクショナリ表のスキーマ名は，MASTER になります。

ディクショナリ表（実表）とディクショナリ表の関係を次の図に示します。

図 B-1　ディクショナリ表（実表）とディクショナリ表の関係



[説明]

表およびインデクスの定義情報や，DB エリアの情報は，ディクショナリ表（実表）に格納されます。ディクショナリ表（実表）から検索用に作成されるビュー表が，ディクショナリ表になります。

HADB ユーザは，ディクショナリ表を検索できます。しかし，ディクショナリ表（実表）は検索できません。

HADB ユーザがSELECT 文でディクショナリ表を検索すると，表およびインデクスの定義情報や，DB エリアの情報を確認できます。ディクショナリ表を検索する場合は，スキーマ名にMASTER を指定してください。

(1)　ディクショナリ表の一覧

ディクショナリ表の一覧を次の表に示します。

表 B-1　ディクショナリ表の一覧

項番	ディクショナリ表の名称	格納されている情報	ディクショナリ表に対応するディクショナリ表（実表）の表 ID	ビュー定義に含まれる副問合せの数※	外への参照の有無
1	SQL_TABLES	実表，ビュー表，ディクショナリ表（実表），システム表（実表），ディクショナリ表，およびシステム表の表定義情報が格納されています。1 行に 1 表分の情報が格納されています。	0x00020001	1	○
2	SQL_COLUMNS	実表，ビュー表，ディクショナリ表（実表），システム表（実表），ディクショナリ表，およびシステム表の列の定義情報が格納されています。1 行に 1 列分の情報が格納されています。	0x00020002	1	○
3	SQL_DIV_TABLE	実表，ディクショナリ表（実表），およびシステム表（実表）を格納している DB エリアなどの情報が格納されています。1 行に 1 表分の情報が格納されています。	0x00020004	1	○
4	SQL_INDEXES	実表のインデクスの定義情報が格納されています。また，ディクショナリ表（実表），およびシステム表（実表）の B-tree インデクスの定義情報も格納されています。1 行に 1 インデクス分の情報が格納されています。	0x00020003	1	○
5	SQL_DIV_INDEX	実表のインデクスが格納されている DB エリアなどの情報が格納されています。また，ディクショナリ表（実表），およびシステム表（実表）の B-tree インデクスが格納されている DB エリアなどの情報も格納されています。1 行に 1 インデクス分の情報が格納されています。	0x00020005	1	○
6	SQL_DBAREAS	DB エリアの定義情報が格納されています。1 行に 1DB エリア分の情報が格納されています。	0x00020006	0	×

項番	ディクショナリ表の名称	格納されている情報	ディクショナリ表に対応するディクショナリ表（実表）の表 ID	ビュー定義に含まれる副問合せの数※	外への参照の有無
7	SQL_SCHEMATA	スキーマに関する定義情報が格納されています。1 行に 1 スキーマ分の情報が格納されています。	0x00020007	1	○
8	SQL_VIEWS	ビュー表の定義情報が格納されています。1 行に 1 ビュー表分が格納されています。	0x00020008	1	○
9	SQL_VIEW_TABLE_USAGE	ビュー表の基表の情報が格納されています。1 行に 1 基表分の情報が格納されています。	0x00020009	1	○
10	SQL_VIEW_OBJECT	システムが使用する情報が格納されています。 なお，ディクショナリ表は存在しません。ディクショナリ表（実表）だけ存在します。	0x0002000A	－	－
11	SQL_DEFINE_SOURCE	SQL 文の情報が格納されています。1 行に 1SQL テキスト分の情報が格納されています。	0x0002000B	2	○
12	SQL_DEFINE_ENVIRONMENT	ビュー定義時およびビュー定義変更時の環境情報が格納されています。1 行に 1 つのビュー表の環境情報が格納されています。	0x0002000C	2	○
13	SQL_USERS	HADB ユーザの情報が格納されています。1 行に 1HADB ユーザの情報が格納されています。	0x0002000D	1	×
14	SQL_TABLE_CONSTRAINTS	実表の制約に関する情報が格納されています。1 行に 1 制約分の情報が格納されています。	0x0002000E	1	○
15	SQL_INDEX_COLINF	インデクス構成列に関する情報が格納されています。1 行にインデクス構成列 1 列分の情報が格納されています。	0x0002000F	1	○
16	SQL_KEY_COLUMN_USAGE	主キーと外部キーの構成列に関する情報が格納されています。1 行に主キーと外部キー	0x00020010	1	○

項番	ディクショナリ表の名称	格納されている情報	ディクショナリ表に対応するディクショナリ表（実表）の表 ID	ビュー定義に含まれる副問合せの数※	外への参照の有無
		の構成列 1 列分の情報が格納されています。			
17	SQL_REFERENTIAL_CONSTRAINTS	参照制約に関する情報が格納されています。1 行に 1 参照制約分の情報が格納されています。	0x00020011	1	○
18	SQL_TABLE_PRIVILEGES	表に対するアクセス権限に関する情報が格納されています。1 行に，ある権限付与者が与えた 1 認可識別子の 1 表分の情報が格納されています。	0x00020012	1	○
19	SQL_AUDITS	CREATE AUDIT 文で定義した監査対象定義に関する情報が格納されています。1 行に，1 監査対象定義の情報が格納されています。	0x00020013	0	×

（凡例）

○：ディクショナリ表のビュー定義に，外への参照を行う副問合せを含みます。そのため，該当するディクショナリ表は，マルチ集合値式の副問合せに指定できません。

×：ディクショナリ表のビュー定義に，外への参照を行う副問合せを含みません。

－：該当しません。

注※

該当するディクショナリ表を SQL 文の副問合せに指定した場合に，副問合せの入れ子の数として加算する必要がある数を示しています。

メモ

- ディクショナリ表中には，ディクショナリ表の表定義情報およびインデクス定義情報も格納されています。
- ディクショナリ表のデータ型は，HADB の SQL の仕様に従っています。

(2) ディクショナリ表が作成されるタイミング

ディクショナリ表は，次に示すタイミングで自動的に作成されます。

- データベースの初期設定が完了したあとに，初めて HADB サーバを開始したとき
- HADB サーバをバージョンアップしたとき

(3) HADB ユーザが参照できるディクショナリ表の範囲

HADB ユーザがディクショナリ表を参照する際，持っている権限によって参照できる情報が異なります。HADB ユーザが参照できるディクショナリ表の情報を次の表に示します。

なお，ディクショナリ表を基表に指定したビュー表の場合も，持っている権限によって参照できる情報が異なります。そのため，同じビュー表を検索しても，ビュー表を定義した HADB ユーザと，ビュー表に対するアクセス権限を付与された HADB ユーザでは，参照できる情報が異なることがあります。

表 B-2　HADB ユーザが参照できるディクショナリ表の情報

ディクショナリ表の名称	参照対象の情報	ディクショナリ表を参照する HADB ユーザが持っている権限			
		DBA 権限	監査管理権限	アクセス権限	CONNECT 権限
SQL_TABLES	自分	○	○	○	○
	他ユーザ	○	○	△※1	×
	MASTER	○	○	○	○
SQL_COLUMNS	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	○	○	○	○
SQL_DIV_TABLE	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	−	−	−	−
SQL_INDEXES	自分	○	○	○	○
	他ユーザ	○	○	△※1	×
	MASTER	○	○	○	○
SQL_DIV_INDEX	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	−	−	−	−
SQL_DBAREAS	−	○	○	○	○
SQL_SCHEMATA	自分	○	○	○	○
	他ユーザ	○	○	△※1	×
	MASTER	○	○	○	○
SQL_VIEWS	自分	○	○	○	○
	他ユーザ	○	×	△※1，※3	×
	MASTER	○	○	○	○

ディクショナリ表の名称	参照対象の情報	ディクショナリ表を参照する HADB ユーザが持っている権限			
		DBA 権限	監査管理権限	アクセス権限	CONNECT 権限
SQL_VIEW_TABLE_USAGE	自分	○	○	○	○
	他ユーザ	○	○	△※1，※3	×
	MASTER	○	○	○	○
SQL_DEFINE_SOURCE	自分	○	○	○	○
	他ユーザ	○	×	△※1，※3	×
	MASTER	−	−	−	−
SQL_DEFINE_ENVIRONMENT	自分	○	○	○	○
	他ユーザ	○	×	△※1，※3	×
	MASTER	○	○	○	○
SQL_USERS	自分	○	○	−	○
	他ユーザ	○	○	−	×
	MASTER	−	−	−	−
SQL_TABLE_CONSTRAINTS	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	−	−	−	−
SQL_INDEX_COLINF	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	−	−	−	−
SQL_KEY_COLUMN_USAGE	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	−	−	−	−
SQL_REFERENTIAL_CONSTRAINTS	自分	○	○	○	○
	他ユーザ	○	×	△※1	×
	MASTER	−	−	−	−
SQL_TABLE_PRIVILEGES	自分	△※2	△※2	△※2	△※2
	他ユーザ	△※2	×	△※1，※2	×
	MASTER	−	−	−	−
SQL_AUDITS	自分	×	○	×	×
	他ユーザ	×	○	×	×

ディクショナリ表の名称	参照対象の情報	ディクショナリ表を参照する HADB ユーザが持っている権限			
		DBA 権限	監査管理権限	アクセス権限	CONNECT 権限
	MASTER	×	○	×	×

（凡例）

自分：自分が所有するスキーマおよびスキーマオブジェクトに関する情報（SQL_USERS の場合は自分のユーザ情報）

他ユーザ：ほかの HADB ユーザが所有するスキーマおよびスキーマオブジェクトに関する情報（SQL_USERS の場合はほかの HADB ユーザのユーザ情報）

MASTER：ディクショナリ表およびシステム表に関する情報

○：参照できます。

△：条件付きで参照できます。

×：参照できません。

－：該当しません。

注※1

アクセス権限を持っているスキーマオブジェクトに関する情報と，そのスキーマに関する情報については参照できます。

注※2

対象のスキーマオブジェクトがビュー表で，かつそのビュー表が無効化されている場合，無効化されているビュー表に対するアクセス権限についての情報は，SQL_TABLE_PRIVILEGES から検索することはできません。

注※3

対象のスキーマオブジェクトがビュー表で，かつそのビュー表が無効化されている場合，無効化されているビュー表に関する定義情報を参照することはできません。

付録 B.2　SQL_TABLES の内容

SQL_TABLES には，実表，ビュー表，ディクショナリ表（実表），システム表（実表），ディクショナリ表，およびシステム表の表定義情報が格納されています。1 行に 1 表分の情報が格納されています。

SQL_TABLES の内容を次の表に示します。

表 B-3　SQL_TABLES の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子

項番	列名	データ型	格納されている情報
3	TABLE_TYPE	CHAR(1)	表種別 • 'R'：実表 • 'V'：ビュー表
4	TABLE_ID	SMALLINT	表 ID • 0x00020001～：ディクショナリ表（実表） • 0x000200C9～：システム表（実表） • 0x00020191～：HADB ユーザが定義した実表 • 0x00040001～：ディクショナリ表およびシステム表 • 0x00040401～：HADB ユーザが定義したビュー表
5	N_COLS	SMALLINT	表の構成列数
6	N_INDEX	SMALLINT	表に定義されているインデクス数（全インデクスの合計）
7	CREATE_TIME	TIMESTAMP	表定義時のタイムスタンプ
8	N_NOTNULL	SMALLINT	非ナル値制約が定義されている列数
9	FREE_AREA	SMALLINT	表のページ内の未使用領域の比率
10	FIX_TABLE	CHAR(1)	FIX 表かどうか • 'F'：FIX 表 • 'N'：FIX 表ではない表
11	ROW_LENGTH	SMALLINT	FIX 表の行長 FIX 表ではない表，ディクショナリ表（実表），またはシステム表（実表）の場合は，ナル値が格納されます。
12	IS_BRANCH_ALL	CHAR(1)	表オプションBRANCH ALL の指定 • 'Y'：BRANCH ALL の指定ありの表 • 'N'：BRANCH ALL の指定なしの表
13	N_RANGE_INDEX	SMALLINT	表に定義されているレンジインデクスの数
14	IS_CHUNK	CHAR(1)	チャンク指定のCHUNK の指定 • 'Y'：指定あり（マルチチャンク表） • 'N'：指定なし（シングルチャンク表）
15	N_CHUNK_RESERVED	SMALLINT	作成されるチャンク数の最大値 CHUNK の指定値が格納されます。シングルチャンク表，およびビュー表の場合はナル値が格納されます。
16	N_DEFAULT_COLUMN	SMALLINT	DEFAULT 句を指定した列数 実表の定義時に，DEFAULT 句を指定した列の数が格納されます。

項番	列名	データ型	格納されている情報
			なお，ビュー表，ディクショナリ表（実表），システム表（実表），およびDEFAULT句の指定がない実表の場合は，ナル値が格納されます。
17	N_PRIMARY_KEY_COLUMN	SMALLINT	主キーを構成する列数 実表の定義時に，定義した主キーを構成する列の数が格納されます。 なお，ビュー表，ディクショナリ表（実表），システム表（実表），および主キーが定義されていない実表の場合は，ナル値が格納されます。
18	N_FOREIGN_KEY	SMALLINT	外部キーの数 実表に定義した外部キーの数が格納されます。 なお，次に示す表の場合は，ナル値が格納されます。 • ビュー表 • ディクショナリ表（実表） • システム表（実表） • 外部キーが定義されていない実表
19	N_REFERENCING_KEY	SMALLINT	主キーを参照する外部キーの数 この実表に定義された主キーを参照する外部キーの数が格納されます。 なお，次に示す表の場合は，ナル値が格納されます。 • ビュー表 • ディクショナリ表（実表） • システム表（実表） • 主キーを参照する外部キーが1つもない実表 • 主キーが定義されていない実表
20	N_FOREIGN_KEY_COLUMN	SMALLINT	外部キーを構成する列数 実表に定義した外部キーを構成する列数を合計した数が格納されます。 なお，次に示す表の場合は，ナル値が格納されます。 • ビュー表 • ディクショナリ表（実表） • システム表（実表） • 外部キーが定義されていない実表
21	N_TEXT_INDEX	SMALLINT	表に定義されているテキストインデクスの数 次に示す表の場合は，ナル値が格納されます。 • ビュー表 • ディクショナリ表（実表） • システム表（実表） • テキストインデクスが定義されていない表

項番	列名	データ型	格納されている情報
22	IS_ARCHIVABLE	CHAR(1)	チャンクアーカイブ指定の有無 • 'Y' チャンクアーカイブ指定が指定されている（アーカイブマルチチャンク表である） • ナル値 チャンクアーカイブ指定が指定されていない（アーカイブマルチチャンク表以外の実表である） なお，次に示す表の場合は，ナル値が格納されます。 • ビュー表 • ディクショナリ表（実表） • システム表（実表）
23	ARCHIVE_DIRECTORY_PATH	VARCHAR(400)	アーカイブディレクトリの絶対パス 次に示す表の場合は，ナル値が格納されます。 • ビュー表 • ディクショナリ表（実表） • システム表（実表） • アーカイブマルチチャンク表以外の実表
24	RECREATE_TIME	TIMESTAMP	ALTER VIEW 文の実行日時 ビュー表に対して，ALTER VIEW 文を実行した最新のタイムスタンプが格納されます。 なお，次に示す表の場合は，ナル値が格納されます。 • 実表 • ALTER VIEW 文を実行していないビュー表
25	STORAGE_FORMAT	VARCHAR(32)	表データの格納形式 • 'COLUMN' カラムストア形式 • 'ROW' ローストア形式 なお，ビュー表の場合は，ナル値が格納されます。
26	N_ARRAY_COLUMN	SMALLINT	表に定義されている配列型の列の数 配列型の列が定義されていない表の場合は，ナル値が格納されます。

付録 B.3　SQL_COLUMNS の内容

SQL_COLUMNS には，実表，ビュー表，ディクショナリ表（実表），システム表（実表），ディクショナリ表，およびシステム表の列の定義情報が格納されています。1 行に 1 列分の情報が格納されています。

SQL_COLUMNS の内容を次の表に示します。

表 B-4　SQL_COLUMNS の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	COLUMN_NAME	VARCHAR(100)	列名
4	COLUMN_ID	SMALLINT	列 ID
5	TABLE_ID	SMALLINT	表 ID • 0x00020001～：ディクショナリ表（実表） • 0x000200C9～：システム表（実表） • 0x00020191～：HADB ユーザが定義した実表 • 0x00040001～：ディクショナリ表およびシステム表 • 0x00040401～：HADB ユーザが定義したビュー表
6	DATA_TYPE_CODE	SMALLINT	列のデータ型 • 113（0x71）：DATE • 121（0x79）：TIME • 125（0x7D）：TIMESTAMP • 145（0x91）：VARBINARY • 149（0x95）：BINARY • 193（0xC1）：VARCHAR • 197（0xC5）：CHAR • 225（0xE1）：DOUBLE PRECISION，FLOAT • 229（0xE5）：DECIMAL，NUMERIC • 241（0xF1）：INTEGER • 245（0xF5）：SMALLINT 配列型の列の場合は，要素データ型が格納されます。
7	DATA_LENGTH	SMALLINT	列の定義長 列の定義長は，列のデータ型（配列型の列の場合は，要素データ型）によって異なります。 **列のデータ型が次に示すどれかの場合** 列の定義長がバイト単位で格納されます。 • INTEGER • SMALLINT • CHAR • VARCHAR • DATE • DOUBLE PRECISION • FLOAT • BINARY • VARBINARY

項番	列名	データ型	格納されている情報
			列のデータ型が TIME(*p*)の場合 次に示す計算式で求めた値が格納されます。 3＋↑*p*÷2↑ *p*：0，3，6，9 または 12 列のデータ型が TIMESTAMP(*p*)の場合 次に示す計算式で求めた値が格納されます。 7＋↑*p*÷2↑ *p*：0，3，6，9 または 12 列のデータ型が DECIMAL，NUMERIC の場合 最初の 1 バイト目に位取りが格納され，2 バイト目に精度が格納されます。 CREATE VIEW 文の問合せ式本体の選択式に，四則演算の除算結果のデータ型がDECIMAL 型となる値式が含まれている場合，次に示す 2 つの値が不一致になることがあります。 • ここに格納されている位取りの値 • ビュー表の検索時に，ビュー表を等価変換した内部導出表の導出列の位取りの値 詳細については，マニュアル『HADB　SQL リファレンス』の『除算結果のデータ型が DECIMAL 型の場合の留意事項』の『ビュー表を検索する場合』を参照してください。
8	MAX_ELEMENT	SMALLINT	配列型の列の最大要素数 配列型の列の場合 2～30,000 の整数が格納されます。 配列型以外の列の場合 1 が格納されます。
9	IS_NULLABLE	CHAR(1)	列に対する非ナル値制約定義の指定 • 'Y' ナル値を許容する（非ナル値制約定義の指定なし） • 'N' ナル値を許容しない（非ナル値制約定義の指定あり） FIX 表の場合は，すべての列が'N'となります。
10	COLUMN_OFFSET	SMALLINT	列オフセット（表の先頭から何バイトの位置で列が始まっているかの情報） FIX 表ではない表（表オプションBRANCH ALL の指定なし）の場合は，ナル値が格納されます。
11	BRANCH	CHAR(1)	列定義のBRANCH の指定値 • 'Y'：YES • 'N'：NO • 'A'：AUTO

項番	列名	データ型	格納されている情報
			• '0'：VARCHAR 型の列，またはVARBINARY 型の列でBRANCH の指定なし VARCHAR 型およびVARBINARY 型以外の列の場合は，ナル値が格納されます。 また，カラムストア表の列の場合もナル値が格納されます。
12	DEFAULT_VALUE	VARCHAR(32000)	DEFAULT 句に指定された既定値 実表の定義時にDEFAULT 句に指定された既定値が格納されます。 • 定数 定数を指定した場合に格納されます。詳細については，「表 B-5　定数を指定した場合にDEFAULT_VALUE の列に格納される値」を参照してください。 • CURRENT_DATE CURRENT_DATE を指定した場合に格納されます。 • CURRENT_TIME(p) CURRENT_TIME(p)を指定した場合に格納されます。p は小数秒精度を示します。 p ＝ 0 の場合は，CURRENT_TIME となります。 p ＝ 3，6，9 および 12 の場合は，CURRENT_TIME(p)となります。 • CURRENT_TIMESTAMP(p) CURRENT_TIMESTAMP(p)を指定した場合に格納されます。p は小数秒精度を示します。 p ＝ 0 の場合は，CURRENT_TIMESTAMP となります。 p ＝ 3，6，9 および 12 の場合は，CURRENT_TIMESTAMP(p)となります。 • CURRENT_USER CURRENT_USER を指定した場合に格納されます。 • ナル値 NULL を指定した場合に格納されます。 なお，ビュー表の列，ディクショナリ表（実表）の列，システム表（実表）の列，およびDEFAULT 句の指定がない実表の列の場合は，ナル値が格納されます。
13	IS_DEFAULT_COLUMN	CHAR(1)	DEFAULT 句の指定有無 • 'Y' 実表の定義時にDEFAULT 句を指定した列 • ナル値 実表の定義時にDEFAULT 句を指定していない列 ビュー表の列，ディクショナリ表（実表）の列，およびシステム表（実表）の列の場合も，ナル値が格納されます。

項番	列名	データ型	格納されている情報
14	IS_PRIMARY_KEY_COLUMN	CHAR(1)	主キーを構成する列かどうか • 'Y' 主キーを構成する列 • ナル値 主キーを構成しない列 ビュー表の列，ディクショナリ表（実表）の列，およびシステム表（実表）の列の場合も，ナル値が格納されます。
15	PRIMARY_KEY_COLUMN_SEQUENCE_NUMBER	SMALLINT	主キーの構成順序 主キーを構成する列の順序が，1 から昇順で格納されます。 なお，ビュー表の列，ディクショナリ表（実表）の列，システム表（実表）の列，および主キーを構成しない実表の列の場合は，ナル値が格納されます。
16	N_FOREIGN_KEY_COLUMN	SMALLINT	この列を使用している外部キーの数 この列を使用している外部キーの数が格納されます。 なお，次に示す列の場合は，ナル値が格納されます。 • ビュー表の列 • ディクショナリ表（実表）の列 • システム表（実表）の列 • 外部キーを構成しない実表の列
17	IS_ARCHIVE_RANGE_COLUMN	CHAR(1)	アーカイブレンジ列かどうか • 'Y' アーカイブレンジ列である • ナル値 アーカイブレンジ列ではない なお，次に示す列の場合は，ナル値が格納されます。 • ビュー表の列 • ディクショナリ表（実表）の列 • システム表（実表）の列
18	COMPRESSION_TYPE	VARCHAR(32)	列定義のCOMPRESSION TYPE の指定値 カラムストア表の定義時に，圧縮方式指定に指定された内容が格納されます。 なお，次に示す列の場合は，ナル値が格納されます。 • ビュー表の列 • ディクショナリ表（実表）の列 • システム表（実表）の列 • ローストア表の列

DEFAULT 句に定数を指定した場合に，DEFAULT_VALUE の列に格納される値を次の表に示します。

表 B-5　定数を指定した場合に DEFAULT_VALUE の列に格納される値

項番	既定値に仮定されるデータ型		DEFAULT_VALUE の列に格納される値	
			データ長（単位：バイト）	格納される値の文字形式の例
1	数データ	INTEGER	指定された既定値を文字形式に変換した長さ	12345
2		SMALLINT		12345
3		DECIMAL		12.345
4		DOUBLE PRECISION		-1234567890.1234567E-123
5	文字データ	CHAR	*指定された既定値を文字形式に変換した長さ* + 2	'ABCD'
6		VARCHAR		
7	日時データ	DATE	16	DATE'*YYYY-MM-DD*'
8		TIME(*p*)	• *p* が 0 のとき： 14 • *p* が 0 以外のとき： 15 + *p*	TIME'*hh*:*mm*:*ss*.*nn...n*' ※1，※2
9		TIMESTAMP(*p*)	• *p* が 0 のとき： 30 • *p* が 0 以外のとき： 31 + *p*	TIMESTAMP'*YYYY-MM-DD*△*hh*:*mm*:*ss*.*nn....n*' ※1，※2
10	バイナリデータ	BINARY	• 2 進形式のバイナリデータの場合： *指定された既定値のデータ長* × 8 + 3 • 16 進形式のバイナリデータの場合： *指定された既定値のデータ長* × 2 + 3	• 2 進形式のバイナリデータの場合： B'01010101' • 16 進形式のバイナリデータの場合： X'010203'
11		VARBINARY		

（凡例）

p：小数秒精度（0，3，6，9 または 12）

YYYY：年

MM：月

DD：日

△：空白

hh：時

mm：分

ss：秒

nn...n：小数秒

注※1

DEFAULT 句を指定した列の小数秒精度が 0 の場合は，秒（*ss*）と小数秒（*nn...n*）の間のピリオド（ . ）は省略されます。

注※2

既定値の小数秒精度が，DEFAULT 句を指定した列の小数秒（*nn...n*）の桁数に満たない場合は，満たない桁数分，右側に 0 が補われます。

既定値の小数秒精度が，DEFAULT 句を指定した列の小数秒（*nn...n*）の桁数を超える場合は，超えた桁数分，値が切り捨てられます。

付録 B.4　SQL_DIV_TABLE の内容

SQL_DIV_TABLE には，実表，ディクショナリ表（実表），およびシステム表（実表）を格納している DB エリアの情報などが格納されています。1 行に 1 表分の情報が格納されています。

SQL_DIV_TABLE の内容を次の表に示します。

表 B-6　SQL_DIV_TABLE の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	TABLE_ID	SMALLINT	表 ID • 0x00020001～：ディクショナリ表（実表） • 0x000200C9～：システム表（実表） • 0x00020191～：HADB ユーザが定義した実表
4	DBAREA_ID	SMALLINT	表が格納されている DB エリアの DB エリア ID

付録 B.5　SQL_INDEXES の内容

SQL_INDEXES には，実表のインデクスの定義情報が格納されています。また，ディクショナリ表（実表），およびシステム表（実表）の B-tree インデクスの定義情報も格納されています。1 行に 1 インデクス分の情報が格納されています。

SQL_INDEXES の内容を次の表に示します。

表 B-7　SQL_INDEXES の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	インデクスが定義されている表の表識別子

項番	列名	データ型	格納されている情報
3	INDEX_NAME	VARCHAR(100)	インデクス識別子 主キーに対応するインデクスの場合，制約名と同じ名称のインデクス識別子が格納されます。
4	INDEX_ID	SMALLINT	インデクス ID • 0x00030001～：ディクショナリ表（実表）のインデクス • 0x00030191～：システム表（実表）のインデクス • 0x00030321～：HADB ユーザが定義した実表のインデクス
5	TABLE_ID	SMALLINT	インデクスが定義されている表の表 ID • 0x00020001～：ディクショナリ表（実表） • 0x000200C9～：システム表（実表） • 0x00020191～：HADB ユーザが定義した実表
6	COLUMN_COUNT	SMALLINT	インデクス構成列数
7	COLUMN_ID_LIST	VARCHAR(32)	インデクス構成列 ID リスト※
8	CREATE_TIME	TIMESTAMP	インデクス定義時のタイムスタンプ
9	UNIQUE_TYPE	CHAR(1)	B-tree インデクスの種類（ユニークインデクスかどうか） • 'Y'：ユニークインデクス • 'N'：非ユニークインデクス
10	FREE_AREA	SMALLINT	インデクスのページ内の未使用領域の比率 レンジインデクスの場合は，ナル値が格納されます。
11	INDEX_KEY_EXPAND	CHAR(1)	インデクスを構成する列の長さの合計が 256 バイト以上のインデクスかどうか • 'Y' インデクスを構成する列の長さの合計が 256 バイト以上のインデクス • 'N' インデクスを構成する列の長さの合計が 255 バイト以下のインデクス
12	INDEX_TYPE	CHAR(1)	インデクスの種別 • 'B'：B-tree インデクス • 'T'：テキストインデクス • 'R'：レンジインデクス
13	IS_PRIMARY_KEY	CHAR(1)	主キーに対応するインデクスかどうか • 'Y' 主キーに対応するインデクス • ナル値 主キーに対応しないインデクス

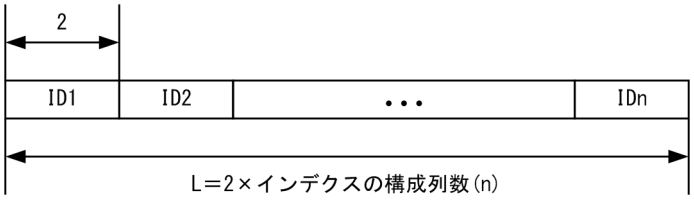
項番	列名	データ型	格納されている情報
			ディクショナリ表（実表）のインデクス，およびシステム表（実表）のインデクスの場合も，ナル値が格納されます。
14	IS_CHUNK_SKIP	CHAR(1)	チャンクのスキップができるレンジインデクスかどうか • 'Y' チャンクのスキップができるレンジインデクス • ナル値 チャンクのスキップができないレンジインデクス 次に示すインデクスの場合は，ナル値が格納されます。 • レンジインデクス以外のインデクス • バージョン 02-02 より前のバージョンで定義されたレンジインデクス
15	IS_EXCLUDE_NULL_VALUES	CHAR(1)	ナル値除外指定の指定有無 • 'Y' ナル値除外指定が指定されているインデクス • ナル値 ナル値除外指定が指定されていないインデクス
16	IS_ARCHIVE_RANGE	CHAR(1)	HADB サーバが自動的に定義したレンジインデクスかどうか • 'Y' アーカイブレンジ列に対して，HADB サーバが自動的に定義したレンジインデクス • ナル値 'Y'の条件を満たさないレンジインデクス ただし，アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更した場合は，HADB サーバが自動的に定義したレンジインデクスであっても，ナル値が格納されます。 なお，次に示すインデクスの場合は，ナル値が格納されます。 • ディクショナリ表（実表）のインデクス • システム表（実表）のインデクス
17	IS_TEXT_CORRECTION_RULE	CHAR(1)	テキストインデクス表記ゆれ補正指定の有無 • 'Y' テキストインデクス表記ゆれ補正指定が指定されているテキストインデクス • ナル値 テキストインデクス表記ゆれ補正指定が指定されていないテキストインデクス なお，次に示すインデクスの場合は，ナル値が格納されます。

項番	列名	データ型	格納されている情報
			• ディクショナリ表（実表）のインデクス • システム表（実表）のインデクス • テキストインデクス以外のインデクス
18	IS_TEXT_WORDCONTEXT	CHAR(1)	テキストインデクスワード検索指定の有無 • 'Y' テキストインデクスワード検索指定が指定されているテキストインデクス • ナル値 テキストインデクスワード検索指定が指定されていないテキストインデクス なお，次に示すインデクスの場合は，ナル値が格納されます。 • ディクショナリ表（実表）のインデクス • システム表（実表）のインデクス • テキストインデクス以外のインデクス
19	TEXT_DELIMITER_TYPE	CHAR(1)	テキストインデクス区切り文字指定の種類 • 'D' テキストインデクス区切り文字指定に，DEFAULT が指定されているテキストインデクス（テキストインデクス区切り文字指定を省略して，DEFAULT が仮定された場合も含む） • 'A' テキストインデクス区切り文字指定に，ALL が指定されているテキストインデクス なお，次に示すインデクスの場合は，ナル値が格納されます。 • テキストインデクスワード検索指定が指定されていないテキストインデクス • ディクショナリ表（実表）のインデクス • システム表（実表）のインデクス • テキストインデクス以外のインデクス

注※

インデクス構成列 ID リストの形式を次の図に示します。

図 B-2　インデクス構成列 ID リストの形式



IDm＝ { 昇順指定，レンジインデクス，またはテキストインデクス：m番目の列IDを格納
降順指定：m番目の列ID×(-1)を格納　　（単位:バイト）

ID1，ID2，…，IDnは，インデクス定義時の指定順に格納

付録 B.6　SQL_DIV_INDEX の内容

SQL_DIV_INDEX には，実表のインデクスが格納されている DB エリアなどの情報が格納されています。また，ディクショナリ表（実表），およびシステム表（実表）の B-tree インデクスが格納されている DB エリアなどの情報も格納されています。1 行に 1 インデクス分の情報が格納されています。

SQL_DIV_INDEX の内容を次の表に示します。

表 B-8　SQL_DIV_INDEX の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	インデクスが定義されている表の表識別子
3	INDEX_NAME	VARCHAR(100)	インデクス識別子
4	INDEX_ID	SMALLINT	インデクス ID • 0x00030001～：ディクショナリ表（実表）のインデクス • 0x00030191～：システム表（実表）のインデクス • 0x00030321～：HADB ユーザが定義した実表のインデクス
5	DBAREA_ID	SMALLINT	インデクスが格納されている DB エリアの DB エリア ID

付録 B.7　SQL_DBAREAS の内容

SQL_DBAREAS には，DB エリアの定義情報が格納されています。1 行に 1DB エリア分の情報が格納されています。

SQL_DBAREAS の内容を次の表に示します。

表 B-9　SQL_DBAREAS の内容

項番	列名	データ型	格納されている情報
1	DBAREA_NAME	VARCHAR(100)	DB エリア名
2	DBAREA_ID	SMALLINT	DB エリア ID
3	DBAREA_TYPE	CHAR(4)	DB エリアの種類 • 'MAST'：マスタディレクトリ用 DB エリア • 'DICT'：ディクショナリ用 DB エリア • 'DATA'：データ用 DB エリア • 'WORK'：作業表用 DB エリア • 'STBL'：システム表用 DB エリア
4	PAGE_SIZE	SMALLINT	DB エリアのページサイズ（単位：バイト） 作業表用 DB エリアの場合，adbinit コマンドの初期設定オプションのadb_init_wrk_page_size オペランドで指定したページサイズが格納されます。なお，サーバ定義adb_dbarea_wrk_page_size オペランドを指定してページサイズを変更していても，adbinit コマンドで指定したページサイズが格納されます。
5	SEGMENT_SIZE	SMALLINT	DB エリアのセグメントサイズ（単位：ページ）
6	N_TABLE	SMALLINT	DB エリアに格納されている実表の数 マスタディレクトリ用 DB エリア，および作業表 DB エリアの場合は 0 が格納されます。
7	N_INDEX	SMALLINT	DB エリアに格納されているインデクスの数（全インデクスの合計） マスタディレクトリ用 DB エリア，および作業表 DB エリアの場合は 0 が格納されます。

付録 B.8　SQL_SCHEMATA の内容

SQL_SCHEMATA には，スキーマに関する定義情報が格納されています。1 行に 1 スキーマ分の情報が格納されています。

SQL_SCHEMATA の内容を次の表に示します。

表 B-10　SQL_SCHEMATA の内容

項番	列名	データ型	格納されている情報
1	SCHEMA_OWNER	VARCHAR(100)	スキーマの所有者
2	SCHEMA_DEFINER	VARCHAR(100)	スキーマ定義を実行した HADB ユーザの認可識別子
3	SCHEMA_NAME	VARCHAR(100)	スキーマ名

項番	列名	データ型	格納されている情報
4	CREATE_TIME	TIMESTAMP	スキーマ定義時のタイムスタンプ

付録 B.9　SQL_VIEWS の内容

SQL_VIEWS には，ビュー表に関する定義情報が格納されています。1 行に 1 ビュー表分の情報が格納されています。

SQL_VIEWS の内容を次の表に示します。

表 B-11　SQL_VIEWS の内容

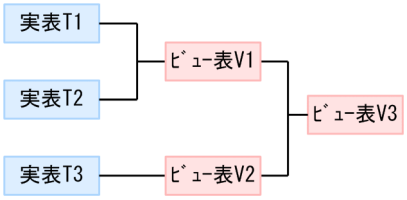
項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	VIEW_ID	SMALLINT	ビュー ID
4	IS_UPDATABLE	CHAR(1)	ビュー表の更新可否 'Y'：更新可能ビュー表 'N'：読み取り専用ビュー表
5	VIEW_OBJECT_SIZE	SMALLINT	ビューオブジェクト長
6	DEFINE_SOURCE_ID	INTEGER	SQL 文 ID
7	IS_INVALID	CHAR(1)	ビュー表が無効化されているかどうかの情報 • 'Y' ビュー表が無効化されている状態 なお，ビュー表が無効化されていない場合は，ナル値が格納されます。
8	VIEW_LEVEL	SMALLINT	ビューレベル

付録 B.10　SQL_VIEW_TABLE_USAGE の内容

SQL_VIEW_TABLE_USAGE には，ビュー表の基表の情報が格納されています。1 行に 1 基表分の情報が格納されています。

なお，定義したビュー表の基表がビュー表の場合は，そのビュー表の基表の情報も格納されます。ビュー表の基表が実表になるまで，すべての情報が格納されます。例を次の図に示します。

図 B-3　基表がビュー表の場合に格納される情報の例



[説明]

ビュー表V3 の基表は，ビュー表V1 とビュー表V2 です。そのため，ビュー表V3 の基表の情報としては，ビュー表V1 とビュー表V2 だけでなく，実表T1，実表T2 および実表T3 の情報も，SQL_VIEW_TABLE_USAGE に格納されます。

SQL_VIEW_TABLE_USAGE の内容を次の表に示します。

表 B-12　SQL_VIEW_TABLE_USAGE の内容

項番	列名	データ型	格納されている情報
1	VIEW_SCHEMA	VARCHAR(100)	ビュー表のスキーマ名
2	VIEW_NAME	VARCHAR(100)	ビュー表の表識別子
3	VIEW_ID	SMALLINT	ビュー ID
4	TABLE_SCHEMA	VARCHAR(100)	基表のスキーマ名
5	TABLE_NAME	VARCHAR(100)	基表の表識別子
6	TABLE_TYPE	CHAR(1)	基表の種類 • 'R'：実表 • 'V'：ビュー表
7	IS_DIRECT	CHAR(1)	直接基表として指定した表かどうかの情報 • 'Y'：直接指定 • 'N'：直接指定ではない

付録 B.11　SQL_DEFINE_SOURCE の内容

SQL_DEFINE_SOURCE には，各 SQL 文の情報が格納されています。1 行に 1SQL テキスト分の情報が格納されています。

SQL_DEFINE_SOURCE の内容を次の表に示します。

表 B-13　SQL_DEFINE_SOURCE の内容

項番	列名	データ型	格納されている情報
1	SOURCE_ID	INTEGER	SQL 文 ID
2	SOURCE_TYPE	CHAR(1)	SQL 文種別

項番	列名	データ型	格納されている情報
			• 'V'：ビュー定義
3	DEFINE_SOURCE	VARCHAR(64000)	SQL 文 SQL 文は入力された形で格納されます。
4	SOURCE_SIZE	SMALLINT	SQL 文の実長

注

ディクショナリ表SQL_DEFINE_SOURCE を検索した場合，HADB ユーザが定義したビュー表の情報だけが検索結果として出力されます。

スキーマ名にHADB およびMASTER を指定して検索しても，検索結果は0 件となります（エラーにはなりません）。

付録 B.12　SQL_DEFINE_ENVIRONMENT の内容

SQL_DEFINE_ENVIRONMENT には，ビュー定義時およびビュー定義変更時の環境情報が格納されています。1 行に 1 つのビュー表の環境情報が格納されています。

SQL_DEFINE_ENVIRONMENT の内容を次の表に示します。

表 B-14　SQL_DEFINE_ENVIRONMENT の内容

項番	列名	データ型	格納されている情報
1	RESOURCE_TYPE	CHAR(1)	定義情報種別 • 'V'：ビュー表
2	RESOURCE_ID	SMALLINT	定義情報 ID • ビューID
3	DELETE_RESERVED_WORD	VARCHAR(32000)	CREATE VIEW の実行時に削除されていた予約語※ 予約語を削除していない場合は，ナル値となります。
4	DEFINE_VR	CHAR(8)	定義時のバージョン 次に示す形式で格納されます。 '*VV-RR*' • *VV*：バージョン番号 • *RR*：リビジョン番号
5	RECREATE_VR	CHAR(8)	ALTER VIEW 文実行時の HADB サーバのバージョン ビュー表に対して，ALTER VIEW 文を最後に実行したときの HADB サーバのバージョン情報が，次に示す形式で格納されます。 '*VV-RR*' • *VV*：バージョン番号

項番	列名	データ型	格納されている情報
			• *RR*：リビジョン番号 なお，ビュー表に対して，ALTER VIEW 文を実行していない場合は，ナル値が格納されます。

注※

DELETE_RESERVED_WORD 列には，予約語削除機能環境で定義した削除予約語が，サーバ定義で指定した順にコンマ区切り（スペースなし）で格納されます。

予約語削除機能環境で削除予約語がACTION，CATALOG の順に定義され，ビュー表が 2 つ定義されている場合の格納例を次に示します。

（例）SQL_DEFINE_ENVIRONMENT の検索

```
SELECT * FROM "MASTER"."SQL_DEFINE_ENVIRONMENT"
```

検索結果

RESOURCE_TYPE	RESOURCE_ID	DELETE_RESERVED_WORD	DEFINE_VR	RECREATE_VR
'V'	262385	ACTION,CATALOG	01-01	NULL
'V'	262386	ACTION,CATALOG	01-01	NULL

付録 B.13　SQL_USERS の内容

SQL_USERS には，HADB ユーザの情報が格納されています。1 行に 1HADB ユーザの情報が格納されています。

SQL_USERS の内容を次の表に示します。

表 B-15　SQL_USERS の内容

項番	列名	データ型	格納されている情報
1	USER_NAME	VARCHAR(100)	HADB ユーザの認可識別子
2	USER_ID	SMALLINT	HADB ユーザのユーザ ID
3	DBA_PRIVILEGE	CHAR(1)	DBA 権限の有無 'Y'：DBA 権限あり 'N'：DBA 権限なし
4	CONNECT_PRIVILEGE	CHAR(1)	CONNECT 権限の有無 'Y'：CONNECT 権限あり 'N'：CONNECT 権限なし
5	SCHEMA_PRIVILEGE	CHAR(1)	スキーマ定義権限の有無 'Y'：スキーマ定義権限あり 'N'：スキーマ定義権限なし

項番	列名	データ型	格納されている情報
6	CREATE_TIME	TIMESTAMP	HADB ユーザの作成日時
7	AUDIT_ADMIN_PRIVILEGE	CHAR(1)	監査管理権限の有無 'Y'：監査管理権限あり 'N'：監査管理権限なし
8	AUDIT_VIEWER_PRIVILEGE	CHAR(1)	監査参照権限の有無 'Y'：監査参照権限あり 'N'：監査参照権限なし

付録 B.14　SQL_TABLE_CONSTRAINTS の内容

SQL_TABLE_CONSTRAINTS には，実表の制約に関する情報が格納されています。1 行に 1 制約の情報が格納されています。

SQL_TABLE_CONSTRAINTS の内容を次の表に示します。

表 B-16　SQL_TABLE_CONSTRAINTS の内容

項番	列名	データ型	格納されている情報
1	CONSTRAINT_SCHEMA	VARCHAR(100)	制約を含むスキーマ名
2	CONSTRAINT_NAME	VARCHAR(100)	制約名
3	CONSTRAINT_TYPE	CHAR(1)	制約の種別 • 'P'：主キー • 'F'：外部キー
4	TABLE_SCHEMA	VARCHAR(100)	制約が定義されている表のスキーマ名
5	TABLE_NAME	VARCHAR(100)	制約が定義されている表の表識別子

付録 B.15　SQL_INDEX_COLINF の内容

SQL_INDEX_COLINF には，インデクスの構成列に関する情報が格納されています。1 行にインデクス構成列 1 列分の情報が格納されています。

SQL_INDEX_COLINF の内容を次の表に示します。

表 B-17　SQL_INDEX_COLINF の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	インデクスが定義されている表の表識別子

項番	列名	データ型	格納されている情報
3	INDEX_NAME	VARCHAR(100)	インデクス識別子
4	INDEX_ID	SMALLINT	インデクス ID
5	INDEX_ORDER	SMALLINT	インデクスを構成する列の順序 インデクスを構成する列の順序が，1 から昇順に格納されます。
6	COLUMN_NAME	VARCHAR(100)	インデクス構成列名
7	ASC_DESC	CHAR(1)	B-tree インデクスのキー値の並び順 • 'A' ASC が指定されています（並び順は昇順です）。 • 'D' DESC が指定されています（並び順は降順です）。 なお，B-tree インデクス以外のインデクスの場合は，ナル値が格納されます。

付録 B.16　SQL_KEY_COLUMN_USAGE の内容

SQL_KEY_COLUMN_USAGE には，主キー・外部キーの構成列に関する情報が格納されています。1 行に主キー・外部キーの構成列 1 列分の情報が格納されています。

SQL_KEY_COLUMN_USAGE の内容を次の表に示します。

表 B-18　SQL_KEY_COLUMN_USAGE の内容

項番	列名	データ型	格納されている情報
1	CONSTRAINT_SCHEMA	VARCHAR(100)	制約を含むスキーマ名
2	CONSTRAINT_NAME	VARCHAR(100)	制約名
3	TABLE_SCHEMA	VARCHAR(100)	制約が定義されている表のスキーマ名
4	TABLE_NAME	VARCHAR(100)	制約が定義されている表の表識別子
5	COLUMN_NAME	VARCHAR(100)	制約が定義されている列の列名
6	COLUMN_ORDER	SMALLINT	制約が定義されている列の定義順序

付録 B.17　SQL_REFERENTIAL_CONSTRAINTS の内容

SQL_REFERENTIAL_CONSTRAINTS には，参照制約に関する情報が格納されています。1 行に 1 参照制約分の情報が格納されています。

SQL_REFERENTIAL_CONSTRAINTS の内容を次の表に示します。

表 B-19　SQL_REFERENTIAL_CONSTRAINTS の内容

項番	列名	データ型	格納されている情報
1	CONSTRAINT_SCHEMA	VARCHAR(100)	参照制約を含むスキーマ名
2	CONSTRAINT_NAME	VARCHAR(100)	参照制約の制約名
3	TABLE_SCHEMA	VARCHAR(100)	参照制約が定義されている表のスキーマ名
4	TABLE_NAME	VARCHAR(100)	参照制約が定義されている表の表識別子
5	PRIMARY_CONSTRAINT_SCHEMA	VARCHAR(100)	外部キーが参照する制約を含むスキーマ名
6	PRIMARY_CONSTRAINT_NAME	VARCHAR(100)	外部キーが参照する制約名
7	REFERENCED_TABLE_SCHEMA	VARCHAR(100)	被参照表のスキーマ名
8	REFERENCED_TABLE_NAME	VARCHAR(100)	被参照表の表識別子
9	IS_DISABLE	CHAR(1)	参照制約チェック抑止指定の有無 • 'Y' 参照制約のチェックを行いません。

付録 B.18　SQL_TABLE_PRIVILEGES の内容

SQL_TABLE_PRIVILEGES には，表に対するアクセス権限に関する情報が格納されています。1 行に，ある権限付与者が与えた 1 認可識別子の 1 表分の情報が格納されています。

SQL_TABLE_PRIVILEGES の内容を次の表に示します。

表 B-20　SQL_TABLE_PRIVILEGES の内容

項番	列名	データ型	格納されている情報
1	GRANTOR	VARCHAR(100)	アクセス権限を付与した HADB ユーザの認可識別子
2	GRANTEE	VARCHAR(100)	アクセス権限が付与された HADB ユーザの認可識別子，または'PUBLIC'
3	GRANTEE_TYPE	CHAR(1)	アクセス権限が付与された HADB ユーザの種別 • 'I' HADB ユーザの認可識別子，またはPUBLIC
4	OWNER	VARCHAR(100)	アクセス権限の対象となる表を所有する HADB ユーザの認可識別子
5	TABLE_SCHEMA	VARCHAR(100)	アクセス権限の対象となる表のスキーマ名
6	TABLE_NAME	VARCHAR(100)	アクセス権限の対象となる表の表識別子

項番	列名	データ型	格納されている情報
7	GRANT_TIME	TIMESTAMP	アクセス権限が付与された時刻 ある権限付与者が，1 認可識別子に対して，表のアクセス権限を初めて与えたときの時刻が格納されます。
8	SELECT_PRIVILEGE	CHAR(1)	SELECT 権限を持っているかどうか • 'G' 付与権付きのSELECT 権限を持っている • 'Y' 付与権なしのSELECT 権限を持っている • 'N' SELECT 権限を持っていない
9	INSERT_PRIVILEGE	CHAR(1)	INSERT 権限を持っているかどうか • 'G' 付与権付きのINSERT 権限を持っている • 'Y' 付与権なしのINSERT 権限を持っている • 'N' INSERT 権限を持っていない
10	UPDATE_PRIVILEGE	CHAR(1)	UPDATE 権限を持っているかどうか • 'G' 付与権付きのUPDATE 権限を持っている • 'Y' 付与権なしのUPDATE 権限を持っている • 'N' UPDATE 権限を持っていない
11	DELETE_PRIVILEGE	CHAR(1)	DELETE 権限を持っているかどうか • 'G' 付与権付きのDELETE 権限を持っている • 'Y' 付与権なしのDELETE 権限を持っている • 'N' DELETE 権限を持っていない
12	TRUNCATE_PRIVILEGE	CHAR(1)	TRUNCATE 権限を持っているかどうか • 'G' 付与権付きのTRUNCATE 権限を持っている • 'Y' 付与権なしのTRUNCATE 権限を持っている • 'N' TRUNCATE 権限を持っていない
13	REFERENCES_PRIVILEGE	CHAR(1)	REFERENCES 権限を持っているかどうか • 'G'

項番	列名	データ型	格納されている情報
			付与権付きのREFERENCES 権限を持っている • 'Y' 付与権なしのREFERENCES 権限を持っている • 'N' REFERENCES 権限を持っていない
14	IMPORT_TABLE_PRIVILEGE	CHAR(1)	IMPORT TABLE 権限を持っているかどうか • 'G' 付与権付きのIMPORT TABLE 権限を持っている • 'Y' 付与権なしのIMPORT TABLE 権限を持っている • 'N' IMPORT TABLE 権限を持っていない
15	REBUILD_INDEX_PRIVILEGE	CHAR(1)	REBUILD INDEX 権限を持っているかどうか • 'G' 付与権付きのREBUILD INDEX 権限を持っている • 'Y' 付与権なしのREBUILD INDEX 権限を持っている • 'N' REBUILD INDEX 権限を持っていない
16	GET_COSTINFO_PRIVILEGE	CHAR(1)	GET COSTINFO 権限を持っているかどうか • 'G' 付与権付きのGET COSTINFO 権限を持っている • 'Y' 付与権なしのGET COSTINFO 権限を持っている • 'N' GET COSTINFO 権限を持っていない
17	EXPORT_TABLE_PRIVILEGE	CHAR(1)	EXPORT TABLE 権限を持っているかどうか • 'G' 付与権付きのEXPORT TABLE 権限を持っている • 'Y' 付与権なしのEXPORT TABLE 権限を持っている • 'N' EXPORT TABLE 権限を持っていない
18	MERGE_CHUNK_PRIVILEGE	CHAR(1)	MERGE CHUNK 権限を持っているかどうか • 'G' 付与権付きのMERGE CHUNK 権限を持っている • 'Y' 付与権なしのMERGE CHUNK 権限を持っている • 'N' MERGE CHUNK 権限を持っていない

項番	列名	データ型	格納されている情報
19	CHANGE_CHUNK_COMMENT_PRIVILEGE	CHAR(1)	CHANGE CHUNK COMMENT 権限を持っているかどうか • 'G' 付与権付きのCHANGE CHUNK COMMENT 権限を持っている • 'Y' 付与権なしのCHANGE CHUNK COMMENT 権限を持っている • 'N' CHANGE CHUNK COMMENT 権限を持っていない
20	CHANGE_CHUNK_STATUS_PRIVILEGE	CHAR(1)	CHANGE CHUNK STATUS 権限を持っているかどうか • 'G' 付与権付きのCHANGE CHUNK STATUS 権限を持っている • 'Y' 付与権なしのCHANGE CHUNK STATUS 権限を持っている • 'N' CHANGE CHUNK STATUS 権限を持っていない
21	ARCHIVE_CHUNK_PRIVILEGE	CHAR(1)	ARCHIVE CHUNK 権限を持っているかどうか • 'G' 付与権付きのARCHIVE CHUNK 権限を持っている • 'Y' 付与権なしのARCHIVE CHUNK 権限を持っている • 'N' ARCHIVE CHUNK 権限を持っていない
22	UNARCHIVE_CHUNK_PRIVILEGE	CHAR(1)	UNARCHIVE CHUNK 権限を持っているかどうか • 'G' 付与権付きのUNARCHIVE CHUNK 権限を持っている • 'Y' 付与権なしのUNARCHIVE CHUNK 権限を持っている • 'N' UNARCHIVE CHUNK 権限を持っていない

注

ビュー表が無効化されている場合，そのビュー表に対するアクセス権限に関する情報は検索できません。

■ディクショナリ表およびシステム表に対するアクセス権限に関する情報

ディクショナリ表およびシステム表に対するアクセス権限に関する情報は，SQL_TABLE_PRIVILEGES には格納されません。ディクショナリ表およびシステム表に対するアクセス権限に関する情報は，次のように仮定されます。

- SELECT_PRIVILEGE（SELECT 権限を持っているかどうか）
  'G'：付与権付きのSELECT 権限を持っている
- EXPORT_TABLE_PRIVILEGE（EXPORT TABLE 権限を持っているかどうか）
  'G'：付与権付きのEXPORT TABLE 権限を持っている
- 上記以外のアクセス権限を持っているかどうか
  'N'：持っていない

付録 B.19　SQL_AUDITS の内容

SQL_AUDITS には，CREATE AUDIT 文で定義した監査対象定義の情報が格納されています。1 行に，1 監査対象定義の情報が格納されています。

SQL_AUDITS の内容を次の表に示します。

表 B-21　SQL_AUDITS の内容

項番	列名	データ型	格納されている情報
1	OPERATION_TYPE	VARCHAR(30)	対象操作種別 'ANY'：すべての監査対象操作
2	EVENT_TYPE	VARCHAR(30)	システムで使用
3	OBJECT_TYPE	VARCHAR(30)	システムで使用
4	OBJECT_SCHEMA	VARCHAR(100)	システムで使用
5	OBJECT_NAME	VARCHAR(100)	システムで使用
6	USER_NAME	VARCHAR(100)	システムで使用
7	EVENT_RESULT	VARCHAR(30)	システムで使用
8	AUDIT_TYPE	VARCHAR(30)	監査証跡種別 'EVENT'：イベントの最終結果
9	IS_EXCLUSION_AUDIT	CHAR(1)	システムで使用

付録 B.20　ディクショナリ表の参照時に排他が取得される実表

SQL 文を実行してディクショナリ表を参照すると，ディクショナリ表（実表）およびシステム表（実表）の排他が取得されます。ディクショナリ表の参照時に排他が取得される実表については，「2.10.5　ディクショナリ表およびシステム表の参照時に排他が取得される実表」を参照してください。

付録 B.21　ディクショナリ表（実表）の B-tree インデクス

ディクショナリ表（実表）には，次の表に示す B-tree インデクスが定義されています。

表 B-22　ディクショナリ表（実表）の B-tree インデクスの一覧

項番	ディクショナリ表（実表）	インデクス構成列	B-tree インデクス名
1	SQL_TABLES	TABLE_SCHEMA	SQLINDEXM01
2		TABLE_NAME	
3		TABLE_ID	SQLINDEXS01
4	SQL_COLUMNS	TABLE_SCHEMA	SQLINDEXM02
5		TABLE_NAME	
6		COLUMN_NAME	
7		TABLE_ID	SQLINDEXS02
8	SQL_DIV_TABLE	TABLE_SCHEMA	SQLINDEXM05
9		TABLE_NAME	
10		DBAREA_ID	SQLINDEXS04
11	SQL_INDEXES	TABLE_SCHEMA	SQLINDEXM03
12		INDEX_NAME	
13		TABLE_SCHEMA	SQLINDEXM04
14		TABLE_NAME	
15		INDEX_ID	SQLINDEXS03
16	SQL_DIV_INDEX	TABLE_SCHEMA	SQLINDEXM06
17		TABLE_NAME	
18		TABLE_SCHEMA	SQLINDEXM07
19		INDEX_NAME	
20		DBAREA_ID	SQLINDEXS05
21	SQL_DBAREAS	DBAREA_NAME	SQLINDEXS06
22		DBAREA_ID	SQLINDEXS07
23		N_TABLE	SQLINDEXM08
24		DBAREA_ID	
25	SQL_SCHEMATA	SCHEMA_OWNER	SQLINDEXS08
26		SCHEMA_NAME	SQLINDEXS09
27	SQL_VIEWS	TABLE_SCHEMA	SQLINDEXM09

項番	ディクショナリ表（実表）	インデクス構成列	B-tree インデクス名
28		TABLE_NAME	
29		VIEW_ID	SQLINDEXS10
30	SQL_VIEW_TABLE_USAGE	VIEW_SCHEMA	SQLINDEXM10
31		VIEW_NAME	
32		TABLE_SCHEMA	SQLINDEXM11
33		TABLE_NAME	
34		VIEW_ID	SQLINDEXS11
35	SQL_VIEW_OBJECT	VIEW_ID	SQLINDEXM12
36		OBJECT_NO	
37	SQL_DEFINE_SOURCE	SOURCE_ID	SQLINDEXS12
38	SQL_DEFINE_ENVIRONMENT	RESOURCE_TYPE	SQLINDEXM13
39		RESOURCE_ID	
40	SQL_USERS	USER_NAME	SQLINDEXS13
41	SQL_TABLE_CONSTRAINTS	CONSTRAINT_SCHEMA	SQLINDEXM14
42		CONSTRAINT_NAME	
43		TABLE_SCHEMA	SQLINDEXM15
44		TABLE_NAME	
45	SQL_INDEX_COLINF	TABLE_SCHEMA	SQLINDEXM16
46		INDEX_NAME	
47		TABLE_SCHEMA	SQLINDEXM17
48		TABLE_NAME	
49	SQL_KEY_COLUMN_USAGE	CONSTRAINT_SCHEMA	SQLINDEXM18
50		CONSTRAINT_NAME	
51		COLUMN_NAME	
52		TABLE_SCHEMA	SQLINDEXM19
53		TABLE_NAME	
54		COLUMN_NAME	
55	SQL_REFERENTIAL_CONSTRAINTS	CONSTRAINT_SCHEMA	SQLINDEXM20
56		CONSTRAINT_NAME	
57		TABLE_SCHEMA	SQLINDEXM21
58		TABLE_NAME	

項番	ディクショナリ表（実表）	インデクス構成列	B-tree インデクス名
59		PRIMARY_CONSTRAINT_SCHEMA	SQLINDEXM22
60		PRIMARY_CONSTRAINT_NAME	
61		REFERENCED_TABLE_SCHEMA	SQLINDEXM23
62		REFERENCED_TABLE_NAME	
63	SQL_TABLE_PRIVILEGES	TABLE_SCHEMA	SQLINDEXM24
64		TABLE_NAME	
65		GRANTEE	
66		GRANTOR	
67		GRANTOR	SQLINDEXS14
68		GRANTEE	SQLINDEXS15
69	SQL_AUDITS	OBJECT_TYPE	SQLINDEXM25
70		OBJECT_SCHEMA	
71		OBJECT_NAME	

付録 B.22　ディクショナリ表の検索

SELECT 文でディクショナリ表を検索できます。ディクショナリ表の認可識別子はMASTER です。

なお，ディクショナリ表を検索したら，すぐにCOMMIT 文を実行してトランザクションを終了させてください。COMMIT 文を実行しないと，ディクショナリ表の排他が解放されません。ただし，ディクショナリ表を検索したあとで，adbsql コマンドを終了する場合，ディクショナリ表の排他は解放されるため，COMMIT 文の実行は不要です。

また，ディクショナリ表を検索する HADB ユーザがDBA 権限を持っているかどうかによって，検索できる情報が異なります。詳細については，「付録 B.1　ディクショナリ表の概要」の「(3)　HADB ユーザが参照できるディクショナリ表の範囲」を参照してください。

ディクショナリ表を検索するSELECT 文の指定例を次に示します。

(1)　表 ID から表の表名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","TABLE_NAME"
  FROM "MASTER"."SQL_TABLES"
    WHERE "TABLE_ID"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。表 ID を入力してください。

(2)　ローストア表の一覧を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA", "TABLE_NAME"
  FROM "MASTER"."SQL_TABLES"
    WHERE "STORAGE_FORMAT" = 'ROW'
```

(3)　カラムストア表の一覧を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA", "TABLE_NAME"
  FROM "MASTER"."SQL_TABLES"
    WHERE "STORAGE_FORMAT" = 'COLUMN'
```

(4)　インデクス ID からインデクスのインデクス名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","INDEX_NAME"
  FROM "MASTER"."SQL_INDEXES"
    WHERE "INDEX_ID"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。インデクス ID を入力してください。

(5)　インデクス ID からインデクスが定義された表の表名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","TABLE_NAME"
  FROM "MASTER"."SQL_INDEXES"
    WHERE "INDEX_ID"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。インデクス ID を入力してください。

(6) インデクス ID からインデクスが定義された列の列名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”TABLE_NAME”,”COLUMN_NAME”
  FROM ”MASTER”.”SQL_INDEX_COLINF” WHERE ”INDEX_ID” = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。インデクス ID を入力してください。

(7) 全レンジインデクスのインデクス名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”INDEX_NAME”
  FROM ”MASTER”.”SQL_INDEXES”
    WHERE ”INDEX_TYPE”=’R’
```

(8) 表名から表に定義された全インデクス名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”INDEX_NAME”
  FROM ”MASTER”.”SQL_INDEXES”
    WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(9) 表名から表を使用する全ビュー表のビュー表名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”VIEW_SCHEMA”,”VIEW_NAME”
  FROM ”MASTER”.”SQL_VIEW_TABLE_USAGE”
    WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(10)　スキーマ名から定義された全表の表名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","TABLE_NAME","TABLE_TYPE"
  FROM "MASTER"."SQL_TABLES"
    WHERE "TABLE_SCHEMA"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。スキーマ名を入力してください。

なお，TABLE_TYPE が'R'の場合は実表を，'V'の場合はビュー表を意味しています。

(11)　スキーマ名から定義された全インデクスのインデクス名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","INDEX_NAME","INDEX_TYPE"
  FROM "MASTER"."SQL_INDEXES"
    WHERE "TABLE_SCHEMA"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。スキーマ名を入力してください。

なお，INDEX_TYPE が'B'の場合は B-tree インデクスを，'T'の場合はテキストインデクスを，'R'の場合はレンジインデクスを意味しています。

(12)　インデクス名からインデクスが格納されている DB エリア名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "MASTER"."SQL_DBAREAS"."DBAREA_NAME"
  FROM "MASTER"."SQL_DBAREAS","MASTER"."SQL_DIV_INDEX"
    WHERE "MASTER"."SQL_DIV_INDEX"."TABLE_SCHEMA"=?
      AND "MASTER"."SQL_DIV_INDEX"."INDEX_NAME"=?
      AND "MASTER"."SQL_DBAREAS"."DBAREA_ID"="MASTER"."SQL_DIV_INDEX"."DBAREA_ID"
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求にはインデクス識別子を入力してください。

(13)　全マルチチャンク表に作成されるチャンク数の最大値を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_NAME”,”N_CHUNK_RESERVED”
  FROM ”MASTER”.”SQL_TABLES”
    WHERE ”IS_CHUNK” = ’Y’
      ORDER BY ”TABLE_NAME”
```

(14)　DB エリアに格納されている表の表名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”TABLE_NAME” FROM ”MASTER”.”SQL_DIV_TABLE”
  WHERE ”DBAREA_ID”=
   (SELECT ”DBAREA_ID” FROM ”MASTER”.”SQL_DBAREAS” WHERE ”DBAREA_NAME”=?)
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。入力要求には DB エリア名を入力してください。

(15)　DB エリアに格納されているインデクスのインデクス名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”INDEX_NAME” FROM ”MASTER”.”SQL_DIV_INDEX”
  WHERE ”DBAREA_ID”=
   (SELECT ”DBAREA_ID” FROM ”MASTER”.”SQL_DBAREAS” WHERE ”DBAREA_NAME”=?)
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。入力要求には DB エリア名を入力してください。

(16)　ビュー表の更新可否を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”IS_UPDATABLE” FROM ”MASTER”.”SQL_VIEWS”
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求にはビュー表の表識別子を入力してください。

検索結果が’Y’の場合は，更新可能ビュー表であるため更新できます。’N’の場合は，読み取り専用ビュー表であるため更新できません。

(17) HADB ユーザが持っているユーザ権限とスキーマ操作権限を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

なお，DBA 権限を持っている HADB ユーザは，すべての HADB ユーザが持っているユーザ権限情報，およびスキーマ操作権限情報を調べられます。DBA 権限を持っていない HADB ユーザは，自分（HADB サーバに接続中の認可識別子の HADB ユーザ）が持っているユーザ権限情報，およびスキーマ操作権限情報を調べられます。

指定例

```
SELECT "DBA_PRIVILEGE","CONNECT_PRIVILEGE","SCHEMA_PRIVILEGE"
  FROM "MASTER"."SQL_USERS" WHERE "USER_NAME"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。入力要求には，HADB ユーザの認可識別子を入力してください。

検索結果の各列の値が'Y'の場合は，対象のユーザ権限およびスキーマ操作権限情報を持っています。'N'の場合は，対象のユーザ権限およびスキーマ操作権限情報を持っていません。

(18) 主キーに対応するインデクスのインデクス名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","INDEX_NAME" FROM "MASTER"."SQL_INDEXES"
  WHERE "TABLE_SCHEMA" = ? AND "TABLE_NAME" = ? AND "IS_PRIMARY_KEY" = 'Y'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(19) チャンクのスキップができるレンジインデクスかどうかを調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "IS_CHUNK_SKIP" FROM "MASTER"."SQL_INDEXES"
  WHERE "TABLE_SCHEMA" = ? AND "INDEX_NAME" = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求にはインデクス識別子を入力してください。

検索結果が'Y'の場合は，チャンクのスキップができるレンジインデクスです。ナル値の場合は，チャンクのスキップができないレンジインデクス，またはレンジインデクス以外のインデクスです。

(20)　主キーと外部キーを構成する列名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "COLUMN_NAME"
  FROM "MASTER"."SQL_KEY_COLUMN_USAGE"
    WHERE "CONSTRAINT_SCHEMA" = ? AND "CONSTRAINT_NAME" = ?
      ORDER BY "COLUMN_ORDER" ASC
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求には制約を含むスキーマ名を，2 番目の入力要求には制約名を入力してください。

(21)　主キーを参照する外部キーを調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "CONSTRAINT_SCHEMA","CONSTRAINT_NAME"
  FROM "MASTER"."SQL_REFERENTIAL_CONSTRAINTS"
    WHERE "PRIMARY_CONSTRAINT_SCHEMA" = ?
      AND "PRIMARY_CONSTRAINT_NAME" = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求には制約を含むスキーマ名を，2 番目の入力要求には制約名を入力してください。

(22)　ビュー表が無効化されているかどうかを調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "IS_INVALID" FROM "MASTER"."SQL_VIEWS"
  WHERE "TABLE_SCHEMA"=? AND "TABLE_NAME"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはビュー表のスキーマ名を，2 番目の入力要求にはビュー表の表識別子を入力してください。

検索結果が'Y'の場合，ビュー表は無効化されています。ナル値の場合，ビュー表は無効化されていません。

(23)　表に対するアクセス権限を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”GRANTOR”,
       ”SELECT_PRIVILEGE”,”INSERT_PRIVILEGE”,”UPDATE_PRIVILEGE”,
       ”DELETE_PRIVILEGE”,”TRUNCATE_PRIVILEGE”,
       ”REFERENCES_PRIVILEGE”,”IMPORT_TABLE_PRIVILEGE”,
       ”REBUILD_INDEX_PRIVILEGE”,”GET_COSTINFO_PRIVILEGE”,
       ”EXPORT_TABLE_PRIVILEGE”,”MERGE_CHUNK_PRIVILEGE”,
       ”CHANGE_CHUNK_COMMENT_PRIVILEGE”,”CHANGE_CHUNK_STATUS_PRIVILEGE”,
       ”ARCHIVE_CHUNK_PRIVILEGE”,”UNARCHIVE_CHUNK_PRIVILEGE”
  FROM ”MASTER”.”SQL_TABLE_PRIVILEGES”
    WHERE ”GRANTEE” IN (’PUBLIC’, ?)
      AND ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはアクセス権限が付与された HADB ユーザの認可識別子を，2 番目の入力要求には表のスキーマ名を，3 番目の入力要求には表の表識別子を入力してください。

GRANTOR 列には，アクセス権限を付与した HADB ユーザの認可識別子が出力されます。

そのほかの列には，各アクセス権限を持っているかどうかが出力されます。検索結果が’G’の場合は，該当の表に対するアクセス権限を付与権付きで持っています。’Y’の場合は，該当の表に対するアクセス権限（付与権なし）を持っています。’N’の場合は，該当の表に対するアクセス権限を持っていません。

(24) 実表の種類（マルチチャンク表，アーカイブマルチチャンク表，カラムストア表かどうか）を調べる場合

指定例に示すSELECT 文でディクショナリ表を検索すると，次のことを確認できます。

- 実表がシングルチャンク表なのか，それともマルチチャンク表なのか
- 実表がレギュラーマルチチャンク表なのか，それともアーカイブマルチチャンク表なのか
- 実表がローストア表なのか，それともカラムストア表なのか

指定例

```
SELECT ”TABLE_SCHEMA”,”TABLE_NAME”,”IS_CHUNK”,”IS_ARCHIVABLE”,”STORAGE_FORMAT”
  FROM ”MASTER”.”SQL_TABLES”
    WHERE ”TABLE_SCHEMA” = ? AND ”TABLE_NAME” = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

検索結果の例

```
TABLE_SCHEMA  TABLE_NAME  IS_CHUNK  IS_ARCHIVABLE  STORAGE_FORMAT
------------  ----------  --------  -------------  --------------
ADBUSER01     T1                Y              *  COLUMN
```

この例では，実表"ADBUSER01"."T1"（以降，表T1 と表記）の種類が表示されています。

- IS_CHUNK 列には，表T1 がマルチチャンク表かどうかが表示されます。'Y' が表示された場合，表T1 はマルチチャンク表です。'N' が表示された場合，表T1 はシングルチャンク表です。
- IS_ARCHIVABLE 列には，表T1 がアーカイブマルチチャンク表かどうかが表示されます。'Y' が表示された場合，表T1 はアーカイブマルチチャンク表です。ナル値が表示された場合，表T1 はアーカイブマルチチャンク表ではありません。この例の*は，ナル値を意味しています。
- STORAGE_FORMAT 列には，表T1 がローストア表なのか，カラムストア表なのかが表示されます。'COLUMN' が表示された場合，表T1 はカラムストア表です。'ROW' が表示された場合，表T1 はローストア表です。

この例の場合，表T1 は，レギュラーマルチチャンク表かつカラムストア表であることがわかります。

(25) アーカイブマルチチャンク表のアーカイブディレクトリを調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”ARCHIVE_DIRECTORY_PATH” FROM ”MASTER”.”SQL_TABLES”
  WHERE ”TABLE_SCHEMA” = ? AND ”TABLE_NAME” = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(26) アーカイブマルチチャンク表のアーカイブレンジ列の列名を調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”COLUMN_NAME” FROM ”MASTER”.”SQL_COLUMNS”
  WHERE ”TABLE_SCHEMA” = ? AND ”TABLE_NAME” = ? AND ”IS_ARCHIVE_RANGE_COLUMN” = 'Y'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(27) アーカイブマルチチャンク表のアーカイブレンジ列に定義されたレンジインデクスを調べる場合

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”IX”.”TABLE_SCHEMA”,”IX”.”INDEX_NAME”
  FROM ”MASTER”.”SQL_INDEXES” ”IX”
      ,”MASTER”.”SQL_INDEX_COLINF” ”IC”
      ,”MASTER”.”SQL_COLUMNS” ”CL”
  WHERE ”IX”.”TABLE_SCHEMA” = ?
    AND ”IX”.”TABLE_NAME” = ?
```

```
    AND "IX"."INDEX_TYPE" = 'R'
    AND "IX"."INDEX_ID" = "IC"."INDEX_ID"
    AND "IC"."TABLE_SCHEMA" = "CL"."TABLE_SCHEMA"
    AND "IC"."TABLE_NAME" = "CL"."TABLE_NAME"
    AND "IC"."COLUMN_NAME" = "CL"."COLUMN_NAME"
    AND "CL"."IS_ARCHIVE_RANGE_COLUMN" = 'Y'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(28) 実表の定義情報を調べる場合

実表を再定義する際にCREATE TABLE 文の指定内容がわからない場合は，指定例 1 および指定例 2 のSELECT 文をそれぞれ実行し，ディクショナリ表を検索してください。

なお，SELECT 文をadbsql コマンドで実行する場合は，-s オプションを指定してください。SELECT 文の検索結果を，標準出力に CSV 形式で出力できます。

メモ

adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』を参照してください。

指定例 1

```
SELECT "MT"."TABLE_SCHEMA"                          AS "スキーマ名"
      ,"MT"."TABLE_NAME"                            AS "表識別子"
      ,"MT"."FREE_AREA"                             AS "表未使用領域指定"
      ,"MT"."FIX_TABLE"                             AS "FIX指定"
      ,"MT"."IS_BRANCH_ALL"                         AS "BRANCH ALL指定"
      ,"MT"."IS_CHUNK"                              AS "チャンク指定"
      ,"MT"."N_CHUNK_RESERVED"                      AS "チャンクの最大値"
      ,"MT"."IS_ARCHIVABLE"                         AS "チャンクアーカイブ指定"
      ,"MT"."ARCHIVE_DIRECTORY_PATH"                AS "アーカイブディレクトリ"
      ,"MT"."STORAGE_FORMAT"                        AS "表データの格納形式"
      ,"MR"."DBAREA_NAME"                           AS "表格納DBエリア名"
      ,"MC"."COLUMN_ID"                             AS "列ID"
      ,"MC"."COLUMN_NAME"                           AS "列名"
      ,CONVERT("MC"."DATA_TYPE_CODE", CHAR(9), '0XXXXXXX')
                                                    AS "データ型コード"
      ,CONVERT("MC"."DATA_LENGTH", CHAR(9), '0XXXXXXX')
                                                    AS "データ長"
      ,"MC"."IS_NULLABLE"                           AS "ナル値可否"
      ,"MC"."BRANCH"                                AS "ブランチ指定"
      ,"MC"."DEFAULT_VALUE"                         AS "既定値"
      ,"MC"."IS_DEFAULT_COLUMN"                     AS "DEFAULT句指定"
      ,"MC"."PRIMARY_KEY_COLUMN_SEQUENCE_NUMBER" AS "主キー構成順序"
      ,"MC"."IS_ARCHIVE_RANGE_COLUMN"               AS "アーカイブレンジ列指定"
      ,"MC"."COMPRESSION_TYPE"                      AS "列データの圧縮方式"
FROM "MASTER"."SQL_TABLES"     "MT"
    ,"MASTER"."SQL_DIV_TABLE" "MD"
    ,"MASTER"."SQL_DBAREAS"   "MR"
    ,"MASTER"."SQL_COLUMNS"   "MC"
```

```
WHERE "MT"."TABLE_SCHEMA" = ?
  AND "MT"."TABLE_NAME" = ?
  AND "MT"."TABLE_TYPE" = 'R'
  AND "MT"."TABLE_ID" = "MD"."TABLE_ID"
  AND "MD"."DBAREA_ID" = "MR"."DBAREA_ID"
  AND "MT"."TABLE_SCHEMA" = "MC"."TABLE_SCHEMA"
  AND "MT"."TABLE_NAME" = "MC"."TABLE_NAME"
ORDER BY "MT"."TABLE_SCHEMA" ASC
        ,"MT"."TABLE_NAME"   ASC
        ,"MC"."COLUMN_ID"    ASC
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

指定例 2

```
SELECT "MT"."TABLE_SCHEMA"            AS "スキーマ名"
      ,"MT"."TABLE_NAME"              AS "表識別子"
      ,"MT"."CONSTRAINT_SCHEMA"       AS "制約を含むスキーマ名"
      ,"MT"."CONSTRAINT_NAME"         AS "制約名"
      ,"MT"."CONSTRAINT_TYPE"         AS "制約の種別"
      ,"MC"."COLUMN_NAME"             AS "列名"
      ,"MC"."COLUMN_ORDER"            AS "定義順序"
      ,"MR"."REFERENCED_TABLE_SCHEMA" AS "被参照表のスキーマ名"
      ,"MR"."REFERENCED_TABLE_NAME"   AS "被参照表の表識別子"
      ,"MR"."IS_DISABLE"              AS "参照制約チェック抑止指定"
FROM "MASTER"."SQL_TABLE_CONSTRAINTS" "MT"
INNER JOIN "MASTER"."SQL_KEY_COLUMN_USAGE" "MC"
  ON "MT"."CONSTRAINT_SCHEMA" = "MC"."CONSTRAINT_SCHEMA"
 AND "MT"."CONSTRAINT_NAME" = "MC"."CONSTRAINT_NAME"
LEFT OUTER JOIN "MASTER"."SQL_REFERENTIAL_CONSTRAINTS" "MR"
  ON "MT"."CONSTRAINT_SCHEMA" = "MR"."CONSTRAINT_SCHEMA"
 AND "MT"."CONSTRAINT_NAME" = "MR"."CONSTRAINT_NAME"
WHERE "MT"."TABLE_SCHEMA" = ?
  AND "MT"."TABLE_NAME" = ?
ORDER BY "MT"."TABLE_SCHEMA"      ASC
        ,"MT"."TABLE_NAME"        ASC
        ,"MT"."CONSTRAINT_SCHEMA" ASC
        ,"MT"."CONSTRAINT_NAME"   ASC
        ,"MC"."COLUMN_ORDER"      ASC
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(29)　インデクスの定義情報を調べる場合

インデクスを再定義する際にCREATE INDEX 文の指定内容がわからない場合は，次に示すSELECT 文でディクショナリ表を検索してください。

なお，SELECT 文をadbsql コマンドで実行する場合は，-s オプションを指定してください。SELECT 文の検索結果を，標準出力に CSV 形式で出力できます。

メモ

adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQLの実行)』を参照してください。

指定例

```
SELECT "MI"."TABLE_SCHEMA"             AS "スキーマ名"
      ,"MI"."TABLE_NAME"               AS "表識別子"
      ,"MI"."INDEX_NAME"               AS "インデクス識別子"
      ,"MC"."INDEX_ORDER"              AS "インデクス構成列順序"
      ,"MC"."COLUMN_NAME"              AS "列名"
      ,"MC"."ASC_DESC"                 AS "昇順・降順"
      ,"MI"."UNIQUE_TYPE"              AS "UNIQUE"
      ,"MI"."FREE_AREA"                AS "インデクス未使用領域指定"
      ,"MI"."INDEX_TYPE"               AS "インデクスタイプ指定"
      ,"MI"."IS_EXCLUDE_NULL_VALUES"   AS "ナル値除外指定"
      ,"MI"."IS_TEXT_CORRECTION_RULE"  AS "テキストインデクス表記ゆれ補正指定"
      ,"MI"."IS_TEXT_WORDCONTEXT"      AS "テキストインデクスワード検索指定"
      ,"MI"."TEXT_DELIMITER_TYPE"      AS "テキストインデクス区切り文字指定"
      ,"MR"."DBAREA_NAME"              AS "インデクス格納DBエリア名"
FROM "MASTER"."SQL_INDEXES"       "MI"
    ,"MASTER"."SQL_DIV_INDEX"     "MD"
    ,"MASTER"."SQL_DBAREAS"       "MR"
    ,"MASTER"."SQL_INDEX_COLINF"  "MC"
WHERE "MI"."TABLE_SCHEMA" = ?
  AND "MI"."TABLE_NAME" = ?
  AND "MI"."INDEX_ID" = "MD"."INDEX_ID"
  AND "MI"."INDEX_ID" = "MC"."INDEX_ID"
  AND "MD"."DBAREA_ID" = "MR"."DBAREA_ID"
ORDER BY "MI"."TABLE_SCHEMA" ASC
        ,"MI"."TABLE_NAME"   ASC
        ,"MI"."INDEX_NAME"   ASC
        ,"MC"."INDEX_ORDER"  ASC
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(30)　アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる場合

ALTER TABLE 文で，レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する際，アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる必要があります。

アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる場合は，次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","INDEX_NAME" FROM "MASTER"."SQL_INDEXES" "IX"
  WHERE "TABLE_SCHEMA" = ?
    AND "TABLE_NAME" = ?
```

```
    AND "INDEX_TYPE" = 'R'
    AND EXISTS(SELECT * FROM "MASTER"."SQL_INDEX_COLINF" "IC"
                WHERE "IX"."TABLE_SCHEMA" = "IC"."TABLE_SCHEMA"
                  AND "IX"."INDEX_NAME" = "IC"."INDEX_NAME"
                  AND "IC"."COLUMN_NAME" = ?)
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。

- 最初の入力要求には，レギュラーマルチチャンク表のスキーマ名を入力してください。
- 2 番目の入力要求には，レギュラーマルチチャンク表の表識別子を入力してください。
- 3 番目の入力要求には，アーカイブレンジ列として指定する列の列名を入力してください。

(31)　ビュー表の定義情報を調べる場合

ビュー表を再定義する際に，CREATE VIEW 文の指定内容がわからない場合は，次に示すSELECT 文でディクショナリ表を検索してください。

なお，SELECT 文をadbsql コマンドで実行する場合は，-s オプションを指定してください。SELECT 文の検索結果を，標準出力に CSV 形式で出力できます。

> **メモ**
>
> adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQL の実行)』を参照してください。

指定例

```
SELECT "MV"."TABLE_SCHEMA"          AS "スキーマ名"
      ,"MV"."TABLE_NAME"            AS "表識別子"
      ,"MS"."DEFINE_SOURCE"         AS "ビュー定義文"
      ,"ME"."DELETE_RESERVED_WORD" AS "削除予約語"
      ,"ME"."DEFINE_VR"             AS "定義バージョン"
FROM "MASTER"."SQL_VIEWS" "MV"
    ,"MASTER"."SQL_DEFINE_SOURCE" "MS"
    ,"MASTER"."SQL_DEFINE_ENVIRONMENT" "ME"
WHERE "MV"."TABLE_SCHEMA" = ?
  AND "MV"."TABLE_NAME" = ?
  AND "MV"."DEFINE_SOURCE_ID" = "MS"."SOURCE_ID"
  AND "MV"."VIEW_ID" = "ME"."RESOURCE_ID"
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはビュー表のスキーマ名を，2 番目の入力要求にはビュー表の表識別子を入力してください。

検索結果の例

```
ｽｷｰﾏ名      表識別子  ﾋﾞｭｰ定義文                                  削除予約語  定義ﾊﾞｰｼﾞｮﾝ
----------  --------  ----------------------------------------  ----------  ----------
ADBUSER01   V11       create view v11 as select * from t1                   03-03
```

ビュー定義文に表示されている内容が，ビュー表V11（スキーマ名ADBUSER01）のCREATE VIEW 文の指定内容です。

上記のSELECT 文でディクショナリ表を検索したあとで，ビュー表を再定義する場合の注意事項を次に示します。

- SELECT 文の検索結果のうち，「"削除予約語"」がナル値ではない場合は，ビュー表を前回定義したときと同じ予約語削除の指定が，ビュー表の再定義時に必要になることがあります。
- ALTER TABLE 文による列名変更の影響で，ビュー表の無効化を解除する場合は，SELECT 文の検索結果を基に作成する定義系 SQL について，列名の修正が必要になることがあります。
- ALTER TABLE 文による表名変更の影響で，ビュー表の無効化を解除する場合は，SELECT 文の検索結果を基に作成する定義系 SQL について，表識別子の修正が必要になることがあります。

(32) 依存するビュー表の一覧，およびそれらのビュー表の定義情報を調べる場合

依存するビュー表の一覧と，それらのビュー表の定義情報を調べる場合は，次に示すSELECT 文でディクショナリ表を検索してください。

なお，SELECT 文をadbsql コマンドで実行する場合は，-s オプションを指定してください。SELECT 文の検索結果を，標準出力に CSV 形式で出力できます。

> **メモ**
>
> adbsql コマンドについては，マニュアル『HADB　コマンドリファレンス』の『adbsql（SQLの実行)』を参照してください。

指定例

```
SELECT "MV"."TABLE_SCHEMA"         AS "スキーマ名"
      ,"MV"."TABLE_NAME"           AS "表識別子"
      ,"MV"."IS_INVALID"           AS "無効状態"
      ,"MV"."VIEW_LEVEL"           AS "ビューレベル"
      ,"MS"."DEFINE_SOURCE"        AS "ビュー定義文"
      ,"ME"."DELETE_RESERVED_WORD" AS "削除予約語"
      ,"ME"."DEFINE_VR"            AS "定義バージョン"
 FROM  "MASTER"."SQL_VIEW_TABLE_USAGE" "MVU"
     ,"MASTER"."SQL_VIEWS" "MV"
     ,"MASTER"."SQL_DEFINE_SOURCE" "MS"
     ,"MASTER"."SQL_DEFINE_ENVIRONMENT" "ME"
 WHERE "MVU"."TABLE_SCHEMA"=?
   AND "MVU"."TABLE_NAME" = ?
```

```
   AND "MV"."TABLE_SCHEMA"      = "MVU"."VIEW_SCHEMA"
   AND "MV"."TABLE_NAME"        = "MVU"."VIEW_NAME"
   AND "MV"."DEFINE_SOURCE_ID" = "MS"."SOURCE_ID"
   AND "ME"."RESOURCE_TYPE"     = 'V'
   AND "MV"."VIEW_ID"           = "ME"."RESOURCE_ID"
 ORDER BY "ビューレベル" ASC
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求には表のスキーマ名を，2 番目の入力要求には表の表識別子を入力してください。

指定した表に依存するビュー表の一覧と，それらのビュー表の定義情報が表示されます。

検索結果の例

```
ｽｷｰﾏ名     表識別子  無効状態  ﾋﾞｭｰﾚﾍﾞﾙ  ﾋﾞｭｰ定義文                                削除予約語  定義ﾊﾞｰｼﾞｮﾝ
---------  --------  --------  --------  ---------------------------------------  ----------  ----------
ADBUSER01  V12                1  create view v12 as select * from t1                      03-03
ADBUSER01  V11                1  create view v11 as select * from t1                      03-03
ADBUSER01  V3                 2  create view v3 as select * from v12,v21                  03-03
```

上記のビュー表（V12，V11，V3）が，？パラメタで指定した表に依存するビュー表です。

ビュー定義文に表示されている内容が，各ビュー表のCREATE VIEW 文の指定内容です。

上記のSELECT 文でディクショナリ表を検索したあとで，ビュー表を再定義する場合の注意事項を次に示します。

- SELECT 文の検索結果のうち，「"削除予約語"」がナル値ではない場合は，ビュー表を前回定義したときと同じ予約語削除の指定が，ビュー表の再定義時に必要になることがあります。
- ALTER TABLE 文による列名変更の影響で，ビュー表の無効化を解除する場合は，SELECT 文の検索結果を基に作成する定義系 SQL について，列名の修正が必要になることがあります。
- ALTER TABLE 文による表名変更の影響で，ビュー表の無効化を解除する場合は，SELECT 文の検索結果を基に作成する定義系 SQL について，表識別子の修正が必要になることがあります。

(33)　依存するビュー表の一覧を調べる場合

依存するビュー表の一覧を調べる場合は，次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "MV"."TABLE_SCHEMA"         AS "スキーマ名"
      ,"MV"."TABLE_NAME"           AS "表識別子"
      ,"MV"."IS_INVALID"           AS "無効状態"
FROM  "MASTER"."SQL_VIEW_TABLE_USAGE" "MVU"
     ,"MASTER"."SQL_VIEWS" "MV"
WHERE "MVU"."TABLE_SCHEMA"=?
   AND "MVU"."TABLE_NAME" = ?
   AND "MV"."TABLE_SCHEMA"      = "MVU"."VIEW_SCHEMA"
   AND "MV"."TABLE_NAME"        = "MVU"."VIEW_NAME"
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求には表のスキーマ名を，2 番目の入力要求には表の表識別子を入力してください。

指定した表に依存するビュー表の一覧が表示されます。

検索結果の例

```
ｽｷｰﾏ名      表識別子  無効状態
---------  --------  --------
ADBUSER01  V12
ADBUSER01  V3
ADBUSER01  V11
```

上記のビュー表（V12，V3，V11）が，？パラメタで指定した表に依存するビュー表です。

(34) テキストインデクスに指定したインデクスオプションを調べる場合

テキストインデクスに指定したインデクスオプションを調べる場合は，次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA"
      ,"INDEX_NAME"
      ,"INDEX_TYPE"
      ,"IS_TEXT_CORRECTION_RULE"
      ,"IS_TEXT_WORDCONTEXT"
      ,"TEXT_DELIMITER_TYPE"
  FROM "MASTER"."SQL_INDEXES"
    WHERE "TABLE_SCHEMA" = ?
      AND "INDEX_NAME" = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求にはインデクス識別子を入力してください。

(35) ビュー表の基表である実表とビュー表を調べる場合

ビュー表の基表である実表とビュー表を調べる場合は，次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT "TABLE_SCHEMA","TABLE_NAME"
  FROM "MASTER"."SQL_VIEW_TABLE_USAGE"
    WHERE "VIEW_SCHEMA"=? AND "VIEW_NAME"=? AND "IS_DIRECT"='Y'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはビュー表のスキーマ名を，2 番目の入力要求にはビュー表の表識別子を入力してください。

(36) 監査人の認可識別子と監査人が持っている監査権限を調べる方法

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”USER_NAME”,”AUDIT_ADMIN_PRIVILEGE”,”AUDIT_VIEWER_PRIVILEGE”
  FROM ”MASTER”.”SQL_USERS”
    WHERE ”AUDIT_ADMIN_PRIVILEGE”='Y' OR ”AUDIT_VIEWER_PRIVILEGE”='Y'
```

検索結果の例

```
USER_NAME      AUDIT_ADMIN_PRIVILEGE  AUDIT_VIEWER_PRIVILEGE
-------------  ---------------------  ----------------------
ADBAUDITADMIN                      Y                       N
ADBAUDITOR                         N                       Y
```

監査人の一覧が表示されます。

USER_NAME 列には，監査人の認可識別子が表示されます。

AUDIT_ADMIN_PRIVILEGE 列が'Y'の場合は，その監査人は監査管理権限を持っています。

AUDIT_VIEWER_PRIVILEGE 列が'Y'の場合は，その監査人は監査参照権限を持っています。

(37) adbinit コマンド実行時に指定した作業表用 DB エリアのページサイズを調べる方法

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”PAGE_SIZE”
  FROM ”MASTER”.”SQL_DBAREAS”
    WHERE ”DBAREA_NAME”='ADBWRK'
```

adbinit コマンド実行時に指定した作業表用 DB エリアのページサイズが，PAGE_SIZE 列にバイト単位で出力されます。

(38) 更新行のカラム化機能の対象となる表を調べる方法（B-tree インデクスを定義していないカラムストア表を調べる方法）

次に示すSELECT 文でディクショナリ表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”TABLE_NAME”
  FROM ”MASTER”.”SQL_TABLES”
    WHERE ”STORAGE_FORMAT” = 'COLUMN'
      AND ”TABLE_ID”
           NOT IN (
                   SELECT ”TABLE_ID”
                     FROM ”MASTER”.”SQL_INDEXES”
```

```
                    WHERE ”INDEX_TYPE” = ’B’
                )
```

更新行のカラム化機能の対象となる表（B-tree インデクスを定義していないカラムストア表）の一覧が表示されます。

付録 C　システム表

ここでは，システム表に格納されている情報，システム表に定義されている B-tree インデクス，およびシステム表の検索方法について説明します。

付録 C.1　システム表の概要

システム表には，実表およびインデクスのコスト情報や，マルチチャンク表のチャンク情報が格納されます。

システム表は，実表とビュー表の 2 種類があります。

- システム表（実表）

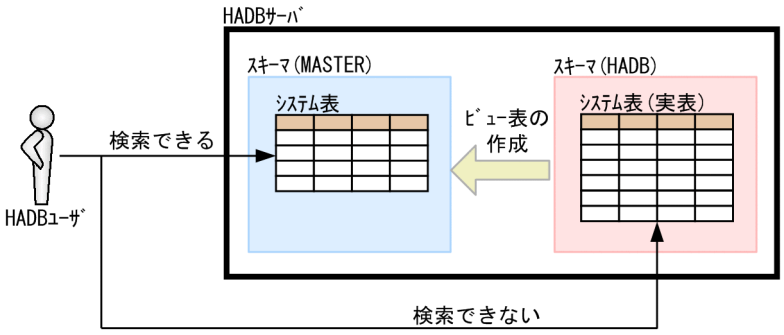
  このマニュアルでは，システム表の実表を，**システム表（実表）**と表記しています。システム表（実表）のスキーマ名は，HADB になります。

- システム表（ビュー表）

  このマニュアルでは，システム表のビュー表を，**システム表**と表記しています。システム表のスキーマ名は，MASTER になります。

システム表（実表）とシステム表の関係を次の図に示します。

図 C-1　システム表（実表）とシステム表の関係



[説明]

実表およびインデクスのコスト情報や，マルチチャンク表のチャンク情報は，システム表（実表）に格納されます。システム表（実表）から検索用に作成されるビュー表が，システム表になります。

HADB ユーザは，システム表を検索できます。しかし，システム表（実表）は検索できません。

HADB ユーザがSELECT 文でシステム表を検索することで，コスト情報の収集日時，チャンクの作成日時などの情報を確認できます。システム表を検索する場合は，スキーマ名にMASTER を指定してください。

(1)　システム表の一覧

システム表の一覧を次の表に示します。

表 C-1　システム表の一覧

項番	システム表の名称	格納されている情報	システム表に対応するシステム表（実表）の表 ID	ビュー定義に含まれる副問合せの数※	外への参照の有無
1	STATUS_TABLES	実表に関するコスト情報が格納されています。1 行に 1 実表分の情報が格納されています。	0x000200C9	1	○
2	STATUS_COLUMNS	列に関するコスト情報が格納されています。1 行に 1 列分の情報が格納されています。	0x000200CA	1	○
3	STATUS_INDEXES	インデクスに関するコスト情報が格納されています。1 行に 1 インデクス分の情報が格納されています。 なお，レンジインデクスのコスト情報は格納されません。	0x000200CB	1	○
4	STATUS_CHUNKS	マルチチャンク表に関するチャンク情報が格納されています。1 行に 1 チャンク分の情報が格納されています。	0x000200CC	1	○
5	STATUS_SYNONYM_DICTIONARIES	同義語辞書に関する情報が格納されています。1 行に 1 辞書分の情報が格納されています。	0x000200CD	0	×

（凡例）

○：システム表のビュー定義に，外への参照を行う副問合せを含みます。そのため，該当するシステム表は，マルチ集合値式の副問合せに指定できません。

×：システム表のビュー定義に，外への参照を行う副問合せを含みません。

注※

該当するシステム表を SQL 文の副問合せに指定した場合に，副問合せの入れ子の数として加算する必要がある数を示しています。

(2)　システム表が作成されるタイミング

システム表は，次に示すタイミングで自動的に作成されます。

- データベースの初期設定が完了したあとに，初めて HADB サーバを開始したとき
- HADB サーバをバージョンアップしたとき

(3) HADB ユーザが参照できるシステム表の範囲

HADB ユーザがシステム表を参照する際，持っている権限によって参照できる情報が異なります。HADB ユーザが参照できるシステム表の情報を次の表に示します。

なお，システム表を基表に指定したビュー表の場合も，持っている権限によって参照できる情報が異なります。そのため，同じビュー表を検索しても，ビュー表を定義した HADB ユーザと，ビュー表に対するアクセス権限を付与された HADB ユーザでは，参照できる情報が異なることがあります。

表 C-2　HADB ユーザが参照できるシステム表の情報

システム表の名称	参照対象の情報	システム表を参照する HADB ユーザが持っている権限			
		DBA 権限	監査管理権限	アクセス権限	CONNECT 権限
STATUS_TABLES	自分	○	○	○	○
	他ユーザ	○	×	△※	×
	MASTER	－	－	－	－
STATUS_COLUMNS	自分	○	○	○	○
	他ユーザ	○	×	△※	×
	MASTER	－	－	－	－
STATUS_INDEXES	自分	○	○	○	○
	他ユーザ	○	×	△※	×
	MASTER	－	－	－	－
STATUS_CHUNKS	自分	○	○	○	○
	他ユーザ	○	×	△※	×
	MASTER	－	－	－	－
STATUS_SYNONYM_DICTIONARIES	自分	○	○	○	○
	他ユーザ	○	×	△※	×
	MASTER	－	－	－	－

（凡例）

自分：自分が所有するスキーマおよびスキーマオブジェクトに関する情報

他ユーザ：ほかの HADB ユーザが所有するスキーマおよびスキーマオブジェクトに関する情報

MASTER：ディクショナリ表およびシステム表に関する情報

○：参照できます。

△：条件付きで参照できます。

×：参照できません。

－：該当しません。

注※

アクセス権限を持っているスキーマオブジェクトに関する情報と，そのスキーマに関する情報については参照できます。

付録 C.2　STATUS_TABLES の内容

STATUS_TABLES には，実表に関するコスト情報が格納されています。1 行に 1 実表分の情報が格納されています。

STATUS_TABLES の内容を次の表に示します。

表 C-3　STATUS_TABLES の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	COLLECT_TIME	TIMESTAMP	コスト情報の収集日時
4	N_ROW	INTEGER	処理対象表の行数 IS_N_ROW_COMPLETE 列の値が'N'の場合，コスト情報の収集処理が打ち切られた時点で，分析が完了している行数が格納されます。
5	COLLECT_VR	CHAR(8)	コスト情報を収集したときの HADB サーバのバージョン※
6	IS_COMPLETE	CHAR(1)	コスト情報を収集した際に，「処理対象表に格納されているデータの特徴」の取得処理が完了しているかどうか※ • 'Y'：完了している • 'N'：完了していない コスト情報を収集する時間（コスト情報収集オプションadb_getcst_collect_time の指定値）を超えたため，コスト情報の収集処理が途中で打ち切られています。
7	IS_N_ROW_COMPLETE	CHAR(1)	「処理対象表の行数」の取得処理が完了しているかどうか※ • 'Y'：完了している • 'N'：完了していない コスト情報を収集する時間（コスト情報収集オプションadb_getcst_collect_time の指定値）を超えたため，コスト情報の収集処理が途中で打ち切られています。

注※

次の条件をすべて満たす表の場合，STATUS_TABLES のこの列にナル値が格納されます。

- バージョン 04-03 より前の HADB サーバで，コスト情報を収集している表
- バージョン 04-03 以降の HADB サーバでは，コスト情報を収集していない表

付録 C.3　STATUS_COLUMNS の内容

STATUS_COLUMNS には，実表の列に関するコスト情報が格納されています。1 行に 1 列分の情報が格納されています。

STATUS_COLUMNS の内容を次の表に示します。

表 C-4　STATUS_COLUMNS の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	COLUMN_NAME	VARCHAR(100)	列名
4	DATA_TYPE_CODE	SMALLINT	列のデータ型
5	DATA_LENGTH	SMALLINT	列のデータ定義長 詳細については，「表 B-4　SQL_COLUMNS の内容」の『DATA_LENGTH』を参照してください。
6	COLUMN_VALUE_MAX	VARCHAR(100)	列の値の最大値 列のデータ型が配列型の場合，配列要素の値の最大値が格納されます。 IS_MAXMIN_COMPLETE 列の値が'N'の場合，コスト情報の収集処理が打ち切られた時点で，処理対象となった列の値の最大値が格納されます。 列データがない場合，または列データがNULLだけの場合，この列にはナル値が格納されます。
7	COLUMN_VALUE_MIN	VARCHAR(100)	列の値の最小値 列のデータ型が配列型の場合，配列要素の値の最小値が格納されます。 IS_MAXMIN_COMPLETE 列の値が'N'の場合，コスト情報の収集処理が打ち切られた時点で，処理対象となった列の値の最小値が格納されます。 列データがない場合，または列データがNULLだけの場合，この列にはナル値が格納されます。

項番	列名	データ型	格納されている情報
8	N_NULL	INTEGER	ナル値の数 列のデータ型が配列型の場合，列値（配列要素の値ではありません）がナル値となる行数が格納されます。 この列が含まれる表に対応するSTATUS_TABLESのIS_COMPLETE 列の値が'N'の場合，コスト情報の収集処理が打ち切られた時点で，分析が完了している処理対象列のナル値の数が格納されます。
9	IS_MAXMIN_COMPLETE	CHAR(1)	「列の値の最大値」，および「列の値の最小値」の取得処理が完了しているかどうか※1 • 'Y'：完了している • 'N'：完了していない コスト情報を収集する時間（コスト情報収集オプションadb_getcst_collect_time の指定値）を超えたため，コスト情報の収集処理が途中で打ち切られています。
10	MAX_ELEMENT	SMALLINT	配列型の列の最大要素数※2 配列型の列の場合，最大要素数が格納されます。配列型以外の列の場合，1 が格納されます。
11	ARRAY_DATA_LENGTH_AVG	SMALLINT	配列要素のデータ長の平均値 可変長のデータの場合は，実データ長の平均値が格納されます。配列要素の値がナル値の場合は，その配列要素は平均値の計算に含まれません。 すべての配列データが次のどちらかの条件に該当する場合は，この列には 0 が格納されます。 • 配列要素の値がすべてナル値の場合 • 空の配列データの場合 なお，次のどれかの条件に該当する場合は，この列にはナル値が格納されます。 • 配列型以外の列の場合 • 列データがない場合 • 列データがナル値だけの場合
12	N_ARRAY_NULL_ELEMENT	INTEGER	配列要素にナル値を含む行数 次のどれかの条件に該当する場合は，この列にはナル値が格納されます。 • 配列型以外の列の場合 • 列データがない場合 • 列データがナル値だけの場合
13	N_ARRAY_ELEMENT_AVG	SMALLINT	配列データの配列要素数の平均値 次のどれかの条件に該当する場合は，この列にはナル値が格納されます。

項番	列名	データ型	格納されている情報
			• 配列型以外の列の場合 • 列データがない場合 • 列データがナル値だけの場合

注※1

次の条件をすべて満たす表の場合，STATUS_COLUMNS のIS_MAXMIN_COMPLETE 列にはナル値が格納されます。

- バージョン 04-03 より前の HADB サーバで，コスト情報を収集している表
- バージョン 04-03 以降の HADB サーバでは，コスト情報を収集していない表

注※2

次の条件をすべて満たす表の場合，STATUS_COLUMNS のMAX_ELEMENT 列にはナル値が格納されます。

- バージョン 05-04 より前の HADB サーバで，コスト情報を収集している表
- バージョン 05-04 以降の HADB サーバでは，コスト情報を収集していない表

付録 C.4　STATUS_INDEXES の内容

STATUS_INDEXES には，インデクスに関するコスト情報が格納されています。1 行に 1 インデクス分の情報が格納されています。

STATUS_INDEXES の内容を次の表に示します。

表 C-5　STATUS_INDEXES の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	INDEX_NAME	VARCHAR(100)	インデクス識別子
4	INDEX_TYPE	CHAR(1)	インデクスの種別 'B'：B-tree インデクス 'T'：テキストインデクス
5	COLLECT_TIME	TIMESTAMP	コスト情報の収集日時

付録 C.5　STATUS_CHUNKS の内容

STATUS_CHUNKS には，マルチチャンク表に関するチャンク情報が格納されています。1 行に 1 チャンク分の情報が格納されています。

STATUS_CHUNKS の内容を次の表に示します。

表 C-6　STATUS_CHUNKS の内容

項番	列名	データ型	格納されている情報
1	TABLE_SCHEMA	VARCHAR(100)	スキーマ名
2	TABLE_NAME	VARCHAR(100)	表識別子
3	CHUNK_ID	INTEGER	チャンク ID
4	CREATE_TIME	TIMESTAMP	チャンクの作成日時 • チャンクをマージしている場合は，マージ元チャンクのうち，最も古い「チャンクの作成日時」が格納されます。 • adbimport コマンドで待機状態で作成されたあと，adbchgchunkstatus コマンドで通常状態に変更されたチャンクの場合は，「adbchgchunkstatus コマンドを実行した日時」が格納されます。 • adbchgchunkstatus コマンドのカレントチャンク切り替え動作によって作成されたチャンクの場合は，「adbchgchunkstatus コマンドを実行した日時」が格納されます。 なお，次に示す場合はナル値が格納されます。 • adbimport コマンドで待機状態のチャンクを作成した場合
5	SWAP_TIME	TIMESTAMP	カレントチャンクが切り替わった日時 • チャンクをマージしている場合は，マージ元チャンクのうち，最も新しい「カレントチャンクが切り替わった日時」が格納されます。 • adbimport コマンドで待機状態で作成されたあと，adbchgchunkstatus コマンドで通常状態に変更されたチャンクであり，かつ通常状態に変更されたときにカレントチャンクに切り替わらなかったチャンクの場合は，「adbchgchunkstatus コマンドを実行した日時」が格納されます。 なお，次に示す場合はナル値が格納されます。 • 新たなチャンクが作成されていない場合（カレントチャンクが切り替わっていない場合） • チャンクをマージしたときに，マージ元チャンクにカレントチャンクが含まれていた場合 • adbimport コマンドで待機状態のチャンクを作成した場合
6	CHUNK_COMMENT	VARCHAR(1024)	チャンクに設定されたコメント コメントが設定されていない場合は，ナル値が格納されます。
7	CHUNK_STATUS	VARCHAR(32)	チャンクの状態

項番	列名	データ型	格納されている情報
			• 'Normal' 通常状態 • 'Pending Delete' 削除仕掛中 • 'Wait' 待機状態
8	ARCHIVE_STATUS	VARCHAR(32)	アーカイブ状態のチャンクかどうか • 'Archive' アーカイブ状態のチャンク • ナル値 アーカイブ状態ではないチャンク
9	STORAGE_FORMAT	VARCHAR(32)	チャンク内のデータ格納形式 • 'COLUMN' カラムストア形式 • 'ROW' ローストア形式

付録 C.6　STATUS_SYNONYM_DICTIONARIES の内容

STATUS_SYNONYM_DICTIONARIES には，同義語辞書に関する情報が格納されています。1 行に 1 辞書分の情報が格納されています。

STATUS_SYNONYM_DICTIONARIES の内容を次の表に示します。

表 C-7　STATUS_SYNONYM_DICTIONARIES の内容

項番	列名	データ型	格納されている情報
1	SYNONYM_DICTIONARY_NAME	VARCHAR(120)	同義語辞書名
2	CREATE_TIME	TIMESTAMP	同義語辞書の作成日時
3	BINARY_PATH	VARCHAR(510)	同義語辞書ファイルの格納ディレクトリのパス名
4	BINARY_FILE_NAME	VARCHAR(130)	同義語辞書ファイル名
5	BINARY_FILE_NAME_IGNORECASE	VARCHAR(130)	表記ゆれ補正検索（IGNORECASE 指定）に対応している同義語辞書ファイル名 表記ゆれ補正オプションにCASESENSITIVE を指定した場合は，ナル値が格納されます。 また，表記ゆれ補正オプションの指定を省略した場合も，ナル値が格納されます。

項番	列名	データ型	格納されている情報
6	BINARY_FILE_NAME_SORTCODE	VARCHAR(130)	表記ゆれ補正検索（SORTCODE 指定）に対応している同義語辞書ファイル名 表記ゆれ補正オプションにCASESENSITIVE を指定した場合は，ナル値が格納されます。 また，表記ゆれ補正オプションの指定を省略した場合も，ナル値が格納されます。
7	CORRECTION_RULE	VARCHAR(32)	表記ゆれ補正オプションの指定値 • 'CASESENSITIVE' 表記ゆれ補正検索の指定なし • 'CORRECTIONRULE' 表記ゆれ補正検索の指定あり 表記ゆれ補正オプションの指定を省略した場合は，'CASESENSITIVE'が格納されます。
8	N_SYNONYM_GROUP	SMALLINT	同義語グループ数
9	SYNONYM_DICTIONARY_COMMENT	VARCHAR(1024)	コメント コメントを指定していない場合は，ナル値が格納されます。

付録 C.7　システム表の参照時に排他が取得される実表

SQL 文を実行してシステム表を参照すると，ディクショナリ表（実表）およびシステム表（実表）の排他が取得されます。システム表の参照時に排他が取得される実表については，「2.10.5　ディクショナリ表およびシステム表の参照時に排他が取得される実表」を参照してください。

付録 C.8　システム表（実表）の B-tree インデクス

システム表（実表）には，次の表に示す B-tree インデクスが定義されています。

表 C-8　システム表（実表）の B-tree インデクスの一覧

項番	システム表	インデクス構成列	B-tree インデクス名
1	STATUS_TABLES	TABLE_SCHEMA	STATUSINDEXM01
2		TABLE_NAME	
3	STATUS_COLUMNS	TABLE_SCHEMA	STATUSINDEXM02
4		TABLE_NAME	
5		COLUMN_NAME	
6	STATUS_INDEXES	TABLE_SCHEMA	STATUSINDEXM03

項番	システム表	インデクス構成列	B-tree インデクス名
7		INDEX_NAME	
8		TABLE_SCHEMA	STATUSINDEXM04
9		TABLE_NAME	
10	STATUS_CHUNKS	TABLE_SCHEMA	STATUSINDEXM05
11		TABLE_NAME	
12		CHUNK_ID	
13	STATUS_SYNONYM_DICTIONARIES	SYNONYM_DICTIONARY_NAME	STATUSINDEXS01

付録 C.9　システム表の検索

SELECT 文でシステム表を検索できます。システム表を検索することで，コスト情報およびチャンクの情報を確認できます。システム表の認可識別子はMASTER です。

なお，システム表を検索したら，すぐにCOMMIT 文を実行してトランザクションを終了させてください。COMMIT 文を実行しないと，システム表の排他が解放されません。ただし，システム表を検索したあとで，adbsql コマンドを終了する場合，システム表の排他は解放されるため，COMMIT 文の実行は不要です。

また，システム表を検索するときに，REPEATABLE READ のトランザクションからシステム表を検索すると，更新前の古いシステム表の情報を参照するおそれがあります。そのため，次のどちらかの方法で検索してください。

- READ COMMITTED のトランザクションからシステム表を検索する

  adbsql サブコマンドの#SET TRAN_ISOLV でREAD_COMMITTED を指定したあとで，システム表を検索してください。
- 検索の直前にCOMMIT 文を実行する

  REPEATABLE READ のトランザクションからシステム表を検索する場合は，COMMIT 文を実行したあとで，システム表を検索してください。

システム表を検索する HADB ユーザがDBA 権限を持っているかどうかによって，検索できる情報が異なります。詳細については，「付録 C.1　システム表の概要」の「(3)　HADB ユーザが参照できるシステム表の範囲」を参照してください。

システム表を検索するSELECT 文の指定例を次に示します。

(1)　コスト情報を収集した全実表の表名と収集日時を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”TABLE_NAME”,”COLLECT_TIME”
  FROM ”MASTER”.”STATUS_TABLES”
```

(2) コスト情報を収集した全インデクスのインデクス名と収集日時を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”TABLE_SCHEMA”,”INDEX_NAME”,”COLLECT_TIME”
  FROM ”MASTER”.”STATUS_INDEXES”
```

(3) 表名から表内の全チャンクの情報を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”CHUNK_ID”,”CHUNK_COMMENT”,”CHUNK_STATUS”,”CREATE_TIME”,”SWAP_TIME”
  FROM ”MASTER”.”STATUS_CHUNKS” WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(4) チャンク ID からチャンクの情報を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT * FROM ”MASTER”.”STATUS_CHUNKS”
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=? AND ”CHUNK_ID”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を，3 番目の入力要求にはチャンク ID を入力してください。

(5) 指定した期間に作成されたチャンクを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT * FROM ”MASTER”.”STATUS_CHUNKS”
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
    AND ”CREATE_TIME” BETWEEN ? AND ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を，3 番目の入力要求には期間の開始日時を，4 番目の入力要求には期間の終了日時を入力してください。

上記のSELECT 文では，次のチャンクが検索対象となります。

- 指定した期間に作成されたチャンク
- バックグラウンドインポートで待機状態のチャンクとして作成されたあとで，指定した期間に adbchgchunkstatus コマンドでチャンクの状態を通常状態に変更されたチャンク

ただし，次のチャンクは検索対象外です。

- 指定した期間にバックグラウンドインポートで待機状態のチャンクとして作成されたチャンク

(6) チャンクの作成日時から最新のチャンク ID を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”CHUNK_ID” FROM ”MASTER”.”STATUS_CHUNKS” TC
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
    AND ”CREATE_TIME”=(SELECT MAX(”CREATE_TIME”)
        FROM ”MASTER”.”STATUS_CHUNKS”
          WHERE TC.”TABLE_SCHEMA”=”TABLE_SCHEMA”
            AND TC.”TABLE_NAME”=”TABLE_NAME”)
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

上記のSELECT 文では，次の日時のうち，最も新しい日時のチャンクが検索対象となります。

- チャンクが作成された日時
- バックグラウンドインポートで待機状態のチャンクとして作成されたあとで，adbchgchunkstatus コマンドで通常状態のチャンクに変更されたときの日時

ただし，次のチャンクは検索対象外です。

- 過去に通常状態になったことがない待機状態のチャンク

(7) チャンクの作成日時から最古のチャンク ID を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”CHUNK_ID” FROM ”MASTER”.”STATUS_CHUNKS” TC
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
    AND ”CREATE_TIME”=(SELECT MIN(”CREATE_TIME”)
```

```
        FROM "MASTER"."STATUS_CHUNKS"
          WHERE TC."TABLE_SCHEMA"="TABLE_SCHEMA"
            AND TC."TABLE_NAME"="TABLE_NAME")
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

上記のSELECT 文では，次の日時のうち，最も古い日時のチャンクが検索対象となります。

- チャンクが作成された日時
- バックグラウンドインポートで待機状態のチャンクとして作成されたあとで，adbchgchunkstatus コマンドで通常状態のチャンクに変更されたときの日時

ただし，次のチャンクは検索対象外です。

- 過去に通常状態になったことがない待機状態のチャンク

(8) 指定した期間内に作成されたチャンクやカレントチャンクから切り替わったチャンクのチャンク ID を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS"
  WHERE "TABLE_SCHEMA"=? AND "TABLE_NAME"=?
    AND (("CREATE_TIME" BETWEEN ? AND ?)
      OR ("SWAP_TIME" BETWEEN ? AND ?))
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を，3 番目の入力要求にはチャンクの作成日時の開始日時を，4 番目の入力要求にはチャンクの作成日時の終了日時を，5 番目の入力要求にはカレントチャンクが切り替わった日時の開始日時を，6 番目の入力要求にはカレントチャンクが切り替わった日時の終了日時を入力してください。

上記のSELECT 文では，次のチャンクが対象となります。

- 指定した期間内に作成されたチャンク
- 指定した期間内にカレントチャンクから切り替わったチャンク
- バックグラウンドインポートで待機状態のチャンクとして作成されたあとで，指定した期間内に adbchgchunkstatus コマンドでチャンクの状態が通常状態に変更されたチャンク

ただし，次のチャンクは検索対象外です。

- 過去に通常状態になったことがない待機状態のチャンク

(9) 指定した日時にデータが格納された可能性があるチャンクを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”CHUNK_ID” FROM ”MASTER”.”STATUS_CHUNKS”
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
     AND ”CREATE_TIME” <= ?
       AND (”SWAP_TIME” IS NULL OR ”SWAP_TIME” >= ?)
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，２番目の入力要求には表識別子を，３番目と４番目の入力要求には指定の日時を入力してください。

上記のSELECT 文では，次のチャンクが対象となります。

- 指定した日時以前に作成されたカレントチャンク
- 指定した日時以前に作成されて，かつ指定した日時以降にカレントチャンクから切り替わったチャンク
- バックグラウンドインポートで待機状態のチャンクとして作成されたあとで，指定した日時に adbchgchunkstatus コマンドでチャンクの状態が通常状態に変更されたチャンク

ただし，次のチャンクは検索対象外です。

- 過去に通常状態になったことがない待機状態のチャンク

(10) 作成されたチャンクの数を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT COUNT(*) FROM MASTER.STATUS_CHUNKS
  WHERE ”SCHEMA_NAME” = ? AND ”TABLE_NAME” = ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，２番目の入力要求には表識別子を入力してください。

(11) 指定したコメントが含まれているチャンクを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”CHUNK_ID”,”CHUNK_COMMENT” FROM ”MASTER”.”STATUS_CHUNKS”
  WHERE ”TABLE_SCHEMA”= ? AND ”TABLE_NAME”= ? AND ”CHUNK_COMMENT” LIKE ?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を，3 番目の入力要求にはチャンクに指定したコメントを入力してください。

(12) 削除仕掛中のチャンクを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS"
  WHERE "TABLE_SCHEMA"=? AND "TABLE_NAME"=?
    AND "CHUNK_STATUS"='Pending Delete'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(13) 通常状態のチャンクを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS"
  WHERE "TABLE_SCHEMA"=? AND "TABLE_NAME"=?
    AND "CHUNK_STATUS"='Normal'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(14) 待機状態のチャンクを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT "CHUNK_ID" FROM "MASTER"."STATUS_CHUNKS"
  WHERE "TABLE_SCHEMA"=? AND "TABLE_NAME"=?
    AND "CHUNK_STATUS"='Wait'
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

(15) チャンクがアーカイブ状態かどうかを調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT ”ARCHIVE_STATUS” FROM ”MASTER”.”STATUS_CHUNKS”
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=? AND ”CHUNK_ID”=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を，3 番目の入力要求にはチャンク ID を入力してください。

SELECT 文の実行結果にArchive が出力された場合は，チャンクがアーカイブ状態であることを意味しています。

(16)　カレントチャンクの情報を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT * FROM ”MASTER”.”STATUS_CHUNKS”
  WHERE ”TABLE_SCHEMA”=? AND ”TABLE_NAME”=?
    AND ”CREATE_TIME” IS NOT NULL AND ”SWAP_TIME” IS NULL
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。最初の入力要求にはスキーマ名を，2 番目の入力要求には表識別子を入力してください。

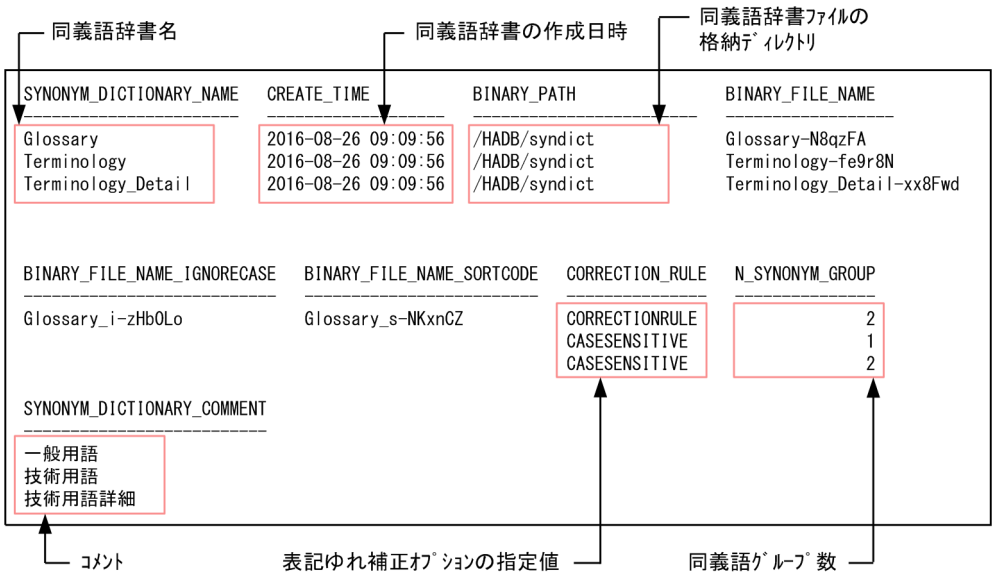
(17)　すべての同義語辞書に関する情報を調べる場合

次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT * FROM ”MASTER”.”STATUS_SYNONYM_DICTIONARIES”
```

検索結果の例



説明のない出力項目については，「付録 C.6　STATUS_SYNONYM_DICTIONARIES の内容」で説明されているSTATUS_SYNONYM_DICTIONARIES の列情報の説明を参照してください。

上記の検索結果の表示形式は，実際に出力される表示形式とは異なるところがあります。

(18)　特定の同義語辞書に関する情報を調べる場合

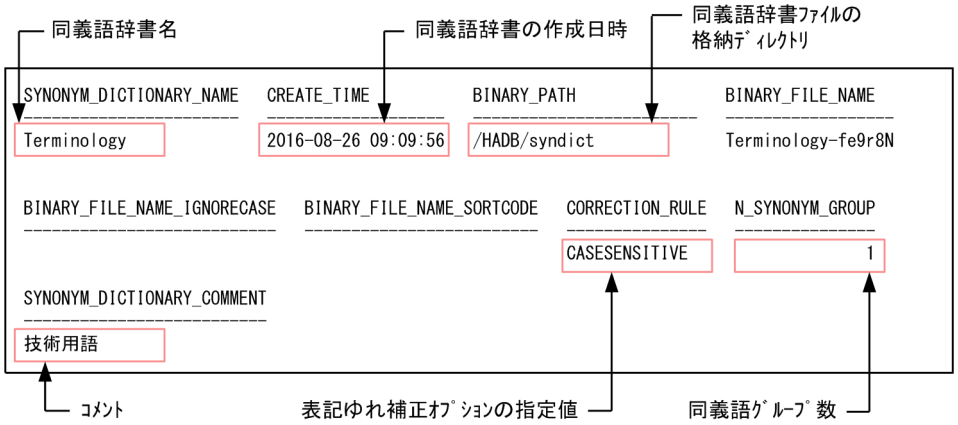
次に示すSELECT 文でシステム表を検索してください。

指定例

```
SELECT * FROM "MASTER"."STATUS_SYNONYM_DICTIONARIES"
  WHERE "SYNONYM_DICTIONARY_NAME"=?
```

adbsql コマンドで上記のSELECT 文を実行した場合，？パラメタに対する入力データの入力要求があります。同義語辞書名を入力してください。

検索結果の例



説明のない出力項目については，「付録 C.6　STATUS_SYNONYM_DICTIONARIES の内容」で説明されているSTATUS_SYNONYM_DICTIONARIES の列情報の説明を参照してください。

上記の検索結果の表示形式は，実際に出力される表示形式とは異なるところがあります。

付録 D　HADB の最大値と最小値

ここでは，システム構成およびデータベースに関する最大値と最小値について，一覧で示します。

付録 D.1　システム構成に関する最大値と最小値

HADB のシステム構成に関する最大値と最小値を次の表に示します。

表 D-1　HADB のシステム構成に関する最大値と最小値

項番	項目	最小値	最大値	単位
1	総 HADB ユーザ数	1	30,000	ユーザ
2	総 DB エリア数	5	1,024	個
3	マスタディレクトリ用 DB エリア数		1	個
4	ディクショナリ用 DB エリア数		1	個
5	システム表用 DB エリア数		1	個
6	作業表用 DB エリア数	1	12※1	個
7	データ用 DB エリア数	1	1,014	個
8	DB エリア名	1	30	バイト
9	DB エリアファイル相対パス名	1	100	バイト
10	ディクショナリ用 DB エリアの容量	−	16	テラバイト
11	システム表用 DB エリアの容量	−	16	テラバイト
12	作業表用 DB エリアの容量	−	作業表用 DB エリアのページサイズ×4,294,967,295	バイト
13	データ用 DB エリアの容量	−	128	エクサバイト
14	データ用 DB エリアの DB エリアファイルサイズ	−	128	ペタバイト
15	データ用 DB エリアの 1DB エリア中の構成ファイル数	1	1,024	個
16	1 作業表のサイズ	−	作業表用 DB エリアのページサイズ×4,294,967,295	バイト
17	HADB ユーザが定義できる実表の数	0	4,096	個
18	HADB ユーザが定義できるビュー表の数	0	30,000	個
19	HADB ユーザが定義できるインデクス数	0	8,192	個
20	1DB エリア中の実表の数	0	200	個

項番	項目	最小値	最大値	単位
21	1DB エリア中のインデクス数	0	400	個
22	最大文ハンドル数		4,095	個
23	最大同時接続数※2		1,024	個
24	ページサイズ（作業表用 DB エリア以外）	4,096	32,768	バイト
25	作業表用 DB エリアのページサイズ	32,768	33,554,432	バイト
26	マスタログファイル数		1	個
27	ユーザログファイル数	1	10,240	個
28	1DB エリア中のチャンク数	1	30,000	個
29	1 つの実表に作成できるチャンク数	1	30,000	個
30	マルチノード機能を使用する場合のノード数	2	12	ノード
31	プライマリノード数	1	1	ノード
32	セカンダリノード数	1	3	ノード
33	ワーカーノード数	0	8	ノード
34	HADB ユーザが登録できる同義語辞書の数	0	50	個

（凡例）

－：該当しません。

注※1

マルチノード機能を使用する場合の最大値になります。各ノードに作業表用 DB エリアを 1 個作成できるため，最大 12 個となります。

注※2

マルチノード機能を使用する場合，マルチノード構成の全 HADB サーバの最大同時接続数になります。

付録 D.2　データベースに関する最大値と最小値

HADB のデータベースに関する最大値と最小値を次の表に示します。

表 D-2　HADB のデータベースに関する最大値と最小値

項番	項目		最小値	最大値	単位
1	文字データの長さ（定義長）	CHAR	1	32,000	バイト
2		VARCHAR	1	64,000※3	バイト
3	数データの長さ	INTEGER		8	バイト

項番	項目		最小値	最大値	単位
4		SMALLINT	4		バイト
5		DOUBLE PRECISION	8		バイト
6		FLOAT			
7	数データの値の範囲	INTEGER	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	－
8		SMALLINT	-2,147,483,648	2,147,483,647	
9	固定小数点データの精度	DECIMAL	1	38	桁
10		NUMERIC			
11	固定小数点データの位取り	DECIMAL	0	38	桁
12		NUMERIC			
13	浮動小数点データの仮数部	DOUBLE PRECISION	17		桁
14		FLOAT			
15	浮動小数点データの指数部	DOUBLE PRECISION	3		桁
16		FLOAT			
17	日付データの値の範囲	DATE	DATE'0001-01-01'	DATE'9999-12-31'	－
18	時刻データの値の範囲	TIME	TIME'00:00:00.000000000000'	TIME'23:59:59.999999999999'	－
19	時刻印データの値の範囲	TIMESTAMP	TIMESTAMP'0001-01-01 00:00:00.000000000000'	TIMESTAMP'9999-12-31 23:59:59.999999999999'	－
20	時刻データの小数秒精度	TIME	0	12	桁
21	時刻印データの小数秒精度	TIMESTAMP	0	12	桁
22	バイナリデータの長さ（定義長）	BINARY	1	32,000	バイト
23		VARBINARY	1	32,000	バイト
24	配列データの最大要素数	ARRAY	2	30,000	個
25	ラベル付き間隔に指定できる値の範囲(YEAR(S))		-9,998	9,998	－
26	ラベル付き間隔に指定できる値の範囲(MONTH(S))		-119,987	119,987	－
27	ラベル付き間隔に指定できる値の範囲(DAY(S))		-3,652,058	3,652,058	－

項番	項目		最小値	最大値	単位
28	ラベル付き間隔に指定できる値の範囲(HOUR(S))		-87,649,415	87,649,415	−
29	ラベル付き間隔に指定できる値の範囲(MINUTE(S))		-5,258,964,959	5,258,964,959	−
30	ラベル付き間隔に指定できる値の範囲(SECOND(S))		-315,537,897,599	315,537,897,599	−
31	ラベル付き間隔に指定できる値の範囲(MILLISECOND(S))		-315,537,897,599,999	315,537,897,599,999	−
32	ラベル付き間隔に指定できる値の範囲(MICROSECOND(S))		-315,537,897,599,999,999	315,537,897,599,999,999	−
33	ラベル付き間隔に指定できる値の範囲(NANOSECOND(S))		-9,223,372,036,854,775,807	9,223,372,036,854,775,807	−
34	ラベル付き間隔に指定できる値の範囲(PICOSECOND(S))		-9,223,372,036,854,775,807	9,223,372,036,854,775,807	−
35	表中の列数		1	1,000	列
36	表中のインデクス数		0	32	個
37	表中の外部キーの数		0	255	個
38	主キーを参照する外部キーの数		0	255	個
39	制約の列数		1	16	列
40	B-tree インデクスの構成列数		1	16	列
41	レンジインデクスの構成列数		1		列
42	テキストインデクスの構成列数		1		列
43	レンジインデクスを定義するCHARACTER型の列の長さ		1	32	バイト
44	表のページ内未使用領域の比率		0	99	%（パーセント）
45	インデクスのページ内未使用領域の比率		0	99	%（パーセント）
46	識別子の長さ（認可識別子，スキーマ識別子，表識別子，インデクス識別子，列名，相関名，制約名）		1	100	バイト
47	SQL テキスト長		1	16,000,000※1	バイト
48	SQL 文中の表数		1	2,048	個
49	集合演算の数	すべてUNION の場合	0	1,023	個

項番	項目		最小値	最大値	単位
50		EXCEPT または INTERSECT を含む場合	0	63	個
51	SQL 文中のFULL OUTER JOIN の数		0	63	個
52	問合せ指定と表値構成子の指定数		1	1,024	個
53	ソートキーの数		0	64	個
54	グループ化列数		0	64	個
55	集まり導出表の列指定の指定数		1	1,000	個
56	表の結合数		1	64	個
57	表値構成子中の行値構成子の指定数		1	30,000	個
58	行値構成子中の行値構成子要素の指定数		1	1,000	個
59	選択式の数	スカラ副問合せ	1	1	個
60		IN 述語に指定する表副問合せ	1	1	個
61		EXISTS 述語に指定する表副問合せ	1	1,000	個
62		導出表の表副問合せ	1	1,000	個
63		INSERT～SELECT	1	1,000	個
64		上記以外	1	1,000	個
65	挿入列数		1	1,000	個
66	更新列数		1	1,000	個
67	？パラメタ数		0	1,000	個
68	IN 述語中の値式数		1	30,000	個
69	最大行数		0	18,446,744,073,709,551,615	件
70	値式中のスカラ演算と集合関数の数		0	500	個
71	値式を展開したあとの値式中のスカラ演算と集合関数の数		0	10,000	個
72	スカラ関数，ウィンドウ関数およびCASE式の入れ子の数		0	15	個
73	副問合せの入れ子の数		0	32	個
74	ウィンドウ関数のウィンドウ分割句に指定する値式の数		0	16	個

項番	項目	最小値	最大値	単位
75	SQL 文中のスカラ関数RANDOMCURSOR の識別番号の種類数	0	1,000	個
76	SQL 文中のスカラ関数RANDOMROW の識別番号の種類数	0	1,000	個
77	配列要素参照の添え字の指定値	1	30,000	－
78	配列要素参照の識別番号の指定値	1	255	－
79	作業表の最大行長	－	262,076※2	バイト
80	再帰的問合せの最大再帰数に指定できる値の範囲	0	32,767	－
81	問合せ指定中に集合関数として指定した，被集約引数が異なるDISTINCT 集合関数の数	0	64	個
82	問合せ指定中のARRAY_AGG 集合関数の指定数	0	256	個
83	問合せ指定中のLISTAGG 集合関数の指定数	0	64	個
84	SQL 文中の？パラメタのデータ長の合計	0	32,000,000	バイト
85	ビュー表のビューレベルの範囲	1	33	－
86	SQL メインノード数	1	1	ノード
87	SQL サブノード数	1	11	ノード

（凡例）

－：該当しません。

注※1

CREATE VIEW 文の場合は，64,000 バイトが最大値になります。

注※2

作業表用 DB エリアのページサイズが 262,144 バイト以下の場合，作業表用 DB エリアのページサイズから 68 バイトを引いた値が最大値になります。

注※3

次に示す個所に，データ長が 32,000 バイトを超えるVARCHAR 型は指定できません。

- ALTER TABLE 文の列定義に指定するデータ型
- CREATE TABLE 文の列定義に指定するデータ型
- 表関数列リストに指定するデータ型

- スカラ関数CAST に指定する変換後のデータ型
- スカラ関数CONVERT に指定する変換後のデータ型

付録 E　HADB サーバで起動するプロセス

adbstart コマンドを実行すると，次に示す名称の HADB サーバのサーバプロセスが起動します。

- adbsrvd

起動されるサーバプロセス数は 1 つで，adbstop コマンドを実行するとサーバプロセスが終了します。

なお，HADB のコマンドを実行すると，コマンドプロセスが起動します。コマンドプロセスの名称は，コマンドの名称と同じになります。

用語解説

数字

1 セグメントの容量

1 セグメントの大きさをメガバイト単位で表すときの用語です。

次に示すコマンドのオプションで，データ用 DB エリアの 1 セグメントの容量をメガバイト単位で指定します。

- adbinit コマンドの初期設定オプションadbinitdbarea の-s オプション
- adbmodarea コマンドの DB エリア追加・変更オプションadbaddarea の-s オプション

英字

CONNECT 権限

HADB ユーザが HADB サーバに接続する際に必要となるユーザ権限のことです。HADB サーバを利用するための前提となるユーザ権限であり，ほかの権限を持っていても，CONNECT 権限を持っていないと HADB サーバに接続できません。

DBA 権限

HADB ユーザおよび権限（ユーザ権限とスキーマ操作権限）を管理する際に必要となるユーザ権限のことです。

DB エリア

表やインデクス，およびそれらの定義情報などを格納する論理的な領域のことです。DB エリアは用途によって次のように分類されます。

- マスタディレクトリ用 DB エリア
- ディクショナリ用 DB エリア
- システム表用 DB エリア
- 作業表用 DB エリア
- データ用 DB エリア

DB エリアファイル

DB エリアを構成するファイルです。

DB ディレクトリ

DB エリアファイルやシステムログファイルなど，データベースに関するファイル群を格納するディレクトリのことです。

HADB 管理グループ

サーバマシンの OS に設定する HADB 専用のグループのことです。HADB 管理グループは，次に示す OS ユーザが所属するプライマリグループになります。

- HADB 管理者
- HADB 管理グループに所属する OS ユーザ

HADB 管理グループに所属する OS ユーザ

HADB 管理者とは別に，HADB を管理する OS ユーザのことです。HADB 管理者が所有するサーバディレクトリおよび DB ディレクトリにアクセスできます。また，HADB の一部のコマンドを実行できます。

HADB 管理者

HADB を管理する OS ユーザのことです。HADB 管理者はサーバディレクトリおよび DB ディレクトリの所有者になります。また，HADB のすべてのコマンドを実行できます。

HADB クライアント

クライアント・サーバシステムで動作する HADB の，クライアント側システムのことです。

HADB サーバ

クライアント・サーバシステムで動作する HADB の，サーバ側システムのことです。

HADB ユーザ

HADB サーバに接続して，データのインポート，データの検索・更新，HADB ユーザの管理などを実施できるユーザのことです。HADB ユーザには，各種操作の実施に必要な権限を付与できます。

OS ユーザ

OS にログインして，OS の機能を利用できるユーザのことです。HADB サーバの運用で使用する OS ユーザには，次の 3 種類があります。

- スーパユーザ
- HADB 管理者
- HADB 管理グループに所属する OS ユーザ

SQL サブノード

SQL パラレル実行機能を適用した検索系 SQL を実行した際に，SQL メインノードからの要求を受けて SQL 文の処理の一部を実行するノードのことです。プライマリノードの HADB サーバが SQL サブノードを決定します。

SQL 処理リアルスレッド

SQL 実行時に使用される処理リアルスレッドのことです。

SQL トレース機能

SQL 文の実行履歴情報（SQL トレース情報）をファイル（SQL トレースファイル）に出力する機能のことです。

SQL トレース情報

SQL トレース機能によって取得される SQL 文の実行履歴情報のことです。実行した SQL 文の実行時間や，アクセスパスの情報などが SQL トレース情報として出力されます。SQL トレース情報は次のことに使用します。

- SQL 文のエラー原因の調査
- SQL 文のチューニング

SQL パラレル実行機能

検索系 SQL（SELECT 文）の処理を複数のノードで並列実行する機能のことです。SQL パラレル実行機能を使用することによって，検索系 SQL の処理時間が短縮できます。SQL パラレル実行機能は，マルチノード機能を使用している場合に利用できます。

SQL メインノード

SQL パラレル実行機能を適用した検索系 SQL を実行した際に，検索系 SQL の処理を制御するノードのことです。また，HADB クライアントとのデータのやり取りも実行します。プライマリノードの HADB サーバが SQL メインノードを決定します。

ア行

アーカイブ状態

アーカイブされたデータがアーカイブファイルに格納されている状態のことです。

アーカイブディレクトリ

アーカイブファイルを格納するディレクトリです。

アーカイブファイル

アーカイブされたデータを格納しているファイルです。

アーカイブマルチチャンク表

バックグラウンドインポート機能およびチャンクアーカイブ機能を使用できる実表のことです。

アーカイブレンジ列

アーカイブマルチチャンク表の検索時に，検索範囲を絞り込むために使用する列のことです。アーカイブマルチチャンク表には，アーカイブレンジ列を必ず定義する必要があります。日時データの列などを，アーカイブレンジ列として使用できます。

アクセス権限

スキーマオブジェクト（実表，ビュー表）にアクセスする際に必要となる権限のことです。スキーマオブジェクトのデータを SQL 文で検索したり，スキーマオブジェクトをコマンドで操作したりするには，スキーマオブジェクトに対するアクセス権限が必要になります。アクセス権限には，SELECT 権限，INSERT 権限，IMPORT TABLE 権限など，幾つかの種類があります。

アンロード

無効な行データを除いたシステム表のデータを，アンロードファイルに出力する処理のことです。

アンロードファイル

システム表の再編成処理中に一時的に作成されるファイルのことです。

依存権限

ほかの HADB ユーザから付与されたアクセス権限のことです。

（例）

- HADB ユーザ A が表X.T1 に対して持っているアクセス権限をP1 とします。
- HADB ユーザ A が HADB ユーザ B に付与した，表X.T1 に対するアクセス権限をP2 とします。
- HADB ユーザ B が HADB ユーザ C に付与した，表X.T1 に対するアクセス権限をP3 とします。

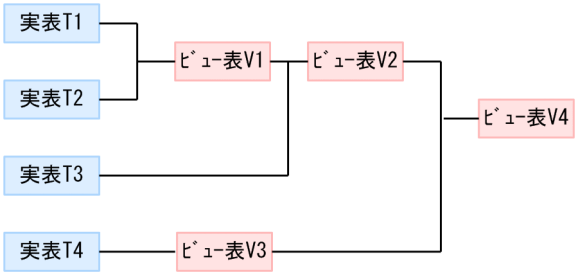
この場合，アクセス権限P1 の依存権限は，アクセス権限P2，P3 になります。アクセス権限P2 の依存権限は，アクセス権限P3 になります。アクセス権限P3 には，依存権限はありません。

例えば，アクセス権限P1 が取り消された場合，依存権限であるアクセス権限P2 とP3 も取り消されます。アクセス権限P2 が取り消された場合，依存権限であるアクセス権限P3 も取り消されます。

依存するビュー表

DROP TABLE 文による実表の削除や，DROP VIEW 文によるビュー表の削除などによって，影響を受けるビュー表のことです。依存するビュー表の例を次の図に示します。

依存するビュー表の例



[説明]

- 実表T1 に依存するビュー表は，ビュー表V1，V2，V4 になります。
- 実表T2 に依存するビュー表は，ビュー表V1，V2，V4 になります。
- 実表T3 に依存するビュー表は，ビュー表V2，V4 になります。
- 実表T4 に依存するビュー表は，ビュー表V3，V4 になります。
- ビュー表V1 に依存するビュー表は，ビュー表V2，V4 になります。
- ビュー表V2 に依存するビュー表は，ビュー表V4 になります。
- ビュー表V3 に依存するビュー表は，ビュー表V4 になります。
- ビュー表V4 に依存するビュー表はありません。

実表の削除や，ビュー表の削除以外にも，依存するビュー表が影響を受ける操作があります。依存するビュー表が影響を受ける操作については，「2.1.2　ビュー表」の「(3)　ビュー表の無効化」を参照してください。

一意性制約

列値（または複数の列の列値の組み合わせ）が重複することを許さない制約のことです。

インデクス再作成

実表に定義されたインデクスを再作成することです。インデクス再作成には，adbidxrebuild コマンドを使用します。

カ行

外部キー

ほかの実表の主キーを参照する列（または複数の列の組み合わせ）のことです。主キーと外部キーによって，データベース内の表と表の関係を定義できます。

カラムストア形式

データを列単位でデータベースに格納する形式のことをカラムストア形式といいます。カラムストア形式の場合，表の各列のデータが列ごとにまとまってデータベースに格納されます。

カラムストア表

表データの格納形式にカラムストア形式を選択して定義した表のことです。表データの格納形式には，ローストア形式とカラムストア形式の 2 つがあり，表を定義するときに表データの格納形式を指定することができます。

カレントチャンク

作成されたチャンクのうち，データが追加されるチャンクのことです。実表内に 1 つだけ存在します。

次の場合に，カレントチャンクが変更されます。

- バックグラウンドインポートを実行して新たなチャンクが作成された場合
- カレントチャンクを含む複数のチャンクをマージした場合

また，次の場合に，カレントチャンクが変更されることがあります。

- チャンクの状態を変更した場合

監査権限

監査権限とは，次に示す 2 つの権限の総称です。監査証跡機能を使用する場合に，監査権限が必要になります。

- 監査管理権限
  監査管理権限とは，監査証跡機能の運用を行う際に必要となる権限です。
- 監査参照権限
  監査参照権限とは，監査証跡を参照する際に必要となる権限です。

監査証跡機能

HADB ユーザがデータベースにアクセスした際のアクセス記録や，HADB ユーザが実行したコマンドの操作記録などを，監査証跡としてファイルに出力する機能のことです。例えば，HADB ユーザが表にアクセスした場合，アクセスした時刻，アクセスした HADB ユーザの認可識別子，操作内容，操作対象のスキーマオブジェクトなどが，操作記録（監査証跡）として出力されます。出力された監査証跡を参照すると，「いつ」，「誰が」，「どのような操作を実行し」，「どのスキーマオブジェクトにアクセスしたか」，などを調査できます。

監査証跡ファイル

HADB ユーザがデータベースにアクセスした際の操作記録や，HADB ユーザが実行したコマンドの実行記録などが格納されている監査証跡を出力するファイルのことです。

監査対象イベント

監査証跡の出力対象となる操作のことです。監査対象イベントには，監査証跡機能が有効な場合は監査証跡が必ず出力される必須監査イベントと，監査証跡を出力するかどうかを選択できる選択監査イベントがあります。

監査対象定義

監査証跡を出力するイベント，HADB ユーザ，スキーマオブジェクトの組み合わせを定義したものです。CREATE AUDIT 文で定義した情報を監査対象定義といいます。

監査人

監査権限を持っている HADB ユーザのことです。監査人は，監査証跡機能の運用や，出力された監査証跡を調査してデータベースの監査を実施します。

擬似スレッド

1 つのリアルスレッド上で複数の処理を並行に実行させるために，HADB 上で実装されているスレッドのことです。1 つのリアルスレッド上で，複数の擬似スレッドが動作することで処理を並列に実行します（同時に処理を実行できる擬似スレッドは 1 つだけです）。

基本行

分岐行が作成される場合に，分岐先の情報を持つ行のことです。分岐行が作成されない場合は，基本行には 1 行のデータが格納されます。

共有ディスクの SCSI リザーブ

共有ディスクのデータを保護するための，HA モニタの機能です。HA モニタによる共有ディスクのデータ保護方法には，次のどちらかを選択する必要があります。推奨する方法は，系のリセットです。

- 系のリセット
- 共有ディスクの SCSI リザーブ

系のリセットおよび共有ディスクの SCSI リザーブの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『系のリセット』および『共有ディスクの SCSI リザーブ』を参照してください。

クライアントグループ機能

複数の HADB クライアントを 1 つのグループにまとめて，グループで使用する接続数や処理リアルスレッド数の範囲を設定する機能のことです。

クライアント定義

HADB クライアントの実行環境を指定する定義のことです。

クライアント定義の集中管理機能

各 HADB クライアントで個々に管理しているクライアント定義を，HADB サーバ上で一括管理できる機能のことです。

クライアントディレクトリ

1 つのクライアントプロセスに関するファイル群を格納するディレクトリのことです。

グローバルバッファ

ディスク上の DB エリアに格納されているデータを入出力するためのバッファです。グローバルバッファは共有メモリ上に確保されます。

系のリセット

共有ディスクのデータを保護するための，HA モニタの機能です。HA モニタによる共有ディスクのデータ保護方法には，次のどちらかを選択する必要があります。推奨する方法は，系のリセットです。

- 系のリセット
- 共有ディスクの SCSI リザーブ

系のリセットおよび共有ディスクの SCSI リザーブの詳細については，マニュアル『HA モニタ Linux(R)(x86)編』の『系のリセット』および『共有ディスクの SCSI リザーブ』を参照してください。

権限

HADB ユーザが，HADB サーバに対して各種操作を実行する際に必要となります。HADB の権限は，大きく分けて次に示す 3 種類があります。

- ユーザ権限
- スキーマ操作権限
- アクセス権限

HADB ユーザは，複数の権限を持つことができます。必要な権限を持っていない場合，操作を実行できません。

検索系 SQL

表のデータを検索するときに使用する SQL 文のことです。SELECT 文が該当します。

更新行のカラム化機能

カラムストア表にローストア形式で格納されたデータを，HADB が自動的にカラムストア形式に変換する機能のことです。カラムストア表に対してINSERT 文またはUPDATE 文を実行すると，追加・更新されたデータはローストア形式でカラムストア表に格納されます。この機能を有効にすると，ローストア形式で格納されたデータが，自動的にカラムストア形式に変換されます。

更新系 SQL

表のデータを追加，削除および更新するときに使用する SQL 文の総称です。次に示す SQL 文が該当します。

- INSERT 文
- DELETE 文
- UPDATE 文
- PURGE CHUNK 文
- TRUNCATE TABLE 文

コールドスタンバイ構成

業務処理を行うサーバマシン（実行系）のほかに，待機用のサーバマシン（待機系）を準備するシステム構成のことです。実行系で障害が発生して業務が続行できない場合，待機系を実行系に切り替えて，業務を続行します。

コスト情報

HADB サーバが表の検索処理を最適化するために，実表，B-tree インデクスおよびテキストインデクスから収集した検索方法に関する情報のことです。adbgetcst コマンドを実行すると，コスト情報が収集されて，システム表に格納されます。

コネクションハンドル

コネクション（HADB クライアントと HADB サーバとの接続）を管理するオブジェクトです。

コマンドステータスファイル

コマンドの実行状況や中断状態などを管理するファイルです。

サ行

サーバディレクトリ

HADB サーバの実行に必要なファイルを格納するディレクトリのことです。1 つのサーバプロセスに関するファイル群を格納します。

作業表用 DB エリア

SQL の実行時に HADB が内部的に作成する表（作業表）を格納する DB エリアです。

削除仕掛中のチャンク

チャンクのマージ処理が中断した場合に，削除されないで残ってしまった不要なチャンク（マージ元チャンク）のことです。または，システム表（実表）の再編成処理が中断した場合に，削除されないで残ってしまった不要なチャンク（再編成前のデータ）のことです。

チャンクに含まれるデータは，操作系 SQL の対象外となります。ただし，PURGE CHUNK 文と TRUNCATE TABLE 文によるチャンクの削除はできます。なお，adbchgchunkstatus コマンドで，チャンクの状態を変更できません。

参照制約

表と表の関係（データの参照整合性）を保つ制約のことです。実表の定義時に，特定の列（外部キー）に定義できます。

参照表

参照制約，および外部キーを定義した実表のことです。

システム表

システム表（実表）から検索用に作成されるビュー表のことです。システム表のスキーマ名は MASTER です。

システム表の再編成

システム表内の無効な行データの領域を解放することです。

システム表用 DB エリア

システム表およびシステム表のインデクスを格納する DB エリアです。

システム表（実表）

次の情報を格納する表です。

- 実表，B-tree インデクスおよびテキストインデクスのコスト情報
- マルチチャンク表のチャンク情報

システム表（実表）を HADB ユーザは検索できません。システム表（実表）に格納された情報を確認する場合，システム表を検索します。

システムログ

HADB が自動的に取得するデータベースの更新履歴情報です。システムログは，障害発生時にデータベースを回復するために使用されます。

システムログファイル

システムログが出力されるファイルのことです。

主キー

表中の行を一意（ユニーク）に識別するための列（または複数の列の組み合わせ）のことです。主キーを定義することによって，実表のデータの一意性を保つことができます。主キーを定義すると，主キーを構成する列には，次の制約が適用されます。

- 一意性制約
- 非ナル値制約

処理リアルスレッド

HADB サーバの開始時に確保される，SQL 文，データインポートの実行時などに使用されるリアルスレッドのことです。

シングルチャンク表

1 表に 1 つのチャンクだけを作成できる実表のことです。シングルチャンク表の場合，バックグラウンドインポート機能，およびチャンクアーカイブ機能を使用できません。

スキーマ

表やインデクスを包括する論理的な概念です。HADB では，1HADB ユーザは 1 つのスキーマだけを所有できます。

スキーマオブジェクト

スキーマに定義できる要素のことです。表やインデクスが該当します。

スキーマ操作権限

HADB ユーザのスキーマを管理するときに必要となる権限のことです。スキーマ操作権限の種類を次に示します。

- スキーマ定義権限

スキーマ定義権限

HADB ユーザがスキーマ，表，およびインデクスを定義・削除する際に必要となるスキーマ操作権限のことです。

ステータスファイル

HADB サーバの稼働状況や終了モードなどを管理するファイルです。

セカンダリノード

マルチノード機能で，検索系 SQL や一部のコマンドを処理する HADB サーバがあるノードのことです。HA モニタの待機系に対応しています。

プライマリノードでノード障害が発生した場合，セカンダリノードの 1 つがプライマリノードに切り替わります。

セグメント

データの格納単位の 1 つで，ファイル内の領域を表やインデクスに割り当てる単位です。セグメントは連続したページから構成されます。

セグメントサイズ

1 セグメントの大きさをページ数単位で表すときの用語です。

セグメントサイズは，「*1 セグメントの容量÷ページサイズ*」で求められます。1 セグメントの容量については，「1 セグメントの容量」を参照してください。

操作系 SQL

表のデータを操作するときに使用する SQL 文（検索系 SQL，および更新系 SQL）の総称です。

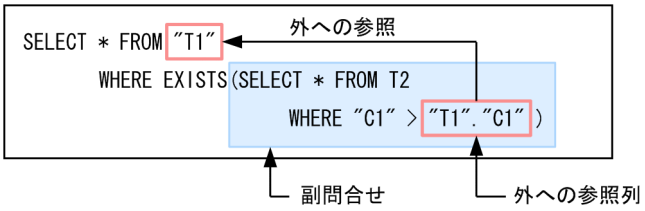
ソートコード

ISO/IEC14651:2011 で規定されている文字の並べ替えや比較を行う際に使用するコードのことです。表記ゆれ補正検索で，表記のゆれを補正して検索する際に使用します。

外への参照列

外側の問合せのFROM 句に指定した表を，副問合せ中の探索条件で参照している列のことです。副問合せ中の探索条件で参照することを，外への参照といいます。外への参照列の例を次の図に示します

外への参照列の例



タ行

待機状態のチャンク

チャンクに含まれるデータが操作系 SQL の対象外となるチャンクのことです。ただし，PURGE CHUNK 文とTRUNCATE TABLE 文によるチャンクの削除はできます。バックグラウンドインポートを実行するときに，作成されるチャンクの状態を指定することで作成されます。また，adbchgchunkstatus コマンドで，通常状態のチャンクに変更できます。なお，待機状態のチャンクは，カレントチャンクにはなりません。

チャンク

1 回のバックグラウンドインポートで実表に格納したデータの固まりのことです。バックグラウンドインポートで格納された実表のデータと，バックグラウンドインポート時に作成されたインデクスのデータで構成されます。バックグラウンドインポートを実行するたびに，新たなチャンクが作成されます。

チャンクには，次に示す状態があります。

- 通常状態のチャンク
- 待機状態のチャンク
- 削除仕掛中のチャンク

また，複数のチャンクをマージすることもできます（チャンクのマージ）。

チャンク ID

チャンクごとに付与される番号のことです。設定される番号は，実表内で一意となります。

チャンクアーカイブ機能

チャンク内のデータを圧縮し，アーカイブしたデータをアーカイブファイルに格納する機能のことです。

チャンクのアーカイブ

チャンクアーカイブ機能によってチャンク内のデータを圧縮し，アーカイブしたデータをアーカイブファイルに格納することを意味しています。

チャンクのアンアーカイブ

チャンクアーカイブ機能によってアーカイブファイル内のデータを伸長し，データ用 DB エリア内にデータを格納することを意味しています。このとき，インデクスも再作成されます。

チャンクのマージ

複数のチャンクを，1 つのチャンクにマージする（まとめる）ことです。チャンクのマージには，adbmergechunk コマンドを使用します。

なお，マージ対象の複数のチャンクを**マージ元チャンク**，マージした 1 つのチャンクを**マージ先チャンク**といいます。

通常状態のチャンク

チャンクに含まれるデータが操作系 SQL の対象となるチャンクのことです。バックグラウンドインポートを実行すると作成されます。また，adbchgchunkstatus コマンドで待機状態のチャンクに変更できます。

定義系 SQL

次に示す場合などに使用する SQL 文の総称です。

- スキーマ，表およびインデクスを定義，削除する場合
- HADB ユーザを作成，削除する場合
- HADB ユーザに権限を付与する場合
- HADB ユーザが持っている権限を取り消す場合

マニュアル『HADB　SQL リファレンス』の『定義系 SQL』で示す SQL 文が該当します。

ディクショナリ表

ディクショナリ表（実表）から検索用に作成されるビュー表のことです。ディクショナリ表のスキーマ名はMASTER です。

ディクショナリ表（実表）

表定義，列定義，インデクス定義といったデータベースの定義情報を格納する表です。ディクショナリ表（実表）を HADB ユーザは検索できません。ディクショナリ表（実表）に格納された情報を確認する場合，ディクショナリ表を検索します。

ディクショナリ用 DB エリア

ディクショナリ表およびディクショナリ表のインデクスを格納する DB エリアです。

データインポート

表にデータを一括して格納することです。データインポートには，adbimport コマンドを使用します。

データの再編成

表内の無効な行データを削除するために，表のデータをいったんエクスポートし，エクスポートしたデータをインポートすることです。データの再編成は，表単位（表の再編成），またはチャンク単位（チャンクの再編成）で実行できます。

データ用 DB エリア

表およびインデクスを格納する DB エリアです。

テーブルスキャン

B-tree インデクスおよびテキストインデクスを使用しないで，実表を検索する方式のことです。

テキストインデクス

表に格納されているデータ（文字列）の出現情報を管理するインデクスのことです。このインデクスは，任意の文字列を含む文字データを検索する場合に使用できます。

統一フォーマット用監査証跡ファイル

JP1/Audit で収集して参照・管理できる形式の監査証跡が出力されたファイルのことです。監査証跡ファイルに出力された監査証跡を，adbconvertaudittrailfile コマンドで変換すると，変換された監査証跡が統一フォーマット用監査証跡ファイルに出力されます。

同義語一覧定義ファイル

同義語辞書に登録する同義語の一覧を記述するファイルです。同義語検索をする場合，同義語一覧定義ファイル中の同義語グループに登録されているすべての同義語が同義語検索の対象になります。

同義語グループ

同義語検索の対象とする同義語の集まりのことです。

同義語検索

文献や論文などの文書データを検索する際，検索対象に指定した語句と，その語句の同義語をまとめて検索する機能のことです。

同義語辞書

同義語の一覧が登録されている辞書のことです。

同義語辞書ファイル

同義語辞書の情報を格納したファイルです。同義語検索の実行時，HADB サーバが同義語辞書ファイルの内容を参照します。

同義語辞書ファイルの格納ディレクトリ

同義語辞書ファイルを格納するディレクトリのことです。サーバ定義の adb_syndict_storage_path オペランドに指定したディレクトリのことです。

統計ログファイル

HADB サーバの稼働情報の一部が出力されるファイルのことです。出力された情報は，adbstat コマンドで確認できます。次の情報が出力されます。

- SQL 文の統計情報

ナ行

認可識別子

HADB に接続するときに使用するユーザ ID のことです。

ノード

HADB サーバがインストールされているサーバマシンと，そのサーバマシンにインストールされている OS，HADB サーバ，および HA モニタを包括したものです。プライマリノードおよびセカンダリノードの場合は，HA モニタの系に対応しています。

ノード障害

プライマリノードまたはセカンダリノードの場合，マルチノード構成からノード（プライマリノードまたはセカンダリノード）の切り離し対象となる障害を意味しています。HA モニタの系障害およびサーバ障害に対応しています。

ワーカーノードの場合，ワーカーノードの HADB サーバが異常終了して，マルチノード構成からワーカーノードが削除される障害を意味しています。

ノードの切り離し

ノード障害が発生したプライマリノードまたはセカンダリノードは，マルチノード構成から切り離されます。これを**ノードの切り離し**といいます。

ノードの復帰

ノード障害が発生してマルチノード構成から切り離されたノード（プライマリノードまたはセカンダリノード）を，障害対策後にマルチノード構成に復帰させることです。マルチノード構成に復帰する際，ノード種別は必ずセカンダリノードになります。

ハ行

バックグラウンドインポート

adbimport コマンドのオプション機能の 1 つです。adbimport コマンド実行時に適用すると，同一表に対してデータの検索とデータインポートが同時に実行できます。

ハッシュ検索

ハッシュテーブル領域を使用する処理のことです。次に示す処理が該当します。

- 表の結合方式のハッシュジョイン
- グループ化の処理方式のグローバルハッシュグループ化
- 副問合せの処理方式のハッシュ実行
- SELECT DISTINCT の処理方式のハッシュ実行
- 集合演算の処理方式のハッシュ実行

被参照表

外部キーによって参照表から参照される，主キーを定義した実表のことです。

非順序実行方式

HADB で採用されているデータベースの検索処理方式です。非順序実行方式では，検索処理が行単位に分割され，並列に実行されます。

非ナル値制約

列値にナル値を許さない制約のことです。

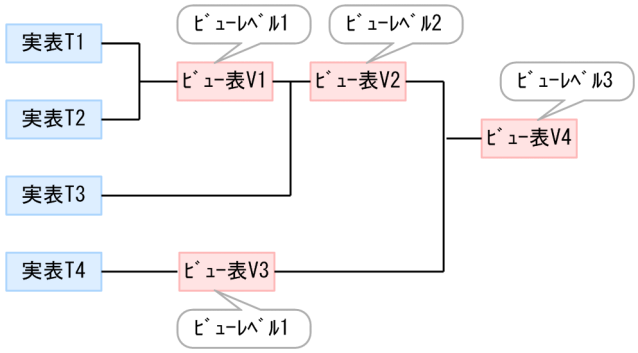
ビュー表

CREATE VIEW 文に指定した問合せ式の結果を，新たな表として定義した仮想的な表のことです。複数の表に対する集合演算の結果をビュー表として定義したり，表中の特定の行や列だけを取り出してビュー表として定義したりすることができます。

ビューレベル

定義したビュー表が，実表から何番目の階層になるのかを示す値のことです。ビューレベルの例を次の図に示します。

ビューレベルの例



[説明]

- ビュー表V1 の基表はすべて実表のため，ビュー表V1 のビューレベルは 1 になります。
- ビュー表V2 の基表は，実表とビューレベル 1 のビュー表のため，ビュー表V2 のビューレベルは 2 になります。
- ビュー表V3 の基表は実表のため，ビュー表V3 のビューレベルは 1 になります。
- ビュー表V4 の基表は，ビューレベル 2 のビュー表とビューレベル 1 のビュー表のため，ビュー表V4 のビューレベルは 3 になります。

表関数導出表

次のシステム定義関数によって導出された表形式のデータ集合のことです。

- ADB_AUDITREAD 関数
- ADB_CSVREAD 関数

表記ゆれ補正検索

テキストデータ中の文字列に「大文字と小文字」，「半角文字と全角文字」，「ひらがなとカタカナ」などの表記ゆれがある場合に，それらの文字列を一括して検索する機能のことです。

付与権

ほかの HADB ユーザにアクセス権限を付与できる権限のことです。

プライマリノード

マルチノード機能の制御を行う HADB サーバがあるノードのことです。HA モニタの実行系に対応しています。

プライマリノードの HADB サーバが，AP またはコマンドの接続要求を受け付けて，負荷の少ないセカンダリノードまたはワーカーノードに検索系 SQL の処理を振り分けます。

更新系 SQL，定義系 SQL，または`adbimport` コマンドなど，データベースの更新が発生する処理はプライマリノードで実行されます。

プライマリノードの切り替え

プライマリノードでノード障害が発生した場合，そのプライマリノードをマルチノード構成から切り離し，セカンダリノードの 1 つをプライマリノードに切り替えることです。

分岐行

基本行が格納されたページとは異なるページに格納された可変長列データ（行）を分岐行といいます。

文ハンドル

実行する SQL を管理するためのオブジェクトです。

ページ

データの格納単位の 1 つで，ディスク I/O 時の最小単位です。ディスク上のデータを読み込むときや，ディスク上にデータを書き込むときは，ページ単位で実行されます。

マ行

マスタディレクトリ

HADB が DB エリアに関する各種情報などを管理するために使用する，システムの内部情報です。

マスタディレクトリ用 DB エリア

マスタディレクトリを格納する DB エリアです。

マルチチャンク表

1 表に複数のチャンクを作成できる実表のことです。マルチチャンク表の場合，バックグラウンドインポート機能を使用できます。マルチチャンク表には，レギュラーマルチチャンク表とアーカイブマルチチャンク表の 2 種類があります。

マルチノード機能

複数の HADB サーバを連携することによって，検索系 SQL の処理の負荷分散を実現する機能のことです。

マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ

マルチノード機能の使用時に同義語検索を行う際に使用される同義語辞書ファイルを格納するディレクトリのことです。サーバ定義のadb_syndict_node_storage_path オペランドに指定したディレクトリのことです。

マルチノード構成

マルチノード機能を使用している複数の HADB サーバから成るシステム構成のことです。

マルチノード構成からのワーカーノードの削除

ワーカーノードの HADB サーバが異常終了した場合，そのワーカーノードはマルチノード構成から削除されます。これを**マルチノード構成からのワーカーノードの削除**といいます。

マルチノード構成へのワーカーノードの追加

マルチノード構成から削除されたワーカーノードを再びマルチノード構成に追加することができます。これを**マルチノード構成へのワーカーノードの追加**といいます。

無効な行データ

削除した行のデータや，更新した行の更新前のデータは，ディスク上から削除されません。このように，ディスク上には存在しますが，検索の対象にはならない行データのことです。行の削除および行の更新を繰り返し実行した場合，表内に無効な行データが増加します。無効な行データが増加すると，次のような影響があります。

- 無効な行データが増加した影響で，データの格納効率が悪くなる
- 検索時に参照するページ数が増えるため，検索性能が低下する

メッセージカタログファイル

HADB が出力するメッセージのテキストを格納したメッセージオブジェクトファイルです。

メッセージログファイル

HADB が出力するメッセージを格納するファイルのことです。

基表

ビュー表の基になる表のことです。CREATE VIEW 文の問合せ式中に指定した表が，基表になります。

ヤ行

ユーザ権限

HADB ユーザが，HADB ユーザを管理したり，HADB サーバに接続したりするときに必要な権限のことです。ユーザ権限の種類を次に示します。

- DBA 権限
- CONNECT 権限

ラ行

リアルスレッド

OS（カーネルとライブラリ）によって管理されるスレッドです。

リロード

システム表の再編成で，アンロードファイル中のデータをシステム表に再格納することです。

レギュラーマルチチャンク表

マルチチャンク表の基本機能であるバックグラウンドインポート機能を使用できる実表のことです。レギュラーマルチチャンク表の場合，チャンクアーカイブ機能は使用できません。

レンジインデクス

表のチャンク内およびセグメント内に格納されているデータの列の値の最小値と最大値（値域）を管理するインデクスのことです。

ローストア形式

データを行単位でデータベースに格納する形式のことをローストア形式といいます。ローストア形式の場合，1 行のデータが 1 レコードとしてデータベースに格納されます。

ローストア表

表データの格納形式にローストア形式を選択して定義した表のことです。表データの格納形式には，ローストア形式とカラムストア形式の 2 つがあり，表を定義するときに表データの格納形式を指定することができます。

ロケーション表

チャンク内のデータをアーカイブしたときに，HADB サーバによって自動的に作成される表です。ロケーション表には，次に示す情報が格納されています。

- チャンク ID
- アーカイブファイルのパス

- アーカイブファイルごとのアーカイブレンジ列の値の範囲（最大値と最小値）

ロケーション表は，HADB サーバが使用するための表で，ユーザが使用することはありません。

1 つのアーカイブマルチチャンク表に対して，1 つのロケーション表が作成されます。

ワ行

ワーカーノード

マルチノード機能で，検索系 SQL や一部のコマンドを処理する HADB サーバがあるノードのことです。

ワーカーノードは，セカンダリノードとは異なり，プライマリノードの予備のノードにはなりません。そのため，プライマリノードで障害が発生しても，ワーカーノードがプライマリノードになることはありません。

ワード検索

英語で書かれたテキストデータから，検索対象に指定した英単語を高速に検索する機能のことです。ワード検索には，次の 2 つの検索方法があります。

- 完全一致検索
- 前方一致検索

索引

記号

？パラメタの情報〔SQL トレース情報〕 1109
/sys/block/デバイス名/queue/add_random〔カーネルパラメタ〕 456
/sys/block/デバイス名/queue/nr_requests〔カーネルパラメタ〕 456
/sys/block/デバイス名/queue/rotational〔カーネルパラメタ〕 456
/sys/block/デバイス名/queue/rq_affinity〔カーネルパラメタ〕 456
/sys/block/デバイス名/queue/scheduler〔カーネルパラメタ〕 456

数字

1 セグメントの容量〔用語解説〕 2160
1 セグメントの容量を決める際の考慮点〔データ用 DB エリア〕 377

A

actcommand オペランド
　コールドスタンバイ構成 1961
　マルチノード機能 1794
acttype オペランド
　コールドスタンバイ構成 1961
　マルチノード機能 1794
adb_audit_log_max_num 875
adb_audit_log_max_size 875
adb_audit_log_path 875
ADB_AUDITREAD 関数を使用した監査証跡の確認 1465
adb_blk_path_wrk 838
adb_core_path 838
ADB_CSVREAD 関数 185
adb_db_path 838
adb_dbarea_wrk_page_size 838
adb_dbarea_wrn_pnt_except_name 862
adb_dbarea_wrn_pnt_use_rate 862
adb_dbbuff_wrktbl_clt_blk_num 842
adb_dbbuff_wrktbl_glb_blk_num 842
adb_log_rec_msg_interval 859
adb_log_usrbuf_num 859
adb_log_usrfile_num 859
adb_log_usrfile_size 859
adb_rpc_port 838
adb_rpc_tcp_keepalive_time 862
adb_rpc_wait_time 862
adb_sql_default_dbarea_shared 867
adb_sql_exe_hashflt_area_size 842
adb_sql_exe_hashgrp_area_size 842
adb_sql_exe_hashtbl_area_size 842
adb_sql_exe_max_rthd_num 842
adb_sql_opt_drvtbl_grping_prior 842
adb_sql_opt_hashjoin_div 842
adb_sql_order_mode 867
adb_sql_prep_dec_div_rs_prior 867
adb_sql_prep_delrsvd_words 867
adb_sql_rngidx_preread 872
adb_sql_tbldef_cache_size 842
adb_sql_text_out 867
adb_sql_trc_accesspath 867
adb_sql_trc_level 867
adb_sql_trc_out 867
adb_sql_trc_param 867
adb_sql_trc_txtfile_size 867
adb_sta_log_max_size 873
adb_sta_log_path 873
adb_sta_log_size_unit 873
adb_syndict_node_storage_path 874
adb_syndict_storage_path 874
adb_sys_max_parallel_exec_num 876
adb_sys_max_users 838
adb_sys_max_users_wrn_pnt 862
adb_sys_memory_limit 842
adb_sys_memory_limit_wrn_pnt 862
adb_sys_multi_max_worker_nodes 876

adb_sys_multi_node_info 876
adb_sys_multi_worker_node_info 876
adb_sys_node_start_wait_time 876
adb_sys_proc_area_max 842
adb_sys_rthd_area_max 842
adb_sys_rthd_num 842
adb_sys_shm_huge_page_size 842
adb_sys_trn_iso_lv 867
adb_sys_uthd_num 842
adbarchivechunk コマンド
メモリ所要量の事前見積もり 319
adbarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 761
adbarchivechunk コマンド実行時のメモリ所要量の求め方 660
adbarchivechunk コマンドの実行時間の短縮〔チューニング〕 1689
adbaudittrail コマンド
--start オプション 1459
--stop オプション 1479
--swap オプション 1473
--write-error オプション 1459
-d オプション 1474
メモリ所要量の求め方 705
adbaudittrail コマンド実行時のメモリ所要量の求め方 705
adbbuff 879
adbbuff オペランドの指定を見直す〔バージョンアップ後の作業〕 948
adbchgchunkcomment コマンド
チャンクのコメントを変更する方法 1257
adbchgchunkcomment コマンド実行時のメモリ所要量の求め方 658
adbchgchunkstatus コマンド
チャンクの状態変更によるカレントチャンクの切り替え 1262
チャンクの状態を変更する運用例 1279
チャンクの状態を変更する方法 1258
adbchgchunkstatus コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 760
adbchgchunkstatus コマンド実行時のメモリ所要量の求め方 658
adbchgchunkstatus コマンドによるチャンクの状態変更 1258
adbclientdefmang コマンド実行時のメモリ所要量の求め方 701
adbclientmang オペランド
クライアント管理定義ファイル 1387
ADBCLTDIR〔環境変数〕 927
adbcltgrp（クライアントグループを設定する場合） 889
adbcltgrp（コマンドグループを設定する場合） 889
ADBCLTLANG〔環境変数〕 927
adbcolumnize コマンド 1440
adbconvertaudittrailfile コマンド実行時のメモリ所要量の求め方 707
adbconvertaudittrailfile コマンドを実行する際の注意事項 1503
adbdbstatus コマンド
B-tree インデクスの状態や使用量を確認したい場合 1075
DB エリアファイルの使用量を確認したい場合 1079
アーカイブファイルのサイズを確認したい場合 1080
システム表（実表）の状態や使用量を確認する 1434
実表の状態や使用量を確認したい場合 1075
チャンクの状態や作成されているチャンク数を確認する方法 1243
データベースの状態や使用量の確認 1074
データベースの使用量を確認したい場合 1074
テキストインデクスの状態や使用量を確認したい場合 1077
レンジインデクスの状態や使用量を確認したい場合 1078
adbdbstatus コマンド実行時のメモリ所要量の求め方 632
ADBDIR〔環境変数〕 927
adbexport コマンド
実表のデータをファイルに出力する方法（データをエクスポートする方法） 1179
チャンク単位にデータをエクスポートする方法 1241

ビュー表のデータをファイルに出力する方法（データをエクスポートする方法） 1205
メモリ所要量の事前見積もり 317
adbexport コマンドが出力するファイルの容量見積もり 783
adbexport コマンド実行時のメモリ所要量の求め方 634
adbexport コマンドの実行時間の短縮〔チューニング〕 1687
adbgetcst コマンド
メモリ所要量の事前見積もり 316
adbgetcst コマンド実行時のメモリ所要量の求め方 627
adbgetcst コマンドの実行時間の短縮〔チューニング〕 1686
adbidxrebuild コマンド
コマンドが再実行できない場合の対処方法 1722
作業用一時ファイルの容量見積もり 791
作業用一時ファイルを格納するディスク容量が小さい場合 1726
実表が更新不可状態になった場合 1753
メモリ所要量の事前見積もり 314
メモリ所要量の求め方 612
adbidxrebuild コマンドが再実行できない場合の対処方法
KFAA50244-E メッセージ，および KFAA50247-E メッセージ以外のメッセージが出力されたとき 1723
KFAA50244-E メッセージが出力されたとき 1722
KFAA50247-E メッセージが出力されたとき 1722
adbidxrebuild コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 739
adbidxrebuild コマンドの実行時間の短縮〔チューニング〕 1685
adbimport コマンド
コマンドが再実行できない場合の対処方法 1722
作業用一時ファイルの容量見積もり 787
作業用一時ファイルを格納するディスク容量が小さい場合 1726
実表が更新不可状態になった場合 1753
実表にデータを格納する方法（データインポート） 1177
メモリ所要量の事前見積もり 313
adbimport コマンドが再実行できない場合の対処方法
KFAA50244-E メッセージ，および KFAA50247-E メッセージ以外のメッセージが出力されたとき 1723
KFAA50244-E メッセージが出力されたとき 1722
KFAA50247-E メッセージが出力されたとき 1722
adbimport コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 729
adbimport コマンド実行時の作業用一時ファイル
インデクス作成用の作業用一時ファイル 787
データ圧縮用の作業用一時ファイル 787
adbimport コマンド実行時のメモリ所要量の求め方 588
adbimport コマンドの実行時間の短縮〔チューニング〕 1685
adbinfoget コマンド〔トラブルシュート情報の取得〕 1712
adbinfosweep コマンド〔トラブルシュート情報の削除〕 1715
adbinit コマンド実行時に指定した作業表用 DB エリアのページサイズを調べる 2131
adbinit コマンド実行時のメモリ所要量の求め方 587
ADBLANG〔環境変数〕 927
adbmergechunk コマンド
作業用一時ファイルの容量見積もり 795
作業用一時ファイルを格納するディスク容量が小さい場合 1729
待機状態のチャンクをマージする場合の注意点 1251
チャンクをマージする方法（チャンク数を減らす方法） 1247
メモリ所要量の事前見積もり 318
メモリ所要量の求め方 642
adbmergechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 743
adbmergechunk コマンド実行時に増加するデータ量の見積もり 805
adbmergechunk コマンド実行中に検索対象となるチャンク 1251

adbmergechunk コマンドによるチャンクのマージ 1248
adbmergechunk コマンドの--purge-chunk オプション指定時の考慮点 1255
adbmergechunk コマンドの実行時間の短縮〔チューニング〕 1688
adbmodarea コマンド
データ用 DB エリアが満杯になった場合 1747
データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加） 1366
データ用 DB エリアを拡張できなくなった場合1751
データ用 DB エリアを再初期化する方法 1367
データ用 DB エリアを削除する方法 1366
データ用 DB エリアを追加する方法 1365
データ用 DB エリアを追加できなくなった場合1747
メモリ所要量の求め方 638
adbmodbuff コマンド
ローカル作業表用バッファを変更する方法 1381
ADBMSGLOGSIZE〔環境変数〕 927
adbreorgsystemdata コマンド
adbreorgsystemdata コマンドと排他制御の関係 1432
adbreorgsystemdata コマンドの待ち状態を解消する方法 1433
アンロードファイルに関するトラブル 1773
アンロードファイルを格納するディスクの空き容量が不足したときの対処方法 1773
アンロードファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法 1773
アンロードファイルを格納するディスク容量が小さい場合 1775
作業用一時ファイルの容量見積もり 801
作業用一時ファイルを格納するディスク容量が小さい場合 1729
システム表（実表）の格納効率を確認する 1435
システム表の再編成 1429
システム表を再編成するタイミング 1430
システム表を再編成する方法 1431
システム表を再編成する理由 1429
ディスクに不要なアンロードファイルが残っている 1774
メモリ所要量の求め方 686
adbreorgsystemdata コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 767
adbreorgsystemdata コマンド実行時の作業用一時ファイルの容量見積もり 801
adbreorgsystemdata コマンド実行時のメモリ所要量の求め方 686
ADBSQLNULLCHAR〔環境変数〕 927
adbsql コマンド
実表のデータをファイルに出力する方法（データをエクスポートする方法） 1179
チャンク単位にデータをエクスポートする方法1241
ビュー表のデータをファイルに出力する方法（データをエクスポートする方法） 1205
adbsql コマンドの実行〔Docker および Kubernetes を使用した HADB の運用〕 2060
adbstart コマンド 1023
HADB サーバの再開始に掛かる処理時間が長い場合の対処方法 1779
adbstat コマンド
adbstat コマンド実行時に統計ログファイルから出力される情報の容量見積もり 1095
HADB サーバの統計情報の運用方法 1082
SQL 文の統計情報の運用方法 1088
グローバルバッファの統計情報の運用方法 1085
コネクションの稼働情報の運用方法 1083
統計解析の運用（HADB サーバの稼働情報の確認） 1082
統計ログファイルの運用 1092
adbstat コマンド実行時のメモリ所要量の求め方 638
adbstop コマンド 1024
adbstop 1024
adbstop --cancel 1025
adbstop --force 1024
adbstop --wait connection 1025
adbstop --wait transaction 1025
adbsyndict コマンド
作業用一時ファイルの容量見積もり 804
メモリ所要量の求め方 704
adbsyndict コマンド実行時の作業用一時ファイルの容量見積もり 804

adbsyndict コマンド実行時のメモリ所要量の求め方 704
ADBSYSLOGLV〔環境変数〕 927
adbunarchivechunk コマンド
コマンドが再実行できない場合の対処方法 1723
作業用一時ファイルの容量見積もり 798
作業用一時ファイルを格納するディスク容量が小さい場合 1729
メモリ所要量の事前見積もり 320
adbunarchivechunk コマンド実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 766
adbunarchivechunk コマンド実行時のメモリ所要量の求め方 667
address オペランド
コールドスタンバイ構成 1960
マルチノード機能 1793
aio-max-nr〔カーネルパラメタ〕 456
ALTER TABLE 文
アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる 2126
実表に列を追加する方法 1169
実表の表名を変更する方法 1172
実表の列名を変更する方法 1170
チャンク数の最大値を変更する方法 1256
ALTER VIEW 文
ビュー表を再作成する方法 1205
AP およびコマンドの処理状況を確認する方法 1069
AP が実行できない場合の対処方法 1718
AP が終了しない場合や AP が異常終了した場合の対処方法 1718
AP に関するトラブル 1718
AP の HADB サーバへの接続に時間が掛かる場合の対処方法 1718
AP の運用〔Docker および Kubernetes を使用した HADB の運用〕 2062
ARCHIVE CHUNK 権限 102
ARRAY_ROWSZ〔計算式〕 413
ARRAY 型の定義 1199
ARRAY 型の列の定義 346

B

B-tree インデクス 66
インデクスが格納されているデータ用 DB エリアを変更する方法 1231
インデクスページスプリット 356
状態や使用量を確認したい場合 1075
定義する列を決めるときの考慮点 351
未完状態になったときの対処方法 1760
B-tree インデクスのインデクスページスプリットの発生を抑える 1225
B-tree インデクスのキー長の求め方 419
B-tree インデクスの再作成 1224
B-tree インデクスの使用量を確認する 1076
B-tree インデクスのセグメントごとの格納ページ数の求め方 416
B-tree インデクスの設計 350
B-tree インデクスの定義〔CREATE INDEX 文〕 1223
B-tree インデクスの未完状態 350
未完状態かどうかを確認する 1075
B-tree インデクスを使用した検索性能の低下を防ぐ 1224
B-tree インデクスを定義する場合の注意点 350
BOM〔サーバ定義〕 903
BRANCH
BRANCH の指定有無を確認する場合 1190
BRANCH ALL
表オプション BRANCH ALL の指定有無を確認する場合 1190
BRANCH〔可変長データ型の列データの分岐指定〕 345

C

CHANGE CHUNK COMMENT 権限 102
CHANGE CHUNK STATUS 権限 102
CLASSPATH〔環境変数〕 927
CONNECT 権限 98
CONNECT 権限〔用語解説〕 2160
core_pattern〔カーネルパラメタ〕 456

CORRECTIONRULE
テキストインデクスの表記ゆれ補正指定の指定 363
CPU 障害〔マルチノード機能〕 1901
CREATE INDEX 文〔インデクスの定義〕 1223
CREATE TABLE 文〔実表の定義〕 1169
CSV ファイル中のデータの検索 185
マルチノード機能の使用時 1926
CSV ファイル中のデータを検索する方法 1397

D

DBA 権限 97
DBA 権限〔用語解説〕 2160
DB エリア 75
インデクスが格納されているデータ用 DB エリアを変更する方法 1231
実表が格納されているデータ用 DB エリアを変更する方法 1195
データ用 DB エリアの空き容量を確保する 1372
データ用 DB エリアの空き容量を確保する方法（実表の再編成） 1375
データ用 DB エリアの空き容量を確保する方法（データの圧縮） 1373
データ用 DB エリアの空き容量を確保する方法（不要なデータの削除） 1376
データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する方法 1370
データ用 DB エリアファイルの格納場所を変更する方法 1368
データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する方法 1368
データ用 DB エリアを拡張する方法（データ用 DB エリアファイルの追加） 1366
データ用 DB エリアを再初期化する方法 1367
データ用 DB エリアを削除する方法 1366
データ用 DB エリアを追加する方法 1365
DB エリア〔用語解説〕 2160
DB エリア ID
表が格納されている DB エリアの DB エリア ID を確認する場合 1190
DB エリアに格納されているインデクスのインデクス名を調べる 2119
DB エリアに格納されている表の表名を調べる 2119
DB エリアに関するトラブル 1747
DB エリアに表およびインデクスを格納できるケース 75
DB エリアに表またはインデクスを格納できないケース 75
DB エリアの格納位置の変更〔マルチノード機能〕 1870
DB エリアの構造（セグメントとページ） 77
DB エリアの自動増分 90
DB エリアの自動増分の失敗が原因の場合 1732
DB エリアの自動増分の発生防止〔チューニング〕 1673
DB エリアの種類 76
DB エリアのファイル構成 86
DB エリアの名称を確認する場合 1190
DB エリアファイル 86
使用量を確認したい場合 1079
DB エリアファイル〔用語解説〕 2160
DB エリアファイル管理部ページ 80
DB エリアファイル管理部用のセグメント 79
DB エリアファイルとマルチノード機能の関係 90
DB エリアファイルに障害が発生した場合の対処 1737
DB エリアファイルの構成例 89
DB ディレクトリ 88
DB ディレクトリ〔用語解説〕 2161
DB ディレクトリ下のファイルに関するトラブル 1737
DB ディレクトリの運用〔コールドスタンバイ構成〕 2013
DB ディレクトリの構成 2079
DB ディレクトリの事前見積もり 304
DB ディレクトリの初期化〔Docker および Kubernetes〕 2052
DEFAULT 句 341
DEFAULT 句の指定内容を確認する場合 1190
DELETE 権限 102
DELETE 文
実表のデータを検索・更新する方法 1173

DELIMITER
ワード検索時の区切り文字の選択 365
disk オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
dmmp_device オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
dockerd 2041
Dockerfile 2043
Docker イメージの生成 2043
Docker および Kubernetes を使用した HADB の運用 2036
Docker および Kubernetes を使用した HADB の運用例 2059
Docker および Kubernetes を使用したシステム構成
Docker および Kubernetes の使用停止 2063
運用例 2059
環境構築 2043, 2054
システム構成例 2038
前提ソフトウェアのセットアップ 2041

E

EX〔排他制御モード〕 129
EXCLUDE NULL VALUES〔ナル値除外指定〕 357
EXCLUSIVE〔排他制御モード〕 129
EXPORT TABLE 権限 102

F

fence_lan オペランド
コールドスタンバイ構成 1960
マルチノード機能 1793
fence_reset オペランド
コールドスタンバイ構成 1960
マルチノード機能 1793
fence_scsi オペランド
コールドスタンバイ構成 1960
マルチノード機能 1793
file-max〔カーネルパラメタ〕 456
FIX 表 340
FOREIGN KEY 343
fs_mount_dir オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
fs_mount_opt オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
fs_name オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
fs_neck オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794

G

GET COSTINFO 権限 102

H

HADB 管理グループ 910
HADB 管理グループ〔用語解説〕 2161
HADB 管理グループに所属する OS ユーザ 93, 911
HADB 管理グループに所属する OS ユーザ〔用語解説〕 2161
HADB 管理グループの設定 910
HADB 管理者 93, 910
HADB 管理者〔用語解説〕 2161
HADB 管理者の設定 910
HADB クライアント 51
HADB クライアント〔用語解説〕 2161
HADB クライアントで管理するクライアント定義ファイル 1389
HADB クライアント利用コンテナ 2039
HADB サーバ 51
HADB サーバの再開始に掛かる処理時間が長い場合の対処方法 1779
運用項目の一覧 1020
開始処理完了の確認 1023
開始方法 1023
開始モード 1023

稼働モード 1028
稼働モードを確認する方法 1030
稼働モードを変更する方法 1029
終了処理完了の確認 1027
終了方法 1024
終了モード 1024
バージョンアップ 933
バージョンアップ（動作確認テスト用） 994
バージョンダウン 978
バージョンダウン（動作確認テスト用） 994
HADB サーバ〔用語解説〕 2161
HADB サーバ開始時間の短縮〔チューニング〕 1664
HADB サーバ開始時にメモリが不足した場合の対処方法 1739
HADB サーバ開始時のメモリ所要量の求め方 472
HADB サーバが異常終了した場合の対処 1709
HADB サーバが異常終了していない場合の対処 1710
HADB サーバ起動コンテナ 2039
HADB サーバで起動するプロセス 2159
HADB サーバの稼働情報の確認 1082
HADB サーバの稼働モードの変更（通常モード） 1461
HADB サーバの再開始に関するトラブル 1779
HADB サーバの自動起動 924
HADB サーバの自動再起動 1030
HADB サーバの使用時に単調増加するファイル 708
HADB サーバの正常開始（オフラインモード） 1458
HADB サーバの正常終了〔監査証跡機能〕 1456
HADB サーバの統計情報の運用方法 1082
HADB サーバのプロセス 2159
HADB サーバのメモリ構成 232
HADB サーバのメモリ所要量の見積もり 464
HADB サーバを開始する 1010
HADB ダンプファイルの容量見積もり 713
HADB の最大値と最小値 2152
HADB のシステム構成 51
HADB ユーザ 94
HADB ユーザ〔用語解説〕 2161
HADB ユーザが参照できるシステム表の範囲 2135
HADB ユーザが参照できるディクショナリ表とシステム表の範囲 113
HADB ユーザが参照できるディクショナリ表の範囲 2085
HADB ユーザが持っているアクセス権限を確認する方法 1360
HADB ユーザが持っているアクセス権限を変更する方法 1363
HADB ユーザが持っているユーザ権限とスキーマ操作権限を確認する方法 1356
HADB ユーザが持っているユーザ権限とスキーマ操作権限を調べる 2120
HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法 1357
HADB ユーザにアクセス権限を付与する方法 1359
HADB ユーザに権限を付与する例 113
HADB ユーザにユーザ権限とスキーマ操作権限を付与する方法 1356
HADB ユーザの管理 1353
HADB ユーザを削除する方法 1354
HADB ユーザを作成する方法 1353
パスワードを変更する方法 1353
HADB ユーザを作成する 1011
HA モニタ 219
HA モニタ〔コールドスタンバイ構成〕
インストール 1959
環境設定 1960
HA モニタ〔マルチノード機能〕
インストール 1792
環境設定 1793
HugePages の適用〔チューニング〕 1664
HugePages を適用した共有メモリの使用量の削減〔チューニング〕 1691
hugetlb_shm_group〔カーネルパラメタ〕 456

I

IMPORT TABLE 権限 102
initial オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
INSERT 権限 102

INSERT 文
実表のデータを検索・更新する方法 1173
ip_neck オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
IP アドレスの変更
コールドスタンバイ構成 2022
マルチノード機能 1914
IP アドレスの変更〔サーバマシンの OS〕 1001
IT Report Utility 921
IT Report Utility の設定 921
ITRU 921

J

JP1/Audit 210
JP1/Audit Management - Manager との連携
監査証跡機能 1500
コールドスタンバイ構成 2034
マルチノード機能 1940

K

KeepAlive 862
KFAA30930-E メッセージが出力された場合
SQL 文実行時またはコマンド実行時 1742
KFAA40005-E メッセージが出力された場合
AP が実行できない 1718
コマンドが実行できない 1721
KFAA40007-E メッセージが出力された場合
HADB サーバ開始時 1739
SQL 文実行時またはコマンド実行時 1742
KFAA41205-E メッセージが出力された場合
再開始できない場合 1894
正常開始できない場合 1893
KFAA50244-E メッセージが出力されたとき
コマンドが再実行できない場合の対処方法 1722
KFAA50247-E メッセージが出力されたとき
コマンドが再実行できない場合の対処方法 1722
KFAA51246-E メッセージが出力された場合
バックグラウンドインポート機能に関するトラブル 1767
KFAA61205-W メッセージが出力された場合 1761
Kubernetes 2036

L

lan_updown オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
LANG〔環境変数〕 927
LD_LIBRARY_PATH〔環境変数〕 927
LVM 設定時の考慮点 454

M

memlock〔カーネルパラメタ〕 456
MERGE CHUNK 権限 102
monbegin_restart オペランド
コールドスタンバイ構成 1960
マルチノード機能 1793
multistandby オペランド
マルチノード機能 1793

N

name オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
nofile〔カーネルパラメタ〕 456
nproc〔カーネルパラメタ〕 456
nr_hugepages〔カーネルパラメタ〕 456

O

OS のコマンドでバックアップから回復する 1042
OS のコマンドでバックアップを取得する 1040
OS の時刻変更 997
OS のホスト名・IP アドレスの変更 1001
コールドスタンバイ構成 2022
マルチノード機能 1914
OS ユーザ 93

OS ユーザ〔用語解説〕 2161

P

PATH〔環境変数〕 927
patrolcommand オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
PCTFREE
B-tree インデクスのインデクスページ内の未使用領域の確保 354
PCTFREE の指定内容を確認する場合 1190
データページ内の未使用領域の確保 344
テキストインデクスのインデクスページ内の未使用領域の確保 360
PRIMARY KEY 342
PROTECTED UPDATE〔排他制御モード〕 129
PU〔排他制御モード〕 129
PURGE CHUNK 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 757

R

READ COMMITTED〔トランザクション隔離性水準〕 124
read-only transaction 128
read/write transaction 127
REBUILD INDEX 権限 102
REFERENCES 権限 102
REPEATABLE READ〔トランザクション隔離性水準〕 124
rmem_default〔カーネルパラメタ〕 456
rmem_max〔カーネルパラメタ〕 456
ROWSZ〔計算式〕 409

S

scsi_device オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
SELECT 権限 102
SELECT 文
実表のデータを検索・更新する方法 1173
SELinux のモードの確認 923
servers ファイルの指定〔コールドスタンバイ構成〕 1961
servers ファイルの指定〔マルチノード機能〕 1794
servers ファイルの指定例〔コールドスタンバイ構成〕 1967, 1976
servers ファイルの指定例〔マルチノード機能〕 1801, 1813, 1817
servexec_retry オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
set 形式〔サーバ定義〕 900
ShadowImage を使用するときの注意事項 1045
SHARED RETRIEVE〔排他制御モード〕 129
SHARED UPDATE〔排他制御モード〕 129
SHM_BUFGLOBAL 474
shmall〔カーネルパラメタ〕 456
SHMMAN 473
shmmax〔カーネルパラメタ〕 456
shmmni〔カーネルパラメタ〕 456
SQL_AUDITS の内容〔ディクショナリ表〕 2112
SQL_COLUMNS〔ディクショナリ表〕 2090
SQL_DBAREAS〔ディクショナリ表〕 2100
SQL_DEFINE_ENVIRONMENT の内容〔ディクショナリ表〕 2104
SQL_DEFINE_SOURCE の内容〔ディクショナリ表〕 2103
SQL_DIV_INDEX〔ディクショナリ表〕 2100
SQL_DIV_TABLE〔ディクショナリ表〕 2096
SQL_INDEX_COLINF の内容〔ディクショナリ表〕 2106
SQL_INDEXES〔ディクショナリ表〕 2096
SQL_KEY_COLUMN_USAGE の内容〔ディクショナリ表〕 2107
SQL_REFERENTIAL_CONSTRAINTS の内容〔ディクショナリ表〕 2107
SQL_SCHEMATA〔ディクショナリ表〕 2101
SQL_TABLE_CONSTRAINTS の内容〔ディクショナリ表〕 2106
SQL_TABLE_PRIVILEGES の内容〔ディクショナリ表〕 2108

SQL_TABLES〔ディクショナリ表〕 2087
チャンクの状態や作成されているチャンク数を確認する方法 1243
SQL_USERS の内容〔ディクショナリ表〕 2105
SQL_VIEW_TABLE_USAGE の内容〔ディクショナリ表〕 2102
SQL_VIEWS の内容〔ディクショナリ表〕 2102
SQL サブノード 229
SQL サブノード〔用語解説〕 2162
SQL 処理リアルスレッド〔用語解説〕 2162
SQL トレース機能〔用語解説〕 2162
SQL トレース機能の運用 1096
SQL トレース機能の使用をやめる場合 1161
SQL トレース情報
出力先ファイルの確認 1155
出力時の運用 1152
出力情報の確認 1153
出力情報の変更 1154
出力の開始 1155
出力の停止 1154
準備作業 1151
障害時の対処方法 1160
チューニング 1158
SQL トレース情報〔出力される情報〕 1097
SQL トレース情報〔用語解説〕 2162
SQL トレース情報の出力例 1143
SQL トレースファイル
バックアップの取得 1153
SQL トレースファイルのバックアップ 1153
SQL トレースファイルの容量見積もり 715
SQL パラレル実行機能 229
運用 1944
準備作業 1944
SQL パラレル実行機能〔用語解説〕 2162
SQL パラレル実行機能に関する情報〔SQL トレース情報〕 1112
SQL 文実行時またはコマンド実行時にメモリが不足した場合の対処方法 1742
SQL 文に関するオペランド（set 形式） 867
SQL 文の基本情報〔SQL トレース情報〕 1117
SQL 文の実行時間の短縮に関するチューニング 1675
SQL 文の実行時間を短縮する方法〔チューニング〕 1676
SQL 文の実行情報〔SQL トレース情報〕 1098
SQL 文の実行待ち状態の発生防止〔チューニング〕 1665
SQL 文の処理時間の短縮〔チューニング〕 1666
SQL 文の統計情報〔SQL トレース情報〕 1114
SQL 文の統計情報の運用方法 1088
SQL メインノード 229
SQL メインノード〔用語解説〕 2162
SQL メインノードと SQL サブノードを確認する場合 1864
SR〔排他制御モード〕 129
standbypri オペランド
マルチノード機能 1794
STATUS_CHUNKS の内容〔システム表〕 2139
チャンクの状態や作成されているチャンク数を確認する方法 1243
STATUS_COLUMNS の内容〔システム表〕 2137
STATUS_INDEXES の内容〔システム表〕 2139
STATUS_SYNONYM_DICTIONARIES の内容〔システム表〕 2141
STATUS_TABLES の内容〔システム表〕 2136
SU〔排他制御モード〕 129
SVP 227
sysdef ファイルの指定〔コールドスタンバイ構成〕 1960
sysdef ファイルの指定〔マルチノード機能〕 1793
sysdef ファイルの指定例〔コールドスタンバイ構成〕 1966, 1975
sysdef ファイルの指定例〔マルチノード機能〕 1800, 1811
syslog〔メッセージ〕 1050
syslog に出力されるメッセージの内容 1053
syslog のアクセス権限の変更 916
syslog の文字コード変換と信頼性向上（拡張 SYSLOG 機能の適用） 1054
syslog へのメッセージの出力抑止 1053
systemd-coredump が有効かどうかを確認する 914

T

termcmd_at_abort オペランド
コールドスタンバイ構成 1960
マルチノード機能 1793
termcommand オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
TEXT WORDCONTEXT
ワード検索用のテキストインデクスの指定 364
threads-max〔カーネルパラメタ〕 456
TLB 1664
TRUNCATE TABLE 文
実表の全行を削除する方法 1174
マルチチャンク表に格納されているデータを一括削除する 1243
TRUNCATE TABLE 文実行時に出力されるユーザログ容量（変数 max_user_log）の求め方 754
TRUNCATE 権限 102
TZ〔環境変数〕 927

U

UNARCHIVE CHUNK 権限 102
uoc_neck オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
UPDATE 権限 102
UPDATE 文
実表のデータを検索・更新する方法 1173
usrcommand オペランド
マルチノード機能 1793

V

VARBINARY 型の列データの分岐指定（BRANCH） 345
VARCHAR 型の列データの分岐指定（BRANCH） 345
vg_neck オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794

W

waitserv_exec オペランド
コールドスタンバイ構成 1961
マルチノード機能 1794
wmem_default〔カーネルパラメタ〕 456
wmem_max〔カーネルパラメタ〕 456

あ

アーカイブ状態〔用語解説〕 2162
アーカイブ状態のチャンク 179
アーカイブする方法 1302
アーカイブディレクトリ 179
アーカイブディレクトリのパス名を確認する場合 1190
アーカイブディレクトリ〔用語解説〕 2162
アーカイブディレクトリに関するトラブル 1769
アーカイブディレクトリの空き容量が不足したときの対処方法 1769
アーカイブファイルに障害が発生したときの対処方法 1769
アーカイブディレクトリの空き容量が不足したときの対処方法 1769
アーカイブディレクトリの事前見積もり 309
アーカイブディレクトリの容量見積もり 775
アーカイブファイル 179
アーカイブファイル〔用語解説〕 2162
アーカイブファイルに障害が発生したときの対処方法 1769
アーカイブマルチチャンク表 64
アーカイブする方法 1302
アーカイブディレクトリの容量見積もり 775
アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する方法 1308
アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる 2126
アンアーカイブする方法 1304
マルチチャンク表を削除する方法 1310
マルチチャンク表を定義する方法 1234
アーカイブマルチチャンク表〔用語解説〕 2163

アーカイブマルチチャンク表かどうかを調べる場合 2122
アーカイブマルチチャンク表のアーカイブディレクトリを調べる 2123
アーカイブマルチチャンク表のアーカイブレンジ列に定義されたレンジインデクスを調べる 2123
アーカイブマルチチャンク表のアーカイブレンジ列の列名を調べる 2123
アーカイブマルチチャンク表の検索 181
アーカイブマルチチャンク表のデータを検索・更新するときの注意事項 1237
アーカイブマルチチャンク表を定義する場合の考慮点 335
アーカイブレンジ列 181
アーカイブレンジ列かどうかを確認する場合 1190
アーカイブレンジ列〔用語解説〕 2163
アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる 2126
空き容量の確保〔データ用 DB エリア〕 1372
実表の再編成 1375
データの圧縮（チャンクのアーカイブ） 1373
不要なデータの削除 1376
アクセス権限 99
ARCHIVE CHUNK 権限 102
CHANGE CHUNK COMMENT 権限 102
CHANGE CHUNK STATUS 権限 102
DELETE 権限 102
EXPORT TABLE 権限 102
GET COSTINFO 権限 102
IMPORT TABLE 権限 102
INSERT 権限 102
MERGE CHUNK 権限 102
REBUILD INDEX 権限 102
REFERENCES 権限 102
SELECT 権限 102
TRUNCATE 権限 102
UNARCHIVE CHUNK 権限 102
UPDATE 権限 102
依存権限 104
ビュー表 111
複数のスキーマオブジェクトにアクセスする場合 110
付与権 104
アクセス権限〔用語解説〕 2163
アクセス権限の管理 1359
HADB ユーザが持っているアクセス権限を確認する方法 1360
HADB ユーザが持っているアクセス権限を変更する方法 1363
HADB ユーザにアクセス権限を付与する方法 1359
アクセス権限の付与権だけを取り消す方法 1362
アクセス権限を取り消す方法 1361
アクセス権限の種類 102
アクセス権限の伝搬 111
アクセス権限の取り消し 107
アクセス権限の付与 104
アクセス権限の付与権だけを取り消す方法 1362
アクセス権限を調べる 2121
アクセス権限を取り消す方法 1361
アクセス処理方式〔データベース〕 119
アクセスパス情報〔SQL トレース情報〕 1108
アクセスパス探索情報ログファイルの容量見積もり 717
アクセスパスの統計情報〔SQL トレース情報〕 1116
アクセスパスの統計情報の出力例と出力項目 1118
アクセス頻度の高い列と低い列がある場合〔表の正規化〕 339
圧縮方式〔カラムストア表〕 328
アンアーカイブする方法 1304
アンインストール〔HADB サーバ〕 1003
アンインストール後に実施する作業〔HADB サーバ〕 1004
アンインストール手順〔HADB サーバ〕 1003
アンロード〔用語解説〕 2163
アンロードファイル
アンロードファイルを格納するディスク容量が小さい場合 1775
アンロードファイル〔用語解説〕 2163

アンロードファイルに関するトラブル　1773
アンロードファイルを格納するディスクの空き容量が不足したときの対処方法　1773
アンロードファイルの格納ディレクトリの事前見積もり　309
アンロードファイルの容量見積もり　786
アンロードファイルを格納するディスクの空き容量が不足したときの対処方法　1773
アンロードファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法　1773

い

異常終了〔HADB サーバ〕　1024
依存権限　104
依存権限〔用語解説〕　2163
依存するビュー表　56
依存するビュー表〔用語解説〕　2163
依存するビュー表を確認する方法　1218
一意性制約　342, 356
一意性制約〔用語解説〕　2164
一意性制約違反状態〔ユニークインデクス〕　1761
一意性制約違反状態かどうかを確認する　1076
一意性制約違反状態かどうかを確認する方法　1762
一意性制約違反状態を解除する方法　1762
一意性制約定義　342
一意性制約に違反した場合の対処方法　1761
一括先読み〔レンジインデクス〕　370
イベントログ　154
入れ替え　987
インストール
HA モニタ〔コールドスタンバイ構成〕　1959
HA モニタ〔マルチノード機能〕　1792
インストール〔Docker 環境〕　2041
インストール〔HADB サーバ〕　908
インストール〔Kubernetes 環境〕　2042
インストール後に実施する作業　921
インストール手順〔HADB サーバ〕　916
インストール前に実施する作業　908
HADB 管理グループに所属する OS ユーザの設定　911
HADB 管理グループの設定　910
HADB 管理者の設定　910
syslog のアクセス権限の変更　916
サーバディレクトリを格納するディレクトリの作成　911
サーバディレクトリを格納するディレクトリの設定の変更　912
通信情報ファイルを格納するディレクトリの作成　912
インデクス
B-tree インデクス　66
インデクスが格納されているデータ用 DB エリアを変更する方法　1231
テキストインデクス　66
レンジインデクス　69
インデクス ID からインデクスが定義された表の表名を調べる場合　2116
インデクス ID からインデクスが定義された列の列名を調べる場合　2117
インデクス ID からインデクスのインデクス名を調べる場合　2116
インデクスオプションの確認
テキストインデクス　1228
インデクスが格納されている DB エリア名を調べる場合　2118
インデクス構成列 ID リストの形式　2096
インデクス再作成〔用語解説〕　2164
インデクスの一覧
システム表（実表）　2142
ディクショナリ表（実表）　2113
インデクスの再作成
B-tree インデクス　1224
テキストインデクス　1226
レンジインデクス　1228
インデクスの削除　1232
インデクスの種類　66
インデクスの定義〔CREATE INDEX 文〕　1223
インデクスの定義情報を調べる場合　2125
インデクスの未完状態
B-tree インデクス　350

テキストインデクス　362
レンジインデクス　371
インデクスの未完状態を解除する〔コールドスタンバイ構成〕　2011
インデクスの未完状態を解除する〔マルチノード機能〕　1875
インデクスページ〔B-tree インデクス〕　80
インデクスページ〔テキストインデクス〕　80
インデクスページ〔レンジインデクス〕　80
インデクスページスプリット
B-tree インデクス　356
テキストインデクス　361
インデクスページスプリットの発生を抑える
B-tree インデクス　1225
テキストインデクス　1227
インデクスページ内の未使用領域の確保（PCTFREE）
B-tree インデクス　354
テキストインデクス　360
インデクスを作成する〔マルチノード機能〕　1875
インポート
実表にデータを格納する方法（データインポート）　1177

う

ウイルス対策ソフトによるスキャン対象の見直し 921
運用項目の一覧〔HADB サーバ〕　1020

え

エクスポート
実表のデータをファイルに出力する方法（データをエクスポートする方法）　1179
チャンク単位にデータをエクスポートする方法1241
ビュー表のデータをファイルに出力する方法（データをエクスポートする方法）　1205
エントリページ　80

お

オフラインモード〔HADB サーバの稼働モード〕 1028

か

カーネルパラメタ〔dockerd〕　2041
カーネルパラメタ〔Docker および Kubernetes を使用したシステム構成〕　2040
カーネルパラメタの設定　926
カーネルパラメタの見積もり　456
改行コード〔サーバ定義〕　901
開始処理完了の確認〔HADB サーバ〕　1023
開始手順〔コールドスタンバイ構成〕　1992
開始手順〔マルチノード構成〕　1833
開始方法〔HADB サーバ〕　1023
開始モード〔HADB サーバ〕　1023
開始モード〔コールドスタンバイ構成〕　1992
開始モード〔マルチノード構成〕　1834
外部キー　343
実表の制約に関する情報（主キーや外部キーの有無）を確認する場合　1190
主キーと外部キーの構成列に関する情報を確認する場合　1190
外部キー〔用語解説〕　2164
外部キーの指定（FOREIGN KEY）　343
回復処理〔データベース〕　154
回復方法〔データベース〕　1037
下位ページ用のセグメント　79
下位ページ用のセグメントで使用する格納ページ数の求め方　416
拡張 SYSLOG 機能　1054
格納位置の変更〔DB エリアファイル〕　1870
格納効率〔システム表（実表）〕　1435
格納先〔サーバ定義の格納先〕　931
稼働モード　1028
確認する方法　1030
変更する方法　1029
稼働モード〔マルチノード機能〕　1838
可変長データ型の列データの分岐指定（BRANCH）　345
カラムストア形式　61
カラムストア形式〔用語解説〕　2164
カラムストア表　59
圧縮方式　328

カラムストア表の列データの圧縮方式を確認する場合　1190
カラムストア表をローストア表に変更する方法1188
実表のデータを検索・更新する方法　1173
実表を定義する方法　1169
制限事項　326
選択基準　325
表の種類（ローストア表またはカラムストア表）を確認する場合　1190
ローストア表をカラムストア表に変更する方法1185
カラムストア表〔用語解説〕　2165
カラムストア表かどうかを調べる場合　2122
カラムストア表の一覧を調べる場合　2116
カラムデータ用セグメント　79
カレントチャンク〔用語解説〕　2165
カレントチャンクの切り替え　166
カレントチャンクの情報を調べる　2149
環境構築〔Docker および Kubernetes を使用した HADB〕　2043, 2054
環境変数に設定する値　927
環境変数に設定する値（HADB サーバ上で AP を開発および実行する場合）　927
環境変数の設定　927
環境変数の設定〔コールドスタンバイ構成〕
HA モニタ　1960
環境変数の設定〔マルチノード機能〕
HA モニタ　1793
監査管理権限　99, 200
監査権限　99, 200
用語解説　2165
監査権限の取り消し　1471
監査権限の付与　1470
監査参照権限　99, 200
監査証跡機能　198
JP1/Audit との連携　1500
イベントの選定　1447
運用例　1482
環境設定〔マルチノード機能〕　1938
検討項目　1447
コールドスタンバイ構成　2031
推奨運用　208
マルチノード機能　1938
有効化と無効化　207
用語解説　2165
監査証跡機能で使用するサーバ定義の指定　1458
監査証跡機能で使用するディスクの設計　1447
監査証跡機能で使用するディレクトリの準備　1457
監査証跡機能で使用するディレクトリの容量見積もり　1451
監査証跡機能と JP1/Audit の連携
adbconvertaudittrailfile コマンドを実行する際の注意事項　1503
環境設定　1510
サーバ構成と前提ソフトウェア　1502
システム構成例　1500
統一フォーマット用監査証跡ファイルの形式と出力項目　1503
監査証跡機能に関するオペランド（set 形式）　875
監査証跡機能の環境設定　1456
監査証跡機能の使用をやめる場合　1479
監査証跡機能の定期運用　1462
監査証跡機能の動作状況の確認　1474
監査証跡機能のトラブルシュート　1476
監査証跡機能の非定期運用　1470
監査証跡機能の無効化　1479
監査証跡機能の有効化　1459
監査証跡の参照
ADB_AUDITREAD 関数を使用した監査証跡の確認　1465
CSV 形式による監査証跡の確認　1467
監査証跡の出力契機と出力項目　1540
監査証跡の出力先ディスク　208
監査証跡の出力先ディレクトリ　202
容量見積もり　780
監査証跡の出力先ディレクトリを作成したディスクが満杯になった場合の対処方法　1476
監査証跡の出力先ディレクトリを作成したディスクで障害が発生した場合の対処方法　1477
監査証跡の長期保存先ディスク　208
監査証跡のデータ容量の見積もり　1450

監査証跡の変換　210
監査証跡の保存先ディスク　208
監査証跡ファイル　202
　用語解説　2165
　リネーム規則　202
監査証跡ファイルの圧縮率の見積もり　1451
監査証跡ファイルの移動（監査証跡の長期保存先ディレクトリへの移動）　1464
監査証跡ファイルの移動（監査証跡の保存先ディレクトリへの移動）　1462
監査証跡ファイルの切り替え　1473
監査証跡ファイルへの書き込みに失敗した場合の処理方式の検討　1454
監査証跡を CSV 形式に変換する　1467
監査証跡を検索するときの表関数導出表の列構成 1521
監査対象イベント　200
　用語解説　2166
監査対象イベントと表関数導出表の列構成　1518
監査対象イベントの一覧と出力項目　1518
監査対象定義
　用語解説　2166
監査対象定義の確認　1473
監査対象定義の検討　1447
監査対象定義の削除　1474
監査対象定義の変更　1474
監査対象の定義　1460
監査人　200
　用語解説　2166
監査人が持っている監査権限を調べる場合　2131
監査人の削除　1471
監査人の作成　1459
監査人の選定　1453
監査人の追加　1470
監査人の認可識別子を調べる場合　2131
監査人の変更　1471
監視対象メッセージ　1057
監視パス〔コールドスタンバイ構成〕　1953
完全一致検索
　ワード検索　193

き

擬似スレッド〔用語解説〕　2166
擬似スレッドの確認　1068
記述形式〔サーバ定義〕　900
記述順序〔サーバ定義〕　900
基本行　80
基本行〔用語解説〕　2166
基本行用セグメント　79
基本行用ページ　80
行 ID ディレクトリページ　80
行 ID リストページ　80
強制終了〔HADB サーバ〕　1024
強制終了〔コールドスタンバイ構成〕　1993
強制終了〔マルチノード構成〕　1835
行長の固定（FIX 指定）　340
行の継続〔サーバ定義〕　902
行の種別ごとの格納ページ数の求め方　408
共有更新モード　129
共有参照モード　129
共有ディスク　1788
共有ディスクの SCSI リザーブ
　コールドスタンバイ構成　1951
　マルチノード機能　227
　用語解説　2166
共有ディスクのデータの保護方法
　コールドスタンバイ構成　1951
　マルチノード機能　227
共有メモリ　232
共有メモリ管理の所要量の求め方（adbinit コマンド実行時）　587
共有メモリ管理の所要量の求め方（HADB サーバ開始時）　473
共有メモリ管理用のメモリ　232
共有メモリの使用状況を確認したい場合　1066
共有メモリの所要量の求め方　466
行を格納するセグメントが割り当てられている状態 350
切り替え
　現用の監査証跡ファイル　1473

<

区切り文字
ワード検索 195
ワード検索時の区切り文字の選択（DELIMITER） 365
クライアントーサーバ間ネットワーク〔コールドスタンバイ構成〕 1953
クライアントーサーバ間ネットワーク〔マルチノード機能〕 1784
クライアント管理定義の指定形式 1388
クライアント管理定義の指定形式の説明 1388
クライアント管理定義ファイル 162, 1387
クライアントグループ 158
クライアントグループ機能 156
クライアントグループ機能を適用する際の考慮点 811
クライアントグループの設定内容の確認 1383
グループごとの接続数・処理リアルスレッド数の設定 158
グループの種類 158
グループの処理リアルスレッド数を設定する際の考慮点 815
グループの接続数を設定する際の考慮点 811
最大同時接続数 158
最大同時接続数に関する警告メッセージの出力 815
使用できる処理リアルスレッドの最大数 159
使用できる処理リアルスレッドの保証数 159
接続保証数 158
クライアントグループ機能〔用語解説〕 2166
クライアントグループ機能で設定できるグループの種類 158
クライアントグループ機能に関するオペランドおよびオプション（コマンド形式） 889
クライアントグループ機能の概要 156
クライアントグループ機能を適用する際の考慮点 811
クライアントグループの設定内容の確認 1383
クライアント定義〔用語解説〕 2166
クライアント定義情報〔SQL トレース情報〕 1107
クライアント定義の作成〔コールドスタンバイ構成〕 1984
クライアント定義の作成〔マルチノード機能〕 1827
クライアント定義の集中管理機能 161
新たにクライアント定義の集中管理機能を使用する方法 1390
クライアント定義の集中管理機能の使用をやめる方法 1396
クライアント定義の集中管理機能の適用内容を変更する方法 1391
使用例 1393
適用例 163
マルチノード機能の使用時 1828
クライアント定義の集中管理機能〔用語解説〕 2167
クライアント定義の集中管理機能で使用するクライアント定義ファイル 1387
クライアント定義の集中管理機能で必要なファイル 1386
クライアント定義の集中管理機能の使用時に必要なファイル
クライアント管理定義ファイル 1387
クライアント定義の集中管理機能を使用する流れ 1385
クライアント定義の集中管理の運用 1385
クライアント定義ファイル
HADB クライアントで管理するクライアント定義ファイル 1389
クライアント定義ファイル［クライアント定義の集中管理機能］ 1387
クライアントディレクトリ〔用語解説〕 2167
クライアントディレクトリの回復 1736
クライアントメッセージログファイル 1049
グループ
クライアントグループ 158
コマンドグループ 158
グループごとの接続数・処理リアルスレッド数の設定 158
グループの処理リアルスレッド数を設定する際の考慮点 815
グループの接続数を設定する際の考慮点 811
グローバル作業表用のグローバルバッファのページ数の見積もり 825
グローバル作業表を作成する SQL 文の実行時間を短縮する方法〔チューニング〕 1678

グローバルバッファ　119
グローバルバッファ〔用語解説〕　2167
グローバルバッファに関するオペランドおよびオプション（コマンド形式）　879
グローバルバッファの統計情報の運用方法　1085
グローバルバッファのページ数の見積もり〔レンジインデクス〕　370
グローバルバッファの見直しによるメモリ使用量の削減〔チューニング〕　1692
グローバルバッファの割り当て〔マルチノード機能〕　1866
グローバルバッファページの所要量の求め方（HADBサーバ開始時）　474
グローバルバッファ用のメモリ　232

け

計画系切り替え
　コールドスタンバイ構成　2017
　マルチノード機能　1846
系切り替え機能　221
系のリセット
　コールドスタンバイ構成　1951
　マルチノード機能　227
　用語解説　2167
系復帰　2016
権限　97
　HADB ユーザが参照できるディクショナリ表とシステム表の範囲　113
　HADB ユーザに権限を付与する例　113
　アクセス権限　99
　監査権限　99
　権限の種類　97
　スキーマ操作権限　98
　ユーザ権限　97
　ユーザ権限とスキーマ操作権限の管理　1356
権限〔用語解説〕　2167
検索
　システム表　2143
　ディクショナリ表　2115
検索〔CSV ファイル中のデータ〕　185
検索系 SQL〔用語解説〕　2167
現用の監査証跡ファイル　202
　adbconvertaudittrailfile コマンド　1503
現用の監査証跡ファイルの切り替え　202, 1473

こ

コアファイルの容量見積もり　712
更新可能ビュー表　55
更新行のカラム化機能　1437
　エラーが発生した場合の対処　1442
　コールドスタンバイ構成　2035
　トラブルシュート　1442
　マルチノード機能　1943
　メンテナンス処理が実行されない場合　1442
　有効にする　1440
更新行のカラム化機能〔用語解説〕　2167
更新行のカラム化機能に必要なファイルの容量見積もり　716
更新行のカラム化機能の対象となる表を調べる　2131
更新行のカラム化機能のメンテナンス処理に必要なユーザログ容量（変数 max_user_log）の求め方 769
更新系 SQL〔用語解説〕　2168
更新処理方式　119
更新不可状態〔実表〕　1753
構成
　DB ディレクトリ　2079
　サーバディレクトリ（インストール時）　2066
　サーバディレクトリ（運用時）　2071
コール　1098
コールドスタンバイ構成
　HADB サーバのインストールおよび環境設定 1959
　HADB サーバへの接続　1995
　JP1/Audit Management - Manager との連携 2034
　運用方法　1949
　サーバ構成　1952
　サーバマシンの OS のホスト名・IP アドレスの変更　2022
　システム構成例　1951
　前提ソフトウェア　1952

ネットワーク構成　1953
バージョンアップ　2024
コールドスタンバイ構成〔用語解説〕　2168
コールドスタンバイ構成の開始手順　1992
コールドスタンバイ構成の終了手順　1993
コールドスタンバイ構成への系復帰　2016
コスト情報〔用語解説〕　2168
コスト情報の収集　1352
コスト情報を収集した全インデクスのインデクス名と収集日時を調べる　2144
コスト情報を収集した全実表の表名と収集日時を調べる　2143
コスト情報を収集する　1018
コスト情報を収集するタイミング　1352
コネクション状態に関する情報〔SQL トレース情報〕　1107
コネクションの稼働情報の運用方法　1083
コネクションハンドル〔用語解説〕　2168
コマンドが応答待ちタイムアウトになる場合の対処方法　1721
コマンドが再実行できない場合の対処方法
adbidxrebuild コマンド　1722
adbimport コマンド　1722
adbunarchivechunk コマンド　1723
コマンドが実行できない場合の対処方法　1721
コマンドが終了しない場合の対処方法　1721
コマンドグループ　158
コマンド形式〔サーバ定義〕　901
コマンド実行時に増加するデータ量の見積もり　805
コマンド実行時の作業用一時ファイルの容量見積もり　787
コマンド実行時の作業用一時ファイルを格納するディスクの空き容量が不足した場合の対処方法　1723
作業用一時ファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法　1725
作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき）　1726
作業用一時ファイルを格納するディスク容量が小さい場合（adbmergechunk コマンド，adbunarchivechunk コマンドまたは adbreorgsystemdata コマンドが中断したとき）　1729
ディスクに不要な作業用一時ファイルが残っている場合　1726
コマンドステータスファイル〔用語解説〕　2168
コマンド同時実行時のシステムログファイルの容量に関する考慮点　810
コマンド同時実行時の排他制御に関する考慮点　809
コマンド同時実行時のメモリ所要量に関する考慮点　810
コマンドに関するトラブル　1721
コマンドによるプライマリノードの切り替え　1846
コマンドの最大同時実行数　806
コマンドの実行〔Docker および Kubernetes を使用した HADB の運用〕　2059
コマンドの実行時間の短縮に関するチューニング　1685
コマンドの実行時に使用する処理リアルスレッド数に関する考慮点　807
コマンドの実行時に使用する処理リアルスレッド数の最大値　807
コマンドの実行時に使用する処理リアルスレッド数を指定するオペランドとコマンドオプション　808
コマンド用環境変数定義の作成〔コールドスタンバイ構成〕　1965
コマンド用環境変数定義の作成〔マルチノード機能〕　1798
コマンドを同時実行する際の考慮点　806
コマンドを同時実行するときの処理リアルスレッド数の目安　809
コメント〔サーバ定義〕　902
コメントを設定する方法〔チャンク〕　1235
コンテナ構成〔Docker および Kubernetes を使用したシステム構成〕　2039
コンマ〔サーバ定義〕　903

さ

サーバ構成と前提ソフトウェア
監査証跡機能と JP1/Audit の連携　1502
サーバ定義
BOM　903
オペランドの指定形式　834

オペランドの内容　838
改行コード　901
格納先　931
記述形式　900
記述順序　900
行の継続　902
コメント　902
コンマ　903
サーバ定義の設計　833
作成方法　931
文法規則　900
変更方法　931
文字列の識別規則　902
サーバ定義およびコマンドオプションの指定値を見直す〔バージョンアップ後の作業〕
adb_arcv_rthd_num　952
adb_cmd_rthd_num　950
adb_export_rthd_num　953
adb_getcst_rthd_num　954
adb_idxrebuild_rthd_num　954
adb_import_rthd_num　956
adb_mergechunk_rthd_num　959
adb_unarcv_rthd_num　961
サーバ定義のオペランド
SQL 文に関するオペランド（set 形式）　867
監査証跡機能に関するオペランド（set 形式）　875
クライアントグループ機能に関するオペランドおよびオプション（コマンド形式）　889
グローバルバッファに関するオペランドおよびオプション（コマンド形式）　879
システム構成に関するオペランド（set 形式）　838
システムログに関するオペランド（set 形式）　859
状態監視に関するオペランド（set 形式）　862
性能に関するオペランド（set 形式）　842
同義語検索に関するオペランド（set 形式）　874
統計情報に関するオペランド（set 形式）　873
マルチノード機能に関するオペランド（set 形式）　876
レンジインデクスに関するオペランド（set 形式）　872
サーバ定義の記述形式
set 形式　900
コマンド形式　901
サーバ定義の作成〔コールドスタンバイ構成〕　1984
サーバ定義の作成〔マルチノード機能〕　1823
サーバ定義ファイル　931
サーバディレクトリ　918
サーバディレクトリ（インストール時）の構成　2066
サーバディレクトリ（運用時）の構成　2071
サーバディレクトリ〔用語解説〕　2168
サーバディレクトリの回復　1735
サーバディレクトリの削除〔アンインストール〕　1003
サーバディレクトリの事前見積もり　308
サーバディレクトリを格納するディレクトリの作成　911
サーバディレクトリを格納するディレクトリの設定の変更　912
サーバの監視コマンドの作成〔コールドスタンバイ構成〕　1966
サーバの監視コマンドの作成〔マルチノード機能〕　1799
サーバの起動コマンドの作成〔コールドスタンバイ構成〕　1965
サーバの起動コマンドの作成〔マルチノード機能〕　1799
サーバの停止コマンドの作成〔コールドスタンバイ構成〕　1966
サーバの停止コマンドの作成〔マルチノード機能〕　1799
サーバプロセス　2159
サーバマシンの OS の時刻変更　997
サーバマシンの OS の時刻を進める方法　997
サーバマシンの OS の時刻を戻す方法　998
サーバマシンの OS のホスト名・IP アドレスの変更　1001
コールドスタンバイ構成　2022
マルチノード機能　1914
サーバメッセージログファイル　1049
サーバメッセージログファイルの容量見積もり　711

再開始
HADB サーバの再開始に掛かる処理時間が長い場合の対処方法 1779
再開始による回復処理の流れ 154
再初期化〔データ用 DB エリア〕 1367
最大値と最小値
システム構成 2152
データベース 2153
最大同時実行数〔コマンド〕 806
最大同時接続数〔クライアントグループ機能〕 158
最大同時接続数〔マルチノード機能〕 1841
最大同時接続数に関する警告メッセージの出力 862
クライアントグループ機能 815
最大要素数の決め方 346
最大要素数の変更 1202
再編成
シングルチャンク表を再編成する方法 1183
マルチチャンク表を再編成する方法〔サンプルシェル〕 1300
マルチチャンク表を再編成する方法〔チャンク単位〕 1287
マルチチャンク表を再編成する方法〔表全体〕 1295
再編成〔システム表〕 1429
再編成前のデータが残っているかどうかを確認する〔システム表〕 1434
再編成要否の確認
シングルチャンク表 1180
マルチチャンク表 1281
作業表用 DB エリア 76
作業表用 DB エリア〔用語解説〕 2168
作業表用 DB エリアの運用 1380
作業表用 DB エリアの運用〔コールドスタンバイ構成〕 2008
作業表用 DB エリアの運用〔マルチノード機能〕 1871
作業表用 DB エリアの設計 381
作業表用 DB エリアの総ページ数の求め方 431
作業表用 DB エリアのファイル構成の変更〔チューニング〕 1674
作業表用 DB エリアの容量見積もり 431
作業表用 DB エリアファイル 87
作業表用 DB エリアファイルの格納場所を変更する方法 1380
作業表用 DB エリアファイルの容量を小さくする方法 1380
作業表用ページ 80
作業表を格納するために必要な基本行用ページ数の求め方 432
作業用一時ファイル
作業用一時ファイルを格納するディスクの空き容量が不足しているかどうかを確認する方法 1725
作業用一時ファイルを格納するディスク容量が小さい場合（adbimport コマンドまたは adbidxrebuild コマンドが中断したとき） 1726
作業用一時ファイルを格納するディスク容量が小さい場合（adbmergechunk コマンド，adbunarchivechunk コマンドまたは adbreorgsystemdata コマンドが中断したとき） 1729
ディスクに不要な作業用一時ファイルが残っている場合 1726
先読み〔レンジインデクス〕 370
削除
チャンク単位にデータを削除する方法 1242
マルチチャンク表に格納されているデータを一括削除する 1243
削除〔Docker イメージ〕 2063
削除〔Kubernetes のリソースオブジェクト〕 2063
削除仕掛中のチャンク 172, 1248
削除仕掛中のチャンク〔用語解説〕 2168
削除仕掛中のチャンクを調べる 2148
作成されたチャンクの数を調べる 2147
作成方法〔サーバ定義〕 931
作成モード〔データインポート〕 1177
差分圧縮〔カラムストア表〕 328
差分ランレングス圧縮〔カラムストア表〕 328
参照制約
参照制約に関する情報を確認する場合 1190
参照制約〔用語解説〕 2169
参照表〔用語解説〕 2169
サンプルシェルを使用して再編成する場合 1300

し

時刻変更〔サーバマシンの OS〕 997
辞書圧縮〔カラムストア表〕 328
辞書情報登録ファイル 1403
  作成する 1403
  指定規則 1425
辞書ページ 80
システム構成 51
システム構成に関するオペランド（set 形式） 838
システム構成に関する最大値と最小値 2152
システム構成例
  Docker および Kubernetes を使用した HADB の運用 2038
  監査証跡機能と JP1/Audit の連携 1500
システム構築の手順 906
システム設計の流れ 302
システム表 76
  システム表の状態や使用量を確認する方法 1434
システム表（実表）〔用語解説〕 2169
システム表（実表）の B-tree インデクス 2142
システム表（実表）の格納効率を確認する 1435
システム表（実表）の状態や使用量を確認する 1434
システム表（実表）の使用量を確認する 1434
システム表〔用語解説〕 2169
システム表が作成されるタイミング 2134
システム表の一覧 2133
システム表の概要 2133
システム表の検索 2143
  カレントチャンクの情報を調べる 2149
  コスト情報を収集した全インデクスのインデクス名と収集日時を調べる 2144
  コスト情報を収集した全実表の表名と収集日時を調べる 2143
  削除仕掛中のチャンクを調べる 2148
  作成されたチャンクの数を調べる 2147
  指定した期間内に作成されたチャンクやカレントチャンクから切り替わったチャンクのチャンク ID を調べる 2146
  指定した期間に作成されたチャンクを調べる 2144
  指定したコメントが含まれているチャンクを調べる 2147
  指定した日時にデータが格納された可能性があるチャンクを調べる 2147
  すべての同義語辞書に関する情報を調べる 2149
  待機状態のチャンクを調べる 2148
  チャンク ID からチャンクの情報を調べる 2144
  チャンクがアーカイブ状態かどうかを調べる 2148
  チャンクの作成日時から最古のチャンク ID を調べる 2145
  チャンクの作成日時から最新のチャンク ID を調べる 2145
  通常状態のチャンクを調べる 2148
  同義語辞書に関する情報を調べる 2150
  表名から表内の全チャンクの情報を調べる 2144
システム表の再編成 1429
  adbreorgsystemdata コマンドの待ち状態を解消する方法 1433
  システム表の再編成と排他制御 1432
  システム表の状態や使用量を確認する方法 1434
  システム表を再編成するタイミング 1430
  システム表を再編成する方法 1431
  システム表を再編成する理由 1429
システム表の再編成〔マルチノード機能〕 1879
システム表の再編成〔用語解説〕 2169
システム表の容量見積もり 443
システム表用 DB エリア 76
システム表用 DB エリア〔用語解説〕 2169
システム表用 DB エリアの容量見積もり 442
システム表用 DB エリアファイル 88
システム用のチャンク 1248
システムログ 154
システムログ〔用語解説〕 2169
システムログに関するオペランド（set 形式） 859
システムログの運用〔マルチノード機能〕 1880
システムログファイル〔用語解説〕 2169
システムログファイルでトラブルが発生した場合の対処 1737
システムログファイルの運用方法 1165
システムログファイルの構成 154

システムログファイルの作成先　154
システムログファイルの容量に関する考慮点〔コマンド同時実行時〕　810
システムログファイルの容量見積もり　718
事前見積もり
adbarchivechunk コマンド実行時のメモリ所要量　319
adbexport コマンド実行時のメモリ所要量　317
adbgetcst コマンド実行時のメモリ所要量　316
adbidxrebuild コマンド実行時のメモリ所要量 314
adbimport コマンド実行時のメモリ所要量　313
adbmergechunk コマンド実行時のメモリ所要量　318
adbunarchivechunk コマンド実行時のメモリ所要量　320
通常運用時のメモリ所要量　312
実行系　221
実行サーバ　1781
実表　54
カラムストア表をローストア表に変更する方法1188
更新不可状態になった場合　1753
実表が格納されているデータ用 DB エリアを変更する方法　1195
実表にデータを格納する方法（データインポート）　1177
実表の運用　1169
実表の定義情報を調べる方法（ディクショナリ表の検索）　1190
実表のデータを検索・更新する方法　1173
実表を削除する方法　1198
実表を定義する方法　1169
状態や使用量を確認したい場合　1075
シングルチャンク表をマルチチャンク表に変更する方法　1176
全行を削除する方法　1174
データをファイルに出力する方法（データをエクスポートする方法）　1179
配列型の列を定義した実表の運用　1199
表定義を変更する方法　1192
表の設計　325
表名を変更する方法　1172
列名を変更する方法　1170
列を追加する方法　1169
列を追加できない場合　1758
ローストア表をカラムストア表に変更する方法1185
実表に関するトラブル　1753
実表にデータをインポートする　1016
実表にデータを格納する〔SQL パラレル実行機能を使用している場合〕　1873
実表にデータを格納する〔マルチノード機能〕　1873
実表の更新不可状態　1753
実表の更新不可状態を解除する〔コールドスタンバイ構成〕　2009
実表の更新不可状態を解除する〔マルチノード機能〕　1874
実表の再編成に必要なファイルの容量見積もり　785
実表の種類　59
実表の定義情報を調べる場合　2124
実表を定義する HADB ユーザを作成する　1011
指定した期間内に作成されたチャンクやカレントチャンクから切り替わったチャンクのチャンク ID を調べる　2146
指定した期間に作成されたチャンクを調べる　2144
指定したコメントが含まれているチャンクを調べる　2147
指定した日時にデータが格納された可能性があるチャンクを調べる　2147
自動再起動〔HADB サーバ〕　1030
自動増分　90
自動増分の失敗が原因の場合　1732
修正版 HADB サーバとの入れ替え　987
コールドスタンバイ構成　2025
マルチノード機能　1921
集中管理
クライアント定義の集中管理機能　161
終了形態〔マルチノード機能〕　1837
終了処理完了の確認〔HADB サーバ〕　1027
終了待機状態　1025
終了待機状態の確認方法　1025
終了待機状態の待ち時間　1025

終了手順〔コールドスタンバイ構成〕 1993
終了手順〔マルチノード構成〕 1835
終了方法〔HADB サーバ〕 1024
終了モード〔HADB サーバ〕 1024
終了モード〔コールドスタンバイ構成〕 1993
主キー 342
実表の制約に関する情報（主キーや外部キーの有無）を確認する場合 1190
主キーと外部キーの構成列に関する情報を確認する場合 1190
主キーの指定内容を確認する場合 1190
主キー〔用語解説〕 2169
主キーと外部キーを構成する列名を調べる 2121
主キーに対応するインデクスのインデクス名を調べる 2120
主キーを参照する外部キーを調べる 2121
縮退運用 221
縮退状態〔メッセージログファイル〕 1052
縮退状態を解除する〔メッセージログファイル〕 1777
出現情報管理部〔テキストインデクス〕 66
出現情報管理部ページ〔テキストインデクス〕 80
出現情報管理部用のセグメント〔テキストインデクス〕 79
出現情報管理部用のセグメントで使用する格納ページ数の求め方 423
出力
実表のデータをファイルに出力する方法（データをエクスポートする方法） 1179
チャンク単位にデータをエクスポートする方法 1241
ビュー表のデータをファイルに出力する方法（データをエクスポートする方法） 1205
取得方法〔バックアップ〕 1032
上位ページ 80
上位ページ用のセグメント 79
上位ページ用のセグメントで使用する格納ページ数の求め方 418
障害
実行系 2015
待機系 2015
障害が発生した場合〔コールドスタンバイ構成〕 2015
障害管理プロセサ 227
障害情報（コアファイル）の容量見積もり 712
障害発生時の対処の流れ 1709
状態監視に関するオペランド（set 形式） 862
使用停止〔Docker〕 2063
使用停止〔Docker および Kubernetes〕 2063
使用停止〔Kubernetes〕 2063
使用できる処理リアルスレッドの最大数〔クライアントグループ機能〕 159
使用できる処理リアルスレッドの保証数〔クライアントグループ機能〕 159
使用率の監視〔データ用 DB エリア〕 1379
処理性能の向上に関するチューニング 1660
処理リアルスレッド〔用語解説〕 2170
処理リアルスレッド数に関する考慮点〔コマンド〕 807
処理リアルスレッドの確保待ちの発生状況を確認する方法 1071
処理リアルスレッドの最大数〔クライアントグループ機能〕 159
処理リアルスレッドの保証数〔クライアントグループ機能〕 159
シングルチャンク表 64
再編成要否の確認 1180
選択基準 331
シングルチャンク表〔用語解説〕 2170
シングルチャンク表を再編成する方法 1183
シングルチャンク表をマルチチャンク表に変更する方法 1176
伸長 183

す

スーパユーザ 93
スキーマ 94
スキーマの運用 1364
スキーマを削除する方法 1364
スキーマを定義する方法 1364
スキーマ，表，およびインデクスを定義する 1014
スキーマ〔用語解説〕 2170
スキーマオブジェクト 94

スキーマオブジェクト〔用語解説〕 2170
スキーマ操作権限 98
HADB ユーザが持っているユーザ権限とスキーマ操作権限を確認する方法 1356
HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法 1357
HADB ユーザにユーザ権限とスキーマ操作権限を付与する方法 1356
スキーマ定義権限 98
スキーマ操作権限〔用語解説〕 2170
スキーマ定義権限 98
スキーマ定義権限〔用語解説〕 2170
スキーマ名から定義された全インデクスのインデクス名を調べる場合 2118
スキーマ名から定義された全表の表名を調べる場合 2118
ステータス情報 154
ステータスファイル〔用語解説〕 2170
ストライプサイズ 454
ストレージ構成〔Docker および Kubernetes を使用したシステム構成〕 2039
ストレージ構成〔コールドスタンバイ構成〕 1954
スレッドの確認 1068

せ

正規化 337
正規表現を使用した検索 190
静止モード〔HADB サーバの稼働モード〕 1028
正常終了〔HADB サーバ〕 1024
正常終了〔コールドスタンバイ構成〕 1993
正常終了〔マルチノード構成〕 1835
正常終了の種類 1025
性能に関するオペランド（set 形式） 842
制約
実表の制約に関する情報（主キーや外部キーの有無）を確認する場合 1190
セカンダリノード 212
セカンダリノード〔用語解説〕 2170
セカンダリノードの HADB サーバだけを強制終了させる 1836
セカンダリノードの HADB サーバだけを正常終了させる 1836
セカンダリノードの削除 1910
セカンダリノードの新規追加 1905
adbinit コマンドを実行する際の留意事項 1906
セグメント 77
セグメント〔用語解説〕 2170
セグメントが解放されるタイミング 80
セグメントが割り当てられるタイミング 80
セグメント管理部ページ 80
セグメントサイズ〔用語解説〕 2171
セグメントの種類 79
セグメントのスキップ〔レンジインデクス〕 69
設計手順〔データベース〕 324
接続保証数〔クライアントグループ機能〕 158
遷移〔排他制御モード〕 130
選択監査イベント 200
前提 OS および前提ソフトウェア〔Docker および Kubernetes を使用したシステム構成〕 2038
前提ソフトウェアのセットアップ〔Docker および Kubernetes〕 2041
前提パッケージの確認 908
前方一致検索
ワード検索 193
占有モード 129

そ

操作系 SQL〔用語解説〕 2171
ソートコード 188
ソートコード〔用語解説〕 2171
外への参照列〔用語解説〕 2171

た

待機系 221
待機サーバ 1781
待機状態のチャンク 172
マルチチャンク表にインポートするデータを一時的に検索対象外にする方法 1238
待機状態のチャンク〔用語解説〕 2171
待機状態のチャンクを調べる 2148

待機状態のチャンクをマージする場合の注意点 1251
待機状態のチャンクを利用した運用〔バックグラウンドインポート〕 1327
体験環境構築ナビ 244
　よくある質問と対処方法 297
単一列インデクス 352
単一列インデクスと複数列インデクスの使い分け 352
単一列インデクスを定義した方がよい場合 352
単調増加するファイル（HADB サーバ） 708

ち

値域〔レンジインデクス〕 69
値域情報部ページ 80
値域情報部用のセグメント 79
チャンク 166
　アーカイブディレクトリの容量見積もり 775
　カレントチャンク 166
　シングルチャンク表をマルチチャンク表に変更する方法 1176
　チャンク ID 166
　チャンク単位にデータをエクスポートする方法1241
　チャンクの状態と実行できる SQL 文・コマンドの関係 1259
　チャンクの状態変更によるカレントチャンクの切り替え 1262
　マージ元チャンクの削除と検索処理・更新処理の関係 1248
　マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成） 1238
チャンク〔削除仕掛中〕 172, 1248
チャンク〔システム用〕 1248
チャンク〔待機状態〕 172
チャンク〔通常状態〕 172
チャンク〔用語解説〕 2171
チャンク ID〔用語解説〕 2172
チャンク ID からチャンクの情報を調べる 2144
チャンクアーカイブ機能 179
チャンクアーカイブ機能〔用語解説〕 2172
チャンクがアーカイブ状態かどうかを調べる 2148
チャンク数の最大値の求め方 334
チャンク数の最大値を大きくする方法〔KFAA51246-E メッセージ〕 1767
チャンク数の最大値を変更する方法 1256
チャンク数を減らす方法〔KFAA51246-E メッセージ〕 1767
チャンク数を変更できないときの対処方法〔KFAA51246-E メッセージ〕 1768
チャンク単位で再編成する場合 1287
チャンク単位にデータを削除する方法 1242
チャンクとレンジインデクスの関係 173
チャンクにコメントを設定する方法 1235
チャンクのアーカイブ 179
チャンクのアーカイブ〔用語解説〕 2172
チャンクのアーカイブを意識した運用例 1340
チャンクのアンアーカイブ 183
チャンクのアンアーカイブ〔用語解説〕 2172
チャンクの概要 166
チャンクのコメントを変更する〔マルチノード機能〕 1877
チャンクのコメントを変更する方法 1257
チャンクの削除 171
チャンクの作成日時から最古のチャンク ID を調べる 2145
チャンクの作成日時から最新のチャンク ID を調べる 2145
チャンクの状態と実行できる SQL 文・コマンドの関係 1259
チャンクの状態変更 172
　運用例 1279
チャンクの状態変更によるカレントチャンクの切り替え 1262
チャンクの状態や作成されているチャンク数を確認する方法 1243
チャンクの状態を変更する〔マルチノード機能〕 1877
チャンクの状態を変更する方法 1258
チャンクのスキップ〔レンジインデクス〕 69, 174
チャンクのスキップができるレンジインデクスかどうかを調べる 2120
チャンクのスキップの可否〔レンジインデクス〕 1230
チャンクのスキップの設定〔レンジインデクス〕 1230

チャンクのマージ　169
待機状態のチャンクをマージする場合の注意点1251
チャンクのマージ〔用語解説〕　2172
チャンクのマージに伴うインデクスのデータ量の一時的な増加　1248
チャンクをアーカイブする方法　1302
チャンクをアンアーカイブする方法　1304
チャンクをマージする〔マルチノード機能〕　1877
チャンクをマージする方法（チャンク数を減らす方法）　1247
中間ページ　80
チューニング
adbarchivechunk コマンドの実行時間の短縮　1689
adbexport コマンドの実行時間の短縮　1687
adbgetcst コマンドの実行時間の短縮　1686
adbidxrebuild コマンドの実行時間の短縮　1685
adbimport コマンドの実行時間の短縮　1685
adbmergechunk コマンドの実行時間の短縮 1688
DB エリアの自動増分の発生防止　1673
HADB サーバ開始時間の短縮（HugePages の適用）　1664
HugePages を適用した共有メモリの使用量の削減　1691
SQL トレース情報　1158
SQL 文の実行時間を短縮する方法　1676
SQL 文の実行待ち状態の発生防止　1665
SQL 文の処理時間の短縮　1666
グローバル作業表を作成する SQL 文の実行時間を短縮する方法　1678
グローバルバッファの見直しによるメモリ使用量の削減　1692
作業表用 DB エリアのファイル構成の変更　1674
チューニングを行う前に確認すること　1675
定義したテキストインデクスの見直し　1663
定義したレンジインデクスの見直し　1660
テーブルスキャンを実行する SQL 文の実行時間を短縮する方法　1682
ハッシュグループ化領域サイズの見直しによる SQL 文の実行時間の短縮　1699
ハッシュジョインが適用される INNER JOIN の結合順序の見直しによる SQL 文の実行時間の短縮　1701
ハッシュテーブル領域サイズの見直しによる SQL 文の実行時間の短縮　1696
ハッシュフィルタ領域サイズの見直しによる SQL 文の実行時間の短縮　1703
表定義プールサイズの見直しによる SQL 文の実行時間の短縮　1706
ユーザログバッファ面数の拡張　1669
ユーザログファイルの縮小化容量の見直し　1671
ユーザログファイルの初期化容量の拡張　1670
レンジインデクスの一括先読み　1660
ローカル作業表用バッファの見直しによるメモリ使用量の削減　1694
ローカル作業表を作成する SQL 文の実行時間を短縮する方法　1680
重複したキー値を検索する〔一意性制約違反状態の解除〕　1762

つ

追加モード〔データインポート〕　1177
追記型の更新処理方式　119
通常運用時のメモリ所要量の事前見積もり　312
通常運用時のメモリ所要量の求め方　499
通常状態のチャンク　172
通常状態のチャンク〔用語解説〕　2172
通常状態のチャンクを調べる　2148
通常モード〔HADB サーバの稼働モード〕　1028
通信障害〔コールドスタンバイ構成〕　2015
通信障害〔マルチノード機能〕　1899
通信情報ファイルを格納するディレクトリの作成 912
通信ポート数を確認する方法〔HADB クライアント〕　1719

て

定義系 SQL〔用語解説〕　2172
定義系 SQL 実行時に出力されるユーザログ容量（変数 max_user_log）の求め方　723
定義したテキストインデクスの見直し〔チューニング〕　1663

定義したレンジインデクスの見直し〔チューニング〕 1660
定義情報を調べる方法〔実表〕 1190
ディクショナリ表 76
ディクショナリ表（実表）〔用語解説〕 2173
ディクショナリ表（実表）の B-tree インデクス 2113
ディクショナリ表〔用語解説〕 2173
ディクショナリ表が作成されるタイミング 2084
ディクショナリ表の一覧 2081
ディクショナリ表の概要 2081
ディクショナリ表の検索 2115
adbinit コマンド実行時に指定した作業表用 DB エリアのページサイズを調べる 2131
DB エリアに格納されているインデクスのインデクス名を調べる 2119
DB エリアに格納されている表の表名を調べる 2119
HADB ユーザが持っているユーザ権限とスキーマ操作権限を調べる 2120
アーカイブマルチチャンク表かどうかを調べる場合 2122
アーカイブマルチチャンク表のアーカイブディレクトリを調べる 2123
アーカイブマルチチャンク表のアーカイブレンジ列に定義されたレンジインデクスを調べる 2123
アーカイブマルチチャンク表のアーカイブレンジ列の列名を調べる 2123
アーカイブレンジ列として指定する列にレンジインデクスが定義されているかどうかを調べる 2126
インデクス ID からインデクスが定義された表の表名を調べる 2116
インデクス ID からインデクスが定義された列の列名を調べる 2117
インデクス ID からインデクスのインデクス名を調べる 2116
インデクスが格納されている DB エリア名を調べる 2118
インデクスの定義情報を調べる場合 2125
カラムストア表かどうかを調べる場合 2122
カラムストア表の一覧を調べる 2116
監査人が持っている監査権限を調べる場合 2131
監査人の認可識別子を調べる場合 2131
更新行のカラム化機能の対象となる表を調べる 2131
実表の定義情報を調べる場合 2124
主キーと外部キーを構成する列名を調べる 2121
主キーに対応するインデクスのインデクス名を調べる 2120
主キーを参照する外部キーを調べる 2121
スキーマ名から定義された全インデクスのインデクス名を調べる 2118
スキーマ名から定義された全表の表名を調べる 2118
全マルチチャンク表に作成されるチャンク数の最大値を調べる 2118
チャンクのスキップができるレンジインデクスかどうかを調べる 2120
ビュー表が無効化されているかどうかを調べる 2121
ビュー表の更新可否を調べる 2119
表 ID から表の表名を調べる 2115
表に対するアクセス権限を調べる 2121
表名から表に定義された全インデクス名を調べる 2117
表名から表を使用する全ビュー表のビュー表名を調べる 2117
マルチチャンク表かどうかを調べる場合 2122
レンジインデクスのインデクス名を調べる 2117
ローストア表の一覧を調べる 2116
ディクショナリ用 DB エリア 76
ディクショナリ用 DB エリア〔用語解説〕 2173
ディクショナリ用 DB エリアの容量見積もり 436
ディクショナリ用 DB エリアファイル 88
ディスクが満杯になったときの対処方法〔マルチノード機能〕 1899
ディスク障害〔マルチノード機能〕 1898
ディスクに関するトラブル 1735
ディスクに不要なアンロードファイルが残っている 1774
ディスクの空き容量に関するトラブル 1731
ディスクの設計 452
LVM 設定時の考慮点 454
ディスク設計時の考慮点 452
ディスクを増設した場合の運用〔マルチノード機能〕 1871

ディレクトリページ　80
ディレクトリページ群　80
ディレクトリページの配置　80
データインポート
　実表にデータを格納する方法（データインポート）　1177
データインポート〔用語解説〕　2173
データインポート単位のデータ管理（チャンク）　166
データエクスポート
　実表のデータをファイルに出力する方法（データをエクスポートする方法）　1179
　チャンク単位にデータをエクスポートする方法1241
　ビュー表のデータをファイルに出力する方法（データをエクスポートする方法）　1205
データの圧縮　179
データの再編成〔用語解説〕　2173
データの追加と削除を繰り返す運用〔バックグラウンドインポート〕　1310
データページ　80
データページ内の未使用領域の確保（PCTFREE）　344
データベース初期化コンテナ　2039
データベースに関する最大値と最小値　2153
データベースのアクセス処理方式　119
データベースの回復　1735
データベースの回復処理　154
データベースの検索処理方式（非順序実行方式）　119
データベースの更新時に出力されるユーザログ容量（変数 max_user_log）の求め方　748
データベースの更新処理方式　119
データベースの作成手順　1007
データベースの状態や使用量の確認〔adbdbstatus コマンド〕　1074
データベースの使用量を確認したい場合〔adbdbstatus コマンド〕　1074
データベースの設計手順　324
データベースのバックアップ　1032
データベースを初期設定する（データ用 DB エリアを作成する）　1008
データ用 DB エリア　76
　空き容量を確保する方法　1372
　空き容量を確保する方法（実表の再編成）　1375
　空き容量を確保する方法（データの圧縮）　1373
　空き容量を確保する方法（不要なデータの削除）　1376
　インデクスが格納されているデータ用 DB エリアを変更する方法　1231
　拡張する方法（データ用 DB エリアファイルの追加）　1366
　拡張できなくなった場合　1751
　再初期化する方法　1367
　削除する方法　1366
　実表が格納されているデータ用 DB エリアを変更する方法　1195
　使用率の監視方法　1379
　追加する方法　1365
　追加できなくなった場合　1747
　データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する方法　1370
　データ用 DB エリアファイルの格納場所を変更する方法　1368
　データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する方法　1368
　満杯になった場合　1747
データ用 DB エリア〔用語解説〕　2173
データ用 DB エリアの 1 セグメントの容量を決める際の考慮点　377
データ用 DB エリアの空き容量を確保する方法（実表の再編成）　1375
データ用 DB エリアの空き容量を確保する方法（データの圧縮）　1373
データ用 DB エリアの空き容量を確保する方法（不要なデータの削除）　1376
データ用 DB エリアの運用　1365
データ用 DB エリアの拡張〔マルチノード機能〕1868
データ用 DB エリアの削除〔マルチノード機能〕1868
データ用 DB エリアの使用率に関する警告メッセージの出力　862
データ用 DB エリアの使用量を削減する
　B-tree インデクス　1225
　テキストインデクス　1227

レンジインデクス　1229
データ用 DB エリアの設計　373
データ用 DB エリアの総ページ数の求め方　382
計算式　382
変数の説明　382
データ用 DB エリアの追加・削除・拡張〔コールドスタンバイ構成〕　2007
データ用 DB エリアの追加〔マルチノード機能〕 1866
データ用 DB エリアのページサイズを決める際の考慮点　376
データ用 DB エリアの容量見積もり　382
データ用 DB エリアファイル　87
格納場所を変更する方法　1368
データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する方法　1370
データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する方法　1368
データ用 DB エリアファイルの追加〔マルチノード機能〕　1868
データ用 DB エリアを作成する　1008
データ用 DB エリアを設計する際の考慮点　373
マルチチャンク表を格納する場合　375
テーブルスキャン〔用語解説〕　2173
テーブルスキャンを実行する SQL 文の実行時間を短縮する方法〔チューニング〕　1682
テキストインデクス　66
インデクスが格納されているデータ用 DB エリアを変更する方法　1231
インデクスページスプリット　361
検索対象データとテキストインデクスの関係　358
出現情報管理部　66
状態や使用量を確認したい場合　1077
定義した方がよいケース　358
定義しない方がよいケース　360
定義する列を決めるときの考慮点　358
定義できる列　360
表記ゆれ補正指定の指定（CORRECTIONRULE）　363
文字列管理部　66
ワード検索時の区切り文字の選択（DELIMITER）　365
ワード検索用のテキストインデクスの指定（TEXT WORDCONTEXT）　364
テキストインデクス〔用語解説〕　2173
テキストインデクスに関するトラブル　1763
テキストインデクスのインデクスページスプリットの発生を抑える　1227
テキストインデクスの確認（指定したインデクスオプションの確認）　1228
テキストインデクスの再作成　1226
テキストインデクスの使用量を確認する　1077
テキストインデクスのセグメントごとの格納ページ数の求め方　421
テキストインデクスの設計　358
テキストインデクスの定義〔CREATE INDEX 文〕　1223
テキストインデクスの未完状態　362
未完状態かどうかを確認する　1077
未完状態になったときの対処方法　1763
テキストインデクスワード検索指定　364
テキストインデクスを使用した検索性能の低下を防ぐ場合　1227
テキストインデクスを定義する場合の注意点　362
テキストデータの検索　187
電源障害〔マルチノード機能〕　1901
電源断に対する考慮点　455

と

統一フォーマット用監査証跡管理ファイル　1510
統一フォーマット用監査証跡の出力先ディレクトリ　210, 1510
容量見積もり　781
統一フォーマット用監査証跡ファイル　210, 1510
ファイルの形式と出力項目　1503
統一フォーマット用監査証跡ファイル〔用語解説〕　2173
統一フォーマット用監査証跡ファイルの切り替え 1510
同義語一覧定義ファイル　1402
作成する　1402
指定規則　1423

用語解説　2174
同義語一覧定義ファイルを再作成する　1422
同義語グループ　1402
追加する　1413
用語解説　2174
同義語検索　191
運用　1399
検索例　1405
コールドスタンバイ構成　2028
準備作業　1399
準備作業〔マルチノード機能〕　1927
チューニングを行う　1419
マルチノード機能　1927
用語解説　2174
同義語検索に関するオペランド（set 形式）　874
同義語辞書　191
辞書名を確認したい　1408
辞書を削除する　1417
辞書を登録する　1415
登録されている同義語を確認する　1407
表記ゆれ補正検索対応かどうかを知りたい　1408
表記ゆれ補正検索対応に変更する　1418
用語解説　2174
同義語辞書に関する情報を調べる　2150
同義語辞書ファイル　1400
格納ディレクトリを変更する　1421
障害が発生したときの対処方法　1770
トラブルシュート　1770
用語解説　2174
同義語辞書ファイルの格納ディレクトリ
空き容量が不足したときの対処方法　1770
事前見積もり　310
容量見積もり　778
同義語辞書ファイルの格納ディレクトリ〔用語解説〕　2174
同義語辞書ファイルの同期　1936
同義語を削除する　1412
同義語を追加する　1410
統計解析の運用　1082
adbstat コマンド実行時に統計ログファイルから出力される情報の容量見積もり　1095
HADB サーバの統計情報の運用方法　1082
SQL 文の統計情報の運用方法　1088
グローバルバッファの統計情報の運用方法　1085
コネクションの稼働情報の運用方法　1083
統計ログファイルの運用　1092
統計情報に関するオペランド（set 形式）　873
統計情報の運用〔コールドスタンバイ構成〕　2014
統計情報の運用〔マルチノード機能〕　1881
統計ログファイル
上書きされない期間の目安　1093
統計ログファイル〔用語解説〕　2174
統計ログファイルから出力される情報の容量見積もり〔adbstat コマンド〕　1095
統計ログファイルの運用　1092
統計ログファイルの出力先を変更する　1094
統計ログファイルの容量見積もり　714
統計ログファイルを削除する　1094
トラブルシュート
コールドスタンバイ構成　2018
実表に列を追加できない場合　1758
マルチノード機能　1893
トラブルシュート情報の削除（adbinfosweep コマンド）　1715
トラブルシュート情報の取得（adbinfoget コマンド）　1712
トランザクションアクセスモード　127
読み書き可能モード　127
読み取り専用モード　128
トランザクション隔離性水準　124
トランザクションがロールバックしたときの状況を確認する方法　1072
トランザクション制御　124
トランザクションの開始　124
トランザクションの終了　124
トランザクションの状況
トランザクションがロールバックしたときの状況を確認する方法　1072

トランザクションの情報〔SQL トレース情報〕 1112
トランザクションの処理状況
処理リアルスレッドの確保待ちの発生状況を確認する方法 1071
排他資源の確保待ちの発生状況を確認する方法 1071
トランザクションの処理状況の確認 1069
AP およびコマンドの処理状況を確認する方法 1069
トランザクションを実行するノード 1841

な

ナル値除外指定の設定（EXCLUDE NULL VALUES） 357
ナル値表示文字列 927

に

認可識別子〔用語解説〕 2174
認可識別子の指定規則 1012
認可識別子の情報〔SQL トレース情報〕 1113

ね

ネットワーク構成〔Docker および Kubernetes を使用したシステム構成〕 2039
ネットワーク構成〔コールドスタンバイ構成〕 1953
ネットワーク構成〔マルチノード機能〕 1784

の

ノード〔用語解説〕 2174
ノード間ネットワーク〔マルチノード機能〕 1785
ノード構成〔Docker および Kubernetes を使用したシステム構成〕 2039
ノード障害 220
セカンダリノード 1885
プライマリノード 1883
ワーカーノード 1890
ノード障害〔用語解説〕 2175
ノードの切り離し 221
ノードの切り離し〔用語解説〕 2175
ノードの構築〔Kubernetes 環境〕 2042
ノードの情報を確認する場合 1862
ノードの復帰 224
手順 1886
ノードの復帰〔用語解説〕 2175

は

バージョンアップ 933
adbbuff オペランドの指定を見直す 948
コールドスタンバイ構成の場合 2024
動作確認テスト用 994
マルチノード機能の場合 1920
バージョンアップ時の注意事項 969
バージョンアップに失敗した場合の対処方法 963
バージョンダウン
旧バージョンでデータベースを再構築する場合 982
旧バージョンのデータベースのバックアップを使用する場合 979
動作確認テスト用 994
バージョンダウン〔HADB サーバ〕 978
排他〔マルチノード機能〕 1845
排他資源の確保待ちの発生状況を確認する方法 1071
排他制御 129
確保される排他資源と排他制御モード 134
システム表の再編成と排他制御 1432
ディクショナリ表およびシステム表の参照時に排他が取得される実表 142
データベースの参照・更新時に排他が取得されるディクショナリ表とシステム表 143
排他資源 131
排他資源が確保される期間 133
排他制御に関する考慮点〔コマンド同時実行時〕 809
排他制御モード 129
排他競合時の動作 129
排他制御モードの同時実行の関係性 129
排他制御モードの種類 129
排他制御モードの遷移 130
配列型
最大要素数の変更 1202
表の検索 1200
表の全行削除 1202

配列型の定義
表の定義 1199
配列型の列の追加 1203
配列型の列の定義 346
配列型の列を定義した実表の運用 1199
配列要素 346
パスワードの指定規則 1013
パスワードを変更する方法〔HADB ユーザ〕 1353
バックアップ〔データベース〕 1032
バックアップからの回復方法 1037
バックアップからの回復方法〔コールドスタンバイ構成〕 1997
バックアップからの回復方法〔マルチノード機能〕 1849
バックアップの運用例（OS コマンドを使用） 1038
バックアップの運用例（ShadowImage を使用） 1045
バックアップの取得方法 1032
バックアップの取得方法〔コールドスタンバイ構成〕 1996
バックアップの取得方法〔マルチノード機能〕 1847
バックアップを取得するタイミング 1032
バックアップを取得する手順 1032
バックグラウンドインポート
マルチチャンク表にデータを格納する方法 1235
バックグラウンドインポート〔用語解説〕 2175
バックグラウンドインポート機能 165
バックグラウンドインポート機能に関するトラブル 1767
作成されたチャンク数を減らす方法 1767
作成されたチャンク数を変更できないときの対処方法 1768
作成されるチャンク数の最大値を大きくする方法 1767
バックグラウンドインポートとは 165
ハッシュグループ化領域 232
ハッシュグループ化領域サイズの見直しによる SQL 文の実行時間の短縮〔チューニング〕 1699
ハッシュ検索〔用語解説〕 2175
ハッシュジョインが適用される INNER JOIN の結合順序の見直しによる SQL 文の実行時間の短縮〔チューニング〕 1701
ハッシュテーブル領域 232
ハッシュテーブル領域サイズの見直しによる SQL 文の実行時間の短縮〔チューニング〕 1696
ハッシュテーブルを使用した検索時に作成される作業表の数 828
ハッシュフィルタ領域 232
ハッシュフィルタ領域サイズの見直しによる SQL 文の実行時間の短縮〔チューニング〕 1703
バッファの運用 1381
ローカル作業表用バッファを変更する方法 1381
バッファの見直しによる SQL 文の実行時間の短縮に関するチューニング 1675

ひ

被参照表〔用語解説〕 2175
非順序実行方式〔用語解説〕 2175
必須監査イベント 200
非ナル値制約 342
非ナル値制約〔用語解説〕 2176
ビュー表 54
アクセス権限 111
アクセス権限を確認する方法 1219
依存するビュー表を確認する方法 1218
運用例 1219
更新可能ビュー表 55
データの検索，追加，更新，削除 1205
データをファイルに出力する方法（データをエクスポートする方法） 1205
ビュー表の運用 1204
ビュー表の更新可否を調べる方法 1207
ビュー表の無効化を解除する方法 1208
ビュー表を再作成する方法 1205
ビュー表を削除する方法 1206
ビュー表を定義する方法 1204
付与権 111
無効化されたビュー表を削除する方法 1217
無効化されているかどうかを確認する方法 1207

基表を確認する方法　1218
読み取り専用ビュー表　55
ビュー表〔用語解説〕　2176
ビュー表が無効化されているかどうかを調べる　2121
ビュー表の更新可否を調べる　2119
ビュー表の無効化　56
ビューレベル　56
ビューレベル〔用語解説〕　2176
表
表の設計　325
表 ID から表の表名を調べる場合　2115
表関数導出表　185
表関数導出表〔ADB_AUDITREAD 関数〕　205
表関数導出表〔用語解説〕　2176
表記ゆれ補正検索　187
表記ゆれ補正検索〔用語解説〕　2176
表記ゆれ補正指定
テキストインデクスの表記ゆれ補正指定の指定（CORRECTIONRULE）　363
標準エラー出力〔メッセージ〕　1049
標準出力〔メッセージ〕　1049
表定義情報　842
表定義プール　842
表定義プールサイズの見直しによる SQL 文の実行時間の短縮〔チューニング〕　1706
表定義を変更する方法　1192
表データの格納形式　59
表に対するアクセス権限を調べる　2121
表の格納効率を向上させる場合〔表の正規化〕　338
表の種類　54
表の正規化　337
アクセス頻度の高い列と低い列がある場合　339
表の格納効率を向上させる場合　338
表名から表内の全チャンクの情報を調べる　2144
表名から表に定義された全インデクス名を調べる場合　2117
表名から表を使用する全ビュー表のビュー表名を調べる場合　2117
表名を確認する場合　1190
表名を変更する方法　1172
表を設計する際の流れ　325

ふ

ファイル構成〔DB エリア〕　86
ファイルディスクリプタが不足した場合の対処方法　1477
複数列インデクス　352
複数列インデクスを使用するときの留意事項　353
複数列インデクスを定義した方がよい場合　353
付与権　104
付与権〔用語解説〕　2177
付与権だけを取り消す方法　1362
プライマリノード　212
プライマリノード〔用語解説〕　2177
プライマリノードの HADB サーバだけを強制終了させる　1836
プライマリノードの HADB サーバだけを正常終了させる　1835
プライマリノードの切り替え　221
コマンドによる切り替え　1846
プライマリノードの切り替え〔用語解説〕　2177
フルバックアップ　1032
プローブパケット　862
プロセス　2159
プロセス共通メモリ　232
プロセスメモリ　232
プロセスメモリの所要量の求め方　471
ブロックスペシャルファイル　86
データ用 DB エリアファイル（ブロックスペシャルファイル）の格納場所を変更する方法　1370
データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する方法　1368
マルチノード機能　90
分岐行　80
分岐行〔用語解説〕　2177
分岐行用セグメント　79
分岐行用ページ　80
分岐指定〔可変長データ型の列データ〕　345
文ハンドル〔用語解説〕　2177

文法規則〔サーバ定義〕 900

へ

ページ 80
ページ〔用語解説〕 2177
ページが解放されるタイミング 84
ページが割り当てられるタイミング 84
ページグループ 85
ページグループサイズを確認する〔レンジインデクス〕 1078
ページサイズを決める際の考慮点〔データ用 DB エリア〕 376
ページの種類 80
変更方法〔サーバ定義〕 931

ほ

保護更新モード 129
ホスト名の変更
コールドスタンバイ構成 2022
マルチノード機能 1914
ホスト名の変更〔サーバマシンの OS〕 1001

ま

マージ先チャンク〔用語解説〕 2172
マージチャンク 169
チャンクのマージに伴うインデクスのデータ量の一時的な増加 1248
マージ元チャンクの削除と検索処理・更新処理の関係 1248
マージ元チャンク〔用語解説〕 2172
マージ元チャンクの削除と検索処理・更新処理の関係 1248
前処理表〔排他資源〕 131
マスタディレクトリ〔用語解説〕 2177
マスタディレクトリ用 DB エリア 76
マスタディレクトリ用 DB エリア〔用語解説〕 2177
マスタディレクトリ用 DB エリアの容量見積もり 435
マスタディレクトリ用 DB エリアファイル 88
マスタログファイル 154
マスタログファイルの容量の求め方 718
マニフェストファイル（yaml ファイル）の準備 2054
マルチチャンク表 64
再編成要否の確認 1281
シングルチャンク表をマルチチャンク表に変更する方法 1176
選択基準 331
チャンク単位にデータをエクスポートする方法 1241
マルチチャンク表〔用語解説〕 2177
マルチチャンク表かどうかを調べる場合 2122
マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成） 1238
マルチチャンク表に作成されるチャンク数の最大値を調べる 2118
マルチチャンク表にデータを格納する方法（バックグラウンドインポート） 1235
マルチチャンク表の運用 1234
アーカイブマルチチャンク表をレギュラーマルチチャンク表に変更する方法 1308
チャンク数の最大値を変更する方法 1256
チャンク単位にデータを削除する方法 1242
チャンクのコメントを変更する方法 1257
チャンクの状態や作成されているチャンク数を確認する方法 1243
チャンクの状態を変更する方法 1258
チャンクをマージする方法（チャンク数を減らす方法） 1247
データを一括削除する方法 1243
バックグラウンドインポートとチャンクを意識した運用例 1（定期的にデータの追加と削除を繰り返す運用） 1310
バックグラウンドインポートとチャンクを意識した運用例 2（待機状態のチャンクを利用した運用） 1327
マルチチャンク表にインポートするデータを一時的に検索対象外にする方法（待機状態のチャンクの作成） 1238
マルチチャンク表にデータを格納する方法（バックグラウンドインポート） 1235
マルチチャンク表を削除する方法 1310
マルチノード機能 1876

レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する方法 1306
マルチチャンク表を再編成する方法
サンプルシェルを使用する場合 1300
チャンク単位で再編成する場合 1287
表全体を再編成する場合 1295
マルチチャンク表を定義する場合の考慮点 332
マルチチャンク表を定義する方法 1234
マルチノード機能 212
DB エリアファイル 90
HADB サーバの環境設定 1792
HADB サーバへの接続 1840
JP1/Audit Management - Manager との連携 1940
運用方法 1780
サーバ構成 1783
サーバマシンの OS のホスト名・IP アドレスの変更 1914
システム構成例 1782
ストレージ構成 1787
前提ソフトウェア 1783
同義語辞書ファイルの格納ディレクトリ 2178
ネットワーク構成 1784
バージョンアップ 1920
ブロックスペシャルファイル 90
マルチノード機能〔用語解説〕 2178
マルチノード機能で使用する同義語辞書ファイルの格納ディレクトリ
事前見積もり 310
容量見積もり 782
マルチノード機能に関するオペランド（set 形式） 876
マルチノード機能を使用するシステムへの移行 1916
マルチノード構成 212
ノードの切り離し 221
ノードの復帰 224
マルチノード構成〔用語解説〕 2178
マルチノード構成からのワーカーノードの削除 226
マルチノード構成からのワーカーノードの削除〔用語解説〕 2178
マルチノード構成の HADB サーバの開始 1833
マルチノード構成の HADB サーバの終了 1835
マルチノード構成へのノードの復帰 1886
マルチノード構成へのワーカーノードの追加 226
マルチノード構成へのワーカーノードの追加〔用語解説〕 2178
満杯状態 1732

み

未完状態
B-tree インデクス 350
テキストインデクス 362
レンジインデクス 371
未完状態かどうかを確認する〔B-tree インデクス〕 1075
未完状態かどうかを確認する〔テキストインデクス〕 1077
未完状態かどうかを確認する〔レンジインデクス〕 1078
未完状態になったときの対処方法〔B-tree インデクス〕 1760
未完状態になったときの対処方法〔テキストインデクス〕 1763
未完状態になったときの対処方法〔レンジインデクス〕 1765
未使用領域の確保（PCTFREE）
B-tree インデクスのインデクスページ内 354
データページ内 344
テキストインデクスのインデクスページ内 360
見積もり
グローバル作業表用のグローバルバッファのページ数 825
ハッシュテーブルを使用した検索時に作成される作業表の数 828
ローカル作業表用バッファのページ数 826
見積もり式
adbarchivechunk コマンド実行時のメモリ所要量 660
adbaudittrail コマンド実行時のメモリ所要量 705
adbchgchunkcomment コマンド実行時のメモリ所要量 658

adbchgchunkstatus コマンド実行時のメモリ所要量 658
adbclientdefmang コマンド実行時のメモリ所要量 701
adbconvertaudittrailfile コマンド実行時のメモリ所要量 707
adbdbstatus コマンド実行時のメモリ所要量 632
adbexport コマンドが出力するファイルの容量見積もり 783
adbexport コマンド実行時のメモリ所要量 634
adbgetcst コマンド実行時のメモリ所要量 627
adbidxrebuild コマンド実行時の作業用一時ファイルの容量見積もり 791
adbidxrebuild コマンド実行時のメモリ所要量 612
adbimport コマンド実行時の作業用一時ファイルの容量 787
adbimport コマンド実行時のメモリ所要量 588
adbinit コマンド実行時のメモリ所要量 587
adbmergechunk コマンド実行時の作業用一時ファイルの容量見積もり 795
adbmergechunk コマンド実行時のメモリ所要量 642
adbmodarea コマンド実行時のメモリ所要量 638
adbreorgsystemdata コマンド実行時の作業用一時ファイルの容量見積もり 801
adbreorgsystemdata コマンド実行時のメモリ所要量 686
adbstat コマンド実行時のメモリ所要量 638
adbsyndict コマンド実行時の作業用一時ファイルの容量見積もり 804
adbsyndict コマンド実行時のメモリ所要量 704
adbunarchivechunk コマンド実行時の作業用一時ファイルの容量見積もり 798
adbunarchivechunk コマンド実行時のメモリ所要量 667
B-tree インデクスのキー長（KEYSZ） 419
B-tree インデクスのセグメントごとの格納ページ数 416
HADB サーバ開始時のメモリ所要量 472
HADB サーバのメモリ所要量 464
HADB ダンプファイルの容量 713
SQL トレースファイルの容量 715
アクセスパス探索情報ログファイルの容量 717
アンロードファイルの容量見積もり 786
カーネルパラメタ 456
下位ページ用のセグメントで使用する格納ページ数 416
監査証跡の出力先ディレクトリの容量見積もり 780
行の種別ごとの格納ページ数 408
共有メモリの所要量 466
更新行のカラム化機能に必要なファイルの容量 716
サーバメッセージログファイルの容量 711
作業表用 DB エリアの総ページ数 431
作業表用 DB エリアの容量 431
作業表を格納するために必要な基本行用ページ数 432
システム表用 DB エリアの容量 442
システムログファイルの容量 718
実表の再編成に必要なファイルの容量見積もり 785
出現情報管理部用のセグメントで使用する格納ページ数 423
上位ページ用のセグメントで使用する格納ページ数 418
障害情報（コアファイル）の容量 712
通常運用時のメモリ所要量 499
ディクショナリ用 DB エリアの容量 436
データ用 DB エリアの総ページ数 382
データ用 DB エリアの容量 382
テキストインデクスのセグメントごとの格納ページ数 421
統一フォーマット用監査証跡の出力先ディレクトリの容量見積もり 781
統計ログファイルの容量 714
プロセスメモリの所要量 471
マスタディレクトリ用 DB エリアの容量 435
マスタログファイルの容量 718
メモリ所要量の事前見積もり 311
文字列管理部用のセグメントで使用する格納ページ数 422
ユーザログファイル数 770
ユーザログファイルの容量 722

レンジインデクスを格納するセグメント数　424

む

無圧縮〔カラムストア表〕　328
無効化されたビュー表を削除する方法　1217
無効情報管理用ページ　61, 80
無効な行データ〔用語解説〕　2178

め

メッセージ〔監視対象〕　1057
メッセージカタログファイル〔用語解説〕　2178
メッセージの確認　1049
メッセージの出力先　1049
メッセージの出力抑止〔syslog〕　1053
メッセージの出力レベル　1053
メッセージログの参照方法（メッセージログの出力先）　1050
メッセージログの出力先　1050
メッセージログファイル
　縮退状態　1052
メッセージログファイル〔用語解説〕　2178
メッセージログファイルに関するトラブル　1777
メッセージログファイルの運用　1050
メッセージログファイルの切り替え　1050
メッセージログファイルの縮退状態を解除する方法　1777
メモリが不足した場合の対処方法
　HADB サーバ開始時　1739
　SQL 文実行時またはコマンド実行時　1742
メモリ構成〔HADB サーバ〕　232
メモリ使用状況を確認したい場合〔リアルスレッドごと〕　1067
メモリ使用量に関する警告メッセージの出力　862
メモリ使用量の削減に関するチューニング　1691
メモリ所要量に関する考慮点〔コマンド同時実行時〕　810
メモリ所要量の事前見積もり　311
メモリに関するトラブル　1739
メモリの使用状況の確認　1065
メモリの使用状況を確認したい場合　1065
メモリの使用状況を確認する場合〔コールドスタンバイ構成〕　2006
メモリの使用状況を確認する場合〔マルチノード機能〕　1863
メンテナンス処理　1443
メンテナンスモード〔HADB サーバの稼働モード〕　1028

も

文字コード
　HADB クライアントで使用する文字コード　927
　HADB サーバで使用する文字コード　927
　OS の文字コード　927
文字列管理部〔テキストインデクス〕　66
文字列管理部ページ〔テキストインデクス〕　80
文字列管理部用のセグメント〔テキストインデクス〕79
文字列管理部用のセグメントで使用する格納ページ数の求め方　422
文字列の識別規則〔サーバ定義〕　902
基表　56
基表〔用語解説〕　2178
モニタモード　220

ゆ

ユーザ権限　97
　CONNECT 権限　98
　DBA 権限　97
　HADB ユーザが持っているユーザ権限とスキーマ操作権限を確認する方法　1356
　HADB ユーザが持っているユーザ権限とスキーマ操作権限を取り消す方法　1357
　HADB ユーザにユーザ権限とスキーマ操作権限を付与する方法　1356
ユーザ権限〔用語解説〕　2179
ユーザコマンドの作成　1799
ユーザとスキーマ　93
ユーザ認証　96
ユーザの種類（OS ユーザと HADB ユーザ）　93
ユーザログ　154
ユーザログバッファ面数の拡張〔チューニング〕　1669

ユーザログファイル　154
ユーザログファイル数の求め方　770
ユーザログファイルの縮小化容量の見直し〔チューニング〕　1671
ユーザログファイルの初期化容量の拡張〔チューニング〕　1670
ユーザログファイルの容量の求め方　722
ユニークインデクス　356
ユニークインデクスが一意性制約違反状態かどうかを確認する　1076

よ

要素データ型　346
読み書き可能トランザクション　127
読み書き可能モード〔トランザクションアクセスモード〕　127
読み取り専用トランザクション　128
読み取り専用ビュー表　55
読み取り専用モード〔トランザクションアクセスモード〕　128

ら

ランレングス圧縮〔カラムストア表〕　328

り

リアルスレッド〔用語解説〕　2179
リアルスレッド固有メモリ　232
リアルスレッドの確認　1068
リーフページ　80
リセットパス〔コールドスタンバイ構成〕　1954
リセットパス〔マルチノード機能〕　1786
リソースオブジェクトの生成　2054
リソースの事前見積もり　304
リソースの使用状況の監視　1057
リターンコード〔adbstart コマンド〕　1023
リターンコード〔adbstop コマンド〕　1027
リロード〔用語解説〕　2179

る

ルートページ　80

れ

レギュラーファイル　86
　データ用 DB エリアファイルをレギュラーファイルからブロックスペシャルファイルに変更する方法　1368
レギュラーマルチチャンク表　64
　マルチチャンク表を削除する方法　1310
　マルチチャンク表を定義する方法　1234
　レギュラーマルチチャンク表をアーカイブマルチチャンク表に変更する方法　1306
レギュラーマルチチャンク表〔用語解説〕　2179
列データの圧縮方式〔カラムストア表〕　328
列データページ　80
列の既定値の設定　341
列のデータ型と列の定義長を確認する場合　1190
列名
　列名を確認する場合　1190
　列名を変更する方法　1170
列を追加する方法　1169
列を追加できない場合〔トラブルシュート〕　1758
レンジインデクス　69
　インデクスが格納されているデータ用 DB エリアを変更する方法　1231
　状態や使用量を確認したい場合　1078
　セグメントのスキップ　69
　チャンクのスキップ　69
　定義した方がよいケース　367
　定義しない方がよいケース　369
　定義する際の考慮点　367
　定義できない列　370
レンジインデクス〔チャンク〕　173
レンジインデクス〔用語解説〕　2179
レンジインデクス専用のグローバルバッファのページ数の見積もり　370
レンジインデクスに関するオペランド（set 形式）872
レンジインデクスに関するトラブル　1765
レンジインデクスの一括先読み　370
　一括先読みしたレンジインデクスのデータの掃き出しを防ぐ方法　371
　一括先読みを行うメリット　370

レンジインデクスの一括先読み〔チューニング〕 1660
レンジインデクスのインデクス名を調べる場合 2117
レンジインデクスの概要 70
レンジインデクスの確認（チャンクのスキップの可否） 1230
レンジインデクスの再作成 1228
レンジインデクスの再定義（チャンクのスキップの設定） 1230
レンジインデクスの使用量を確認する 1078
レンジインデクスの設計 367
レンジインデクスの値域の更新（セグメント） 73
レンジインデクスの値域の更新（チャンクとセグメント） 177
レンジインデクスの定義〔CREATE INDEX 文〕 1223
レンジインデクスの特徴 71
レンジインデクスの特徴（チャンクのスキップ） 175
レンジインデクスのページグループサイズを確認する 1078
レンジインデクスの未完状態 371
未完状態かどうかを確認する 1078
未完状態になったときの対処方法 1765
レンジインデクスを格納するセグメント数の求め方 424
レンジインデクスを使用した検索処理（チャンクのスキップ） 174
レンジインデクスを使用した検索性能の低下を防ぐ場合 1229
レンジインデクスを定義する場合の注意点 371

ろ

ローカル作業表 826
ローカル作業表用バッファ 232
ローカル作業表用バッファのページ数の見積もり 826
ローカル作業表用バッファの見直しによるメモリ使用量の削減〔チューニング〕 1694
ローカル作業表用バッファを変更する方法 1381
ローカル作業表を作成する SQL 文の実行時間を短縮する方法〔チューニング〕 1680
ローストア形式 59
ローストア形式〔用語解説〕 2179
ローストア表 59
カラムストア表をローストア表に変更する方法 1188
実表のデータを検索・更新する方法 1173
実表を定義する方法 1169
選択基準 325
表の種類（ローストア表またはカラムストア表）を確認する場合 1190
ローストア表の一覧を調べる場合 2116
ローストア表をカラムストア表に変更する方法 1185
ローストア表〔用語解説〕 2179
ローデータ用セグメント 79
ロールバックしたときの状況を確認する方法 1072
ロケーション表 182
ロケーション表〔用語解説〕 2179
ロケーション表のチャンク数 375

わ

ワーカーノード 212
ワーカーノード〔用語解説〕 2180
ワーカーノードの HADB サーバだけを強制終了させる 1837
ワーカーノードの HADB サーバだけを正常終了させる 1837
ワーカーノードの最大数 876
ワーカーノードの削除 1913
ワーカーノードの新規追加 1912
ワーカーノードの追加手順 1891
ワーカーノードのマルチノード構成からの削除 226
ワーカーノードのマルチノード構成への追加 226
ワード検索 193
完全一致検索 193
区切り文字 195
前方一致検索 193
ワード検索〔用語解説〕 2180
ワード検索用のテキストインデクスの指定（TEXT WORDCONTEXT） 364
割り当て管理用ページ 80

株式会社 日立製作所
〒 100-8280　東京都千代田区丸の内一丁目 6 番 6 号