

HiRDB データ連動拡張機能

HiRDB Datareplicator Extension Version 10

解説・手引・操作書

3020-6-574

---

## 前書き

### ■ 対象製品

●適用 OS : Red Hat Enterprise Linux 6 (64-bit x86\_64), Red Hat Enterprise Linux 7 (64-bit x86\_64)

P-8462-1LA1 HiRDB Datareplicator Extension Version 10 10-00

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

### ■ 輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

### ■ 商標類

HITACHI, HiRDB, Cosminexus, HA モニタ, OpenTP1, TPBroker, uCosminexus, VOS3/LS, VOS3/US, VOS3/XS, XDM は、株式会社 日立製作所の商標または登録商標です。

Access は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

ActiveX は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

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

Excel は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

IBM, DataStage, MetaBroker, MetaStage および QualityStage は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM, DB2 は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM, HACMP は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM, OS/390 は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM, PowerHA は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Itanium は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。

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

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

Visual C++ は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows NT は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

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

## ■ 発行

2018 年 10 月 3020-6-574

## ■ 著作権

All Rights Reserved. Copyright (C) 2018, Hitachi, Ltd.

## 変更内容

### 変更内容 (3020-6-574) HiRDB Datareplicator Extension Version 10 10-00

追加・変更内容	変更箇所
マニュアルの体裁を変更しました。	—

単なる誤字・脱字などはお断りなく訂正しました。

### 変更内容 (3020-6-361-50) HiRDB Datareplicator Extension Version 8 08-06

追加・変更内容
次の OS に対応しました。 <ul style="list-style-type: none"><li>• AIX V7.1</li><li>• Red Hat Enterprise Linux AS 4(x86)</li><li>• Red Hat Enterprise Linux ES 4(x86)</li><li>• Red Hat Enterprise Linux AS 4(AMD64 &amp; Intel EM64T)</li><li>• Red Hat Enterprise Linux ES 4(AMD64 &amp; Intel EM64T)</li><li>• Red Hat Enterprise Linux 5 Advanced Platform (x86)</li><li>• Red Hat Enterprise Linux 5 (x86)</li><li>• Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)</li><li>• Red Hat Enterprise Linux 5(AMD/Intel 64)</li><li>• Windows 7</li></ul>

### 変更内容 (3020-6-361-40) HiRDB Datareplicator Extension Version 8 08-04

追加・変更内容
Windows Vista に対応しました。
抽出側 Datareplicator Extension が通信障害を検知した際、障害内容に関係なく無条件にコネクションリトライできるようになりました。 これに伴い、cm_errno_check オペランドを追加しました。
データ連動回復機能に、システムログファイルによる回復方法を追加しました。 これに伴い、次のオペランドを追加しました。 <ul style="list-style-type: none"><li>• recover_info_send オペランド</li><li>• recover_info_send_interval オペランド</li></ul>
コネクションリトライ時のリトライ間隔を指定できるようになりました。 これに伴い、retry_interval オペランドを追加しました。
Oracle Database 11g に対応しました。

## 変更内容 (3020-6-361-30) HiRDB Datareplicator Extension Version 8 08-03

追加・変更内容
Datareplicator の稼働プラットフォームに Windows Server 2008 を追加しました。
DECIMAL 型の精度を拡張し、38 けたまで定義できるようになりました。 これに伴い、decimal_max_precision オペランドを追加しました。
監査証跡の取得対象になる操作（監査対象イベント）が発生した場合に、監査証跡を取得するかどうかを指定できるようになりました。 これに伴い、hirdb_audit_trail オペランドを追加しました。
Datareplicator で使用するファイルのチェックを初期化時又は起動時に実施できるようになりました。 これに伴い、resource_chk_err オペランドを追加しました。
参照制約動作に CASCADE が指定された表をデータ連動できるようになりました。 これに伴い、次のオペランドを追加しました。 <ul style="list-style-type: none"><li>• check_pending オペランド</li><li>• control_reference_trigger オペランド</li></ul>

## はじめに

このマニュアルは、HiRDB Datareplicator Extension Version 10（以降、このマニュアルでは、「Datareplicator Extension」と表記します）の機能と使い方について説明したものです。

Datareplicator Extension を使用するためには、次に示すプログラムプロダクトが必要です。

- HiRDB Datareplicator Version 10

以降、このマニュアルでは、上記の製品を「Datareplicator」と表記します。

### ■ 対象読者

Datareplicator Extension を使ってデータ連動システムを構築／運用するシステム管理者、システム設計者、プログラマ、及びオペレータの方々を対象としています。

このマニュアルの記述は、適用 OS の基礎的な知識、抽出側 DBMS 及び反映側 DBMS の知識があることを前提としています。

また、このマニュアルは、「HiRDB データ連動機能 HiRDB Datareplicator Version 10」のマニュアルを理解されていることを前提としています。

### ■ 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

#### HiRDB (UNIX 用マニュアル)

- HiRDB Version 10 システム導入・設計ガイド (UNIX(R)用) (3020-6-552)
- HiRDB Version 10 システム定義 (UNIX(R)用) (3020-6-554)
- HiRDB Version 10 システム運用ガイド (UNIX(R)用) (3020-6-556)
- HiRDB Version 10 コマンドリファレンス (UNIX(R)用) (3020-6-558)
- インナレプリカ機能 HiRDB Staticizer Option Version 10 (3020-6-563)
- HiRDB Version 10 ディザスタリカバリシステム 構築・運用ガイド (3020-6-564)

#### HiRDB (Windows 用マニュアル)

- HiRDB Version 10 システム導入・設計ガイド (Windows(R)用) (3020-6-553)
- HiRDB Version 10 システム定義 (Windows(R)用) (3020-6-555)
- HiRDB Version 10 システム運用ガイド (Windows(R)用) (3020-6-557)

- HiRDB Version 10 コマンドリファレンス (Windows(R)用) (3020-6-559)

## HiRDB (Windows, UNIX 共通マニュアル)

- HiRDB Version 10 解説 (3020-6-551)
- HiRDB Version 10 UAP 開発ガイド (3020-6-560)
- HiRDB Version 10 SQL リファレンス (3020-6-561)
- HiRDB Version 10 メッセージ (3020-6-562)
- HiRDB Version 10 XDM/RD E2 接続機能 (3020-6-565)
- HiRDB Version 10 バッチ高速化機能 (3020-6-567)
- HHiRDB 全文検索プラグイン HiRDB Text Search Plug-in Version 9 (3020-6-481-30)
- HiRDB XML 拡張機能 HiRDB XML Extension Version 9 (3020-6-480-30)
- HiRDB データ連動機能 HiRDB Datareplicator Version 10 (3020-6-573)

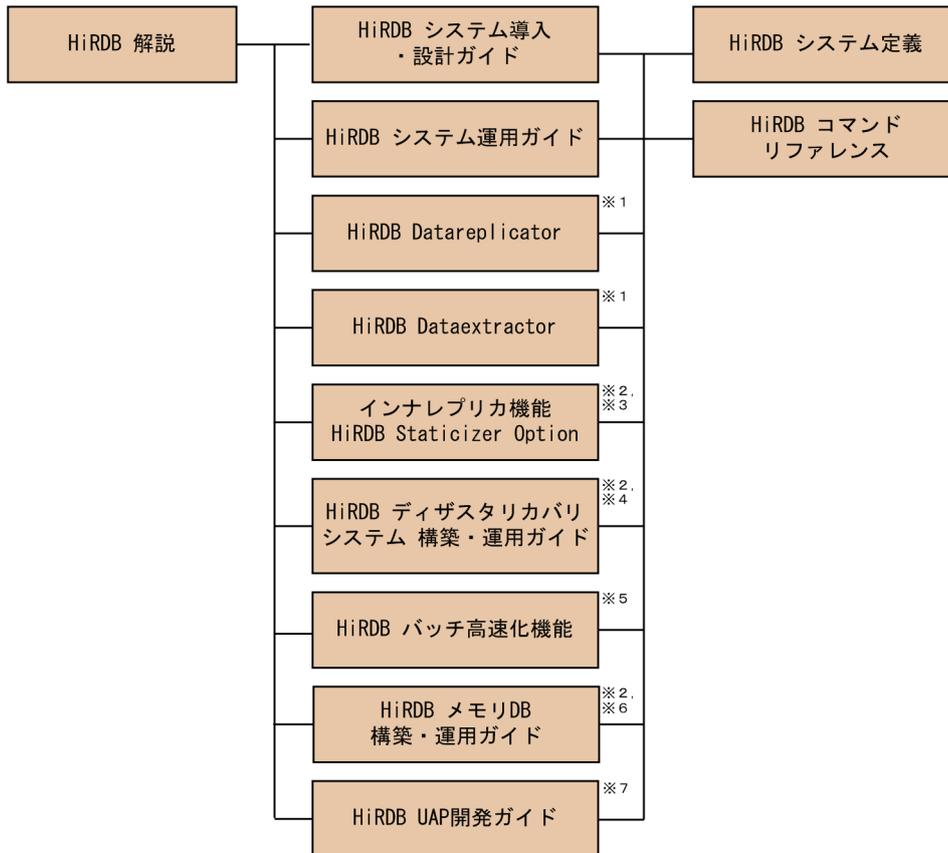
なお、本文中で使用している HiRDB Version 10 のマニュアル名は、Version 10, (UNIX(R)用), (Windows(R)用) を省略して表記しています。使用しているプラットフォームに応じて UNIX 用または Windows 用のマニュアルを参照してください。

## ■ 利用者ごとの関連マニュアル

HiRDB のマニュアルをご利用になる場合、利用者ごとに次のようにお読みください。

また、より理解を深めるために、左側のマニュアルから順にお読みいただくことをお勧めします。

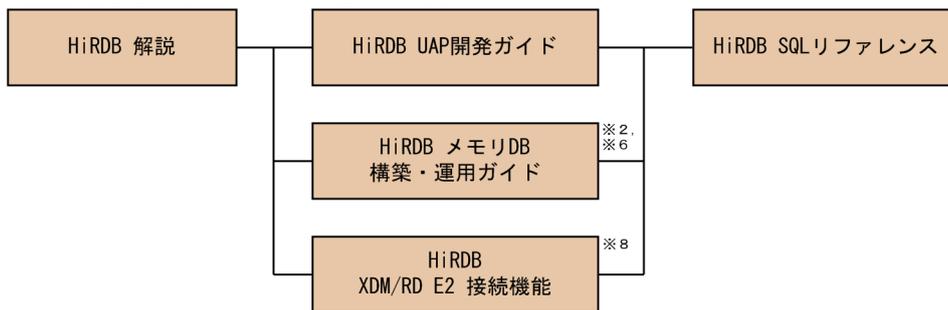
システム管理者が利用するマニュアル



表の作成者が利用するマニュアル



UAP作成者、およびUAP実行者が利用するマニュアル



- 注※1 レプリケーション機能を使用してデータ連携をする場合にお読みください。
- 注※2 UNIX用マニュアルです。Windows用はありません。
- 注※3 インナレプリカ機能を使用する場合にお読みください。
- 注※4 ディザスタリカバリシステムを構築する場合にお読みください。
- 注※5 インメモリデータ処理によるバッチ高速化を行う場合にお読みください。
- 注※6 メモリDBを使用する場合にお読みください。
- 注※7 OLTPシステムと連携する場合は必ずお読みください。
- 注※8 XDM/RD E2接続機能を使用して、XDM/RD E2のデータベースを操作する場合にお読みください。

## ■ このマニュアルでの表記

このマニュアルでは製品名称及び名称について次のように表記しています。ただし、それぞれのプログラムについての表記が必要な場合はそのまま表記しています。

製品名称または名称	表記	
HiRDB Server Version 10	HiRDB/シングルサーバ	HiRDB 又は HiRDB サーバ
	HiRDB/パラレルサーバ	
HiRDB/Developer's Kit Version 10	HiRDB/Developer's Kit	HiRDB クライアント
HiRDB/Developer's Kit Version 10(64)		
HiRDB/Run Time Version 10	HiRDB/Run Time	
HiRDB/Run Time Version 10(64)		
HiRDB Datareplicator Version 10	HiRDB Datareplicator	
HiRDB Dataextractor Version 10	HiRDB Dataextractor	
HiRDB Advanced High Availability Version 10	HiRDB Advanced High Availability	
HiRDB Accelerator Version 10	HiRDB Accelerator	
HiRDB Non Recover Front End Server Version 10	HiRDB Non Recover FES	
HiRDB Staticizer Option Version 10	HiRDB Staticizer Option	
HiRDB Text Search Plug-in Version 9	HiRDB Text Search Plug-in	
HiRDB XML Extension Version 9	HiRDB XML Extension	
Microsoft <sup>(R)</sup> SQL Server	SQL Server	
HP-UX 11i V3 (IPF)	HP-UX または HP-UX (IPF)	
AIX V7.1	AIX	
AIX V7.2		
Linux <sup>(R)</sup>	Linux	
Red Hat Enterprise Linux <sup>(R)</sup> 6 (64-bit x86_64)	Linux 6	Linux
Red Hat Enterprise Linux <sup>(R)</sup> 7 (64-bit x86_64)	Linux 7	
Red Hat Enterprise Linux <sup>(R)</sup> 6 (64-bit x86_64)	Linux (EM64T)	
Red Hat Enterprise Linux <sup>(R)</sup> 7 (64-bit x86_64)		
Red Hat Enterprise Linux <sup>(R)</sup> 6 (64-bit x86_64)	Linux 6 (64-bit x86_64)	Linux 6

製品名称または名称	表記		
Red Hat Enterprise Linux(R) 7 (64-bit x86_64)	Linux 7 (64-bit x86_64)	Linux 7	
Microsoft(R) Windows Server(R) 2012 Standard	Windows Server 2012		
Microsoft(R) Windows Server(R) 2012 Datacenter			
Microsoft(R) Windows Server(R) 2012 R2 Standard	Windows Server 2012 R2		
Microsoft(R) Windows Server(R) 2012 R2 Datacenter			
Microsoft(R) Windows Server(R) 2016 Standard	Windows Server 2016 Standard	Windows Server 2016	
Microsoft(R) Windows Server(R) 2016 Datacenter	Windows Server 2016 Datacenter		
Microsoft(R) Windows(R) 7 Professional	Windows 7 Professional	Windows 7	
Microsoft(R) Windows(R) 7 Enterprise	Windows 7 Enterprise		
Microsoft(R) Windows(R) 7 Ultimate	Windows 7 Ultimate		
Microsoft(R) Windows(R) 7 Professional (x64)	Windows 7 (x64)		
Microsoft(R) Windows(R) 7 Enterprise (x64)			
Microsoft(R) Windows(R) 7 Ultimate (x64)			
Microsoft(R) Windows(R) 7 Professional (x64)	Windows 7 Professional (x64)		
Microsoft(R) Windows(R) 7 Enterprise (x64)	Windows 7 Enterprise (x64)		
Microsoft(R) Windows(R) 7 Ultimate (x64)	Windows 7 Ultimate (x64)		
Windows(R) 8.1 (Core Edition)	Windows 8.1 (Core Edition)		Windows 8.1
Windows(R) 8.1 Pro	Windows 8.1 Pro		
Windows(R) 8.1 Enterprise	Windows 8.1 Enterprise		
Windows(R) 8.1 (Core Edition) (x64)	Windows 8.1 (Core Edition) (x64)		
Windows(R) 8.1 Pro (x64)	Windows 8.1 Pro (x64)		

製品名称または名称	表記	
Windows(R) 8.1 Enterprise (x64)	Windows 8.1 Enterprise (x64)	
Windows(R) 10 Home	Windows 10 Home	Windows 10
Windows(R) 10 Pro	Windows 10 Pro	
Windows(R) 10 Enterprise	Windows 10 Enterprise	
Windows(R) 10 Home (x64)	Windows 10 Home (x64)	
Windows(R) 10 Pro (x64)	Windows 10 Pro (x64)	
Windows(R) 10 Enterprise (x64)	Windows 10 Enterprise (x64)	
Oracle9i	Oracle	
Oracle 10g		
Oracle Database 11g		
VOS3/XS, VOS3/US, VOS3/LS, VOS3/FS, VOS3/AS, VOS3/ES1 及びこれらの OS が稼働するシステム	VOS3	
VOS1/LS, VOS1/FS, VOS1/ES2 及びこれらの OS が稼働するシステム	VOS1	

- Windows Server 2012, および Windows Server 2016 を総称して Windows Server と表記します。また, Windows Server, Windows 7, Windows 8.1, および Windows 10 を総称して Windows と表記します。
- HP-UX, AIX, 及び Linux を総称して UNIX と表記します。

## ■ このマニュアルで使用する略語

このマニュアルで使う英略語の一覧を次に示します。

英略語	英字の表記
AP	<u>A</u> pplication <u>P</u> rogram
BLOB	<u>B</u> inary <u>L</u> arge <u>O</u> bject
CD-ROM	<u>C</u> ompact <u>D</u> isc - <u>R</u> ead <u>O</u> nly <u>M</u> emory
DB	<u>D</u> atab <u>a</u> se
DBMS	<u>D</u> atab <u>a</u> se <u>M</u> anagement <u>S</u> ystem

英略語	英字の表記
EBCDIC	<u>E</u> xtended <u>B</u> inary <u>C</u> oded <u>D</u> ecimal <u>I</u> nterchange <u>C</u> ode
EBCDIK	<u>E</u> xtended <u>B</u> inary <u>C</u> oded <u>D</u> ecimal <u>I</u> nterchange <u>K</u> ana code
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
IPF	<u>I</u> tanium <sub>(R)</sub> <u>P</u> rocessor <u>F</u> amily
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandard
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
OS	<u>O</u> perating <u>S</u> ystem
PC	<u>P</u> ersonal <u>C</u> omputer
PP	<u>P</u> rogram <u>P</u> roduct
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol / <u>I</u> nternet <u>P</u> rotocol
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
WS	<u>W</u> orkstation
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage

## ■ パス名の表記

- パス名の区切りは「/」で表記しています。Windows 版 Datareplicator Extension を使用している場合はマニュアル中の「/」を「¥」に置き換えてください。ただし、Windows 版と UNIX 版でパス名が異なる場合は、それぞれのパス名を表記しています。
- HiRDB 運用ディレクトリのパスを\$PDDIR と表記します。ただし、Windows 版と UNIX 版でパス名が異なるため、それぞれを表記する場合、Windows 版は%PDDIR%と表記します。例を次に示します。

UNIX 版：\$PDDIR/client/lib/

Windows 版：%PDDIR%¥CLIENT¥UTL¥

## ■ ログの表記

### ●Windows 版の場合

Windows のイベントビューアで表示されるアプリケーションログをイベントログと表記します。イベントログは、次の方法で参照できます。

## 〈手順〉

1. [スタート] - [プログラム] - [管理ツール (共通)] - [イベントビューア] を選択します。
2. [ログ] - [アプリケーション] を選択します。

アプリケーションログが表示されます。「ソース」の列が「HiRDB Datareplicator」になっているのが HiRDB Datareplicator が出力したメッセージです。

## ●UNIX 版の場合

OS のログを syslogfile と表記します。syslogfile は、/etc/syslog.conf でログ出力先に指定しているファイルです。一般的には、次のファイルが syslogfile となります。

OS	ファイル
HP-UX	/var/adm/syslog/syslog.log
AIX	/var/adm/ras/syslog
Linux	/var/log/messages

## ■ Windows の操作説明で使う表記

Windows の操作説明で使う記号を次に示します。

記号	意味
[ ]	ボタンやテキストボックスなど、画面に表示されている要素を示します。
[ ] - [ ]	画面に表示されるメニューやアイコンなどを選択する操作を示します。

Windows の用語「ディレクトリ」と「フォルダ」は、「ディレクトリ」に統一して表記します。

## ■ 図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

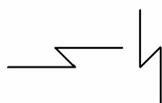
### ●プログラム



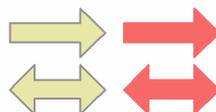
### ●工程、作業項目の流れ



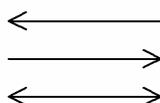
### ●通信回線



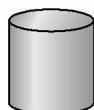
### ●データの流れ



### ●制御の流れ



### ●ファイル



## ■ このマニュアルで使用する記号

形式及び説明で使用する記号を次に示します。ここで説明する文法記述記号は、説明のための記号なので実際には記述しないでください。

文法記述記号	意 味
[ ]	この記号で囲んでいる項目は省略できることを示します。 (例) set hirdbusr = HiRDB 接続認可識別子 [/パスワード] これは、hirdbusr オペランドで、/パスワードを省略できることを示します。
	この記号で区切られた項目は、選択できることを示します。 (例) set keepalive = true   <u>false</u> これは、keepalive オペランドの設定値として、true 又は false のどちらかを選択できることを示します。
{ }	この記号で囲まれている複数の項目のうち、一つを選択できることを示します。 (例) set startmode = { trn   tbl   spd } これは、startmode オペランドの設定値として、trn, tbl, 又は spd のどれかを選択できることを示します。
_(下線)	この下線記号で示す項目は、省略したときに仮定される値であることを示します。 (例) set keepalive = true   <u>false</u> これは、keepalive オペランドを省略すると、set keepalive = false と仮定されることを示します。
{ { } }…	この記号で囲まれた項目を一つの単位として、繰り返して指定できることを示します。 (例) { { name フィールド名称 [ const 初期値 ] } }… これは、"name フィールド名称 [ const 初期値 ]" を複数回繰り返して指定できることを示します。
~	この記号の後に指定する値の属性を示します。
< >	指定する値の構文要素記号を示します。
《 》	値を指定しなかったときに Datareplicator Extension が仮定する値（省略時仮定値）を示します。
(( ))	指定できる値の範囲を示します。

## ■ このマニュアルで使用する構文要素記号

指定する値の説明で使う記号です。指定する文字列には、基本的に半角文字を使ってください。全角文字を使う場合は、半角文字数で換算してください。

構文要素記号	意 味
<英字>	アルファベット (A~Z, a~z), _(下線), - (ハイフン)
<英字記号>	アルファベット (A~Z, a~z), #, @, ¥
<英数字>	<英字>, 数字 (0~9)
<英数字記号>	<英字記号>, 数字 (0~9)

構文要素記号	意味
<符号なし整数>	数字 (0~9)
<16進数字>	数字 (0~9), A~F, a~f
<識別子>	先頭がアルファベット (A~Z, a~z) の<英数字>
<記号名称>	先頭が<英字記号>の<英数字記号>
<文字列>	任意の文字の配列
<パス名>	記号名称, ¥又は/, 及び. (ピリオド)
<ファイル名>	アルファベット (A~Z, a~z), 数字 (0~9),. (ピリオド),_(下線), -(ハイフン), 及び@で構成される文字列 (最大 30 文字) 注 すべての半角文字で指定してください。UNIX の場合は、英字の大文字と小文字は区別されます。パス名は、Datareplicator Extension をインストールする OS の規則に依存します。

## ■ このマニュアルで使用する計算式の記号

このマニュアルで使用する計算式の記号の意味を次に示します。

記号	意味
↑ ↑	計算結果の値を小数点以下で切り上げることを示します。 (例) $\uparrow 34 \div 3 \uparrow$ の計算結果は「12」になります。
↓ ↓	計算結果の値を小数点以下で切り下げることを示します。 (例) $\downarrow 34 \div 3 \downarrow$ の計算結果は「11」になります。
MAX	計算結果の最も大きい値を選ぶことを示します。 (例) MAX (3×6, 4 + 7) の計算結果は「18」になります。
MIN	計算結果の最も小さい値を選ぶことを示します。 (例) MIN (3×6, 4 + 7) の計算結果は「11」になります。

## ■ KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ 1,024 バイト, 1,024<sup>2</sup> バイト, 1,024<sup>3</sup> バイト, 1,024<sup>4</sup> バイトです。

## ■ HiRDB Version 9 と HiRDB Version 10 の製品体系の違い

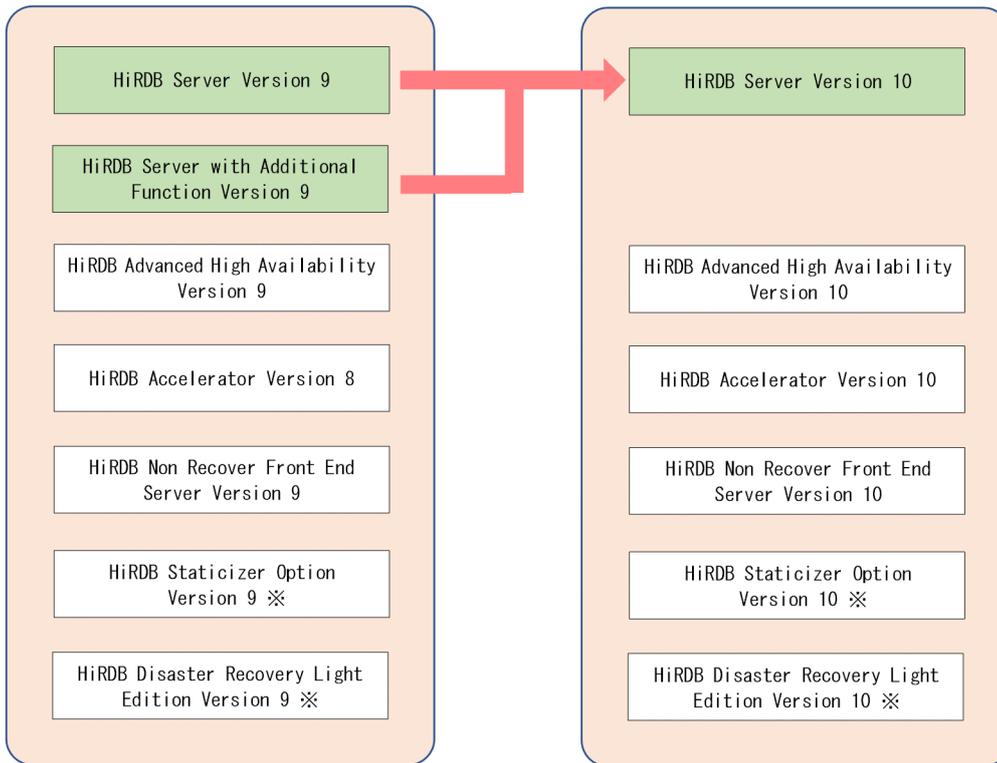
HiRDB Version 10 では、製品体系を次のように変更しました。

- HiRDB Server with Additional Function を廃止し、HiRDB Server with Additional Function の機能を HiRDB Server に統合しました。

HiRDB Version 9 と HiRDB Version 10 の製品体系の違いを次に示します。

●HiRDB Version 9

●HiRDB Version 10



注※ UNIX 版でだけ使用できる製品です。

# 目次

前書き	2
変更内容	4
はじめに	6

<b>1</b>	<b>概要</b>	<b>21</b>
1.1	特長	22
1.1.1	Datareplicator Extension の目的	22
1.2	データ連動について	24
1.2.1	データ連動システムの組み合わせ	24
1.3	ソフトウェア構成	25
1.3.1	Oracle から HiRDB へデータ連動する場合のソフトウェア構成	25
1.3.2	SQL Server から ODBC ドライバを使用して HiRDB へデータ連動する場合のソフトウェア構成	26
1.3.3	HiRDB から Oracle へデータ連動する場合のソフトウェア構成	26
1.3.4	HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合のソフトウェア構成	27
1.4	データ連動システムの構築手順	29
<b>2</b>	<b>環境の設定</b>	<b>30</b>
2.1	Datareplicator Extension に関連する製品	31
2.1.1	抽出側システムに必要な製品	31
2.1.2	反映側システムに必要な製品	31
2.2	Datareplicator Extension のインストール (UNIX)	33
2.2.1	インストール前の準備	33
2.2.2	インストールの実行	33
2.2.3	Datareplicator Extension のアンインストール	33
2.3	ディレクトリ構成 (UNIX)	34
2.4	環境変数の設定 (UNIX)	35
2.4.1	抽出側 Datareplicator Extension の環境変数の設定 (UNIX)	35
2.4.2	反映側 Datareplicator Extension の環境変数の設定 (UNIX)	36
2.5	通信環境の設定 (UNIX)	37
2.6	プロセスファイルの作成 (UNIX)	38
2.7	データ連動リソース管理ユーザの登録 (UNIX)	39
2.7.1	データ連動リソース管理ユーザに対する権限付与	39
2.7.2	キューモニタプロセスの起動	39
2.7.3	データ連動リソース管理ユーザの名称制限	40
2.7.4	その他	40

2.8	Datareplicator 管理者の登録 (UNIX)	41
2.9	Datareplicator Extension のインストール (Windows)	42
2.9.1	インストール前の準備	42
2.9.2	インストールの実行	42
2.9.3	インストール時に登録される内容	44
2.9.4	Datareplicator Extension のアンインストール	45
2.10	ディレクトリ構成 (Windows)	46
2.11	環境変数の設定 (Windows)	47
2.11.1	抽出側 Datareplicator Extension の環境変数の設定 (Windows)	47
2.11.2	反映側 Datareplicator Extension の環境変数の設定 (Windows)	48
2.12	通信環境の設定 (Windows)	49
2.13	ODBC 環境の設定 (Windows)	50
2.13.1	ODBC データソースの作成	50
2.13.2	各 ODBC ドライバの注意点	50
2.14	データ連動リソース管理ユーザの登録 (Windows)	52
2.14.1	データ連動リソース管理ユーザに対する権限付与	52
2.14.2	キューモニタプロセスの起動 (Oracle の場合)	53
2.14.3	データ連動リソース管理ユーザの名称制限	53
2.14.4	その他	53
2.15	Datareplicator 管理者の登録 (Windows)	55

### **3 データ連動機能 56**

3.1	抽出側 Datareplicator Extension の抽出処理	57
3.1.1	抽出処理の概要	57
3.1.2	抽出処理のタイミング	58
3.1.3	更新情報の取り出し	59
3.1.4	データ型の変換	59
3.1.5	更新情報及びトランザクション情報の出力	61
3.1.6	抽出間隔の制御	62
3.1.7	データ連動リソースの作成	62
3.2	反映側 Datareplicator Extension の反映処理	66
3.2.1	反映処理の概要	66
3.2.2	反映処理の方式	66
3.2.3	反映処理のタイミング	66
3.2.4	時系列情報の取得	67
3.3	Oracle, SQLServer から抽出を行う場合の滞留監視機能	68
3.4	イベント機能	69

<b>4</b>	<b>システム設計 70</b>	
4.1	抽出側 DB と反映側 DB との対応の設計	71
4.1.1	抽出側 DB の条件	71
4.1.2	データ型の対応の設計	71
4.1.3	文字コードの対応の設計	81
4.1.4	時系列情報表の作成	81
<b>5</b>	<b>定義 84</b>	
5.1	抽出側 Datareplicator Extension を使用する場合の定義 (Oracle の場合)	85
5.1.1	抽出システム定義	85
5.1.2	抽出環境定義	91
5.1.3	送信環境定義	93
5.1.4	抽出定義	94
5.2	抽出側 Datareplicator Extension を使用する場合の定義 (SQL Server の場合)	98
5.2.1	抽出システム定義	98
5.2.2	抽出環境定義	103
5.2.3	送信環境定義	105
5.2.4	抽出定義	106
5.3	反映側 Datareplicator Extension を使用する場合の定義	110
5.3.1	反映システム定義	110
5.3.2	反映環境定義	113
5.3.3	反映定義	117
<b>6</b>	<b>運用 121</b>	
6.1	抽出側 Datareplicator Extension の起動と終了 (Oracle の場合)	122
6.1.1	抽出側 Datareplicator Extension の起動	122
6.1.2	抽出側 Datareplicator Extension の終了	123
6.2	抽出側 Datareplicator Extension の運用 (Oracle の場合)	124
6.2.1	抽出対象表の運用	124
6.2.2	イベント機能を使用する場合の運用	125
6.2.3	抽出側 Datareplicator Extension 運用時の注意	126
6.3	抽出側 Datareplicator Extension の起動と終了 (SQL Server の場合)	127
6.3.1	抽出側 Datareplicator Extension の起動	127
6.3.2	抽出側 Datareplicator Extension の終了	128
6.4	抽出側 Datareplicator Extension の運用 (SQL Server の場合)	129
6.4.1	抽出対象表の運用	129
6.4.2	イベント機能を使用する場合の運用	130
6.4.3	抽出側 Datareplicator Extension 運用時の注意	131
6.5	反映側 Datareplicator Extension の起動と終了	133

- 6.5.1 反映側 Datareplicator Extension の起動 133
- 6.5.2 反映側 Datareplicator Extension の終了 133
- 6.5.3 反映側 Datareplicator Extension 運用時の注意 133
- 6.6 ファイルの二重化を使うときの運用 134
- 6.6.1 ファイルの一重化から二重化への変更 134
- 6.6.2 二重化するファイルの変更 135
- 6.6.3 ファイルの二重化から一重化への変更 137

## 7 コマンドの文法 139

コマンドの概要 140

hderesstateO (データ連動リソースの状態の取得) 142

hderesstateS (データ連動リソースの状態の取得) 147

## 8 障害時の対処 152

- 8.1 抽出側 Datareplicator Extension の障害時の対処 153
- 8.1.1 抽出側システムでエラーが発生した場合の対処 (Oracle の場合) 153
- 8.1.2 抽出側システムでエラーが発生した場合の対処 (SQL Server の場合) 155
- 8.2 反映側 Datareplicator Extension の障害時の対処 158
- 8.3 データ連動回復機能 159

## 付録 161

- 付録 A Datareplicator Extension 定義の予約語 162
- 付録 B プロセスファイルの作成手順及び作成例 166
- 付録 B.1 プロセスファイルの作成手順 166
- 付録 B.2 プロセスファイル作成シェルの実行例 167

## 索引 169

# 1

## 概要

この章では、Datareplicator Extension の特長、データ連動の概要、ソフトウェア構成、及びデータ連動システムの構築手順について説明します。

## 1.1 特長

---

Datereplicator Extension は、他社 DBMS と HiRDB システムのデータを自動的に連動できるようにする製品です。

### 1.1.1 Datereplicator Extension の目的

今まで、HiRDB システム同士、及びメインフレームと HiRDB システムがデータ連動するための製品として、Datereplicator がありました。

しかし、近年システムのマルチベンダ化が進み、上記のようなデータ連動だけではなく、他ベンダの DB とデータ連動したいという意向が高まってきました。

その意向を受けたのが、Datereplicator の拡張機能である **Datereplicator Extension** です。Datereplicator Extension を使うと、他社 DBMS (Oracle 及び SQL Server) と HiRDB システムとのデータ連動ができます。

#### (1) データ連動を適用できる業務システム

他社 DBMS からの更新内容の抽出は、トリガ及びキューテーブルを使用します。そのため、HiRDB システム同士のデータ連動に比べて、次の点で性能が低下する可能性があります。キューテーブルについては、[\[3.1.1 抽出処理の概要\]](#) を参照してください。

- 更新 AP に、従来の HiRDB システム同士のデータ連動にかかる更新オーバーヘッドに加えて、トリガ内で実行する処理オーバーヘッドが加わります。
- HiRDB システムからの抽出時のように、ブロッキングされた複数の更新データを一括して取り出すことはできないので、更新データを一件ずつ取り出す必要があります。

したがって、他社 DBMS とのデータ連動は、バッチなどの大量更新をするシステムではなく、オンライン系システムへの適用をお勧めします。

#### (2) データ連動できるシステム

Datereplicator Extension を使用すると、次に示す組み合わせでデータ連動できます。

- HiRDB のデータ連動システムと Oracle<sup>※1</sup> の DB とのデータ連動
- HiRDB のデータ連動システムと SQL Server<sup>※2</sup> の DB とのデータ連動

注※1

Windows 版 Oracle を反映側 DB にする場合は ODBC ドライバを使用します。

注※2

SQL Server とデータ連動する場合は ODBC ドライバを使用します。

データ連動システムの組み合わせについては、「[1.2.1 データ連動システムの組み合わせ](#)」を参照してください。

### (3) データ連動システム用語

データ連動システムのうち、更新されるデータベースがあるシステムを**抽出側システム**、抽出側システムの Datareplicator Extension を**抽出側 Datareplicator Extension** といいます。

抽出側システムのデータベースを**抽出側 DB** といい、このうち、HiRDB を **HiRDB** 又は**抽出側 HiRDB** といいます。

また、抽出側システムから自動的に更新内容が反映されるシステムを**反映側システム**、反映側システムの Datareplicator Extension を**反映側 Datareplicator Extension** といいます。

反映側システムのデータベースを**反映側 DB** といい、このうち、HiRDB を **反映側 HiRDB** といいます。

## 1.2 データ連動について

---

### 1.2.1 データ連動システムの組み合わせ

Datereplicator Extension を使うと、HiRDB と Oracle 又は HiRDB と SQL Server とがデータ連動できます。抽出側 DB 及び反映側 DB として使用できる DBMS を次の表に示します。これらの DBMS はどの組み合わせでも使用できます。ただし、抽出側 DB も反映側 DB も HiRDB の場合、Datereplicator Extension は必要ありません。

表 1-1 抽出側 DB, 反映側 DB に使用できる DBMS

システム	DBMS (適用 OS)
抽出側 DB	<ul style="list-style-type: none"><li>• HiRDB (UNIX, Windows)</li><li>• Oracle (UNIX, Windows)</li><li>• SQL Server (Windows)</li></ul>
反映側 DB	<ul style="list-style-type: none"><li>• HiRDB (UNIX, Windows)</li><li>• Oracle (UNIX, Windows<sup>※</sup>)</li><li>• SQL Server (Windows)</li></ul>

#### 注※

データ連動に ODBC を使用します。

データ連動するときに必要な関連製品については、「[2.1 Datereplicator Extension に関連する製品](#)」を参照してください。

## 1.3 ソフトウェア構成

データ連動システムのシステム構成には、「1.2.1 データ連動システムの組み合わせ」で示した DB の組み合わせの分だけの種類があります。ここでは、代表的なシステム構成のデータ連動システムのソフトウェア構成について説明します。説明するシステム構成は次のとおりです。

- Oracle から HiRDB へデータ連動するシステム構成
- SQL Server から HiRDB へデータ連動するシステム構成
- HiRDB から Oracle へデータ連動するシステム構成
- HiRDB から ODBC ドライバを使用して SQL Server へデータ連動するシステム構成

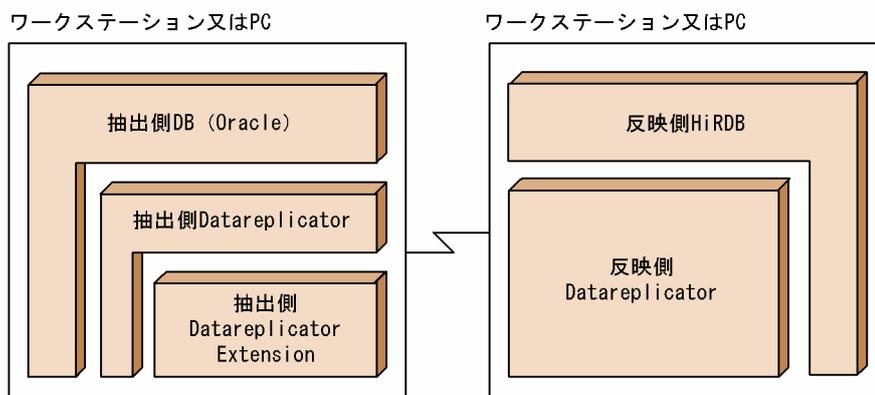
それぞれのシステム構成によって、必要なソフトウェアが異なります。システム構成と必要なソフトウェアについては、「2.1 Datareplicator Extension に関連する製品」を参照してください。

ここに示す Datareplicator Extension 側のシステム構成は、UNIX と Windows で共通です。

### 1.3.1 Oracle から HiRDB へデータ連動する場合のソフトウェア構成

Oracle から HiRDB へデータ連動する場合のソフトウェア構成を次の図に示します。

図 1-1 Oracle から HiRDB へデータ連動する場合のソフトウェア構成



#### (1) 抽出対象の Oracle と抽出側 Datareplicator Extension の対応

Oracle から HiRDB へデータ連動する場合の、抽出対象の Oracle と抽出側 Datareplicator Extension の対応を次に示します。

- Oracle と抽出側 Datareplicator Extension との対応  
Oracle : 抽出側 Datareplicator Extension = 1 : 1

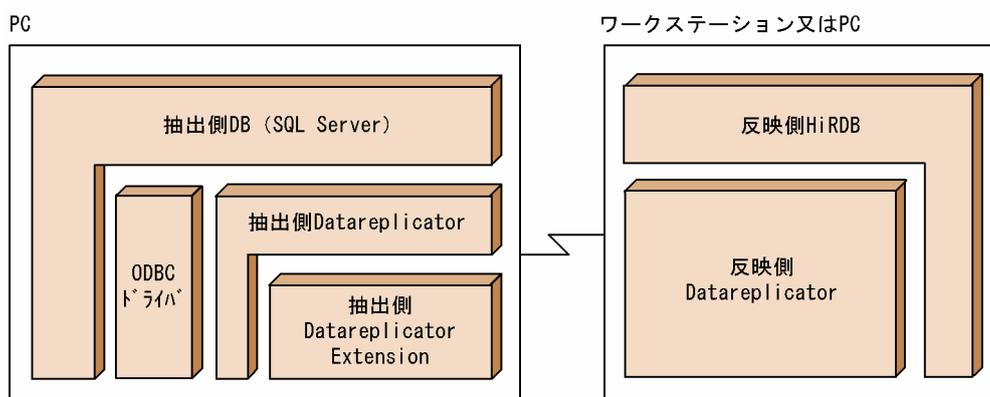
UNIX 版 Datareplicator Extension の場合は、ほかの HiRDB から更新内容を抽出するシステムと Oracle から更新内容を抽出するシステムを、同一マシン上で構築できます。

Windows 版 Datareplicator Extension の場合は、抽出側 Datareplicator の運用ディレクトリを指定する環境変数 HDEPATH をシステム環境変数として設定するため、ほかの HiRDB から更新内容を抽出するシステムと Oracle から更新内容を抽出するシステムを、同一マシン上では構築できません。

### 1.3.2 SQL Server から ODBC ドライバを使用して HiRDB へデータ連動する場合のソフトウェア構成

SQL Server から ODBC ドライバを使用して HiRDB へデータ連動する場合のソフトウェア構成を次の図に示します。

図 1-2 SQL Server から ODBC ドライバを使用して HiRDB へデータ連動する場合のソフトウェア構成



#### (1) 抽出対象の SQL Server と抽出側 Datareplicator Extension の対応

SQL Server から ODBC ドライバを使用して HiRDB へデータ連動する場合の、抽出対象の SQL Server と抽出側 Datareplicator Extension の対応を次に示します。

- SQL Server と抽出側 Datareplicator Extension との対応  
SQL Server : 抽出側 Datareplicator Extension = 1 : 1

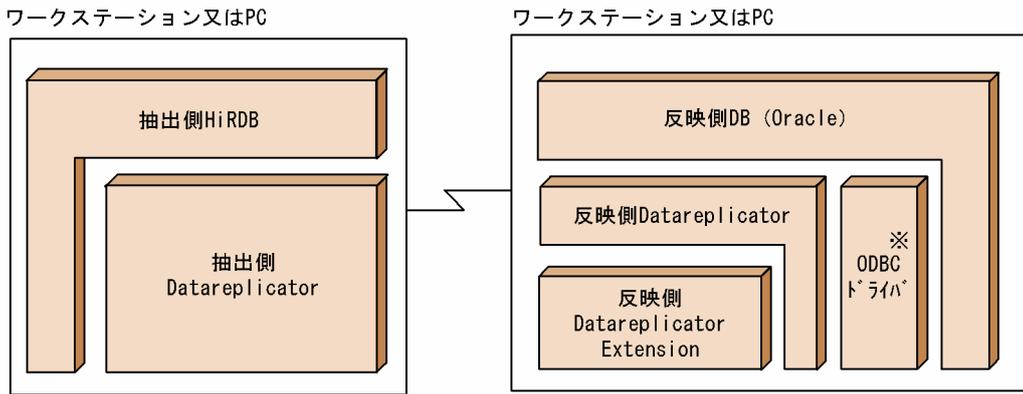
UNIX 版 Datareplicator Extension の場合は、ほかの HiRDB から更新内容を抽出するシステムと SQL Server から更新内容を抽出するシステムを、同一マシン上で構築できます。

Windows 版 Datareplicator Extension の場合は、抽出側 Datareplicator の運用ディレクトリを指定する環境変数 HDEPATH をシステム環境変数として設定するため、ほかの HiRDB から更新内容を抽出するシステムと SQL Server から更新内容を抽出するシステムを、同一マシン上では構築できません。

### 1.3.3 HiRDB から Oracle へデータ連動する場合のソフトウェア構成

HiRDB から Oracle へデータ連動する場合のソフトウェア構成を次の図に示します。

図 1-3 HiRDB から Oracle へデータ連動する場合のソフトウェア構成



注※

Windows 版 Oracle の場合は、ODBC ドライバが必要です。

## (1) 反映対象の Oracle と反映側 Datareplicator Extension の対応

HiRDB から Oracle にデータ連動する場合の、反映対象の Oracle と反映側 Datareplicator Extension の対応を次に示します。

- Oracle と反映側 Datareplicator Extension との対応

Oracle : 反映側 Datareplicator Extension = 1 : 1

UNIX 版 Datareplicator Extension の場合は、ほかの HiRDB へ更新内容を反映するシステムと Oracle へ更新内容を反映するシステムを、同一マシン上で構築できます。

Windows 版 Datareplicator Extension の場合は、反映側 Datareplicator の運用ディレクトリを指定する環境変数 HDSPATH をシステム環境変数として設定するため、ほかの HiRDB へ更新内容を反映するシステムと Oracle へ更新内容を反映するシステムを、同一マシン上では構築できません。

## (2) 抽出側 HiRDB の形態

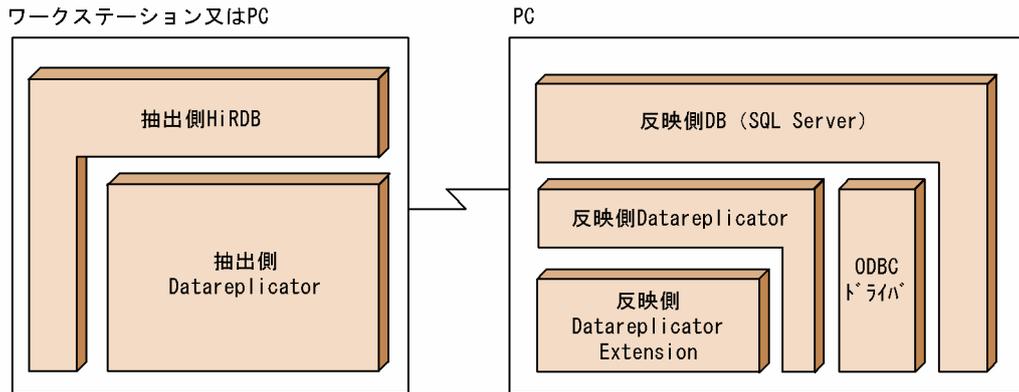
HiRDB から Oracle へデータ連動する場合は、HiRDB システム同士のデータ連動と同様に、抽出側 HiRDB がシングルサーバの場合とパラレルサーバの場合とでソフトウェア構成が異なります。

抽出側 HiRDB がシングルサーバの場合の形態、及び抽出側 HiRDB がパラレルサーバの場合の形態については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 1.3.4 HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合のソフトウェア構成

HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合のソフトウェア構成を次の図に示します。

図 1-4 HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合のソフトウェア構成



## (1) 反映対象の SQL Server と反映側 Datareplicator Extension の対応

HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合の、反映対象の SQL Server と反映側 Datareplicator Extension の対応を次に示します。

- SQL Server と反映側 Datareplicator Extension との対応

SQL Server : 反映側 Datareplicator Extension = 1 : 1

UNIX 版 Datareplicator Extension の場合は、ほかの HiRDB へ更新内容を反映するシステムと SQL Server へ更新内容を反映するシステムを、同一マシン上で構築できます。

Windows 版 Datareplicator Extension の場合は、反映側 Datareplicator の運用ディレクトリを指定する環境変数 HDSPATH をシステム環境変数として設定するため、ほかの HiRDB へ更新内容を反映するシステムと SQL Server へ更新内容を反映するシステムを、同一マシン上では構築できません。

## (2) 抽出側 HiRDB の形態

HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合は、HiRDB システム同士のデータ連動と同様に、抽出側 HiRDB がシングルサーバの場合とパラレルサーバの場合とでソフトウェア構成が異なります。

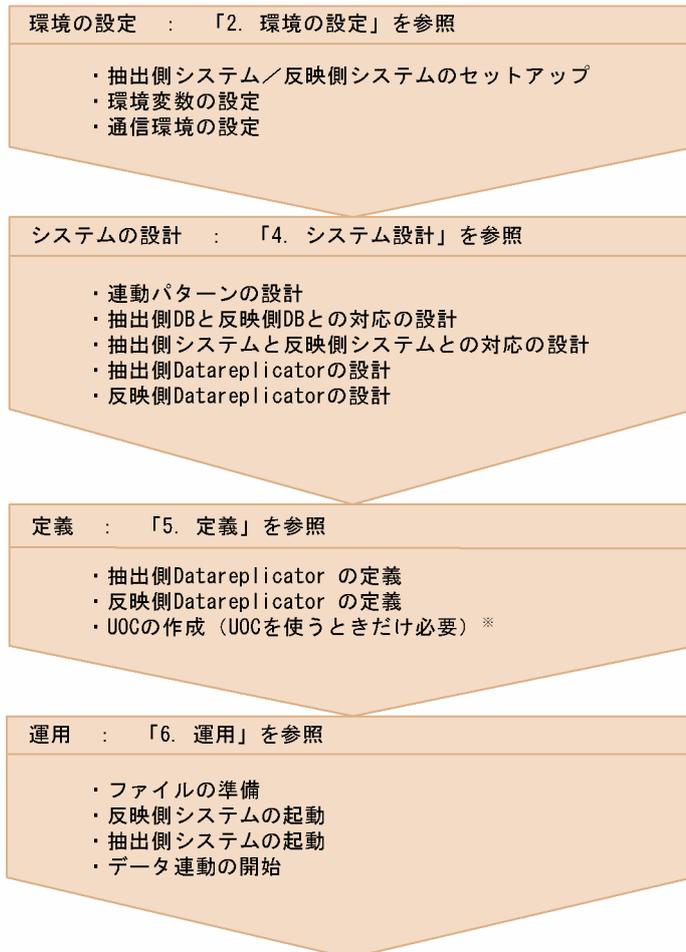
抽出側 HiRDB がシングルサーバの場合の形態及び抽出側 HiRDB がパラレルサーバの場合の形態については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

## 1.4 データ連動システムの構築手順

Datereplicator Extension を導入してから、データ連動するまでに必要なシステム構築手順を次の図に示します。HiRDB システムの構築手順と運用方法については、次に示すマニュアルを参照してください。

- ・「HiRDB システム導入・設計ガイド」
- ・「HiRDB システム運用ガイド」

図 1-5 データ連動するまでに必要なシステム構築手順



### 注

このマニュアルでは、各構築手順で Oracle 及び SQL Server とのデータ連動時に差異がある項目についてだけ記載しています。

各構築手順の基本的な項目及び HiRDB とのデータ連動時と差異がない項目については、マニュアル「HiRDB データ連動機能 HiRDB Datereplicator」を参照してください。

### 注※

UOC 及びメッセージについては、マニュアル「HiRDB データ連動機能 HiRDB Datereplicator」を参照してください。

# 2

## 環境の設定

この章では、Datareplicator Extension に関連する製品、Datareplicator のインストール手順と作成されるディレクトリ、環境変数の設定、及び通信環境の設定について説明します。環境を設定する方法は、UNIX 版 Datareplicator Extension (2.2～2.8) と Windows 版 Datareplicator Extension (2.9～2.15) に分けて説明します。

## 2.1 Datareplicator Extension に関連する製品

Datareplicator Extension を使うときに必要なソフトウェアについて説明します。

- 抽出側システムに必要な製品
- 反映側システムに必要な製品

### 2.1.1 抽出側システムに必要な製品

抽出側システムに必要な製品の一覧を表 2-1 と表 2-2 に示します。

表 2-1 抽出側システムに必要な製品の一覧 (UNIX)

抽出側 DB	製品名	備考
OS	HP-UX, Solaris, AIX, 又は Linux	どれか一つが必要です。
前提 PP	Datareplicator	データ連動するときに必要です。
Oracle	Oracle	抽出側 DB が Oracle の場合に必要です。

表 2-2 抽出側システムに必要な製品の一覧 (Windows)

抽出側 DB	製品名	備考
OS	Windows 2000, Windows Server 2003, Windows XP, Windows Server 2008, 又は Windows Vista	どれか一つが必要です。
前提 PP	Datareplicator	データ連動するときに必要です。
SQL Server	Microsoft SQL Server 7.0	抽出側 DB が SQL Server の場合に必要です。
	Microsoft SQL Server 2000	
	SQL Server (ODBC ドライバ)	ODBC ドライバを使用してデータ連動する場合に必要です。
Oracle	Oracle	抽出側 DB が Oracle の場合に必要です。

### 2.1.2 反映側システムに必要な製品

反映側システムに必要な製品の一覧を表 2-3 と表 2-4 に示します。

表 2-3 反映側システムに必要な製品の一覧 (UNIX)

反映側 DB	製品名	備考
OS	HP-UX, Solaris, 又は AIX	どれか一つが必要です。
前提 PP	Datareplicator	データ連動するときに必要です。

反映側 DB	製品名	備考
Oracle	Oracle	反映側 DB が Oracle の場合に必要です。

表 2-4 反映側システムに必要な製品の一覧 (Windows)

反映側 DB	製品名	備考
OS	Windows 2000, Windows Server 2003, Windows XP, Windows Server 2008, 又は Windows Vista	どれか一つが必要です。
前提 PP	Datareplicator	データ連動するときに必要です。
周辺 PP	Visual C++ 4.2, Visual C++ 5.x, 又は Visual C++ 6.0	反映情報編集 UOC を使用するときが必要です。
SQL Server	Microsoft SQL Server 7.0	反映側 DB が SQL Server の場合に必要です。
	Microsoft SQL Server 2000	
	SQL Server (ODBC ドライバ)	ODBC ドライバを使用してデータ連動する場合に必要です。
Oracle	Oracle	反映側 DB が Oracle の場合に必要です。
	Oracle ODBC driver	ODBC ドライバを使用してデータ連動する場合に必要です。

## 2.2 Datareplicator Extension のインストール (UNIX)

---

UNIX 版 Datareplicator Extension をインストールする方法について説明します。

### 2.2.1 インストール前の準備

Datareplicator Extension をインストールする前に、次に示す作業が必要になります。

- 稼働環境の確認
- ユーザ権限の確認

#### (1) 稼働環境の確認

次に示す内容を確認してください。

- Datareplicator Extension をインストールするマシンのディスクの空き容量が十分であることを確認してください。
- Datareplicator Extension が動作するのに必要な製品がインストールされていることを確認してください。

#### (2) ユーザ権限の確認

インストールを実行する利用者がスーパーユーザであることを確認してください。

### 2.2.2 インストールの実行

UNIX 版 Datareplicator Extension をインストールする前に、あらかじめ UNIX 版 Datareplicator をインストールしておく必要があります。

[[2.1 Datareplicator Extension に関連する製品](#)] を参照して、UNIX 版 Datareplicator Extension を使うために必要な製品をインストールしてください。UNIX 版 Datareplicator Extension のインストールは、日立 PP インストーラで行います。

### 2.2.3 Datareplicator Extension のアンインストール

UNIX 版 Datareplicator Extension のアンインストールは、日立 PP インストーラで実施してください。

## 2.3 ディレクトリ構成 (UNIX)

---

Datareplicator Extension は、Datareplicator のインストールディレクトリにインストールされます。Datareplicator のインストールディレクトリについては、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

## 2.4 環境変数の設定 (UNIX)

製品をインストール後、Datareplicator Extension を実行する前に環境変数を設定する必要があります。ここでは、設定する環境変数を抽出側 Datareplicator Extension と反映側 Datareplicator Extension に分けて説明します。

### 2.4.1 抽出側 Datareplicator Extension の環境変数の設定 (UNIX)

Oracle とデータ連動する場合は、Datareplicator の環境変数及び Oracle のアプリケーションを使用するのに必要な環境変数を設定してください。Oracle とデータ連動する場合の環境変数の設定内容を次の表に示します。

表 2-5 Oracle とデータ連動する場合の環境変数の設定内容 (抽出側 Datareplicator (UNIX))

環境変数	設定内容
PATH	Datareplicator のコマンドライブラリの名称を指定します。*1
HDEPATH	抽出側 Datareplicator の運用ディレクトリを指定します。*2
Oracle 用環境変数*3	Oracle の OCI アプリケーション実行に必要な環境変数を設定します。

#### 注※1

コマンドライブラリの名称は、/opt/hirdbds/bin/ です。Datareplicator のコマンドライブラリについては、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

#### 注※2

Datareplicator の運用ディレクトリの指定方法については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

#### 注※3

Oracle 用環境変数の設定内容については、Oracle のマニュアルを参照してください。

なお、ライブラリパスを指定する環境変数に 256 バイト以上の値を指定した場合、抽出側 Datareplicator Extension は指定値不正エラーとなります。256 バイト以上の値を指定する必要がある場合には、node\_shlibpath オペランドを指定してください。node\_shlibpath オペランドの詳細については、「5.1.1 抽出システム定義」の node\_shlibpath オペランドを参照してください。

なお、次の場合は、Oracle 用環境変数の NLS\_LANG の設定を必ずしてください。

- 抽出対象表名及び列名が 2 バイト文字コードの場合
- 抽出対象列属性に CHAR, VARCHAR, NCHAR 又は NVARCHAR2 型がある場合

NLS\_LANG の設定をしない場合は、抽出データが文字化けする場合があります、データ連動の動作を保証できません。

## 2.4.2 反映側 Datareplicator Extension の環境変数の設定 (UNIX)

Oracle とデータ連動する場合は、Datareplicator の環境変数及び Oracle のアプリケーションを使用するのに必要な環境変数を設定してください。Oracle とデータ連動する場合の環境変数の設定内容を次の表に示します。

表 2-6 Oracle とデータ連動する場合の環境変数の設定内容 (反映側 Datareplicator (UNIX))

環境変数	設定内容
PATH	Datareplicator のコマンドライブラリの名称を指定します。*1
HDSPATH	反映側 Datareplicator の運用ディレクトリを指定します。*2
Oracle 用環境変数*3	Oracle の OCI アプリケーション実行に必要な環境変数を設定します。

### 注※1

コマンドライブラリの名称は、「opt/hirdbds/bin/」です。Datareplicator のコマンドライブラリについては、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 注※2

Datareplicator の運用ディレクトリの指定方法については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 注※3

Oracle 用環境変数の設定内容については、Oracle のマニュアルを参照してください。

なお、次の場合は、Oracle 用環境変数の NLS\_LANG の設定を必ずしてください。

- 抽出対象表名及び列名が 2 バイト文字コードの場合
- 抽出対象列属性に CHAR, VARCHAR, NCHAR 又は NVARCHAR2 型がある場合

NLS\_LANG の設定をしない場合は、抽出データが文字化けする場合があります、データ連動の動作を保証できません。

## 2.5 通信環境の設定 (UNIX)

---

Datareplicator Extension を実行する前に、通信環境を設定する必要があります。

通信環境の設定については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

## 2.6 プロセスファイルの作成 (UNIX)

Oracle とデータ連動する場合、インストール時に作成されたアーカイブライブラリを基に、Oracle を稼働させるユーザ環境下でプロセスファイルを作成する必要があります。プロセスファイルは、次のタイミングで必ず作成してください。

- Datareplicator Extension のインストール又はバージョンアップ後
- Datareplicator のバージョンアップ後
- データ連動の対象となる Oracle のバージョンアップ後又はパッチ対策後

上記のタイミングでプロセスファイルを作成しなかった場合は、動作を保証できません。

インストール時に作成されるアーカイブライブラリ及びアーカイブライブラリを基にして作成するプロセスファイルを表 2-7 と表 2-8 に示します。

表 2-7 インストール時に作成されるアーカイブライブラリ及びアーカイブライブラリを基にして作成するプロセスファイル (抽出側 Datareplicator Extension)

アーカイブライブラリ	プロセスファイル	プロセスファイル名
libhdemasterO.a	hdemasterO	抽出マスタプロセスファイル
libhdecaptureO.a	hdecaptureO	抽出プロセスファイル
libhdeprepO.a	hdeprepO	抽出定義プリプロセスコマンド
libhdeeventO.a	hdeeventO	イベント発行コマンド
libhderesstateO.a	hderesstateO	データ連動リソース状態表示コマンド

表 2-8 インストール時に作成されるアーカイブライブラリ及びアーカイブライブラリを基にして作成するプロセスファイル (反映側 Datareplicator Extension)

アーカイブライブラリ	プロセスファイル	プロセスファイル名
libhdsdefservO.a	hdsdefservO	反映定義サーバプロセスファイル
libhdssqleO.a	hdssqleO	反映 SQL プロセスファイル

プロセスファイルは、Oracle が提供するサンプルメイクファイルを使用して作成します。

プロセスファイルの作成手順及び作成例については、「[付録 B プロセスファイルの作成手順及び作成例](#)」を参照してください。

## 2.7 データ連動リソース管理ユーザの登録 (UNIX)

Oracle から HiRDB へデータ連動する場合、Oracle に対して Datareplicator が割り当てるリソース（データ連動リソース）を管理するためのユーザを事前に登録しておく必要があります。

データ連動リソースの作成については「3.1.7 データ連動リソースの作成」を参照してください。

### 2.7.1 データ連動リソース管理ユーザに対する権限付与

データ連動リソース管理ユーザに対しては、次に示す権限を与えておく必要があります。

権限種別	説明	権限を与えていない場合
DBA 権限	任意のユーザが所有する抽出対象表に対するトリガの定義を可能とするための権限です。 <b>権限付与例</b> CONNECT system/manager ; GRANT DBA TO 管理ユーザ名 ;	hdeprepO コマンド実行時のトリガ作成処理でエラーとなります。 権限を与えた後、hdeprepO を再実行してください。
キューテーブル管理者権限	キューテーブル管理者としてキューテーブルを作成するための権限です。 <b>権限付与例</b> CONNECT system/manager ; GRANT AQ_ADMINISTRATOR_ROLE TO 管理ユーザ名 ;	hdestartO コマンド (-i オプションを指定) 実行時のキューテーブル作成処理でエラーとなります。 権限を与えた後、hdestartO コマンド (-i オプションを指定) を再実行してください。
キューテーブルパッケージ実行権限	キューテーブル操作パッケージを実行するための権限です。 <b>権限付与例</b> CONNECT system/manager ; GRANT EXECUTE ON DBMS_AQADM TO 管理ユーザ名 ; GRANT EXECUTE ON DBMS_AQ TO 管理ユーザ名 ;	hdeprepO コマンド実行時のトリガ作成処理でエラーとなります。 権限を与えた後、hdeprepO を再実行してください。

### 2.7.2 キューモニタープロセスの起動

Datareplicator では、DEQUEUE 済みメッセージ（更新データ）を一定時間経過後に削除する指定を Oracle に対して発行しています。このため、キューメッセージに対して時間監視を行うキューモニタープロセスを起動しておく必要があります。キューモニタープロセスの起動方法を次に示します。

#### 方法 1：Oracle 初期化パラメタファイルへのパラメタ追加による起動

Oracle 初期化パラメタファイルに次のパラメタを追加します。

```
aq_tm_processes=1          (設定値は1以上を指定してください)
```

パラメタ追加後、Oracle を再起動してください。再起動後に SQL\*Plus で "show parameters aq\_tm\_processes" を実行して、パラメタ設定値が 1 以上になっていることを確認するか、Oracle のログから QMUn (x は 0~9 の数字) のプロセスが起動されたことを確認してください。

#### 方法 2 : SQL \* Plus によるパラメタの追加による起動

SQL \* Plus を起動し、"ALTER SYSTEM" コマンドを使用してパラメタを追加します。

```
ALTER SYSTEM SET aq_tm_processes=1 (設定値は1以上を指定してください)
```

パラメタ追加後、SQL\*Plus を再起動してください。再起動後に SQL\*Plus で "show parameters aq\_tm\_processes" を実行して、パラメタ設定値が 1 以上になっていることを確認するか、Oracle のログから QMUn (x は 0~9 の数字) のプロセスが起動されたことを確認してください。

キューモニタープロセスが起動されなかった場合、DEQUEUE 済みメッセージがキューテーブルから削除されません。そして、キューテーブルが満杯になってアプリケーションにエラーが返されます。パラメタの詳細については Oracle のマニュアルを参照してください。

### 2.7.3 データ連動リソース管理ユーザの名称制限

Oracle から HiRDB へデータ連動する場合は、抽出システム定義の oracleusr オペランドでデータ連動リソース管理ユーザ名及びパスワードを指定します。

データ連動リソース管理ユーザ名及びパスワードの長さは、1~8 文字にしてください。

データ連動リソース管理ユーザ名及びパスワードの長さが 1~8 文字でない場合は、抽出環境初期化時にオペランドエラーとなります。

### 2.7.4 その他

Oracle 上のすべてのデータ連動リソースをユーザごと削除できるようにするために、データ連動リソース管理ユーザは次のようにすることをお勧めします。

- 既存のユーザを使用するのではなく、新規に登録する。
- データ連動リソース以外のオブジェクトを割り当てない。

## 2.8 Datareplicator 管理者の登録 (UNIX)

---

抽出側 Datareplicator が \$HDEPATH 下に割り当てるリソースの管理や Datareplicator に対するコマンドを実行するための管理者を OS に対して登録します。

HiRDB システム同士のデータ連動の場合は、抽出側 Datareplicator の管理者は HiRDB の実行ユーザと同じユーザグループに属していなければなりません。他社 DBMS とデータ連動する場合は、特に制約はありません。

## 2.9 Datareplicator Extension のインストール (Windows)

---

Windows 版 Datareplicator Extension をインストールする方法について説明します。

### 2.9.1 インストール前の準備

Datareplicator Extension をインストールする前に、次に示す作業が必要になります。

- 稼働環境の確認
- ユーザ権限の確認

#### (1) 稼働環境の確認

次に示す内容を確認してください。

- Datareplicator Extension をインストールするマシンのディスクの空き容量が十分であることを確認してください。
- Datareplicator Extension が動作するのに必要な製品がインストールされていることを確認してください。

#### (2) ユーザ権限の確認

インストールを実行するユーザに Administrator の権限があることを確認してください。

### 2.9.2 インストールの実行

[2.1 Datareplicator Extension に関連する製品] を参照して、Windows 版 Datareplicator Extension を使うために必要な製品をインストールしてください。

Datareplicator と Datareplicator Extension のどちらを先にインストールしてもかまいませんが、必ず同じディレクトリにインストールしてください。

また、Datareplicator Extension のインストール完了後、Windows を再起動する必要があります。

#### (1) インストールの実行手順

インストール用の統合 CD-ROM からインストールする手順を次に示します。インストールを実行する前に、Datareplicator とすべての Windows アプリケーションを終了させてください。

1. HiRDB の統合 CD-ROM を挿入して、hcd\_inst.exe を実行します。

日立総合インストーラが起動します。

2. インストールプログラムが起動されます。画面の指示に従って [次へ>] ボタンをクリックします。

3. 名前と会社名を入力して、[次へ>] ボタンをクリックします。

表示されている名前と会社名をそのまま使う場合は、入力は不要です。[次へ>] ボタンをクリックします。

4. インストール先ディレクトリを指定し、[次へ>] ボタンをクリックします。標準のインストール先は、次に示すディレクトリです。

**Datareplicator がインストール済みの場合**

Datareplicator をインストールしたディレクトリ

**Datareplicator がインストールされていない場合**

Windows のインストール先ドライブ：¥Program Files¥HITACHI¥hirdbds

上記のディレクトリ以外にインストールする場合は、[参照...] ボタンをクリックして、インストール先のディレクトリ※<sup>1</sup> を指定します。指定したディレクトリが存在しない場合、作成するかどうかを確認するメッセージが表示されます。

5. コンポーネントを選択して、[次へ>] ボタンをクリックします。

- [Oracle 抽出機能] を選択すると、Oracle から HiRDB へデータ連動する場合の抽出側 Datareplicator Extension がインストールされます。※<sup>2</sup>
- [SQL Server 抽出機能] を選択すると、SQL Server から HiRDB へデータ連動する場合の抽出側 Datareplicator Extension がインストールされます。※<sup>2</sup>
- [ODBC 反映機能] を選択すると、HiRDB から ODBC ドライバを使用して Oracle 又は SQL Server へデータ連動する場合の反映側 Datareplicator Extension がインストールされます。※<sup>3</sup>

6. 登録するプログラムフォルダ名又はグループ名として [HiRDB Datareplicator] が表示されるので、[次へ>] ボタンをクリックします。

表示されている名前以外のグループを登録する場合は、名前を指定します。また、既存のグループに登録する場合は、既存のグループの中から選択します。

7. ファイルのコピーを開始する前に 現在の設定が表示されるので、[次へ>] ボタンをクリックします。

Datareplicator Extension のインストールが開始します。

8. Datareplicator Extension のインストールが完了したら、いったん Windows を再起動します。

#### 注※1

必ず Datareplicator と同じディレクトリへインストールしてください。Datareplicator と異なるディレクトリへインストールした場合、プロセスの起動時にエラーとなります。

#### 注※2

抽出側 Datareplicator Extension を使用する場合、Datareplicator のインストールで抽出側 Datareplicator がインストールされている必要があります。

### 注※3

反映側 Datareplicator Extension を使用する場合、Datareplicator のインストールで反映側 Datareplicator がインストールされている必要があります。

## 2.9.3 インストール時に登録される内容

インストール時に登録される内容を次に示します。

### (1) サービス

コンポーネントの選択で [Oracle 抽出機能] 又は [SQL Server 抽出機能] を選択した場合、インストール時にサービスが登録されます。

インストール時に登録されるサービスの種類、及び [コントロールパネル] の [サービス] アイコンを開いたときに表示される名称とスタートアップの設定を次の表に示します。

表 2-9 インストール時に登録されるサービス

種類	表示される名称	スタートアップの種類※	ログオン※
Oracle 抽出サービス	HiRDB Datareplicator (Source Site Oracle)	手動	システムアカウント
SQL Server 抽出サービス	HiRDB Datareplicator (Source Site SQL Server)	手動	システムアカウント

### 注※

インストール後にコントロールパネルの [サービス] アイコンを開いて変更できます。システムアカウントから Administrators 権限を持つユーザのアカウントに変更できます。

これらのサービスが登録されると、Datareplicator のインストール時に登録される「HiRDB Datareplicator (Source site)」は、スタートアップの種類が「無効」になります。

### (2) アイコン

[HiRDB Datareplicator] フォルダに、はじめにお読みいただくファイル (Datareplicator のインストール先ディレクトリ下の¥Extension Readme.txt) のアイコンが登録されます。

アンインストールには [コントロールパネル] の [アプリケーションの追加と削除] アイコンを使うため、アンインストールプログラムのアイコンは登録されません。

## 2.9.4 Datareplicator Extension のアンインストール

Windows 版 Datareplicator Extension をアンインストールする手順を次に示します。

アンインストールを実行する前に、Datareplicator を必ず停止させてください。

1. [コントロールパネル] にある [アプリケーションの追加と削除] アイコンを開きます。
2. [インストールと削除] タブを選択します。
3. インストールされているアプリケーションの一覧から [HiRDB Datareplicator Extension] を選択して、[追加と削除...] ボタンをクリックします。
4. 画面の指示に従って、[はい] ボタンをクリックします。
5. アンインストールが終了したら、[OK] ボタンをクリックします。

## 2.10 ディレクトリ構成 (Windows)

---

Datareplicator Extension は、Datareplicator のインストールディレクトリにインストールされます。Datareplicator のインストールディレクトリについては、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

## 2.11 環境変数の設定 (Windows)

製品をインストール後、Datareplicator Extension を実行する前に環境変数を設定する必要があります。ここでは、設定する環境変数を抽出側 Datareplicator Extension と反映側 Datareplicator Extension に分けて説明します。

### 2.11.1 抽出側 Datareplicator Extension の環境変数の設定 (Windows)

他社 DBMS とデータ連動する場合は、Datareplicator の環境変数及び他社 DBMS のアプリケーションを使用するのに必要な環境変数を、システム環境変数として設定してください。ユーザ環境変数として設定しても無効となります。

他社 DBMS とデータ連動する場合の環境変数の設定内容を次の表に示します。

表 2-10 他社 DBMS とデータ連動する場合の環境変数の設定内容 (抽出側 Datareplicator (Windows))

環境変数	設定内容
PATH	Datareplicator のコマンドライブラリの名称を指定します。*1
HDEPATH	抽出側 Datareplicator の運用ディレクトリを指定します。*2
Oracle 用環境変数*3	Oracle の OCI アプリケーション実行に必要な環境変数を設定します。
SQL Server 用環境変数*4	SQL Server のアプリケーション実行に必要な環境変数を設定します。

#### 注※1

コマンドライブラリの名称は、インストール時に設定されます。詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

#### 注※2

Datareplicator の運用ディレクトリの指定方法については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

#### 注※3

Oracle 用環境変数の設定内容については、Oracle のマニュアルを参照してください。

#### 注※4

SQL Server 用環境変数の設定内容については、SQL Server のマニュアルを参照してください。

## 2.11.2 反映側 Datareplicator Extension の環境変数の設定 (Windows)

ODBC ドライバを使用してデータ連動する場合は、Datareplicator の環境変数及び ODBC ドライバを使用するのに必要な環境変数を設定してください。また、Datareplicator の環境変数は、システム環境変数として設定してください。ユーザ環境変数として設定しても無効となります。

ODBC ドライバを使用してデータ連動する場合の環境変数の設定内容を次の表に示します。

表 2-11 ODBC ドライバを使用してデータ連動する場合の環境変数の設定内容 (反映側 Datareplicator (Windows))

環境変数	設定内容
PATH	Datareplicator のコマンドライブラリの名称を指定します。※1
HDSPATH	反映側 Datareplicator の運用ディレクトリを指定します。※2
SQL Server 用環境変数※3	SQL Server のアプリケーション実行に必要な環境変数を設定します。

### 注※1

コマンドライブラリの名称は、インストール時に設定されます。詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 注※2

Datareplicator の運用ディレクトリの指定方法については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 注※3

SQL Server 用環境変数の設定内容については、SQL Server のマニュアルを参照してください。

## 2.12 通信環境の設定 (Windows)

---

Datereplicator Extension を実行する前に、通信環境を設定してください。

Datereplicator Extension では、Windows Terminal Service を使用して Datereplicator Extension の運用・操作ができます。これによって、遠隔地のマシンや、コンソールがないマシンに対しても Datereplicator Extension の運用・操作が可能となります。

通信環境の設定の詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datereplicator」を参照してください。

## 2.13 ODBC 環境の設定 (Windows)

---

ODBC ドライバを使用してデータ連動する場合、Datareplicator Extension を実行する前に、ODBC 環境を設定する必要があります。

ここでは、ODBC ドライバの設定内容について説明します。

### 2.13.1 ODBC データソースの作成

ODBC データソースの作成手順を次に示します。

1. ODBC データソースアドミニストレータを実行します。

「コントロールパネル」から「管理ツール」 - 「データソース(ODBC)」の順に選択します。

2. 「システム DSN」タブを選択し、「追加...」ボタンをクリックします。

Datareplicator Extension で使うデータソース名称 (DSN) を、システム DSN として作成します。

3. 使用する ODBC ドライバ (Oracle ODBC driver 又は SQL Server) を選択し、「完了」ボタンをクリックします。

これ以降は、使用する ODBC ドライバごとの作成手順に従って作成します。

### 2.13.2 各 ODBC ドライバの注意点

#### (1) Oracle ODBC driver を使用する場合の注意点

特にありません。

#### (2) SQL Server を使用する場合の注意点

ODBC データソースを作成するときは、次の点に注意してください。

1. 「SQL Server が、ログイン ID の権限の確認を行う方法は？」で、次の各項目を選択した場合の処理を次に示します。

- 「ネットワークへのログイン ID で、Windows の認証メカニズムを使う」を選択した場合

抽出システム定義の sql\_usr オペランド、又は反映システム定義の odbcuser オペランドで指定した ODBC 接続認可識別子及びパスワードは無視されます。ただし、何も指定しないと Datareplicator の定義解析でエラーとなるため、ダミーの文字列を指定してください。

- 「ユーザーが入力する SQL Server 用のログイン ID とパスワードを使う」を選択した場合

抽出システム定義の sql\_usr オペランド、又は反映システム定義の odbcuser オペランドで指定した ODBC 接続認可識別子及びパスワードを使用して SQL Server にアクセスします。

2. 反映対象にできるのは、[規定のデータベースを以下のものに変更する] で選択したデータベースに作成した表だけです。

## 2.14 データ連動リソース管理ユーザの登録 (Windows)

他社 DBMS から HiRDB へデータ連動する場合、他社 DBMS に対して Datareplicator が割り当てるリソース（データ連動リソース）を管理するためのユーザを事前に登録しておく必要があります。

### 2.14.1 データ連動リソース管理ユーザに対する権限付与

データ連動リソース管理ユーザに対しては、次に示す権限を与えておく必要があります。

#### Oracle の場合

権限種別	説明	権限を与えていない場合
DBA 権限	任意のユーザが所有する抽出対象表に対するトリガの定義を可能とするための権限です。 <b>権限付与例</b> CONNECT system/manager ; GRANT DBA TO 管理ユーザ名 ;	hdeprepX コマンド実行時のトリガ作成処理でエラーとなります。 権限を与えた後、hdeprepO を再実行してください。
キューテーブル管理者権限	キューテーブル管理者としてキューテーブルを作成するための権限です。 <b>権限付与例</b> CONNECT system/manager ; GRANT AQ_ADMINISTRATOR_ROLE TO 管理ユーザ名 ;	hdestartX コマンド (-i オプションを指定) 実行時のキューテーブル作成処理でエラーとなります。 権限を与えた後、hdestartO コマンド (-i オプションを指定) を再実行してください。
キューテーブルパッケージ実行権限	キューテーブル操作パッケージを実行するための権限です。 <b>権限付与例</b> CONNECT system/manager ; GRANT EXECUTE ON DBMS_AQADM TO 管理ユーザ名 ; GRANT EXECUTE ON DBMS_AQ TO 管理ユーザ名 ;	hdeprepX コマンド実行時のトリガ作成処理でエラーとなります。 権限を与えた後、hdeprepO を再実行してください。

#### SQL Server の場合

ユーザ	ロール種別	ロール
ログインアカウント	固定サーバロール	制限なし
ユーザアカウント	固定データベースロール	db_ddladmin, db_datareader, db_datawrite

注 権限は、SQL Server 管理コンソールから与えてください。

## 2.14.2 キューモニタープロセスの起動 (Oracle の場合)

Datareplicator では、DEQUEUE 済みメッセージ (更新データ) を一定時間経過後に削除する指定を Oracle に対して発行しています。このため、キューメッセージに対して時間監視を行うキューモニタープロセスを起動しておく必要があります。キューモニタープロセスの起動方法を次に幾つか示します。

### 方法 1 : Oracle 初期化パラメタファイルへのパラメタ追加による起動

Oracle 初期化パラメタファイルに次のパラメタを追加します。

```
aq_tm_processes=1          (設定値は1以上を指定してください)
```

パラメタ追加後、Oracle を再起動してください。再起動後に SQL\*Plus で "show parameters aq\_tm\_processes" を実行して、パラメタ設定値が 1 以上になっていることを確認するか、Oracle のログから QMUx (x は 0~9 の数字) のプロセスが起動されたことを確認してください。

### 方法 2 : SQL \* Plus によるパラメタの追加による起動

SQL \* Plus を起動し、"ALTER SYSTEM" コマンドを使用してパラメタを追加します。

```
ALTER SYSTEM SET aq_tm_processes=1  (設定値は1以上を指定してください)
```

パラメタ追加後、Oracle を再起動してください。再起動後に SQL\*Plus で "show parameters aq\_tm\_processes" を実行して、パラメタ設定値が 1 以上になっていることを確認するか、Oracle のログから QMUx (x は 0~9 の数字) のプロセスが起動されたことを確認してください。

キューモニタープロセスが起動されなかった場合、DEQUEUE 済みメッセージがキューテーブルから削除されません。そして、キューテーブルが満杯になってアプリケーションにエラーが返されます。パラメタの詳細については Oracle のマニュアルを参照してください。

## 2.14.3 データ連動リソース管理ユーザの名称制限

他社 DBMS から HiRDB へデータ連動する場合は、抽出システム定義の oracleusr オペランド (Oracle の場合)、又は sqls\_usr オペランド (SQL Server の場合) でデータ連動リソース管理ユーザ名及びパスワードを指定します。

データ連動リソース管理ユーザ名及びパスワードの長さは、1~8 文字にしてください。

データ連動リソース管理ユーザ名及びパスワードの長さが 1~8 文字でない場合は、抽出環境初期化時にオペランドエラーとなります。

## 2.14.4 その他

他社 DBMS 上のすべてのデータ連動リソースをユーザごと削除できるようにするために、データ連動リソース管理ユーザは次のようにすることをお勧めします。

- 既存のユーザを使用するのではなく、新規に登録する。

- データ連動リソース以外のオブジェクトを割り当てない。

## 2.15 Datareplicator 管理者の登録 (Windows)

---

抽出側 Datareplicator が \$HDEPATH 下に割り当てるリソースの管理や Datareplicator に対するコマンドを実行するための管理者を OS に対して登録します。

Datareplicator 管理者の登録については、「[2.8 Datareplicator 管理者の登録 \(UNIX\)](#)」を参照してください。

# 3

## データ連動機能

この章では、Datareplicator Extension で実現できるデータ連動の、抽出処理及び反映処理の各機能について説明します。

## 3.1 抽出側 Datareplicator Extension の抽出処理

抽出側 Datareplicator Extension の抽出処理について説明します。

### 3.1.1 抽出処理の概要

他社 DBMS (Oracle 又は SQL Server) からの更新情報の抽出には、トリガ及びキューテーブルを使用します。

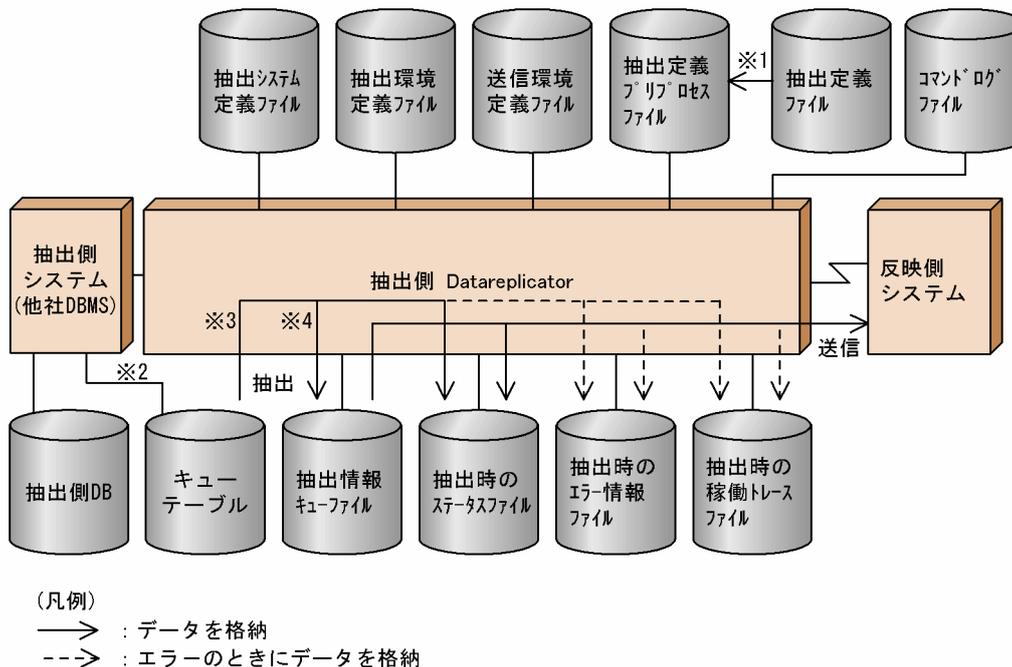
DBMS によって抽出に使用するキューテーブルは異なります。DBMS ごとに使用するキューテーブルを次の表に示します。

表 3-1 各 DBMS が使用するキューテーブル

DBMS	キューテーブル
Oracle	アドバンストキュー (Oracle 機能)
SQL Server	データベースキュー (Datareplicator Extension 機能)

抽出処理の概要を次の図に示します。

図 3-1 抽出処理の概要



注※1

hdeprepX コマンドを実行して、抽出定義プリプロセスファイルを作成

注※2

トリガを実行して、キューテーブルへ更新情報を格納

注※3

キューテーブルに格納されている更新情報の取り出し (DEQUEUE)

注※4

抽出情報キューファイルへ更新情報を格納 (ENQUEUE)

## (1) 抽出

抽出の手順を次に示します。

1. AP などによる更新処理の延長でトリガを実行して更新情報を抽出し、キューテーブルへ格納します (この操作を ENQUEUE といいます)。
2. キューテーブルに格納されている更新情報を取り出し、取り出した更新情報を抽出情報キューファイルへ格納します (この操作を DEQUEUE といいます)。

## (2) 送信

抽出情報キューファイルに格納されている更新情報を読み出し、指定された送信先の反映側システムに送信します。

### 3.1.2 抽出処理のタイミング

抽出処理の開始タイミングと停止タイミングについて説明します。

説明中のコマンド名の *X* は、使用する DBMS によって変わることを意味します。例えば、`hdeprepX` は、使用する DBMS が Oracle の場合は `hdeprepO`、SQL Server の場合は `hdeprepS` となります。

#### (1) 抽出処理の開始タイミング

##### (a) キューテーブルへの更新情報の格納

`hdeprepX` コマンドの実行が完了した時点から開始します。

##### (b) キューテーブルから抽出情報キューファイルへの更新情報の格納

`hdestartX` コマンド (オプション指定なし、又は `-e` オプションを指定) の実行が完了した時点から開始します。

#### (2) 抽出処理の停止タイミング

抽出側 Datareplicator Extension は、`hdestopX` コマンド (オプション指定なし、又は `-e` オプションを指定) を実行して終了します。

他社 DBMS から HiRDB へデータ連動する場合は、HiRDB システム同士のデータ連動時のような抽出対象 DBMS の動作と連動した抽出処理の停止（抽出システム定義の syncterm オペランド及び termlevel オペランドによる同期停止や、Datareplicator の pdrplstop コマンドによる強制停止など）はできません。

### 3.1.3 更新情報の取り出し

キューテーブルに入っている更新情報の取り出しには、DEQUEUE を使用します。

#### (1) 更新情報の有効性

DEQUEUE は、更新元トランザクションが有効（COMMIT 完了済み）となっている更新情報だけを取り出します。

このため、抽出側 Datareplicator Extension が抽出する更新情報はすべて有効トランザクションについてのものだけとなります。

#### (2) 更新情報の取り出し順序

キューテーブルに入っている更新情報は、DEQUEUE を実行した時点で有効となっているすべてのトランザクションを対象として、更新処理の実行順に取り出されます（DEQUEUE のうちの格納順取り出し方法を使用）。

この方法は、DEQUEUE を実行するタイミングによって抽出対象となるトランザクションが異なるため、取り出される更新情報の順序が異なる可能性があります。抽出する更新情報は有効トランザクションについてのものだけなので、同一行に対しての更新順序は保証されます。

### 3.1.4 データ型の変換

抽出側 DB から抽出された更新データのデータ型は、抽出側 Datareplicator Extension によって反映側 DB に反映できるデータ型に変換され、抽出情報キューファイルに格納されます。

データ型の対応については、「[4.1.2 データ型の対応の設計](#)」を参照してください。

なお、抽出側 DB が他社 DBMS の場合、変換ができるデータ型であっても抽出情報キューファイルに格納できないデータがあります。これらのデータを**例外データ**といいます。

例外データを検出した場合は、これらを抽出情報キューファイルに格納できるデータに置換することで、抽出処理を続行できます。

Oracle の場合の例外データを次に示します。

データ型	抽出情報キューファイルに格納できないデータ	理由
NUMBER 型	「表 4-3 SQL Server から抽出するデータのデータ型対応」の注※1 に従ってデータ型を変換した列に対する、各数値型の表現範囲外のデータ。	抽出情報キューファイルに格納するためには、各数値型に相当する形式に変換する必要がありますが、表現範囲外であるため変換できません。
DATE 型	紀元前の日付データ。	抽出情報キューファイルに格納するためには、日付データは 4 バイトのパック形式に変換する必要がありますが、紀元前の日付データは変換できません。

次に SQL Server の場合の例外データを次に示します。

データ型	抽出情報キューファイルに格納できないデータ	理由
CHAR (n) 型	時間又は日付用の情報を格納している CHAR 型データ (n≠8, 10)。	時間及び日付のデータ形式が、'HH:MM:SS'及び'YYYY-MM-DD'のため、n = 8, 10 以外では変換できません。

## (1) 例外データを検出したときの抽出側 Datareplicator Extension の動作

例外データを検出したときの抽出側 Datareplicator Extension の動作は、次に示すどちらかを選択できます。

例外データを検出したときの抽出側 Datareplicator Extension の動作	出力メッセージ
抽出処理を停止します。	KFRB05028-E
例外データを置換して、抽出処理を続行します。	KFRB05029-W

例外データを検出したときの抽出側 Datareplicator Extension の動作は、抽出環境定義の `exterrlevel` オペランドで指定します。

また、例外データを置換して抽出処理を続行する場合、どの値に置換するかは、抽出環境定義の `cvt_datakind` オペランドで指定します。

各オペランドの詳細については、「[5.1.2 抽出環境定義](#)」を参照してください。

## (2) 例外データを置換する場合の置換値

Oracle の場合の例外データを置換する場合の置換値を次に示します。

データ型	置換値		
	ナル値	最大値	最小値
NUMBER 型	ナル値	各数値型の表現範囲の最大値	0
DATE 型	ナル値	9999/12/31 23:59:59	0001/01/01 00:00:00

SQL Server の場合の例外データを置換する場合の置換値を次に示します。

データ型	置換値		
	ナル値	最大値	最小値
CHAR (n) 型	ナル値	9999/12/31	0001/01/01
		23:59:59	00:00:00

### 3.1.5 更新情報及びトランザクション情報の出力

抽出情報キューファイルに対しては、更新情報だけではなく、送信プロセスで実行するトランザクション管理処理に使用する更新トランザクションの開始/終了情報を出力する必要があります。

しかし、データ連動用トリガで取得できる更新情報は更新差分情報だけであり、トランザクションの開始/終了についての情報は取得できないため、他社 DBMS からの更新情報の抽出時は、抽出側 Datareplicator Extension が更新トランザクションの開始/終了情報を付け加えます。

#### (1) トランザクション情報を付け加えるタイミング

抽出側 Datareplicator Extension は、次に示すタイミングで更新トランザクションの開始/終了情報を付け加えます。

抽出側 Datareplicator Extension が付け加えるトランザクション情報	トランザクション情報を付け加えるタイミング
更新トランザクションの開始情報	次のどちらかのタイミング。 <ul style="list-style-type: none"> <li>抽出処理を開始した直後の、最初の更新情報を取り出したとき</li> <li>更新トランザクションの終了情報を付け加えた後、次の更新情報を取り出したとき</li> </ul>
更新トランザクションの終了情報	次のどれかのタイミング。 <ul style="list-style-type: none"> <li>抽出処理の同期制御が実行されていない更新情報の数が、トランザクションの最大更新情報数に達したとき</li> <li>抽出処理の同期制御が実行されていない更新情報がある時点で、EOQ が検知されたとき</li> <li>抽出処理の同期制御が実行されていない更新情報がある時点で、抽出処理の停止要求を受け付けたとき</li> </ul>

抽出側 Datareplicator Extension は、実際に他社 DBMS に対して実行した更新トランザクションとは異なるタイミングで、更新トランザクションの開始/終了情報を付け加えます。したがって、他社 DBMS から HiRDB へのデータ連動は、データ連動が完了した時点でだけ抽出側と反映側の整合性が保証されていれば問題がないような業務に対して適用してください。

なお、データ連動中に反映側 DB を参照する場合の留意点を次に示します。

- 抽出側で実行したトランザクション単位とは異なる単位で反映処理が有効となります。したがって、実際のトランザクション以上に排他範囲が大きくなる可能性があるため、NOWAIT 検索をお勧めします。

- 更新順序は、同一行に対しては保証されますが、表間又は行間に対しては保証されません。

## (2) トランザクションの最大更新情報数の調整

1 トランザクション当たりの最大更新情報数を抽出環境定義の tran\_datanum オペランドで指定できます。

オペランドの詳細については、「5.1.2 抽出環境定義」を参照してください。

### 3.1.6 抽出間隔の制御

DEQUEUE を実行してもキューテーブルに取得ができるデータがなかった場合、次の DEQUEUE を実行するまでの間隔を抽出環境定義の ext\_wait\_interval オペランドで指定できます。

オペランドの詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」の抽出環境定義を参照してください。

### 3.1.7 データ連動リソースの作成

他社 DBMS から HiRDB へデータ連動する場合、データ連動リソース管理ユーザに対してデータ連動に必要なリソースを作成する必要があります。

データ連動リソース管理ユーザについては、「2.7 データ連動リソース管理ユーザの登録 (UNIX)」又は「2.14 データ連動リソース管理ユーザの登録 (Windows)」を参照してください。

## (1) データ連動リソースの一覧

Oracle の場合のデータ連動リソースの一覧を次の表に示します。

表 3-2 Oracle の場合のデータ連動リソースの一覧

リソース種別	リソース名	使用目的	割り当て数	初期作成/再作成のタイミング
オブジェクト型	RPL_EXT_MSGTYPE	キューテーブルに格納するメッセージのデータ型	1	抽出環境初期化時 (hdestartO コマンド (-i オプションを指定) の延長)
キューテーブル	RPL_EXT_QUEUE	各抽出表トリガからの更新情報格納用キュー		
キューテーブル表 (アドバンストキューテーブル)	RPL_EXT_QUEUE_TBL	各抽出表トリガからの更新情報格納用キュー表		
抽出同期管理テーブル	RPL_EXT_SYNC_TBL	抽出同期処理異常終了時の回復情報格納		

リソース種別	リソース名	使用目的	割り当て数	初期作成／再作成のタイミング
抽出順序管理テーブル	RPL_EXT_SORTTBL	同一行に対する更新順序保証のためのソート情報格納		
抽出順序管理テーブル用インデクス	RPL_EXT_SORTIDX	DEQUEUE 性能向上のための抽出順序管理テーブルに対するインデクス		
シーケンス番号	RPL_EXT_SEQ	更新順序保証のためのシーケンス番号		
データ連動用トリガ	RPL_EXT_TRGxxxx (xxxx : 1~4096)	抽出表からの更新情報取得及びキュー格納	抽出表と同数	抽出定義プリプロセスファイルの作成時 (hdeprepO コマンドの延長)

注

削除のタイミングは、どのリソースも、DROP USER...CASCADE を実行してデータ連動リソース管理ユーザを削除したときです。

SQL Server の場合のデータ連動リソースの一覧を次の表に示します。

表 3-3 SQL Server の場合のデータ連動リソースの一覧

リソース種別	リソース名	使用目的	割り当て数	初期作成／再作成のタイミング
キューテーブル	RPL_EXT_QUEUE_TBL	各抽出表トリガからの更新情報格納用キュー	1	抽出環境初期化時(hdestartS コマンド (-i オプションを指定) の延長)
抽出同期管理テーブル	RPL_EXT_SYNCTBL	抽出同期処理異常終了時の回復情報格納	1	
リソースオペランドテーブル	RPL_EXT_RESOPDTBL	抽出システム定義の sqls_msgkeepime の値保証のため	1	
データ連動用トリガ	RPL_EXT_TRGnnnn-XXX*	抽出表からの更新情報取得及びキューテーブル格納	抽出表の数×3	抽出定義プリプロセスファイルの作成時 (hdeprepS コマンドの延長)

注※

nnnn 及び XXX には、次に示す値が入ります。

変数	値	備考
nnnn	1~4096	—
XXX	INS	INSERT 用トリガを意味します。
	UPD	UPDATE 用トリガを意味します。
	DEL	DELETE 用トリガを意味します。

## (2) 更新情報の保存

抽出データ不正などの障害が発生した時に、他社 DBMS と抽出側 Datareplicator Extension のどちらに問題があるかを判断できるように、キューテーブルから更新情報を取り出した後もキューテーブル内に更新情報を保存できます。保存期間は、抽出システム定義の `oracle_msgkeep` (Oracle の場合) オペランド、又は `sqls_msgkeep` (SQL Server の場合) で指定できます。

`oracle_msgkeep` オペランドの詳細については「[5.1.1 抽出システム定義](#)」を、`sqls_msgkeep` オペランドの詳細については「[5.2.1 抽出システム定義](#)」を参照してください。

## (3) データ連動リソースについての注意

次の場合は、データ連動の動作及び抽出側と反映側の整合性を基本的に保証できません。

- 「表 3-2 Oracle の場合のデータ連動リソースの一覧」に示したデータ連動リソースを、Datareplicator Extension が提供するコマンドを使用しないでユーザ自身が作成した場合
- コマンドを使用して作成したリソースに対して、ユーザが独自に定義内容を変更したりリソースを削除したりした場合

ただし、トリガについては、抽出対象表の運用の一環として、ユーザ責任でトリガを削除できます。抽出対象表の運用については、Oracle の場合は「[6.2.1 抽出対象表の運用](#)」を、SQL Server の場合は「[6.4.1 抽出対象表の運用](#)」参照してください。

## (4) トリガの作成

トランザクション処理中にトリガを再作成することでトランザクションの一貫性が不正となるのを防ぐため、データ連動用トリガの作成は、抽出対象表に対する共用排他を取得し、排他取得が成功した場合にだけ実行します。排他取得に失敗した場合は、抽出定義プリプロセスファイルを作成するコマンド (`hdeprepX`) をエラー停止します。

また、トリガを作成していない状態で抽出処理が起動するのを防ぐため、トリガ作成中にエラーが発生した場合は、エラーが発生するまでに作成したすべてのデータ連動用トリガを削除するとともに、抽出定義プリプロセスファイルを使用できないようにします。

## (5) エラー処理

データ連動リソースの作成時に起こるエラー内容及びエラーに対する対処方法を次に示します。

エラー内容	対処方法
オブジェクト型作成エラー	エラーの原因を取り除いた後、再度、抽出環境を初期化してください。
キューテーブル作成エラー	
抽出同期管理テーブル作成エラー	

エラー内容	対処方法
トリガ作成エラー※	エラーの原因を取り除いた後、再度、抽出定義プリプロセスファイルを作成してください。
その他の他社 DBMS 操作エラー	エラーの原因を取り除いた後、再度、抽出環境を初期化するか抽出定義プリプロセスファイルを作成してください。

注

上記のエラーが発生した場合、KFRB00060-E メッセージが出力され、抽出処理はエラー終了します。

注※

エラー処理の過程で、エラーが発生するまでに作成したすべてのデータ連動用トリガを削除するとともに、抽出定義プリプロセスファイルを使用できないようにします。

## 3.2 反映側 Datareplicator Extension の反映処理

反映側 Datareplicator Extension の反映処理について説明します。

### 3.2.1 反映処理の概要

反映処理は、HiRDB システム同士のデータ連動時と差異はありません。

反映処理の概要については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 3.2.2 反映処理の方式

反映側 Datareplicator では、反映処理の方式を次の二つから選べます。どちらの方式にするかは、反映環境定義で指定します。

- トランザクション単位反映方式
- 表単位反映方式

ただし、マルチ FES 機能に対応した反映方式は指定できません。

### 3.2.3 反映処理のタイミング

反映処理の開始タイミングと停止タイミングについて説明します。

#### (1) 反映処理の開始タイミング

反映処理の開始タイミングは、反映側 Datareplicator Extension の起動時には反映処理を開始しないで、何らかの方式で開始が要求された時点で反映処理を開始する**遅延開始**です。

反映処理の開始を要求する方式を次に示します。

#### HiRDB から Oracle へデータ連動する場合

- hdsstart コマンドを実行して、反映側 Datareplicator を起動します。  
ただし、システム起動時の startmode パラメタの指定が spd の場合は反映処理を開始しません。
- hdsrfctl コマンドを実行して、反映処理を再開始します。

#### HiRDB から ODBC ドライバを使用してデータ連動する場合

- hdsstart コマンドを実行します。又は、Datareplicator (Target Site) サービスを開始します。  
ただし、システム起動時の startmode パラメタの指定が spd の場合は反映処理を起動しません。

- hdsrfctl コマンドを実行して、反映処理を再開します。

## (2) 反映処理の停止タイミング

反映側 Datareplicator Extension は、次のどちらかで終了します。

HiRDB から Oracle へデータ連動する場合

- hdsstop コマンドを実行して、反映側 Datareplicator を終了します。
- hdsrfctl コマンドを実行して、反映側 Datareplicator をイベント待ち状態にします。

HiRDB から ODBC ドライバを使用してデータ連動する場合

- Datareplicator (Target Site) サービスを終了します。
- hdsrfctl コマンドを実行して、反映側 Datareplicator をイベント待ち状態にします。

### 3.2.4 時系列情報の取得

反映定義に時系列情報を取得する指定をすると、反映処理の実行時に反映側システムに対する更新情報を時系列情報表に蓄積できます。

時系列情報の取得については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

ただし、時系列情報表の作成については、「[4.1.4 時系列情報表の作成](#)」を参照してください。

### 3.3 Oracle, SQLServer から抽出を行う場合の滞留監視機能

滞留監視機能とは、更新データの滞留時間を監視する機能です。滞留時間とは、抽出プロセスがキューテーブルから更新情報を抽出した時刻と、任意のデータ連動処理が完了した時刻との差のことです。滞留監視機能の概念を次の図に、滞留監視する項目を次の表に示します。

図 3-2 滞留監視機能の概念

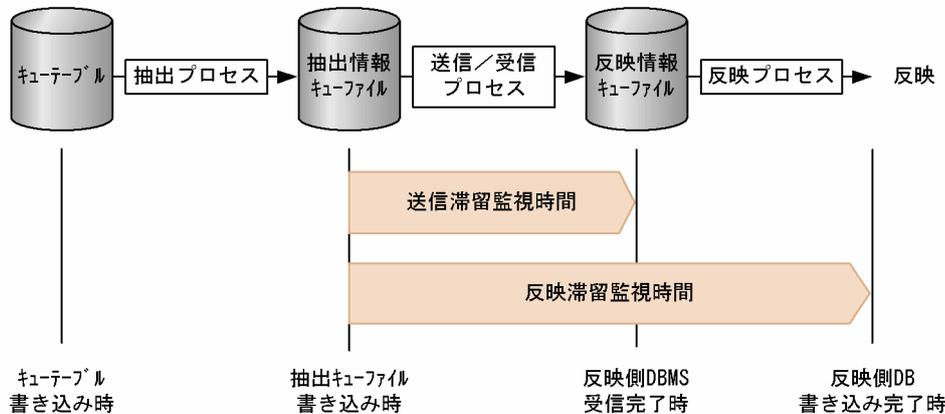


表 3-4 滞留監視する項目

監視項目	説明	監視プロセス
送信滞留時間	抽出プロセスがキューテーブルから更新情報を抽出した時刻と、その更新情報が送信されて反映側 DBMS で受信が完了した時刻との差のことです。 送信プロセスの送信間隔ごとに監視します。複数の更新情報が一度に送信される場合は、それらのうち、最初に送信された更新情報の受信が完了した時刻との差が、送信滞留時間になります。	送信プロセス
反映滞留時間	抽出プロセスがキューテーブルから更新情報を抽出した時刻と、反映側 DB でその更新情報が反映された時刻との差のことです。 反映側 DB でのコミット間隔ごとに監視します。	反映プロセス

#### Datareplicator の滞留監視機能と異なる点

Datareplicator の滞留監視機能では上記以外に「抽出滞留時間」を監視できますが、Datareplicator Extension ではサポートしていません。抽出滞留時間を監視するオペランド (extract\_delay\_limit\_time) を指定しても無効になります。また、Datareplicator では送信滞留時間及び反映滞留時間は、「更新情報がシステムログファイルに格納された時刻」を起点とした滞留時間となるので、注意してください。

Oracle, SQLServer へ反映する場合の滞留監視機能については、Datareplicator の滞留監視機能を参照してください。

## 3.4 イベント機能

---

イベント機能とは、抽出側システムで発行したイベントに従って、反映処理の動作を切り替える機能です。イベント機能を使用すると、反映処理の停止、反映処理の再開始、及び稼働中の反映処理の反映方式の切り替えができます。

イベント機能の詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

抽出側 DB が HiRDB の場合はイベント制御用テーブルに FIX 属性を指定しますが、FIX 属性の指定は HiRDB 独自の指定のため、抽出側 DB が他社 DBMS のときは FIX 属性を指定する必要はありません。

イベント制御用テーブルを作成する SQL 文の例を次に示します。

```
CREATE TABLE "hde_dtbl"("KEY" INTEGER,"EVNO" INTEGER,"SNDID" CHAR(10));  
COMMIT;  
INSERT INTO "hde_dtbl" VALUES (1, 0, '0');  
COMMIT;
```

# 4

## システム設計

この章では、Datareplicator Extension を使用したデータ連動システムを設計する方法について説明します。

## 4.1 抽出側 DB と反映側 DB との対応の設計

---

抽出側 DB と反映側 DB を対応させたシステム設計について説明します。

### 4.1.1 抽出側 DB の条件

他社 DBMS のデータベースからデータを抽出する場合の条件について説明します。

#### (1) 抽出対象表

抽出対象表の種別は実表だけです。

抽出対象表の運用については、「[6.2.1 抽出対象表の運用](#)」を参照してください。

#### (2) 抽出対象の SQL 文

他社 DBMS からの更新情報は、抽出対象表に対して定義したトリガによって抽出されます。抽出の対象となる SQL 文の操作は、トリガを定義するときに指定したトリガイベントに従います。

抽出対象表に対して Datareplicator が定義するトリガイベントを次に示します。

- INSERT
- UPDATE
- DELETE

したがって、上記の SQL 文の操作が抽出の対象となります。

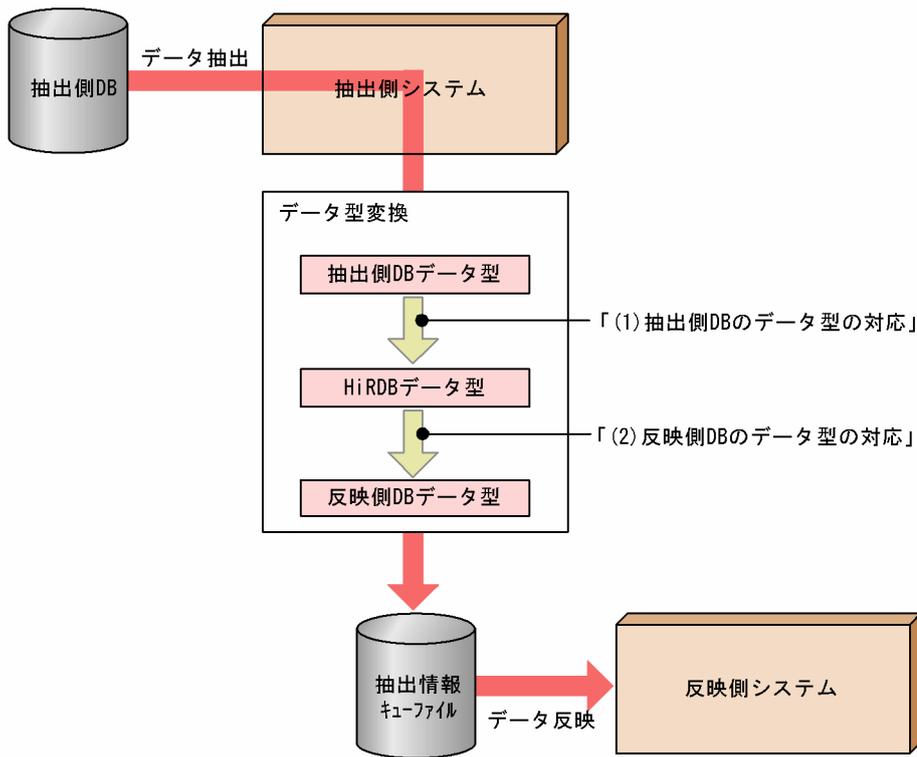
ただし、AP からの抽出対象表に対する更新だけでなく、ほかの表の更新を契機とした自動的な更新やユーティリティの実行などで、内部的にトリガが実行される場合があります。内部的にトリガが実行される契機については、他社 DBMS の仕様に従います。

### 4.1.2 データ型の対応の設計

抽出処理で抽出側 DB から抽出された更新データのデータ型は、抽出側 Datareplicator Extension によって反映側 DB に反映できるデータ型に変換され、抽出情報キューファイルに格納されます。

このとき、抽出側 DB から抽出されたデータは、まず HiRDB データ型に変換されます。そして、HiRDB データ型から反映側 DB に合ったデータ型に変換されて、抽出情報キューファイルに格納されるという流れになります。

図 4-1 データ型の変換



Datereplicator Extension が対応している抽出側 DBMS と反映側 DBMS の組み合わせを次の表に示します。

表 4-1 Datereplicator Extension が対応している抽出側 DBMS と反映側 DBMS の組み合わせ

抽出側 DB	反映側 DB		
	HiRDB (UNIX, Windows)	Oracle (UNIX, Windows ※2)	SQL Server (Windows)
HiRDB (UNIX, Windows)	○※1	○	○
Oracle (UNIX, Windows)	○	×	×
SQL Server (Windows)	○	×	×

(凡例)

- ：データ連動可能
- ×：データ連動不可

注※1

Datereplicator Extension は不要です。

注※2

データ連動に ODBC を使用します。

以降では、各 DBMS のデータ型と HiRDB データ型との対応について説明します。抽出側 DB で使用するデータ型が反映側でどのデータ型に変換されるかは、「(1)抽出側 DB のデータ型の対応」及び「(2)反映側 DB のデータ型の対応」を比較して調べてください。

## (1) 抽出側 DB のデータ型の対応

ここでは、抽出側 DB のデータ型と HiRDB データ型との対応について説明します。

### (a) Oracle データ型から HiRDB データ型への変換

Oracle からデータを抽出した場合、HiRDB データ型には次の表のように変換されます。

抽出対象外のデータを抽出対象として抽出定義に指定した場合は、抽出定義プリプロセスファイルの作成処理でエラーとなります。

表 4-2 Oracle から抽出するデータのデータ型対応

Oracle データ型		対応する HiRDB データ型
データ属性	データの長さ	
VARCHAR2 (n)	$1 \leq n \leq 4000$	MVARCHAR (n)
NVARCHAR2 (n)	$1 \leq n \leq 4000$	NVARCHAR (n)
NUMBER (p, s)	$1 \leq p \leq 38$ $-84 \leq s \leq 127$	SMALLINT, INTEGER, DECIMAL, 又は FLOAT <sup>*1</sup>
LONG	—	抽出対象外
ROWID	—	抽出対象外
DATE	固定長の日付+時間 データ。	DATE, TIME, 又は CHAR (19) <sup>*2</sup>
RAW (n)	$1 \leq n \leq 2000$	BLOB (n)
LONG RAW	—	抽出対象外
CHAR (n)	$1 \leq n \leq 2000$	MCHAR (n), DATE, 又は TIME <sup>*2</sup>
NCHAR (n)	$1 \leq n \leq 2000$	NCHAR (n)
CLOB	—	抽出対象外
NCLOB	—	抽出対象外
BLOB	—	抽出対象外
BFILE	—	抽出対象外
UROWID	—	抽出対象外
ユーザ定義型	—	抽出対象外

(凡例)

－：該当しません。

注 1

抽出対象のデータの n は、データ長 (バイト数) を表します。ただし、NCHAR 及び NVARCHAR については文字数を表します。

また、p は精度、s は位取りを表します。

注 2

反映対象表のデータ型の n は、データ長 (バイト数) を表します。

ただし、NCHAR 及び NVARCHAR については文字数を表します。

注※1

精度及び位取りを基に、次の表に従ってデータ型を決定します。

指定分類	Oracle のデータ	decimal_max_p recision オペラ ンド指定値	条件	HiRDB データ型
精度及び位取りの指 定なし	NUMBER REAL DOUBLE PRECISION	－	なし	FLOAT
	INTEGER SMALLINT DECIMAL NUMERIC	－	なし	INTEGER
精度だけ指定	NUMBER (p) DECIMAL (p) NUMERIC (p)	$m \geq p$	$p \leq 4$	SMALLINT
			$5 \leq p \leq 9$	INTEGER
			$10 \leq p \leq m$	[LARGE] DECIMAL (p)
	$m < p$	－	FLOAT	
FLOAT (p)	－	－	FLOAT	
精度及び位取りを 指定	NUMBER (p, s) DECIMAL (p, s) NUMERIC (p, s)	$m \geq p$	$p \geq m+1,$ $s \geq m+1,$ 又は $p-s \geq m+1$	FLOAT
			$p < s$ かつ $p < m+1$ かつ $s < m+1$ かつ $p-s < m+1$	[LARGE] DECIMAL (s, s)
			$s < 0$ かつ $p < m+1$ かつ $p-s < m+1$	[LARGE] DECIMAL (p-s, 0)

指定分類	Oracle のデータ	decimal_max_p recision オペラ ンド指定値	条件	HiRDB データ型
			上記以外	[LARGE] DECIMAL (p, s)
		m < p	—	FLOAT

(凡例)

—：依存しません。

m：抽出システム定義 decimal\_max\_precision オペランドの指定値です。省略した場合は、29 が仮定されます。m の範囲は、29 から 38 です。

## 注※2

どのデータ型で抽出するかは、抽出定義でユーザが指定します。

抽出定義の詳細については、「5.1.4 抽出定義」を参照してください。

## (b) SQL Server データ型から HiRDB データ型への変換

SQL Server からデータを抽出した場合、HiRDB データ型には次の表のように変換されます。

抽出対象外のデータを抽出対象として抽出定義に指定した場合は、抽出定義プリプロセスファイルの作成処理でエラーとなります。

表 4-3 SQL Server から抽出するデータのデータ型対応

SQL Server データ型		対応する HiRDB データ型
データ属性	データの長さ	
BIGINT	—	LARGE DECIMAL (19, 0) p = 19, s = 0
INT	—	INTEGER
SMALLINT	—	SMALLINT
TINYINT	—	SMALLINT
BIT	—	CHAR (1)
DECIMAL (p, s)	0 < p < m 0 ≤ s ≤ p	<ul style="list-style-type: none"> <li>• p ≤ m の場合、LARGE DECIMAL (p, s) p と s は抽出対象表定義に指定したものと同じです。</li> <li>• p &gt; m の場合、FLOAT</li> </ul>
NUMERIC (p, s)		
MONEY	—	LARGE DECIMAL (19, 4)
SMALLMONEY	—	DECIMAL (10, 4)
FLOAT	—	FLOAT
REAL	—	SMALLFLT
DATETIME	—	DATE, TIME, 又は CHAR (19) *1
SMALLDATETIME		

SQL Server データ型		対応する HiRDB データ型
データ属性	データの長さ	
CHAR (n)	—	CHAR (n) ※1
VARCHAR (n)	—	VARCHAR (n)
TEXT	—	抽出対象外
NCHAR (n)	—※2	NCHAR (n)
NVARCHAR (n)	—※2	NVARCHAR (n)
NTEXT	—	抽出対象外
BINARY	—	BLOB, 又は VARCHAR※1※3
VARBINARY		
IMAGE	—	抽出対象外
SQL_VARIANT	—	VARCHAR (8000)
TIMESTAMP	—	抽出対象外
UNIQUEIDENTIFIER		
ユーザ定義データ型 (ADT)	—	ベースデータ型マッピングに従います。

(凡例)

m : 抽出システム定義 decimal\_max\_precision オペランドの指定値です。省略した場合は、29 が仮定されます。m の範囲は、29 から 38 です。

#### 注※1

どのデータ型で抽出するかは、抽出定義でユーザが指定します。

抽出定義の詳細については、「5.2.4 抽出定義」を参照してください。

#### 注※2

偶数長のシングルバイトデータ、又はマルチバイトデータである必要があります。

#### 注※3

- バイナリデータの最後の情報が「81」から「9F」までの場合、抽出情報キューファイルへの出力時点でこれらが NULL 文字に変換されます。
- BINARY 又は VARBINARY 以外のデータ型の列にバイナリデータを反映する場合、抽出情報キューファイルへの出力時点で「00」以降のデータは破棄されます。例えば、「0x006162630000」というデータを CHAR 型の列に反映した場合、「0x000000…00」というデータになります。

## (2) 反映側 DB のデータ型の対応

ここでは、反映側 DB のデータ型と HiRDB データ型との対応について説明します。

## (a) HiRDB データ型から Oracle (UNIX) データ型への変換

UNIX 版 Oracle のデータ型には、DB 内部のデータ形式を表す内部データ型と、AP 内のデータ形式を表す外部データ型があります。外部データ型は、内部データ型より種類が多く、Oracle は AP 内の外部データ型のデータを内部データ型の列へ変換する多種のデータ変換パターンを持っています。

Datareplicator Extension は、HiRDB から抽出したデータを Oracle の外部データ型へデータ変換して Oracle のデータベースへ反映します。このため、反映対象表の作成時には、外部データ型から変換できる内部データ型を使用してください。

HiRDB から Oracle へデータ連動する場合の、抽出対象表と反映対象表のデータ型の対応を次の表に示します。

### 注意事項

Windows 版の Oracle への反映には、ODBC を使用します。「(b)HiRDB データ型から ODBC のデータ型への変換 (SQL Server, Oracle)」を参照してください。

表 4-4 Oracle に反映するデータのデータ型対応

HiRDB データ型		対応する Oracle データ型
データ属性	データの長さ	
INTEGER	データ長については、マニュアル「HiRDB SQL リファレンス」を参照してください。	INTEGER (4)
SMALLINT		INTEGER (2)
DECIMAL <sup>※1</sup>		CHAR
FLOAT		FLOAT (8)
SMALLFLT		
CHAR <sup>※7</sup>	$1 \leq n \leq 255$	CHAR (n)
MCHAR		
NCHAR	$1 \leq n \leq 127$	CHAR (2n)
CHAR <sup>※2※7</sup>	$n \geq 256$	LONG (n)
MCHAR <sup>※2</sup>		
NCHAR <sup>※2</sup>	$n \geq 128$	LONG (2n)
VARCHAR <sup>※3※7</sup>	$1 \leq n \leq 4000$	VARCHAR2 (n)
MVARCHAR <sup>※3</sup>		
NVARCHAR <sup>※3</sup>	$1 \leq n \leq 2000$	VARCHAR2 (2n)
DATE <sup>※4</sup>	データ長については、マニュアル「HiRDB SQL リファレンス」を参照してください。	CHAR (10)

HiRDB データ型		対応する Oracle データ型
データ属性	データの長さ	
TIME <sup>※4</sup>		CHAR (8)
INTERVAL YEAR TO DATE <sup>※5</sup>		RAW (5)
INTERVAL HOUR TO SECOND <sup>※5</sup>		RAW (4)
BLOB <sup>※6</sup>	$1 \leq n \leq 2147483647$	LONG RAW (n)

#### 注 1

データの長さの n は、データ長 (バイト数) を表します。  
ただし、NCHAR 及び NVARCHAR については文字数を表します。

#### 注 2

反映対象表のデータ型は、Oracle の外部データ型を表します。  
また、( ) 内はデータ長 (バイト数) を表します。

#### 注※1

Oracle の C 言語の OCI (Oracle Call Interface) では、DECIMAL 型のデータを扱えないので、数値文字列として文字型で反映します。なお、この場合の DECIMAL 型の精度の最大値は 38 です。  
数値文字列へ変換する場合は、符号を正規化します。符号の正規化手順を次に示します。

#### 符号の正規化手順

```
X' A' → X' C'
X' B' → X' D'
X' C' → X' C'
X' D' → X' D'
X' E' → X' C'
X' F' → X' F'
```

#### 注※2

LONG 型は探索条件に指定できません。

#### 注※3

HiRDB には 0 バイトのデータを格納できますが、Oracle には 0 バイトのデータを格納できません。  
そのため、Datareplicator では反映先 Oracle に対して 0x00 を格納します。  
したがって、0 バイトのデータを検索するためには、「0x00」を検索条件とする必要があります。この  
ような検索差異を生じさせないために、Oracle ヘデータ連動する場合は、抽出側で 0 バイトのデータ  
を扱わない運用をお勧めします。

#### 注※4

TIME 型は Oracle に存在しないデータ型です。また、DATE 型は、Oracle に存在するデータ型です  
が、HiRDB とはデータ形式が異なり、日付及び時間の情報を持ちます。

Datereplicator Extension では、TIME 型の抽出データを HH:MM:SS 形式の文字列データとして、Oracle の CHAR 型で反映します。また、DATE 型の抽出データを YYYY-MM-DD 形式の文字列データとして、Oracle の CHAR 型で反映します。

ユーザは、TIME 又は DATE 型のどちらかのフォーマットを Oracle の初期化パラメタ NLS\_DATE\_FORMAT に指定することで、抽出データを DATE 型の列へ反映できます。その場合、内部的には列値にフォーマットされていない部分に Oracle が仮定値を入れます（仮定値は、SQL 実行日時によって変化します）。

Oracle は日付及び時間を含めた（システムの仮定値を含めた）条件判定をするため、Oracle が仮定値を入れた状態でその列をキー値とした条件検索を実行すると、条件が合わないことがあります。したがって、TIME 及び DATE 型のマッピングキーを DATE 型の列へ反映しないでください。

注※5

バイナリデータとして反映します。

注※6

- Oracle の LONG RAW 型への反映はできますが、Oracle の BLOB 型へは反映できません。
- BLOB 型を追加更新又は後方削除更新している場合は、データ連動しないでください。抽出側 DB と反映側 DB で不整合が発生します。

注※7

文字集合指定をした列はデータ連動できません。指定した場合、ポートチェックエラーとなります。

## (b) HiRDB データ型から ODBC のデータ型への変換 (SQL Server, Oracle)

ODBC ドライバを使用して Oracle 又は SQL Server へデータ連動する場合は、HiRDB データ型を ODBC のデータ型に変換して ODBC 関数を発行します。このため、反映対象表の作成時には、ODBC のデータ型から変換できるデータ型を使用してください。

HiRDB から ODBC ドライバを使用してデータ連動する場合の、抽出対象表と反映対象表のデータ型の対応を次の表に示します。

表 4-5 ODBC を使用して反映するデータのデータ型の対応

HiRDB データ型		対応する ODBC のデータ型	
データ属性	データの長さ	ODBC SQL データ型	ODBC C データ型
INTEGER	データ長については、マニュアル「HiRDB SQL リファレンス」を参照してください。	SQL_INTEGER	SQL_C_SLONG
SMALLINT		SQL_SMALLINT	SQL_C_SSHORT
DECIMAL※1		SQL_DECIMAL	SQL_C_CHAR
LARGE DECIMAL※1			
FLOAT		SQL_FLOAT	SQL_C_DOUBLE
SMALLFLT		SQL_REAL	SQL_C_FLOAT
CHAR※6	1 ≤ n ≤ 254	SQL_CHAR	SQL_C_CHAR

HiRDB データ型		対応する ODBC のデータ型	
データ属性	データの長さ	ODBC SQL データ型	ODBC C データ型
MCHAR			
NCHAR			
CHAR <sup>※6</sup>	n ≥ 255	SQL_LONGVARCHAR	SQL_C_CHAR
MCHAR			
NCHAR			
VARCHAR <sup>※6</sup>	データ長については、マニュアル「HiRDB SQL リファレンス」を参照してください。		
LONG VARCHAR			
MVARCHAR			
LONG MVARCHAR			
NVARCHAR			
LONG NVARCHAR			
DATE <sup>※2</sup>			SQL_DATE
TIME <sup>※2</sup>		SQL_TIME <sup>※5</sup>	SQL_C_TIME <sup>※5</sup>
INTERVAL YEAR TO DATE <sup>※3</sup>		SQL_BINARY (5)	SQL_C_BINARY
INTERVAL HOUR TO SECOND <sup>※3</sup>		SQL_BINARY (4)	SQL_C_BINARY
BLOB <sup>※4</sup>		SQL_LONGVARBINARY	SQL_C_BINARY

#### 注 1

データの長さの n は、データ長（バイト数）を表します。

ただし、NCHAR については文字数を表します。

#### 注 2

反映対象表のデータ型の（ ）内は、データ長（バイト数）を表します。

#### 注※1

HiRDB の DECIMAL 型の精度は 29 けたまでですが、ODBC の SQL\_DECIMAL 型は 15 けたまでです。16 けた以上のデータを反映したときの処理は、使用する ODBC ドライバの仕様に従います。

#### 注※2

ODBC ドライバが SQL Server の場合は、DATE 及び TIME 型は SQL\_CHAR に変換されます。

#### 注※3

バイナリデータとして反映します。

#### 注※4

BLOB 型を追加更新又は後方削除更新している場合は、データ連動しないでください。抽出側 DB と反映側 DB で不整合が発生します。

#### 注※5

反映先が Oracle の場合は、データ型を Date 型にしてください。

#### 注※6

文字集合指定をした列はデータ連動できません。指定した場合、ポートチェックエラーとなります。

### 4.1.3 文字コードの対応の設計

文字コードの対応の設計方法について説明します。

#### (1) 文字コードの種類

抽出側 DB が Oracle の場合に抽出対象となる文字コードは、シフト JIS、UTF-8 又は EUC です。また、抽出側 DB が HiRDB の場合に抽出対象となる文字コードは、シフト JIS、UTF-8 又は EUC です。

ただし、HiRDB システム同士のデータ連動と同様に、実際に抽出側 DB が上記の文字コード体系で運用されているかどうかのチェックはしません。

#### (2) 文字コードの変換

##### (a) EBCDIK/KEIS から EUC への文字コードの変換規則

Oracle の識別子に指定できるのは 30 バイトまでです。EBCDIK/KEIS から EUC 及び UTF-8 への文字コードの変換後に識別子が 30 バイトを超えた場合は、定義解析時にエラーとなります。このため、データ連動を開始する前に、あらかじめ抽出側システムをチェックして、識別子が制限値を超えないことを確認してください。

### 4.1.4 時系列情報表の作成

時系列情報表を作成する手順について説明します。

#### (1) 時系列情報を取得する表の定義

更新情報、反映日付、反映時間などの時系列情報を取得するためには、反映側 DB 上に、これらの情報を取得するための表（時系列情報表）を作成しておく必要があります。時系列情報表を作成するときには、表名と列構成（列数、列順序、列名）は任意に指定できます。ただし、列の属性は取得する情報に合わせて定義する必要があります。

また、反映側 DB が HiRDB の場合、取得する情報の種類によっては、更新情報フィールド定義の const 句で定数を指定します。

時系列情報表の列の属性について、反映側 DB ごとに説明します。

### (a) 反映側 DB が HiRDB の場合

反映側 DB が HiRDB の場合の、時系列情報表の列の属性と const 句での指定については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### (b) 反映側 DB が Oracle の場合

反映側 DB が Oracle の場合の時系列情報表の列の属性を次の表に示します。

表 4-6 反映側 DB が Oracle の場合の時系列情報表の列の属性

取得する情報	列の属性 (外部データ型)
抽出日付	DATE (日付及び時間の情報を持ちます)
抽出時間	
反映日付	
反映時間	
反映種別	CHAR(5)
マッピングキー	[4.1.2(2)(a) HiRDB データ型から Oracle (UNIX) データ型への変換] の「表 4-4 Oracle に反映するデータのデータ型対応」及びマニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。
更新情報	

上記のデータ型で変換できるデータ型以外で時系列情報表を定義した場合は、時系列情報の取得処理及び反映処理の動作を保証できません。

### (c) ODBC ドライバを使用してデータ連動する場合

ODBC ドライバを使用してデータ連動する場合の時系列情報表の列の属性を次の表に示します。

表 4-7 ODBC ドライバを使用してデータ連動する場合の時系列情報表の列の属性

取得する情報	列の属性
抽出日付	SQL_DATE*
反映日付	
抽出時間	SQL_TIME*
反映時間	
反映種別	SQL_CHAR(5)
マッピングキー	[4.1.2(2)(b) HiRDB データ型から ODBC のデータ型への変換 (SQL Server, Oracle)] の「表 4-5 ODBC を使用して反映するデータのデータ型の対応」を参照してください。

取得する情報	列の属性
更新情報	

注※

ODBC ドライバが SQL Server 又は Oracle ODBC driver の場合は、SQL\_TIMESTAMP となり、日付と時間が同一列に反映されます。

# 5

## 定義

この章では、抽出側 Datareplicator Extension と反映側 Datareplicator Extension にデータ連動システムの設計内容を定義する方法について説明します。

## 5.1 抽出側 Datareplicator Extension を使用する場合の定義 (Oracle の場合)

---

次に示す抽出側 Datareplicator の定義のうち、抽出側 DB に Oracle を使用する場合に必要な定義の指定方法について説明します。

- 抽出システム定義
- 抽出環境定義
- 送信環境定義
- 抽出定義

なお、「オペランドの説明」には、Datareplicator Extension 固有のオペランドの説明だけを記載します。そのほかのオペランドの説明については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 5.1.1 抽出システム定義

#### (1) 機能

抽出システム定義では、抽出側 Datareplicator 全体の稼働環境についての情報を定義します。

#### (2) 形式

- 共通定義部でだけ定義する項目

set hdeid = 抽出側 Datareplicator 識別子

[ set extdef = "抽出環境定義ファイル名" ]

[ set sendcontrol = nodemst | sendmst ]

set sendid01 = 送信先識別子

[ … [ set sendid64 = 送信先識別子 ] ]

[ set senddef01 = "送信環境定義ファイル名" ]

[ … [ set senddef64 = "送信環境定義ファイル名" ] ]

set sendid0001 = 送信先識別子

[ … [ set sendid4096 = 送信先識別子 ] ]

[ set senddef0001 = 送信環境定義ファイル名 ]

[ ... [ set senddef4096 =送信環境定義ファイル名]

[ set errfile\_unique = true | false ]

[ set errfilesz =エラー情報ファイルの最大ファイルサイズ ]

[ set syslogout = true | false ]

[ set syslog\_message\_suppress =メッセージ番号 [, メッセージ番号] ...]

[ set dblocale = { sjis | euc | utf-8 } }

[ set msglocale = { english | sjis-japanese | euc-japanese } }

[ set watchintvl =エラー監視間隔 ]

[ set cmwaittime =通信待ち時間 ]

[ set mstservice =マスターノードマスタ間通信サービス名 ]

[ set extinfo\_num =最大更新情報名数 ]

[ set syncterm = true | false ] ※1

[ set termlevel = { normal | plan | both } } ※1

[ set info\_message\_out = nosuppress | suppress ]

[ set except\_suppress =メッセージ番号 [, メッセージ番号] ...]

[ set int\_trc\_lvl =稼働トレース取得レベル [, 稼働トレース取得範囲]

[ set int\_trc\_filesz =稼働トレースファイルのサイズ ]

[ set int\_trc\_rintvl =稼働トレース情報取得間隔 ]

[ set sendprocnum =送信プロセス最大起動数 ]

[ set smt\_sendintvl =送信マスタプロセス送信間隔 ]

[ set smt\_sendintvl\_scale = minute | second ]

[ set smt\_editbufsize =更新情報編集バッファ長 ]

[ set smt\_readbufnum =更新情報読み込み用 I/O バッファ数 ]

[ set smt\_queue\_read\_wait\_interval =送信プロセスの抽出情報キューファイル読み込み間隔 ]

[ set file\_dupenv =二重化定義ファイル名 ]

[ set nodecontrol = unit | server ] ※2

[ set node\_connection\_accept = true | false ] ※1

[ set connection\_accept\_hostname = 抽出ノードマスタプロセス接続要求受け付けホスト名 ] ※1

[ set connection\_accept\_service = 抽出ノードマスタプロセス接続要求受け付けサービス名 ] ※1

[ set connection\_accept\_waittime = 抽出ノードマスタプロセス接続要求待ち時間 ] ※1

[ set connection\_retry\_time = 抽出ノードマスタプロセス再接続処理時間 ] ※1

set oracleusr = データ連動リソース管理ユーザ名 /パスワード ※3

[ set oracle\_maxextcol = 抽出可能最大列数 ] ※3

[ set oracle\_msgkeepime = DEQUEUE 後のアドバンストキュー内更新情報保存時間 ] ※3

[ set oracle\_aq\_storage = アドバンストキュー作成時の記憶域パラメタ ] ※3

[ set oracle\_sorttbl\_storage = 抽出順序管理テーブル記憶域パラメタ ] ※3

[ set oracle\_sortidx = true | false ] ※3

[ set oracle\_sortidx\_storage = 抽出順序管理テーブルインデクス記憶域パラメタ ] ※3

[ set node\_syslogout = true | false ]

[ set send\_counter\_reset = true | false ]

[ set hirdb\_audit\_trail = all | none ]

[ set decimal\_max\_precision = DECIMAL 型の精度の最大値 ] ※3

[ set resource\_chk\_err = continue | stop ]

[ set recover\_info\_send = true | false ] ※1

[ set recover\_info\_send\_interval = 回復情報送信間隔 ] ※1

[ set cm\_errno\_check = true | false ]

- 個別定義部でだけ定義する項目

個別定義部を定義する場合、個別定義部の開始ヘッダに指定するサーバ名には、固定名称「Oracle」を指定してください。

[ set node\_pddir = 抽出側 HiRDB の PDDIR 環境変数の設定値 ] ※1

[ set node\_pdconfpath = 抽出側 HiRDB の PDCONFPATH 環境変数の設定値 ] ※1

[ set node\_shlibpath = 抽出側 Oracle のライブラリパス ]

[ set node\_host = ノードマスタホスト名 ]

注※1

Datereplicator Extension を使用する場合は、指定しても無視されます。

注※2

Datereplicator Extension では unit だけ指定できます。server は指定できません。

注※3

Datereplicator Extension 固有のオペランドです。

### (3) オペランドの説明

Datereplicator Extension 固有のオペランドについて説明します。

- oracleusr = データ連動リソース管理ユーザ名 /パスワード

Oracle から HiRDB へデータ連動する場合のデータ連動リソース管理ユーザ名とパスワードを指定します。

- データ連動リソース管理ユーザ名  
～ 〈1～8 文字の識別子〉

Oracle から HiRDB へデータ連動する場合のデータ連動リソース管理ユーザ名を指定します。

- /パスワード  
～ 〈1～8 文字の記号名称〉

データ連動リソース管理ユーザのパスワードを指定します。

- oracle\_maxextcol = 抽出可能最大列数  
～ 〈符号なし整数〉 ((7～500)) 《50》

抽出対象表当たりの抽出可能最大列数を指定します。抽出可能最大列数は、更新情報に対して次の算出式に従って算出した結果のうち、最も大きい値を指します。

ただし、算出結果が 7 より小さい値となった場合は、7 を指定してください。

マッピングキー更新あり

抽出可能最大列数 = 4 + マッピングキー数 + 抽出列数

マッピングキー更新なし

抽出可能最大列数 = 4 + 抽出列数

抽出定義でこのオペランドの指定値を超える抽出列数の抽出定義文が指定されている場合は、抽出定義プリプロセスファイルの作成処理で定義エラーとなります。

このオペランドの指定値は、更新処理の性能に影響を及ぼすため、できるだけ上記の計算式に従って算出した値を指定することをお勧めします。

なお、同一表に対して複数の更新情報を抽出する場合は、マッピングキー及び抽出列をマージした更新情報として算出してください。

- **oracle\_msgkeepime = DEQUEUE 後のアドバンストキュー内更新情報保存時間**  
～ 〈符号なし整数〉 ((0~1000)) 《12》 (単位：時間)

アドバンストキューからの取り出し (DEQUEUE) が完了した更新情報の保存時間を指定します。抽出処理に不正が発生した場合のトラブルシュート情報として、取り出し (DEQUEUE) が完了した後も、更新情報をアドバンストキューに一定時間保存しておくことで、原因の調査に役立てることができます。

ただし、このオペランドの値を大きくすることで、必要となるアドバンストキュー容量も増加するため、ユーザ環境に応じて指定する必要があります。

- **oracle\_aq\_storage = アドバンストキュー作成時の記憶域パラメタ**

初期化時にアドバンストキュー作成のために発行する CREATE\_QUEUE\_TABLE プロシジャの storage\_clause オプションに、このオペランドの設定内容を指定することで、アドバンストキュー作成領域をカスタマイズできます。

このオペランドの最大文字数は 511 文字です。省略すると、storage\_clause オプションなしで CREATE\_QUEUE\_TABLE プロシジャが実行されます。

このオペランドの指定値については、Oracle のマニュアルを参照してください。

- **oracle\_sorttbl\_storage = 抽出順序管理テーブル記憶域パラメタ**

初期化時に抽出順序管理テーブル作成のために発行する CREATE TABLE コマンドのオプションに、このオペランドの設定内容を指定することで、抽出順序管理テーブルの作成領域などをカスタマイズできます。

このオペランドの最大文字数は 511 文字です。省略すると、オプションなしで CREATE TABLE コマンドが実行されます。

このオペランドの指定値については、Oracle のマニュアルを参照してください。

- **oracle\_sortidx = true | false**

抽出順序管理テーブルに対してインデックスを作成するかどうかを指定します。

抽出順序管理テーブルに対してインデックスを作成すると、抽出処理の性能は向上しますが、更新 AP の性能が低下する場合があります。

**true**

抽出順序管理テーブルに対してインデックスを作成します。

**false**

抽出順序管理テーブルに対してインデックスを作成しません。

- **oracle\_sortidx\_storage = 抽出順序管理テーブルインデックス記憶域パラメタ**

抽出順序管理テーブルに対するインデックス作成のために発行する CREATE INDEX コマンドのオプションに、このオペランドの設定内容を指定することで、インデックスの作成領域などをカスタマイズできます。

このオペランドの最大文字数は 511 文字です。省略すると、オプションなしで CREATE INDEX コマンドが実行されます。

このオペランドの指定値については、Oracle のマニュアルを参照してください。

• decimal\_max\_precision = DECIMAL 型の精度の最大値

～((29~38)) 《29》

抽出時に DECIMAL 型として抽出する NUMBER, DECIMAL, NUMERIC の精度の最大値を指定します。

38 より大きな値を指定した場合は定義エラーとなります。

このオペランドで指定した精度より大きな精度の NUMBER 列, DECIMAL 列, NUMERIC 列は, FLOAT 型として抽出します。詳細を次の表に示します。

表 5-1 DECIMAL 型の精度の最大値と抽出データ属性の対応

指定分類	抽出対象表のデータ型	m,n の値	反映対象表のデータ型
精度及び位取りの指定なし	NUMBER	-	FLOAT
	DECIMAL NUMERIC		INTEGER
精度だけ指定	NUMBER(m) DECIMAL(m) NUMERIC(m)	$m \leq 4$	SMALLINT
		$5 \leq m \leq 9$	INTEGER
		$10 \leq m \leq p$	[LARGE] DECIMAL(m)
		上記以外	FLOAT
精度及び位取りを指定	NUMBER(m,n) DECIMAL(m,n) NUMERIC(m,n)	次のどれかの条件を満たす場合 $m \geq p+1$ $n \geq p+1$ $m-n \geq p+1$	FLOAT
		次のすべての条件を満たす場合 $m < n$ $m < p+1$ $n < p+1$ $m-n < p+1$	[LARGE] DECIMAL(n,n)
		次のすべての条件を満たす場合 $n < 0$ $m < p+1$ $m-n < p+1$	[LARGE] DECIMAL(m-n,0)
		上記以外	[LARGE] DECIMAL(m,n)

(凡例)

- : 依存しません。

m：精度を示します。

n：位取りを示します。

p：decimal\_max\_precision に指定した値を示します。

- node\_shlibpath = 抽出側 Oracle のライブラリパス

～ 〈1～255 バイトのパス名〉

ライブラリパスを指定する Oracle 用環境変数に 256 バイト以上の値を指定した場合、抽出側 Datareplicator Extension は指定値不正エラーとなります。256 バイト以上の値を指定する必要がある場合には、このオペランドに、255 バイト以内で抽出側 Datareplicator Extension に必要なライブラリパスだけを指定してください。

このオペランドを指定することで、環境変数の指定値不正エラーが発生しなくなります。将来、環境変数の指定値が 256 バイト以上になることが予想される場合には、初期構築時にこのオペランドを指定しておくことをお勧めします。

なお、このオペランドの指定値を有効にするためには、抽出側 Datareplicator Extension を初期化してください。

## 5.1.2 抽出環境定義

### (1) 機能

抽出処理を稼働するために必要な情報を定義します。

### (2) 形式

[ set dsid = データ連動識別子 ]

[ set qufile001 = "抽出情報キューファイル名" ]

[ set qufile002 = "抽出情報キューファイル名"

[ … [ set qufile016 = "抽出情報キューファイル名" ] ] ]

[ set queuesize = 抽出情報キューファイルサイズ ]

[ set logiosize = システムログ I/O バッファサイズ ] ※1

[ set quiosize = 抽出情報キュー I/O バッファサイズ ]

[ set extsuppress = true | false ] ※1

[ set ext\_wait\_interval = 抽出終端検知後の抽出再開始間隔 ]

[ set extract\_level = current\_gen | all\_gen ] ※1

[ set int\_trc\_getv = true | false ]



high

最大値に置換します。

low

最小値に置換します。

例外データの置換については、「3.1.4 データ型の変換」を参照してください。

- tran\_datanum = 1 トランザクション当たりの最大更新情報数

～ 〈符号なし整数〉 ((1~4096)) 《16》

1 トランザクション当たりの最大更新情報数を指定します。

トランザクションの更新情報については、「3.1.5 更新情報及びトランザクション情報の出力」を参照してください。

なお、このオペランドには、抽出情報キューファイルに格納できる範囲の値を指定してください。1 トランザクションが抽出情報キューファイルに格納できない場合、抽出処理がキュー満杯エラーとなります。

## 5.1.3 送信環境定義

### (1) 機能

送信処理を稼働するために必要な情報を定義します。送信環境定義ファイルは送信先識別子ごとに作成する必要があります。

### (2) 形式

[ set sendhdsid = 相手システム識別子 ]

[ set hdeservice = サービス名 ]

[ set hdehost = 送信先ホスト名 ]

[ set protocol = tcp | osi ] ※2

[ set extract\_tselector = T セレクタ ] ※2

[ set nsap\_address = NSAP アドレス ] ※2

[ set senduoc = use | nouse ]

[ set sendintvl = 送信間隔 ]

[ set sendintvl\_scale = minute | second ]

[ set nsndid001 = 送信抑止受信元識別子 ]

[ ... [ set nsndid256 =送信抑止受信元識別子 ] ] ※1

[ set keepalive = true | false ]

[ set retrynum =コネクションリトライ回数 ]

[ set retry\_interval =コネクションリトライ時のリトライ間隔 ]

[ set overwrite = true | false ]

[ set overwrite\_continue = true | false ]

[ set maxtran =同時実行最大トランザクション最大数 ]

[ set maxtrandata =トランザクション内最大更新情報数 ]

[ set readbufnum =送信用の抽出情報キュー I/O バッファ数 ]

[ set editbufsize =更新情報編集バッファ長 ]

[ set prg\_eventno =イベント番号 ] ※1

[ set int\_trc\_getv = true | false ]

[ set queue\_read\_wait\_interval =送信プロセスの抽出情報キューファイル読み込み間隔 ]

[ set recvwatchtime =データ受信用回線監視時間 ]

[ set send\_delay\_limit\_time =送信滞留時間しきい値 ]

[ set reflect\_mode = server | uap ] ※1

[ set eventsync =同期イベントコード ] ※1

[ set eventcntreset =送信データ件数リセットイベントコード ]

注※1

Datareplicator Extension を使用する場合は、指定しても無視されます。

注※2

OSI プロトコルで通信を実行する場合に指定してください。

## 5.1.4 抽出定義

抽出側 Datareplicator の抽出、送信処理に必要な情報を定義します。

## (1) 抽出定義文

データ連動の対象にする表の抽出条件を定義します。

### (a) 形式

```
{ extract 認可識別子.表識別子 ( { 列名 [抽出データ型] * [ { {, 列名 [抽出データ型] * } } ... ] | * } )  
  to 更新情報名  
  key | ukey (列名 [ { {, 列名 } } ... ] ) } ...
```

注※

Datereplicator Extension 固有のオプションです。

### (b) オペランドの説明

Datereplicator Extension 固有のオプションについて説明します。

#### • 抽出データ型

Oracle の一部のデータ型によって指定できるオプションで、Oracle のデータから抽出する情報、及び対応する HiRDB データ型を指定します。

このオプションで指定できるキーワードを次の表に示します。

表 5-2 Oracle の場合に指定できるキーワード

Oracle のデータ型	指定できるキーワード	抽出する情報	対応する HiRDB データ型
DATE	date	日付だけ	DATE
	time	時間だけ	TIME
	char	日付と時間	CHAR (19)
CHAR (10)	date	日付だけ	DATE
CHAR (8)	time	時間だけ	TIME

このオプションを省略すると、次のデータ型で抽出されます。

Oracle のデータ型	HiRDB データ型
DATE	CHAR (19)
CHAR (10)	MCHAR (10)
CHAR (8)	MCHAR (8)

### (c) 注意

- 抽出データ型オプションが異なる複数のデータ型の列を抽出できません。

- 抽出データ型オプションを指定して、CHAR (10) 及び CHAR (8) を抽出する場合、次に示す形式のデータだけを抽出対象とします。その他の形式のデータは、例外データとなります。

Oracle のデータ型	抽出対象とするデータ形式
CHAR (10)	'YYYY-MM-DD' YYYY : 年 (0001~9999) MM : 月 (01~12) DD : 日 (01~31)
CHAR (8)	'HH:MM:SS' HH : 時 (00~23) MM : 分 (00~59) SS : 秒 (00~59)

- 抽出対象表当たりの抽出可能最大列数は、抽出システム定義の oracle\_maxextcol オペランドの指定値に従います。オペランドの詳細については、「5.1.1 抽出システム定義」を参照してください。
- マッピングキーに指定できる Oracle のデータ型及びデータの長さを次に示します。

データ型	データの長さ
NUMBER (n)	n <= 29
CHAR	255 バイト以下
VARCHAR2	255 バイト以下
NCHAR	127 バイト以下
NVARCHAR2	127 バイト以下
DATE	—

## (2) 送信定義文

抽出側 DB から抽出した更新情報を、送信先識別子ごとに選択して送信するための定義をします。送信文は省略できます。送信文の指定方法によって、送信先へ送信される更新情報は異なります。

### (a) 形式

```
{ { send 送信先識別子 from 更新情報名
```

```
  [ where 列名* {比較演算子 定数* | in (定数* [, 定数*] ...)
```

```
    | flike (比較開始位置, 定数*) }
```

```
  [ and 列名* {比較演算子 定数* | in (定数* [, 定数*] ...)
```

```
    | flike (比較開始位置, 定数*) } ] ] ] } ... ] }
```

注※

Datareplicator Extension を使用する場合は、オペランドの設定内容が Datareplicator と異なります。

## (b) オペランドの説明

Datareplicator Extension を使用する場合に、設定内容が異なる点について説明します。

### • 列名

送信対象とする更新データのマッピングキー列を指定します。ただし、選択条件列として指定できるマッピングキー列の属性は、次の表に示す属性に限ります。マッピングキー以外の列、又は次の表に示す属性以外の列を指定すると、定義エラーになります。

表 5-3 選択条件に指定できるマッピングキー列属性

列属性	長さ, 精度	位取り	指定できる長さ
NUMBER(n)	n ≤ 29	n ≤ 29	n ≤ 29, かつ 精度 - 位取り ≤ 29
CHAR(n)	n ≤ 255	—	1 ≤ n ≤ 255
VARCHAR2(n)	n ≤ 255	—	1 ≤ n ≤ 255
NCHAR(n)	n ≤ 255	—	1 ≤ n ≤ 255
NVARCHAR2(n)	n ≤ 255	—	1 ≤ n ≤ 255

### • 定数

選択条件で比較する定数を指定します。

選択条件列の属性と定数との関連を次の表に示します。

表 5-4 選択条件列の属性と定数との関連

列属性	定数	
	文字列定数	真数定数 又は符号なし整数
NUMBER(n)	×	○
CHAR(n)	○	×
VARCHAR2(n)	○	×
NCHAR(n)	○	×
NVARCHAR2(n)	○	×

(凡例)

○：指定できる定数

×：指定できない定数

## 5.2 抽出側 Datareplicator Extension を使用する場合の定義 (SQL Server の場合)

---

次に示す抽出側 Datareplicator の定義のうち、抽出側 DB に SQL Server を使用する場合に必要な定義の指定方法について説明します。

- 抽出システム定義
- 抽出環境定義
- 送信環境定義
- 抽出定義

なお、「オペランドの説明」には、Datareplicator Extension 固有のオペランドの説明だけを記載します。そのほかのオペランドの説明については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 5.2.1 抽出システム定義

#### (1) 機能

抽出システム定義では、抽出側 Datareplicator 全体の稼働環境についての情報を定義します。

#### (2) 形式

- 共通定義部でだけ定義する項目

set hdeid = 抽出側 Datareplicator 識別子

[ set extdef = "抽出環境定義ファイル名" ]

[ set sendcontrol = nodemst | sendmst ]

set sendid01 = 送信先識別子

[ … [ set sendid64 = 送信先識別子 ] ]

[ set senddef01 = "送信環境定義ファイル名" ]

[ … [ set senddef64 = "送信環境定義ファイル名" ] ]

set sendid0001 = 送信先識別子

[ … [ set sendid4096 = 送信先識別子 ] ]

[ set senddef0001 = 送信環境定義ファイル名 ]

[ … [ set senddef4096 =送信環境定義ファイル名 ] ]

[ set errfile\_unique = true | false ]

[ set errfilesz =エラー情報ファイルの最大ファイルサイズ ]

[ set syslogout = true | false ]

[ set syslog\_message\_suppress =メッセージ番号 [, メッセージ番号] … ]

[ set dblocale = { sjis | euc | utf-8 } ]

[ set msglocale = { english | sjis-japanese | euc-japanese } ]

[ set watchintvl =エラー監視間隔 ]

[ set cmwaittime =通信待ち時間 ]

[ set mstservice =マスターノードマスタ間通信サービス名 ]

[ set extinfunum =最大更新情報名数 ]

[ set syncterm = true | false ] ※1

[ set termlevel = { normal | plan | both } ] ※1

[ set info\_message\_out = nosuppress | suppress ]

[ set except\_suppress =メッセージ番号 [, メッセージ番号] … ]

[ set int\_trc\_lvl =稼働トレース取得レベル [, 稼働トレース取得範囲] ]

[ set int\_trc\_filesz =稼働トレースファイルのサイズ ]

[ set int\_trc\_rintvl =稼働トレース情報取得間隔 ]

[ set sendprocnum =送信プロセス最大起動数 ]

[ set smt\_sendintvl =送信マスタプロセス送信間隔 ]

[ set smt\_sendintvl\_scale = minute | second ]

[ set smt\_editbufsize =更新情報編集バッファ長 ]

[ set smt\_readbufnum =更新情報読み込み用 I/O バッファ数 ]

[ set smt\_queue\_read\_wait\_interval =送信プロセスの抽出情報キューファイル読み込み間隔 ]

[ set file\_dupenv =二重化定義ファイル名 ]

[ set nodecontrol = unit | server ] ※2

[ set node\_connection\_accept = true | false ] ※1

[ set connection\_accept\_hostname = 抽出ノードマスタプロセス接続要求受け付けホスト名 ] ※1

[ set connection\_accept\_service = 抽出ノードマスタプロセス接続要求受け付けサービス名 ] ※1

[ set connection\_accept\_waittime = 抽出ノードマスタプロセス接続要求待ち時間 ] ※1

[ set connection\_retry\_time = 抽出ノードマスタプロセス再接続処理時間 ] ※1

[ set sqls\_dsn = DSN 名 ] ※3

[ set sqls\_usr = SQL Server 接続ユーザ ID [/パスワード]] ※3

[ set sqls\_maxextcol = 抽出可能最大列数 ] ※3

[ set sqls\_msgkeeptime = DEQUEUE 後のキューテーブル内更新情報保存時間 ] ※3

[ set sqls\_qtbl\_filegroup = キューテーブルの FILEGROUP 名 ] ※3

[ set sqls\_qtbl\_idx = true | false ] ※3

[ set sqls\_qtblidx\_filegroup = キューテーブルインデクスの FILEGROUP 名 ] ※3

[ set node\_syslogout = true | false ]

[ set send\_counter\_reset = true | false ]

[ set hirdb\_audit\_trail = all | none ]

[ set decimal\_max\_precision = DECIMAL 型の精度の最大値 ] ※3

[ set resource\_chk\_err = continue | stop ]

[ set recover\_info\_send = true | false ] ※1

[ set recover\_info\_send\_interval = 回復情報送信間隔 ] ※1

[ set cm\_errno\_check = true | false ]

- 個別定義部でだけ定義する項目
 

個別定義部を定義する場合、個別定義部の開始ヘッダに指定するサーバ名には、固定名称「SQLSVR」を指定してください。

[ set node\_pddir = 抽出側 HiRDB の PDDIR 環境変数の設定値 ] ※1

[ set node\_pdconfpath = 抽出側 HiRDB の PDCONFPATH 環境変数の設定値 ] ※1

[ set node\_shlibpath = 抽出側 HiRDB の SHLIB\_PATH 環境変数の設定値 ] ※1

[ set node\_host = ノードマスタホスト名 ]

注※1

Datereplicator Extension を使用する場合は、指定しても無視されます。

注※2

Datereplicator Extension では unit だけ指定できます。server は指定できません。

注※3

Datereplicator Extension 固有のオペランドです。

### (3) オペランドの説明

Datereplicator Extension 固有のオペランドについて説明します。

- sqls\_dsn = DSN 名

～ 〈1～31 文字の識別子〉

SQL Server に接続するための DSN 名を指定します。

- sqls\_usr = SQL Server 接続ユーザ ID [/パスワード]

SQL Server 接続ユーザ ID とパスワードを指定します。

- SQL Server 接続ユーザ ID

～ 〈1～8 文字の識別子〉

SQL Server 接続ユーザ ID を指定します。ここでは、ログインユーザの ID とデータベースユーザ ID が同じユーザ ID を指定してください。

sqls\_dsn でユーザ ID が指定されていても、その情報は無視されます。

指定するユーザは、SQL Server でリソースを作成できる DBA 権限、又は次の三つの固定データベースロールを与えておく必要があります。

db\_ddladmin, db\_datareader, db\_datawrite

- /パスワード

～ 〈1～8 文字の記号名称〉

SQL Server 接続ユーザ ID のパスワードを指定します。

- sqls\_maxextcol = 抽出可能最大列数

～ 〈符号なし整数〉 ((10～1024)) 《50》

抽出対象表 1 表当たりの抽出可能最大列数を指定します。更新情報に対して次の算出式に従って算出した結果のうち、最も大きい値を指定します。

ただし、算出結果が 10 より小さい値となった場合は、10 を指定してください。

## マッピングキー更新あり

抽出可能最大列数 = 7 + マッピングキー数 + 抽出列数

## マッピングキー更新なし

抽出可能最大列数 = 7 + 抽出列数

- `sqls_msgkeep` = DEQUEUE 後のキューテーブル内更新情報保存時間

～ 〈符号なし整数〉 ((0~1000)) 《12》 (単位：時間)

キューテーブルからの取り出し (DEQUEUE) が完了した更新情報の保存時間を指定します。

抽出処理に不正が発生した場合のトラブルシュート情報として、取り出し (DEQUEUE) が完了した後も、更新情報をキューテーブルに一定時間保存しておくことで、原因の調査に役立てることができません。

ただし、このオペランドの値を大きくすることで、必要となるキューテーブル容量も増加するため、ユーザ環境に応じて指定する必要があります。

- `sqls_qtbl_filegroup` = キューテーブルの FILEGROUP 名

キュー表、抽出データ制御表とリソースオペランド表を作成する場合の FILEGROUP 名を指定します。

このオペランドが省略された場合、キューテーブル、抽出同期管理テーブル、及びリソースオペランドテーブルが SQL Server のデフォルト FILEGROUP で作成されます。

- `sqls_qtbl_idx` = {true | false}

キューテーブルに対してインデックスを作成するかどうかを指定します。

キューテーブルに対してインデックスを作成すると、抽出処理の性能は向上しますが、更新 AP の性能が低下する場合があります。

### true

キューテーブルに対してインデックスを作成します。

### false

キューテーブルに対してインデックスを作成しません。

このオペランドを省略すると、オプションなしで CREATE INDEX コマンドが実行されます。

このオペランドの指定値については、SQL Server のマニュアルを参照してください。

- `sqls_qtblidx_filegroup` = キューテーブルインデックスの FILEGROUP 名

～ 〈符号なし整数〉 ((0~128))

キューテーブルにインデックスを作成するための FILEGROUP 名を指定します。

このオペランドが省略された場合、キューテーブルのインデックスが SQL Server のデフォルト FILEGROUP で作成されます。

- `decimal_max_precision` = DECIMAL 型の精度の最大値

～ ((29~38)) 《29》

抽出時に DECIMAL 型として抽出する NUMBER, DECIMAL, NUMERIC の精度の最大値を指定します。

38 より大きな値を指定した場合は定義エラーとなります。

このオペランドで指定した精度より大きな精度の NUMBER 列、DECIMAL 列、NUMERIC 列は、FLOAT 型として抽出します。詳細を次の表に示します。

表 5-5 DECIMAL 型の精度の最大値と抽出データ属性の対応

抽出対象表のデータ型	m,n の値	反映対象表のデータ型
DECIMAL(m,n) NUMERIC(m,n)	次のすべての条件を満たす場合 $0 < m \leq p$ $0 \leq n \leq m$	[LARGE] DECIMAL(m,n)
	上記以外	FLOAT

(凡例)

m：精度を示します。

n：位取りを示します。

p：decimal\_max\_precision に指定した値を示します。

## 5.2.2 抽出環境定義

### (1) 機能

抽出処理を稼働するために必要な情報を定義します。

### (2) 形式

[ set dsid = データ連動識別子 ]

[ set qfile001 = "抽出情報キューファイル名" ]

[ set qfile002 = "抽出情報キューファイル名"

[ … [ set qfile016 = "抽出情報キューファイル名" ] ] ]

[ set queuesize = 抽出情報キューファイルサイズ ]

[ set logiosize = システムログ I/O バッファサイズ ] ※1

[ set quiosize = 抽出情報キュー I/O バッファサイズ ]

[ set extsuppress = true | false ] ※1

[ set ext\_wait\_interval = 抽出終端検知後の抽出再開始間隔 ]

[ set extract\_level = current\_gen | all\_gen ] ※1

[ set int\_trc\_getv = true | false ]



high

最大値に置換します。

low

最小値に置換します。

例外データの置換については、「3.1.4 データ型の変換」を参照してください。

- tran\_datanum = 1 トランザクション当たりの最大更新情報数

～ 〈符号なし整数〉 ((1~4096)) 《16》

1 トランザクション当たりの最大更新情報数を指定します。

トランザクションの更新情報については、「3.1.5 更新情報及びトランザクション情報の出力」を参照してください。

なお、このオペランドには、抽出情報キューファイルに格納できる範囲の値を指定してください。1 トランザクションが抽出情報キューファイルに格納できない場合、抽出処理がキュー満杯エラーとなります。

## 5.2.3 送信環境定義

### (1) 機能

送信処理を稼働するために必要な情報を定義します。送信環境定義ファイルは送信先識別子ごとに作成する必要があります。

### (2) 形式

[ set sendhdsid = 相手システム識別子 ]

[ set hdeservice = サービス名 ]

[ set hdehost = 送信先ホスト名 ]

[ set protocol = tcp | osi ] ※2

[ set extract\_tselector = T セレクタ ] ※2

[ set nsap\_address = NSAP アドレス ] ※2

[ set senduoc = use | nouse ]

[ set sendintvl = 送信間隔 ]

[ set sendintvl\_scale = minute | second ]

[ set nsndid001 = 送信抑止受信元識別子 ]

[ ... [ set nsndid256 =送信抑止受信元識別子 ] ] ※1

[ set keepalive = true | false ]

[ set retrynum =コネクションリトライ回数 ]

[ set retry\_interval =コネクションリトライ時のリトライ間隔 ]

[ set overwrite = true | false ]

[ set overwrite\_continue = true | false ]

[ set maxtran =同時実行最大トランザクション最大数 ]

[ set maxtrandata =トランザクション内最大更新情報数 ]

[ set readbufnum =送信用の抽出情報キュー I/O バッファ数 ]

[ set editbufsize =更新情報編集バッファ長 ]

[ set prg\_eventno =イベント番号 ] ※1

[ set int\_trc\_getv = true | false ]

[ set queue\_read\_wait\_interval =送信プロセスの抽出情報キューファイル読み込み間隔 ]

[ set recvwatchtime =データ受信用回線監視時間 ]

[ set send\_delay\_limit\_time =送信滞留時間しきい値 ]

[ set reflect\_mode = server | uap ] ※1

[ set eventsync =同期イベントコード ] ※1

[ set eventcntreset =送信データ件数リセットイベントコード ]

注※1

Datareplicator Extension を使用する場合は、指定しても無視されます。

注※2

OSI プロトコルで通信を実行する場合に指定してください。

## 5.2.4 抽出定義

抽出側 Datareplicator の抽出、送信処理に必要な情報を定義します。

## (1) 抽出定義文

データ連動の対象にする表の抽出条件を定義します。

### (a) 形式

```
{ extract 認可識別子.表識別子 ( { 列名 [抽出データ型] * [ { 列名 [抽出データ型] * } … ] | * } )
```

```
to 更新情報名
```

```
key | ukey (列名 [ { 列名 } … ] ) } …
```

注※

Datereplicator Extension 固有のオプションです。

### (b) オペランドの説明

Datereplicator Extension 固有のオプションについて説明します。

#### • 抽出データ型

SQL Server の一部のデータ型によって指定できるオプションで、SQL Server のデータから抽出する情報、及び対応する HiRDB データ型を指定します。

このオプションで指定できるキーワードを次の表に示します。

表 5-6 SQL Server の場合に指定できるキーワード

SQL Server データ型	指定できるキーワード	抽出する情報	対応する HiRDB データ型
Datetime	date	日付だけ	DATE
	time	時間だけ	TIME
	char	日付と時間	CHAR (19)
smalldatetime	date	日付だけ	DATE
	time	時間だけ	TIME
	char	日付と時間	CHAR (19)
CHAR (10)	date	日付だけ	DATE
CHAR (8)	time	時間だけ	TIME
BINARY (n)	blob	バイナリ	BLOB (n)
	varchar	バイナリ	VARCHAR (2n)
VARBINARY (n)	blob	バイナリ	BLOB (n)
	varchar	バイナリ	VARCHAR (2n)

このオプションを省略すると、次のデータ型で抽出されます。

SQL Server のデータ型	HiRDB データ型
DATETIME	CHAR (19) ※1
SMALLDATETIME	CHAR (19) ※2
CHAR (10)	CHAR (10)
CHAR (8)	CHAR (8)
BINARY (n)	BLOB
VARBINARY (n)	BLOB

注※1

ミリ秒の位は無視されます。

注※2

秒の位が 00 になります。

### (c) 注意

- 抽出データ型オプションが異なる複数のデータ型の列を抽出できません。
- 抽出データ型オプションを指定して、CHAR (10) 及び CHAR (8) を抽出する場合、次に示す形式のデータだけを抽出対象とします。その他の形式のデータは、例外データとなります。

SQL Server のデータ型	抽出対象とするデータ形式
CHAR (10)	'YYYY-MM-DD' YYYY : 年 (0001~9999) MM : 月 (01~12) DD : 日 (01~31)
CHAR (8)	'HH:MM:SS' HH : 時 (00~23) MM : 分 (00~59) SS : 秒 (00~59)

- 抽出対象表当たりの抽出可能最大列数は、抽出システム定義の `sqls_maxextcol` オペランドの指定値に従います。オペランドの詳細については、「5.2.1 抽出システム定義」を参照してください。
- マッピングキーに指定できる SQL Server のデータ型及びデータの長さを次に示します。

データ型	データの長さ
CHAR	255 バイト以下
VARCHAR	255 バイト以下
NCHAR	127 バイト以下
NVARCHAR	127 バイト以下

## (2) 送信定義文

抽出側 DB から抽出した更新情報を、送信先識別子ごとに選択して送信するための定義をします。送信文は省略できます。送信文の指定方法によって、送信先へ送信される更新情報は異なります。

送信定義文のオペランドの設定内容は Datareplicator と同じです。詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

## 5.3 反映側 Datareplicator Extension を使用する場合の定義

次に示す反映側 Datareplicator の定義のうち、Datareplicator Extension を使用する場合に必要な定義の指定方法について説明します。

- 反映システム定義
- 反映環境定義
- 反映定義

なお、「オペランドの説明」には、Datareplicator Extension 固有のオペランドの説明だけを記載します。そのほかのオペランドの説明については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

### 5.3.1 反映システム定義

#### (1) 機能

反映システム定義では、反映側 Datareplicator の稼働環境についての情報を定義します。

#### (2) 形式

[ set hdsid = 反映側 Datareplicator 識別子 ]

[ set hirdbusr = HiRDB 接続認可識別子 [ /パスワード ] ] ※1

[ set protocol1 = tcp | osi ]

[ set protocol2 = tcp | osi ]

set dsid001 = データ連動識別子

[ … [ set dsid128 = データ連動識別子 ] ]

set refenv001 = "反映環境定義ファイル名"

[ … [ set refenv128 = "反映環境定義ファイル名" ] ]

[ set hdsservice = サービス名 ]

[ set reflect\_tselector = T セレクタ ]

[ set hirdb\_audit\_trail = all | uoc | none\_cont | none\_stop ] ※3

[ set keepalive = true | false ]

[ set errfilesz = 反映エラー情報ファイルサイズ ]

[ set syslogout = true | false ]

[ set syslog\_message\_suppress = メッセージ番号 [, メッセージ番号] ... ]

[ set dblocale = { sjis | euc | utf-8 | unknown } ]

[ set msglocale = { english | sjis-japanese | euc-japanese } ]

[ set discintvl = disconnect 発行間隔 ]

[ set info\_message\_out = nosuppress | suppress ]

[ set except\_suppress = メッセージ番号 [, メッセージ番号] ... ]

[ set commitment\_method = fxa\_none | fxa\_sqlc ] ※1

[ set int\_trc\_lvl = 稼働トレース取得レベル [, 稼働トレース取得範囲] ]

[ set int\_trc\_filesz = 稼働トレースファイルのサイズ ]

[ set int\_trc\_rintvl = 稼働トレース情報取得間隔 ]

[ set use\_convertlib = true | false ]

[ set ref\_wait\_interval = 反映プロセスの反映情報キューファイル読み込み間隔 ]

[ set commit\_wait\_time = COMMIT 発行間隔 ]

[ set file\_dupenv = 二重化定義ファイル名 ]

[ set dbkind = { hirdb | oracle | odbc\_oracle | odbc\_sqlserver } ] ※2

set datasource = データソース名 ※2

set oracleusr = Oracle 接続認可識別子 [ /パスワード ] ※2

set odbcusr = ODBC 接続認可識別子 [ /パスワード ] ※2

[ set syncgroup001 = 同期反映グループ名, データ連動識別子 [ {, データ連動識別子} ... ] ] ※1

[ set syncgrp\_discintvl = disconnect 発行待ち時間 ] ※1

[ set syncwait\_limit\_tran\_count = 同期待ち限界トランザクション数 ] ※1

[ set syncwait\_limit\_time = 同期待ち限界時間 ] ※1

[ set reflect\_counter\_reset = true | false ]

[ set resource\_chk\_err = continue | stop ]

注※1

Datereplicator Extension を使用する場合は、指定しても無視されます。

注※2

Datereplicator Extension 固有のオペランドです。

注※3

HiRDB にアクセスする反映情報編集 UOC に対してだけ有効です。

### (3) オペランドの説明

Datereplicator Extension 固有のオペランドについて説明します。

- dbkind = { hirdb | oracle | odbc\_oracle | odbc\_sqlserver }

反映対象となる DBMS を指定します。

**hirdb**

HiRDB を反映対象とします。

**oracle**

Oracle (Windows 版以外) を反映対象とします。

**odbc\_oracle**

Oracle (Windows 版) を反映対象とします。

ODBC ドライバ (Oracle ODBC driver) を使用します。

**odbc\_sqlserver**

SQL Server を反映対象とします。

ODBC ドライバ (SQL Server) を使用します。

ODBC ドライバの設定内容については、「[2.13 ODBC 環境の設定 \(Windows\)](#)」を参照してください。

- datasource = データソース名

ODBC ドライバを使用してデータ連動する場合に指定するオペランドで、接続する ODBC ドライバのシステムデータソース名を指定します。

- oracleusr = Oracle 接続認可識別子 [ /パスワード ]

Oracle とデータ連動する場合に指定するオペランドで、Oracle へ接続するときに使う認可識別子とパスワードを指定します。

- Oracle 接続認可識別子

～ 〈1~8 文字の識別子〉

Oracle へ接続するときに使う認可識別子を指定します。

- /パスワード

～ 〈1～30 文字の記号名称〉

Oracle へ接続するときに使うパスワードを指定します。

パスワードに「\*」を指定すると、反映側 Datareplicator の起動時に、標準入力へのパスワードの入力要求があります。ここで入力したパスワードをそのまま使って、Oracle へ接続されます。

/パスワードを省略すると、パスワードなしとして Oracle に接続されます。

- `odbcusr` = ODBC 接続認可識別子 [ /パスワード ]

ODBC ドライバを使用してデータ連動する場合に指定するオペランドで、反映側システムへ接続するときに使う認可識別子とパスワードを指定します。

- ODBC 接続認可識別子

～ 〈1～8 文字の識別子〉

反映側システムへ接続するときに使う認可識別子を指定します。

- /パスワード

～ 〈1～30 文字の記号名称〉

反映側システムへ接続するときに使うパスワードを指定します。パスワードは省略できます。

## 5.3.2 反映環境定義

### (1) 機能

反映処理を稼働するために必要な情報を定義します。複数のデータ連動識別子をデータ連動の対象とする場合、反映処理はデータ連動識別子ごとに稼働するため、反映環境定義もデータ連動識別子ごとに必要になります。

### (2) 形式

```
set qufile001 = "反映情報キューファイル名"
```

```
set qufile002 = "反映情報キューファイル名"
```

```
[ ... [ set qufile008 = "反映情報キューファイル名" ] ]
```

```
set queuesize = 反映情報キューファイルサイズ
```

```
[ set reffile = "反映定義ファイル名" ]
```

```
set statsfile = "反映ステータスファイル名"
```

```
set statssize = 反映ステータスファイルサイズ
```

```
set unreffile1 = "未反映情報ファイル名 (プライマリ) "
```

```
set unreffile2 = "未反映情報ファイル名 (セカンダリ) "
```

[ set unreffilesz = 未反映情報ファイルサイズ ]

[ set startmode = { trn | tbl | spd } ]

[ set restartmode = initial | continue ]

[ set breaktime = hh:mm ]

[ set breakmode = trn | tbl ]

[ set eventtrn = トランザクション単位反映イベントコード ]

[ set eventtbl = 表単位反映イベントコード ]

[ set eventretrn = トランザクション単位反映再起動イベントコード ]

[ set eventretbl = 表単位反映再起動イベントコード ]

[ set eventspd = 反映処理停止イベントコード ]

[ set eventcntreset = 反映処理数リセットイベントコード ]

[ set defmerge = true | false ]

[ set cmtintvl = 反映処理コミット間隔 ]

[ set trncmtintvl = トランザクション単位反映方式での反映処理コミット間隔 ]

[ set tblcmtintvl = 表単位反映方式での反映処理コミット間隔 ]

[ set tblcheck = true | false ]

set defshmsize = 定義情報格納用共有メモリサイズ

[ set ebcdic\_type = { eck78 | ekk78 | eck83 | ekk83 } ]

[ set shiftspace\_cnv = multi | single ]

[ set undefcode\_cnv = multi | single ]

[ set ref\_data\_backspace = suppress | nosuppress ]

[ set skip\_sqlcode = SQLCODE [, SQLCODE ] ... ] ※1

[ set skip\_mvcelmwarn = true | false ] ※2

[ set sqlerr\_skip\_info = { output | msgoutput | sqloutput | nooutput } ]

[ set extract\_init = check | nocheck ]

```

[ set db_connect_retry_number = DB コネクトリトライ回数 ]
[ set db_connect_retry_interval = DB コネクトリトライ間隔 ]
[ set skip_codecvt_error = true | false ]
[ set int_trc_getl = 稼働トレース取得範囲 ]
[ set device01 = Datareplicator ファイルシステム領域名 [[, 割り当てファイル種別] ...]
[ set device02 = Datareplicator ファイルシステム領域名 [[, 割り当てファイル種別] ...]
  [ ... [ set device09 = Datareplicator ファイルシステム領域名 [[, 割り当てファイル種別]
...]]]]]
[ set ujcodekind = rcv | sam ]
[ set discintvl = disconnect 発行間隔 ]
[ set ref_wait_interval = 反映プロセスの反映情報キューファイル読み込み間隔 ]
[ set commit_wait_time = COMMIT 発行間隔 ]
[ set mapping_key_check = not_null_unique | unique | none ] ※1
[ set control_trigger = execute | not_execute ] ※2
[ set control_reference_trigger = execute | not_execute ] ※2
[ set check_pending = use | nouse ] ※2
[ set reflect_delay_limit_time = 反映滞留時間しきい値 ]
[ set sql_lockerr_retrynum = 排他エラー発生時のトランザクションリトライ回数 ]
[ set xa_recovery_retry_count = トランザクション回復要求のリトライ回数 ] ※2
[ set xa_recovery_retry_interval = トランザクション回復要求のリトライ間隔 ] ※2
[ set reflect_trn_max_sqlnum = 反映トランザクション内最大更新 SQL 数 ]

```

注※1

Datareplicator Extension を使用する場合は、オペランドの設定内容が Datareplicator と異なります。

注※2

Datareplicator Extension を使用する場合は、指定しても無視されます。

### (3) オペランドの説明

Datereplicator Extension を使用する場合に、設定内容が異なる点について説明します。

- skip\_sqlcode = SQLCODE [, SQLCODE] …

#### UNIX 版 Oracle とデータ連動する場合

UNIX 版 Oracle とデータ連動する場合は、SQLCODE ではなく **Oracle エラーコード**を指定します。Oracle エラーコードは最大 32 個指定できます。

Oracle とデータ連動する場合の SQL プロセスは、OCI インタフェースを使用しているため、反映処理中にエラーが発生した場合は、ORA\_xxxxx の形式のエラーとなります。

このオペランドの Oracle エラーコードは、ORA\_xxxxx の xxxxx 部分だけを指定します。Oracle エラーコードは、1~32767 までとし、それ以外の値を指定した場合は動作を保証できません。

次に主なエラーとそのエラーに対応する Oracle エラーコードを示します。

エラー内容	Oracle エラーコード	備考
INSERT 時にマッピングキーに対する一意性制約違反が発生しました。	1	—
UPDATE 及び DELETE 時に該当する行がありません。	1403	Oracle には該当するエラーがないので、1403 にエラーコードがマッピングされています。
デッドロックが発生しました。	60	—

(凡例)

—：特に備考はありません。

#### Windows 版 Oracle とデータ連動する場合

ODBC ドライバを使用して Windows 版 Oracle とデータ連動する場合、ODBC 関数の `SQLERROR()` のネイティブエラーコードから取得した SQLCODE を指定します。指定する SQLCODE については、使用している DBMS のマニュアルを参照してください。

ただし、UPDATE 及び DELETE 時に該当する行がないというエラーをスキップしたい場合は、SQLCODE に 100 を指定してください。ODBC には該当するエラーがないので、100 にエラーコードがマッピングされています。

- mapping\_key\_check = not null unique | unique | none

マッピングキー列に対するユニークチェックを行うときの条件を指定します。ここで指定する値は、反映定義の check 句を省略した場合の仮定値になります。したがって、反映定義に check 句を指定した場合、check 句の指定が優先されます。

#### not\_null\_unique

反映対象表に対して、表 5-7 の内容を満たす一意性が保証されていること、及び一意性が保証されている列が NOT NULL 属性であることを確認します。

## unique

反映対象表に対して、表 5-7 の内容を満たす一意性が保証されていることを確認します。NULL 値に関するチェックは行わないため、完全なデータの一意性は抽出側 DB を更新するユーザプログラムなどで保証する必要があります。

## none

チェックを行いません。データの一意性は抽出側 DB を更新するユーザプログラムなどで保証する必要があります。

## 注

反映対象表が反映側システムにあるかどうかをチェックしない場合、このオペランドの指定値に関係なくマッピングキーのチェックは行いません。反映対象表が反映側システムにあるかどうかのチェックに関しては、反映環境定義の tblcheck オペランドの指定に従います。

ユニークチェックの内容について次の表に示します。

表 5-7 ユニークチェックの内容

チェック項目	チェック内容
一意性の有無	次のどちらかによって一意性が保証されていること。 <ul style="list-style-type: none"><li>一意制約を持つ列が存在する</li><li>一意索引が定義されている</li></ul>
一意性保証列	マッピングキー構成列だけであること。

## 5.3.3 反映定義

### (1) 更新情報フィールド定義

抽出側システムから送信される更新情報の名称と、フィールドの名称を定義します。

更新情報のすべてのフィールドを、並び順を変えないで反映表定義の load 文で指定した列名の順に対応付ける場合、更新情報フィールド定義は省略できます。この場合、反映表定義の load 文では、フィールド名称を省略して、「\*」を指定します。

更新情報フィールド定義は、一つの更新情報に対して一つだけ定義できます。

#### (a) 形式

[[ format 更新情報名

name フィールド名称

[[ const 初期値\* | by 列データ編集関数識別子 [ nocodecnv ] | [ nocodecnv ] ]]

[[ name フィールド名称

```
[{ const 初期値* | by 列データ編集関数識別子 [ nocodecnv ] | [ nocodecnv ]}]
...}]
}}...]
```

注※

Datereplicator Extension を使用する場合は、オペランドの設定内容が Datereplicator と異なります。

## (b) オペランドの説明

Datereplicator Extension を使用する場合に、設定内容が異なる点について説明します。

### • const 初期値

更新情報にはない任意のフィールドを追加する場合に、追加するフィールドの初期値を指定します。初期値として指定できる定数の中で、Datereplicator Extension を使用する場合に差異がある定数を次の表に示します。

表 5-8 const 句に指定できる定数

定数	説明	指定内容
反映日付定数	反映日時を示す定数	reflect_date
反映時間定数		reflect_time
抽出日付定数	抽出日時を示す定数※	extract_date
抽出時間定数		extract_time

注

これらの定数は、次に示す列属性で反映を実行します。

**Oracle とデータ連動する場合：**

Oracle 外部データ型の DATE 型

**ODBC ドライバを使用してデータ連動する場合：**

ODBC データ型の SQL\_TIMESTAMP 型

注※

HiRDB システム同士でデータ連動する場合は、DB に対して実際に更新処理が実行された日時が抽出日時となりますが、Oracle 又は SQL Server から HiRDB へデータ連動する場合は、抽出側 Datereplicator Extension がアドバンストキューから更新情報を取り出した日時が抽出日時となります。

## (2) 反映表定義

更新情報に対する反映対象の表を定義します。一個の更新情報を、複数の反映対象の表に反映する場合には、反映表定義を反映対象の表の数だけ繰り返し定義します。フィールド名称、認可識別子、及び表識別

子の指定方法については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」の反映定義の定義の規則を参照してください。

### (a) 形式

```
{ { load { フィールド名称 { {, フィールド名称 } } … } | * }  
  from 更新情報名  
  { to { { timestamp | sqlconvopt1* | sqlconvopt2* } }  
    [ 認可識別子. ] 表識別子  
    [ check { not_null_unique | unique | none } ]  
    [ with lock ] *  
    | by 'uoc名'  
  } … }
```

注※

Datareplicator Extension を使用する場合は、指定しても無視されます。

## (3) 反映グループ定義

表単位反映方式（表単位分割方式，キーレンジ単位分割方式，又はハッシュ分割方式）で反映処理を並列に実行するときの実行単位にする反映グループを定義します。反映グループは、128 個まで繰り返し定義できます。フィールド名称，認可識別子，及び表識別子の指定方法については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」の反映定義の定義の規則を参照してください。

### (a) 形式

```
{ { group 反映グループ名称  
  by [ 認可識別子. ] 表識別子 { {, [ 認可識別子. ] 表識別子 } } … }  
/**** キーレンジ単位分割方式の定義 ****/  
{ { { in '反映先 FES のホスト名' / '反映先 FES のサーバ識別子'  
      [ / '反映先 FES のポート番号' ] } } *1  
  [ having キーレンジ分割条件文 { {, キーレンジ分割条件文 } } … ] | other }  
} … }
```

/\*\*\*\* ハッシュ分割方式の定義 \*\*\*\*/ ※2

【{{ hash

{ in '反映先 FES のホスト名' / '反映先 FES のサーバ識別子' [ / '反映先 FES のポート番号' ]

[(RD エリア名 {{, RD エリア名}} … | other)]

【{{, '反映先 FES のホスト名' / '反映先 FES のサーバ識別子' [ / '反映先 FES のポート番号' ]

[(RD エリア名 {{, RD エリア名}} … | other)]

}}… ]

| divide into SQL プロセス分割数 }

}}… ]

注※1

Datereplicator Extension を使用する場合は、指定しても無視されます。

注※2

Datereplicator Extension を使用する場合は、ハッシュ分割方式の定義を指定するとエラーとなります。

# 6

## 運用

この章では、Datareplicator Extension を使用したデータ連動システムを運用するための手順、Datareplicator Extension の起動と終了、及び Datareplicator Extension の運用について説明します。

## 6.1 抽出側 Datareplicator Extension の起動と終了 (Oracle の場合)

抽出側 DB が Oracle の場合の抽出側 Datareplicator Extension の起動方法と終了方法について説明します。

### 6.1.1 抽出側 Datareplicator Extension の起動

抽出側システムで `hdestartO` コマンドを実行すると、抽出システム定義、抽出環境定義、及び送信環境定義に従って抽出側 Datareplicator Extension が起動します。

`hdestartO` コマンドは、抽出環境が `hdestartO` コマンド (`-i` オプションを指定) を使用して初期化されている場合にだけ実行でき、`hdestartO` コマンド以外を使用して初期化されている場合は、`hdestartO` コマンドを実行してもエラー終了します。

抽出環境初期化時に使用したコマンドと抽出側 Datareplicator Extension 起動時に使用したコマンドの関係を次に示します。

抽出環境初期化時に使用したコマンド	抽出側 Datareplicator Extension 起動時に使用したコマンド	
	<code>hdestart</code>	<code>hdestartO</code>
<code>hdestart</code>	抽出側 Datareplicator が起動します。	KFRB00061-E エラー
<code>hdestartO</code>	KFRB00061-E エラー	抽出側 Datareplicator Extension が起動します。

#### 注

抽出定義プリプロセスファイルを作成するための `hdeprepO` コマンドも、抽出環境が `hdestartO` コマンド (`-i` オプションを指定) を使用して初期化されている場合にだけ実行でき、`hdestartO` コマンド以外を使用して初期化されている場合は、`hdeprepO` コマンドを実行しても KFRB00061-E エラーとなります。抽出定義プリプロセスファイルの作成に関する注意については、「[6.2.3\(2\)抽出定義プリプロセスファイルの作成に関する注意](#)」を参照してください。

抽出側 DB が Oracle の場合、抽出側 Datareplicator Extension は Oracle 上のキューテーブル (アドバンストキュー) から更新情報を取り出します (DEQUEUE)。このため、抽出側 Datareplicator Extension を起動するときは抽出対象 Oracle が稼働している必要があります。

また、データ連動処理で、キューテーブルへの更新情報を格納するまでの間にエラーが発生した場合は、更新 AP 自体がエラーとなります。

抽出側システムでエラーが発生した場合の対処については、「[8.1.1 抽出側システムでエラーが発生した場合の対処 \(Oracle の場合\)](#)」を参照してください。

## 6.1.2 抽出側 Datareplicator Extension の終了

抽出側 DB が Oracle の場合、抽出側 Datareplicator Extension の抽出プロセスは抽出対象 Oracle の AP として動作します。このため、抽出側 Datareplicator Extension は抽出対象 Oracle を停止する前に停止する必要があります。

## 6.2 抽出側 Datareplicator Extension の運用 (Oracle の場合)

抽出側 DB が Oracle の場合の抽出側 Datareplicator Extension の運用方法について説明します。

### 6.2.1 抽出対象表の運用

次に示す場合の、抽出対象表の運用手順について説明します。

- 抽出対象表を追加、変更、及び削除する場合
- 表の定義変更によって抽出対象外となった表を、再度抽出対象とする場合
- 任意の抽出対象表に対する抽出処理を一時的に抑止する場合

#### (1) 抽出対象表を追加、変更、及び削除する場合

抽出対象表の追加、変更、及び削除する場合の運用手順を次に示します。

1. キューテーブル (アドバンストキュー) に抽出されていない更新情報がないことを、hderesstateO コマンドで確認します。
2. 抽出情報キューファイルに抽出されていない更新情報がないことを、hdestateO コマンドで確認します。
3. 抽出側 Datareplicator Extension を停止します。
4. 抽出定義を更新し、抽出対象表を追加、変更、又は削除します。
5. 抽出定義プリプロセスファイルを作成します。
6. 抽出側 Datareplicator Extension を起動します。

#### (2) 表の定義変更によって抽出対象外となった表を、再度抽出対象とする場合

表の定義変更に関連して、Oracle の処理によってトリガが削除又は無効化された場合は、それ以降、その表からは更新情報を抽出しません。

トリガが削除又は無効化される表の定義変更については、Oracle のマニュアルを参照してください。

表の定義変更によって抽出対象外となった表を、再度抽出対象とする場合の運用手順を次に示します。

1. 抽出側 Datareplicator Extension を停止します。
2. 抽出定義プリプロセスファイルを作成します。
3. 抽出側 Datareplicator Extension を起動します。

ただし、抽出対象表の定義変更から上記の手順で再度抽出対象とするまでの間に、この表に対して更新処理が実行されていた場合、これらの更新処理は抽出されないため、抽出側 DB と反映側 DB が不整合となります。

抽出側 DB と反映側 DB との間に不整合が発生した場合の対処については、「[8.1.1\(5\)抽出側 DB と反映側 DB に不整合が発生した場合](#)」を参照してください。

### (3) 任意の抽出対象表に対する抽出処理を一時的に抑止する場合

任意の抽出対象表に対する抽出処理を一時的に抑止する場合の運用手順を次に示します。

1. 抽出処理を一時的に抑止する抽出対象表のトリガ識別子を `hderesstateO` コマンドで取得します。
2. `DROP TRIGGER` 又は `ALTER TRIGGER` でトリガを削除又は無効化します。
3. 抽出側 `Datareplicator Extension` を停止します。
4. 抽出定義プリプロセスファイルを作成します。
5. 抽出側 `Datareplicator Extension` を起動します。

2.から 3.までの間は、抽出処理が抑止されます。

なお、抽出処理を抑止している間に更新処理が実行されていた場合、これらの更新処理は抽出されないため、抽出側 DB と反映側 DB が不整合となります。

抽出側 DB と反映側 DB との間に不整合が発生した場合の対処については、「[8.1.1\(5\)抽出側 DB と反映側 DB に不整合が発生した場合](#)」を参照してください。

## 6.2.2 イベント機能を使用する場合の運用

イベント機能を使用する場合の運用手順を次に示します。

1. イベント制御用テーブルを作成します。
2. 抽出側 `Datareplicator Extension` を停止します。
3. 抽出定義プリプロセスファイルを作成します。
4. 抽出側 `Datareplicator Extension` を起動します。
5. イベント発行コマンドを実行します。

Oracle から HiRDB へデータ連動する場合も、HiRDB システム同士のデータ連動と同様に、イベント制御用テーブルはユーザが手動で作成する必要があります。

イベント制御用テーブルの構成は、HiRDB システム同士のデータ連動時で、抽出側 HiRDB がシングルサーバの場合と同じです。イベント制御用テーブルの構成については、マニュアル「[HiRDB データ連動機能 HiRDB Datareplicator](#)」を参照してください。

なお、イベント制御用テーブルは、データ連動リソースとして一括して削除できるように、データ連動リソース管理ユーザのリソースとして作成することをお勧めします。

## 6.2.3 抽出側 Datareplicator Extension 運用時の注意

抽出側 Datareplicator Extension を運用するときの注意を次に示します。

### (1) 初期化についての注意

抽出側 DB が Oracle の場合、抽出環境を初期化するとき Oracle に対してデータ連動リソースを作成します。したがって、抽出環境を初期化するときには抽出対象 Oracle が稼働している必要があります。

### (2) 抽出定義プリプロセスファイルの作成に関する注意

抽出定義プリプロセスファイルの作成に関する注意を次に示します。

抽出定義プリプロセスファイルの作成でエラーとなった場合は、エラー要因を排除した後、再度、抽出定義プリプロセスファイルを作成してください。

#### (a) Oracle の稼働状態

抽出側 DB が Oracle の場合、Oracle に対して抽出対象表の定義情報の取得とデータ連動リソースの作成が実行されます。このため、抽出定義プリプロセスファイルを作成するときは抽出対象 Oracle が稼働している必要があります。

#### (b) 抽出側 Datareplicator Extension 稼働中での抽出定義プリプロセスファイルの作成

HiRDB システム同士でデータ連動する場合は、抽出側 Datareplicator 稼働中に抽出定義プリプロセスファイルを作成できます。

しかし、Oracle から HiRDB へデータ連動する場合は、抽出側 Datareplicator Extension 稼働中に抽出定義プリプロセスファイルを作成すると、抽出対象表に定義されているトリガは即時に再作成されますが、抽出側 Datareplicator Extension が認識する抽出定義ファイルは抽出定義プリプロセスファイルに変換されていない（変更内容は次回の起動から認識されるため）という矛盾が生じます。

このため、抽出定義プリプロセスファイルは、抽出側 Datareplicator Extension の停止中に作成する必要があります。

#### (c) 抽出未完了状態での抽出定義プリプロセスファイルの作成

Oracle から HiRDB へデータ連動する場合も、HiRDB システム同士のデータ連動と同様に、抽出されていない更新情報がアドバンストキュー又は抽出情報キューファイルにある状態で抽出定義プリプロセスファイルを作成し、抽出側 Datareplicator Extension を再起動すると、抽出されていない更新情報と抽出側 Datareplicator Extension が認識する抽出定義との間に矛盾が生じます。

このため、抽出定義プリプロセスファイルは、アドバンストキュー及び抽出情報キューファイルの更新情報がすべて抽出されているときに作成する必要があります。

## 6.3 抽出側 Datareplicator Extension の起動と終了 (SQL Server の場合)

抽出側 DB が SQL Server の場合の抽出側 Datareplicator Extension の起動方法と終了方法について説明します。

### 6.3.1 抽出側 Datareplicator Extension の起動

抽出側システムで hdestartS コマンドを実行すると、抽出システム定義、抽出環境定義、及び送信環境定義に従って抽出側 Datareplicator Extension が起動します。

hdestartS コマンドは、抽出環境が hdestartS コマンド (-i オプションを指定) を使用して初期化されている場合にだけ実行でき、hdestartS コマンド以外を使用して初期化されている場合は、hdestartS コマンドを実行してもエラー終了します。

抽出環境初期化時に使用したコマンドと抽出側 Datareplicator Extension 起動時に使用したコマンドの関係を次に示します。

抽出環境初期化時に使用したコマンド	抽出側 Datareplicator Extension 起動時に使用したコマンド	
	hdestart	hdestartS
hdestart	抽出側 Datareplicator が起動します。	KFRB00061-E エラー
hdestartS	KFRB00061-E エラー	抽出側 Datareplicator Extension が起動します。

#### 注

抽出定義プリプロセスファイルを作成するための hdeprepS コマンドも、抽出環境が hdestartS コマンド (-i オプションを指定) を使用して初期化されている場合にだけ実行でき、hdestartS コマンド以外を使用して初期化されている場合は、hdeprepS コマンドを実行しても KFRB00061-E エラーとなります。抽出定義プリプロセスファイルの作成に関する注意については、「[6.2.3\(2\)抽出定義プリプロセスファイルの作成に関する注意](#)」を参照してください。

抽出側 DB が SQL Server の場合、抽出側 Datareplicator Extension はキューテーブルから更新情報を取り出します (DEQUEUE)。このため、抽出側 Datareplicator Extension を起動するときは抽出対象 SQL Server が稼働している必要があります。

また、データ連動処理で、キューテーブルへの更新情報を格納するまでの間にエラーが発生した場合は、更新 AP 自体がエラーとなります。

抽出側システムでエラーが発生した場合の対処については、「[8.1.2 抽出側システムでエラーが発生した場合の対処 \(SQL Server の場合\)](#)」を参照してください。

## 6.3.2 抽出側 Datareplicator Extension の終了

抽出側 DB が SQL Server の場合、抽出側 Datareplicator Extension の抽出プロセスは抽出対象 SQL Server の AP として動作します。このため、抽出側 Datareplicator Extension は抽出対象 SQL Server を停止する前に停止する必要があります。

## 6.4 抽出側 Datareplicator Extension の運用 (SQL Server の場合)

抽出側 DB が SQL Server の場合の抽出側 Datareplicator Extension の運用方法について説明します。

### 6.4.1 抽出対象表の運用

次に示す場合の、抽出対象表の運用手順について説明します。

- 抽出対象表を追加、変更、及び削除する場合
- 任意の抽出対象表に対する抽出処理を一時的に抑止する場合

#### (1) 抽出対象表を追加、変更、及び削除する場合

抽出対象表の追加、変更、及び削除する場合の運用手順は、修正後の抽出列数が抽出システム定義で定義した `sqls_maxextcol` を超えるか超えないかによって異なります。運用手順を次に示します。

##### (a) 抽出列数が `sqls_maxextcol` を超えない場合

抽出列数が `sqls_maxextcol` を超えない場合の運用手順を次に示します。

1. 未抽出の更新情報がないことを、`hderesstateS` コマンドで確認します。
2. 抽出情報キューファイルに抽出されていない更新情報がないことを、`hdestateS` コマンドで確認します。
3. 抽出側 Datareplicator Extension を停止します。
4. 抽出定義を更新し、抽出対象表を追加、変更、又は削除します。
5. 抽出定義プリプロセスファイルを作成します。
6. 抽出側 Datareplicator Extension を起動します。

##### (b) 抽出列数が `sqls_maxextcol` を超える場合

抽出列数が `sqls_maxextcol` を超える場合の運用手順を次に示します。

1. キューテーブルに抽出されていない更新情報がないことを、`hderesstateS` コマンドで確認します。
2. 抽出情報キューファイルに抽出されていない更新情報がないことを、`hdestateS` コマンドで確認します。
3. 反映情報キューファイルが空であることを、`hdsstate` コマンドで確認します。
4. 抽出側 Datareplicator を停止します。
5. 反映側 Datareplicator を停止します。
6. 抽出定義を更新し、抽出対象表を追加、又は変更します。
7. 抽出システム定義の `sqls_maxextcol` を修正し、ファイルを更新します。
8. `hdestartS` コマンド (`-i` オプションを指定) で環境を初期化します。

- 抽出定義プリプロセスファイルを作成します。
- hdsstart コマンド (-i オプションを指定) で反映側 Datareplicator を初期開始します。
- 抽出側 Datareplicator を起動します。

## (2) 任意の抽出対象表に対する抽出処理を一時的に抑止する場合

任意の抽出対象表に対する抽出処理を一時的に抑止する場合の運用手順を次に示します。

- 抽出処理を一時的に抑止する抽出対象表のトリガ識別子を hderesstateS コマンドで取得します。
- DROP TRIGGER でトリガを削除します。DROP TRIGGER で指定するトリガ名は hderesstateS コマンドで取得したものです。
- 抽出側 Datareplicator を停止します。
- 抽出定義プリプロセスファイルを作成します。
- 抽出側 Datareplicator Extension を起動します。

2.から 3.までの間は、抽出処理が抑止されます。

なお、抽出処理を抑止している間に更新処理が実行されていた場合、これらの更新処理は抽出されないため、抽出側 DB と反映側 DB が不整合となります。

抽出側 DB と反映側 DB との間に不整合が発生した場合の対処については、「[8.1.2\(5\)抽出側 DB と反映側 DB に不整合が発生した場合](#)」を参照してください。

## 6.4.2 イベント機能を使用する場合の運用

イベント機能を使用する場合の運用手順を次に示します。

- イベント制御用テーブルを作成します。
- 抽出側 Datareplicator Extension を停止します。
- 抽出定義プリプロセスファイルを作成します。
- 抽出側 Datareplicator Extension を起動します。
- イベント発行コマンドを実行します。

SQL Server から HiRDB へデータ連動する場合も、HiRDB システム同士のデータ連動と同様に、イベント制御用テーブルはユーザが手動で作成する必要があります。

イベント制御用テーブルの構成は、HiRDB システム同士のデータ連動時で、抽出側 HiRDB がシングルサーバの場合と同じです。イベント制御用テーブルの構成については、マニュアル「[HiRDB データ連動機能 HiRDB Datareplicator](#)」を参照してください。

なお、イベント制御用テーブルは、データ連動リソースとして一括して削除できるように、データ連動リソース管理ユーザのリソースとして作成することをお勧めします。

### 6.4.3 抽出側 Datareplicator Extension 運用時の注意

抽出側 Datareplicator Extension を運用するときの注意を次に示します。

#### (1) 初期化についての注意

抽出側 DB が SQL Server の場合、抽出環境を初期化するとき SQL Server に対してデータ連動リソースを作成します。したがって、抽出環境を初期化するときには抽出対象 SQL Server が稼働している必要があります。

#### (2) 抽出定義プリプロセスファイルの作成に関する注意

抽出定義プリプロセスファイルの作成に関する注意を次に示します。

抽出定義プリプロセスファイルの作成でエラーとなった場合は、エラー要因を排除した後、再度、抽出定義プリプロセスファイルを作成してください。

##### (a) SQL Server の稼働状態

抽出側 DB が SQL Server の場合、SQL Server に対して抽出対象表の定義情報の取得とデータ連動リソースの作成が実行されます。このため、抽出定義プリプロセスファイルを作成するときは抽出対象 SQL Server が稼働している必要があります。

##### (b) 抽出側 Datareplicator Extension 稼働中での抽出定義プリプロセスファイルの作成

HiRDB システム同士でデータ連動する場合は、抽出側 Datareplicator 稼働中に抽出定義プリプロセスファイルを作成できます。

しかし、SQL Server から HiRDB へデータ連動する場合は、抽出側 Datareplicator Extension 稼働中に抽出定義プリプロセスファイルを作成すると、抽出対象表に定義されているトリガは即時に再作成されますが、抽出側 Datareplicator Extension が認識する抽出定義ファイルは抽出定義プリプロセスファイルに変換されていない（変更内容は次回の起動から認識されるため）という矛盾が生じます。

このため、抽出定義プリプロセスファイルは、抽出側 Datareplicator Extension の停止中に作成する必要があります。

##### (c) 抽出未完了状態での抽出定義プリプロセスファイルの作成

SQL Server から HiRDB へデータ連動する場合も、HiRDB システム同士のデータ連動と同様に、抽出されていない更新情報がキューテーブル又は抽出情報キューファイルにある状態で抽出定義プリプロセスファイルを作成し、抽出側 Datareplicator Extension を再起動すると、抽出されていない更新情報と抽出側 Datareplicator Extension が認識する抽出定義との間に矛盾が生じます。

このため、抽出定義プリプロセスファイルは、キューテーブル及び抽出情報キューファイルの更新情報がすべて抽出されているときに作成する必要があります。

### (3) キューテーブルに関する注意事項

SQL Server の場合、システムイベントログ及びエラーファイルに KFRB06507-W メッセージが出力されることがあります。この場合、次の手順で抽出側 Datareplicator Extension を初期化する必要があります。

1. キューテーブルに抽出されていない更新情報がないことを、hderesstateS コマンドで確認します。
2. 抽出情報キューファイルに抽出されていない更新情報がないことを、hdestateS コマンドで確認します。
3. 抽出側 Datareplicator を停止します。
4. hdestartS コマンド (-i オプションを指定) で環境を初期化します。
5. 抽出定義プリプロセスファイルを作成します。
6. 抽出側 Datareplicator を起動します。

## 6.5 反映側 Datareplicator Extension の起動と終了

---

抽出側 DB が他社 DBMS の場合の反映側 Datareplicator Extension の起動方法と終了方法について説明します。

### 6.5.1 反映側 Datareplicator Extension の起動

反映側システムで `hdsstart` コマンドを実行すると、反映システム定義と反映環境定義に従って反映側 Datareplicator Extension が起動します。反映環境を初期化してから、反映側 Datareplicator を起動する時には、`hdsstart` コマンドに `-i` オプションを指定して実行します。

### 6.5.2 反映側 Datareplicator Extension の終了

反映側システムで `hdsstop` コマンドを実行すると、反映環境定義に従って反映側 Datareplicator Extension が終了します。反映側 Datareplicator Extension を終了するときは、必ず `hdsstop` コマンドを実行してください。OS のコマンドでプロセスを削除して、反映側 Datareplicator Extension を終了した場合には、動作を保証できません。

### 6.5.3 反映側 Datareplicator Extension 運用時の注意

反映側 Datareplicator Extension を運用するときは、マルチ FES 機能に対応した反映方式は指定できません。

## 6.6 ファイルの二重化を使うときの運用

---

ファイルの二重化に関するファイル構成変更の方法について説明します。

次の三つの構成変更の手順について説明します。この節では、ファイルの二重化に対して、通常の運用のことをファイルの一重化と表記します。

- ファイルの一重化から二重化への変更
- 二重化するファイルの変更
- ファイルの二重化から一重化への変更

ファイルの二重化についての詳細は、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

なお、hderesstate $X$ のようにコマンド名称の末尾に $X$ が付いているものは、抽出側 DB に合わせてコマンド名称を読み替えてください。

### 6.6.1 ファイルの一重化から二重化への変更

ファイルの一重化から二重化に変更するときの手順を次に示します。太字部分は、二重化のための変更を行っている部分です。

#### (1) 抽出側 Datareplicator Extension での変更手順

抽出側 Datareplicator Extension での変更手順を次に示します。

hderesstate $X$ のようにコマンド名称の末尾に $X$ が付いているものは、抽出側 DB に合わせてコマンド名称を読み替えてください。

1. hderesstate $X$  コマンドで、キューテーブルに未抽出の更新情報がないことを確認します。
2. 抽出側 DB を正常停止します。
3. hdestate $X$  コマンドで抽出情報キューファイルの read 位置と write 位置が同じであることを確認します。
4. hdestop $X$  コマンドで Datareplicator Extension を停止します。
5. **二重化定義ファイルを作成します。**
6. **二重化するファイルの物理ファイルを用意します**※。
7. **二重化するファイルの論理ファイルと同じ名称のファイルが存在する場合、そのファイルを削除します。**
8. hdestart $X$  -i コマンドで Datareplicator Extension を初期化します。
9. \$HDEPATH 下に作成された二重化制御ファイルのバックアップを取得します。
10. 系切り替え構成の場合、切り替え先の \$HDEPATH 下に二重化制御ファイルをコピーします。

11. 抽出側 DB を起動します。
12. `hdeprepX` コマンドを実行します。
13. `hdestartX` コマンドで Datareplicator Extension を起動します。

注※

物理ファイルの用意とは、キャラクタ型スペシャルファイルを使用する場合のパーティション割り当て、及びシンボリックリンクの作成を意味します。

## (2) 反映側 Datareplicator Extension での変更手順

反映側 Datareplicator Extension での変更手順を次に示します。

抽出側 Datareplicator Extension での変更手順との基本的な違いは、反映側 DB を停止する必要がないことです。

1. `hdsstate` コマンドで反映情報キューファイルの read 位置と write 位置が同じであることを確認します。
2. `hdsstop` コマンドで Datareplicator Extension を停止します。
3. 二重化定義ファイルを作成します。
4. 二重化するファイルの物理ファイルを用意します※。
5. 二重化するファイルの論理ファイルと同じ名称のファイルが存在する場合、そのファイルを削除します。
6. `hdsstart -i` コマンドで Datareplicator Extension を初期起動します。
7. `$HDSPATH` 下に作成された二重化制御ファイルのバックアップを取得します。
8. 系切り替え構成の場合、切り替え先の `$HDSPATH` 下に二重化制御ファイルをコピーします。

注※

物理ファイルの用意とは、キャラクタ型スペシャルファイルを使用する場合のパーティション割り当て、及びシンボリックリンクの作成を意味します。

なお、反映情報キューファイルと反映ステータスファイルをキャラクタ型スペシャルファイルで作成した後に初めて初期化する場合は、反映側 Datareplicator Extension を `hdsstart -i -f` コマンドで初期開始してください。2 回目以降は `-f` オプションを指定する必要はありません。

### 6.6.2 二重化するファイルの変更

二重化するファイルを追加、変更、削除するときの手順を次に示します。太字部分は、二重化のための変更を行っている部分です。

手順は、基本的にファイルの一重化から二重化への変更と同じです。

## (1) 抽出側 Datareplicator Extension での変更手順

抽出側 Datareplicator Extension での変更手順を次に示します。

1. `hderesstateX` コマンドで、キューテーブルに未抽出の更新情報がないことを確認します。
2. 抽出側 DB を正常停止します。
3. `hdestate` コマンドで抽出情報キューファイルの read 位置と write 位置が同じであることを確認します。
4. `hdestopX` コマンドで Datareplicator Extension を停止します。
5. 二重化定義ファイルを変更します。
6. 二重化するファイルの物理ファイルを用意します<sup>※</sup>。
7. 未使用になる物理ファイルを削除します。
8. 二重化するファイルの論理ファイルと同じ名称のファイルが存在する場合、そのファイルを削除します。
9. `hdestartX -i` コマンドで Datareplicator Extension を初期化します。
10. `$HDEPATH` 下に作成された二重化制御ファイルのバックアップを取得します。
11. 系切り替え構成の場合、切り替え先の `$HDEPATH` 下に二重化制御ファイルをコピーします。
12. 抽出側 DB を起動します。
13. `hdeprepX` コマンドを実行します。
14. `hdestartX` コマンドで Datareplicator Extension を起動します。

注※

物理ファイルの用意とは、キャラクタ型スペシャルファイルを使用する場合のパーティション割り当て、及びシンボリックリンクの作成を意味します。

## (2) 反映側 Datareplicator Extension での変更手順

反映側 Datareplicator Extension での変更手順を次に示します。

抽出側 Datareplicator Extension での変更手順との基本的な違いは、反映側 DB を停止する必要がないことです。

1. `hdsstate` コマンドで反映情報キューファイルの read 位置と write 位置が同じであることを確認します。
2. `hdsstop` コマンドで Datareplicator Extension を停止します。
3. 二重化定義ファイルを変更します。
4. 二重化するファイルの物理ファイルを用意します<sup>※</sup>。
5. 未使用になる物理ファイルを削除します。
6. 二重化するファイルの論理ファイルと同じ名称のファイルが存在する場合、そのファイルを削除します。
7. `hdsstart -i` コマンドで Datareplicator Extension を初期起動します。

- \$HDSPATH 下に作成された二重化制御ファイルのバックアップを取得します。
- 系切り替え構成の場合、切り替え先の\$HDSPATH 下に二重化制御ファイルをコピーします。

#### 注※

物理ファイルの用意とは、キャラクタ型スペシャルファイルを使用する場合のパーティション割り当て、及びシンボリックリンクの作成を意味します。

なお、反映情報キューファイルと反映ステータスファイルをキャラクタ型スペシャルファイルで作成した後初めて初期化する場合は、反映側 Datareplicator Extension を `hdsstart -i -f` コマンドで初期開始してください。2 回目以降は `-f` オプションを指定する必要はありません。

## 6.6.3 ファイルの二重化から一重化への変更

ファイルの二重化から一重化に変更するときの手順を次に示します。太字部分は、一重化に戻すための変更を行っている部分です。

### (1) 抽出側 Datareplicator Extension での変更手順

抽出側 Datareplicator Extension での変更手順を次に示します。

- `hderesstateX` コマンドで、キューテーブルに未抽出の更新情報がないことを確認します。
- 抽出側 DB を正常停止します。
- `hdestate` コマンドで抽出情報キューファイルの read 位置と write 位置が同じであることを確認します。
- `hdestopX` コマンドで Datareplicator Extension を停止します。
- 二重化で使用していた物理ファイルを削除します。
- 各ノードの\$HDEPATH 下に作成された二重化制御ファイルを削除します。
- 抽出システム定義からファイル二重化指定を削除します。
- 抽出側 DB を起動します。
- `hdestartX -i` コマンドで Datareplicator Extension を初期化します。
- `hdeprepX` コマンドを実行します。
- `hdestartX` コマンドで Datareplicator Extension を起動します。

### (2) 反映側 Datareplicator Extension での変更手順

反映側 Datareplicator Extension での変更手順を次に示します。

- `hdsstate` コマンドで反映情報キューファイルの read 位置と write 位置が同じであることを確認します。
- `hdsstop` コマンドで Datareplicator Extension を停止します。
- 二重化で使用していた物理ファイルを削除します。

4. \$HDSPATH 下に作成された二重化構成定義ファイルを削除します。
5. 反映システム定義からファイル二重化指定を削除します。
6. \$HDSPATH 下に作成された二重化制御ファイルを削除します。
7. 系切り替え構成の場合、切り替え先の\$HDEPATH 下の二重化制御ファイルを削除します。
8. hdsstart -i コマンドで Datareplicator Extension を初期起動します。

# 7

## コマンドの文法

この章では、Datareplicator Extension のコマンドの文法について説明します。

## コマンドの概要

ここでは、Datareplicator Extension でデータ連動するときに使用するコマンドについて説明します。

この章で説明しているコマンドには、hdestart $X$ のように末尾が $X$ 記述されているものがあります。この場合、 $X$ にはDBMSごとの識別文字が記述され、その機能は $X$ なしで表されるDatareplicatorのものと同じです。DBMSごとの識別文字を次の表に示します。

表 7-1 DBMS ごとの識別文字

DBMS	識別文字
Oracle	O
SQL Server	S

Datareplicator Extension でデータ連動するときに使うコマンドの一覧を次の表に示します。

この章では、hderesstate $X$ についてだけ、コマンドの詳細を説明します。hderesstate $X$ 以外の抽出側Datareplicatorのコマンド、及び反映側Datareplicator Extensionのコマンドについては、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

表 7-2 Datareplicator Extension でデータ連動するときに使うコマンドの一覧

コマンドの種類	コマンド名	機能	稼働中での実行	
			DBMS 稼働中	Datareplica tor Extension 稼 働中
抽出側 Datareplicator Extension のコマ ンド	hdestart $X$	抽出側 Datareplicator Extension を起動します。 - i オプションを指定すると抽出側 Datareplicator Extension 環境を初期化できます。この場合、起動はしません。	○ <sup>*1</sup>	△ <sup>*2</sup>
	hdestop $X$ <sup>*3</sup>	抽出側 Datareplicator Extension を停止します。	—	○
	hdeevent $X$ <sup>*3</sup>	イベントを発行します。	○	○
	hdestate $X$ <sup>*3</sup>	抽出側 Datareplicator Extension の状態を標準出力に出力します。	—	○
	hdeprep $X$ <sup>*3</sup>	抽出定義を解析して、抽出定義プリプロセスファイルを作成します。	○	△ <sup>*4</sup>
	hderesstate $X$ <sup>*3,*5</sup>	データ連動リソースの状態を標準出力に出力します。	○	○
反映側 Datareplicator Extension のコマ ンド	hdsstart	反映側 Datareplicator Extension を起動します。 - i オプションを指定すると、反映側 Datareplicator Extension 環境を初期化して起動できます。	○	×

コマンドの種類	コマンド名	機能	稼働中での実行	
			DBMS 稼働中	Datareplica tor Extension 稼 働中
	hdsstop <sup>※6</sup>	反映側 Datareplicator Extension を終了します。	—	○
	hdsrfctl <sup>※6</sup>	反映処理方式を制御したり，反映処理だけを再起動したりします。	—	○
	hdsstate <sup>※6</sup>	反映側 Datareplicator Extension の状態を標準出力に出力します。	—	○

(凡例)

- ：稼働中のときだけ実行できます。
- ×：停止中のときだけ実行できます。
- △：停止中のときは実行できます。
- ：関連しません。

注※1

他 DBMS とデータ連動する場合，-i オプション指定は，初期化処理が完了した後に終了します。

注※2

-i オプション指定は，抽出側 Datareplicator Extension の稼働中には実行できません。

注※3

hdestartX コマンドを実行したユーザ名で実行してください。

注※4

稼働中に実行しても，コマンドの実行結果が抽出側 Datareplicator Extension に反映されるのは，抽出側 Datareplicator Extension を停止してから，再開始した時点です（コマンド実行中に稼働している抽出側 Datareplicator Extension には反映されません）。

注※5

Datareplicator Extension 固有のコマンドです。

注※6

hdsstart コマンドを実行したユーザ名で実行してください。

# hderesstateO (データ連動リソースの状態の取得)

## 機能

コマンドを実行した時点のデータ連動リソースの状態を標準出力に出力します。出力される情報を、次の表に示します。

データ連動リソース	出力される情報
メッセージデータ型	出力情報はありません。
キューテーブル (アドバンストキュー)	<ul style="list-style-type: none"><li>• キューテーブルの有無</li><li>• キューテーブル名称</li><li>• キューテーブルテーブル名称</li><li>• キューテーブルからの取り出し (DEQUEUE) が完了した更新データの保存時間</li><li>• キューテーブルに格納されているメッセージ (更新データ) の情報<ul style="list-style-type: none"><li>• 未抽出メッセージ数</li><li>• 先頭の未抽出メッセージがキューテーブルに格納された時間※</li><li>• 最終の未抽出メッセージがキューテーブルに格納された時間※</li><li>• キューテーブルに格納されているメッセージの詳細※</li></ul></li></ul>
データ連動用トリガ	<ul style="list-style-type: none"><li>• 全トリガ数</li><li>• 全トリガ名</li><li>• 全トリガの定義対象表一覧</li><li>• 全トリガの状態 (有効/無効)</li></ul>
抽出同期管理テーブル	<ul style="list-style-type: none"><li>• 抽出同期管理テーブルの有無</li><li>• 抽出同期管理テーブル情報 (抽出情報キューファイルの最終 WRITE 位置)</li></ul>

## 注※

-d オプションを指定したときだけ出力します。

## 形式

```
hderesstateO -u ユーザ名 / パスワード  
[-d [-s 表示開始時間] [-t 表示終了時間] [-n 表示データ件数] ]
```

## オプション

- -u ユーザ名 / パスワード  
データ連動リソース管理ユーザのユーザ名とパスワードを指定します。
- -d  
キューテーブル内に格納されている更新データの詳細を出力します。
- -s 表示開始時間

-d オプションを指定したときにだけ指定できるオプションです。このオプションで指定した時間以降にキューテーブルに格納された更新データの詳細を出力します。

このオプションは、YYYYMMDDHHMMSS の形式で指定します。

このオプションを省略すると、00010101000000 が仮定されます。

- **-t 表示終了時間**

-d オプションを指定したときにだけ指定できるオプションです。このオプションで指定した時間以前にキューテーブルに格納された更新データの詳細を出力します。

このオプションは、YYYYMMDDHHMMSS の形式で指定します。

このオプションを省略すると、99991231235959 が仮定されます。

- **-n 表示データ件数**

-s 及び-t オプションで指定した条件を満たす更新データの最大出力件数を指定します。

このオプションを省略すると、-s 及び-t オプションで指定した条件を満たすすべての更新データを出力されます。

## 規則

- hderesstateO コマンドを実行後、正常終了すると 0 が返されます。異常終了すると 1 が返されます。
- 抽出側 Datareplicator Extension では、hderesstateO コマンドを実行した時点でデータ連動リソースの状態を取得し、標準出力に出力します。このため出力される情報は、実際の抽出側 Datareplicator Extension の状態とは多少ずれが発生することがあります。



番号	説明	出力条件
1.	キューテーブルの有無 exist : キューテーブルあり not exist : キューテーブルなし	条件はありません。
2.	キューテーブル名称	1.が exist の場合。
3.	キューテーブル表名称	
4.	キューテーブルからの取り出し (DEQUEUE) が完了した更新データの保存時間 (単位: 秒)	
5.	キューテーブル内の未抽出メッセージ数	
6.	先頭の未抽出メッセージがキューテーブルに格納された時間	1.が exist で、かつ 5.が 0 以外の場合。
7.	最終の未抽出メッセージがキューテーブルに格納された時間	
8.	データ連動用トリガ数	条件はありません。
9.	データ連動用トリガの詳細 <ul style="list-style-type: none"> <li>先頭 3 カラム目 *の場合はトリガが有効であることを示します。</li> <li>先頭 5~19 カラム目 トリガ名</li> <li>先頭 21 カラム目以降 トリガ定義対象スキーマ名及び表名</li> </ul>	8.が 0 以外の場合。
10.	抽出同期管理テーブルの有無 exist : 抽出同期管理テーブルあり not exist : 抽出同期管理テーブルなし	条件はありません。
11.	抽出同期管理テーブル情報 (抽出情報キューファイルの最終 WRITE 位置)	10.が exist の場合。
12.	出力メッセージ数	-d オプションを指定している場合。
13.	メッセージ ID	12.が 0 以外の場合。
14.	更新順序番号	
15.	メッセージがキューテーブルに格納された時間, トランザクション ID, 及びメッセージの状態  メッセージの状態 'PROCESSED': 既にキューテーブルから取り出されています。 'READY': まだキューテーブルから取り出されていません。 'EXPIRED': キューテーブルに格納できないメッセージのため, 例外キューに格納されています。	
16.	表 ID 及び更新種別  更新種別 INS : INSERT	

番号	説明	出力条件
	UPD : UPDATE (マッピングキー更新なし) UPDM : UPDATE (マッピングキー更新あり) DEL : DELETE	
17.	更新前データ数	12.が0以外で、かつ16.の更新種別がUPDM又はDELの場合。
18.	更新データ順序名及び更新データ長 更新データ長 -1 : ナル値 -1以外 : データ長	17.が0以外の場合。
19.	更新前データ	17.が0以外で、かつ18.が-1以外の場合。
20.	更新後データ数	12.が0以外で、かつ16.の更新種別がINS, UPD, 又はUPDMの場合。
21.	更新データ順序名及び更新データ長 更新データ長 -1 : ナル値 -1以外 : データ長	20.が0以外の場合。
22.	更新後データ	20.が0以外で、かつ21.が-1以外の場合。

## 注

キューテーブルに格納されているメッセージ（更新データ）の詳細は、キューテーブルに格納された時間が古い順に出力されます。ただし、同じ時間にキューテーブルに格納されたメッセージの詳細については、出力順序は保証されません。

# hderesstateS (データ連動リソースの状態の取得)

## 機能

コマンドを実行した時点のデータ連動リソースの状態を標準出力に出力します。出力される情報を、次の表に示します。

データ連動リソース	出力される情報
メッセージデータ型	出力はありません。
キューテーブル	<ul style="list-style-type: none"><li>• キューテーブルの有無</li><li>• キューテーブル名称</li><li>• キューテーブルからの取り出し (DEQUEUE) が完了した更新データの保存時間</li><li>• キューテーブルに格納されているメッセージ (更新データ) の情報<ul style="list-style-type: none"><li>• 未抽出メッセージ数</li><li>• 先頭の未抽出メッセージがキューテーブルに格納された時間*</li><li>• 最終の未抽出メッセージがキューテーブルに格納された時間*</li><li>• キューテーブルに格納されているメッセージの詳細*</li></ul></li></ul>
データ連動用トリガ	<ul style="list-style-type: none"><li>• 全トリガ数</li><li>• 全トリガ名</li><li>• 全トリガの定義対象表一覧</li><li>• 全トリガの状態 (有効/無効)</li></ul>
抽出データ制御表	<ul style="list-style-type: none"><li>• 抽出データ制御表の有無</li><li>• 抽出データ制御表情報 (抽出情報キューファイルの最終 WRITE 位置)</li></ul>

## 注※

-d オプションを指定したときだけ出力します。

## 形式

```
hderesstateS [-D DSN] [-u ユーザID [/ パスワード] ]  
              [-d [-s 表示開始時間] [-t 表示終了時間] [-n 表示データ件数] ]
```

## オプション

- -D DSN  
抽出側の SQL Server の DSN 名を指定します。
- -u ユーザ名 [/ パスワード]  
データ連動リソース管理ユーザのユーザ名とパスワードを指定します。  
このオプションを省略すると、抽出システム定義の sqls\_usr オペランドの値が仮定されます。
- -d  
キューテーブル内に格納されている更新データの詳細を出力します。

- **-s 表示開始時間**

-d オプションを指定したときにだけ指定できるオプションです。このオプションで指定した時間以降にキューテーブルに格納された更新データの詳細を出力します。

このオプションは、YYYYMMDDHHMMSS の形式で指定します。17530101000000 以上の値を指定してください。このオプションを省略すると、17530101000000 が仮定されます。

- **-t 表示終了時間**

-d オプションを指定したときにだけ指定できるオプションです。このオプションで指定した時間以前にキューテーブルに格納された更新データの詳細を出力します。

このオプションは、YYYYMMDDHHMMSS の形式で指定します。ミリ秒には 000 が仮定されます。したがって、20031201074030 と指定した場合、2003-12-01 07:40:30.000 を意味します。この場合、2003-12-01 07:40:30.001 以降に発生した更新情報は検索の範囲外になります。

このオプションを省略すると、99991231235959 が仮定されます。

- **-n 表示データ件数**

-s 及び-t オプションで指定した条件を満たす更新データの最大出力件数を指定します。

このオプションを省略すると、-s 及び-t オプションで指定した条件を満たすすべての更新データを出力されます。

## 規則

- hderesstateS コマンドを実行後、正常終了すると 0 が返されます。異常終了すると 1 が返されます。
- 抽出側 Datareplicator Extension では、hderesstateS コマンドを実行した時点でデータ連動リソースの状態を取得し、標準出力に出力します。このため出力される情報は、実際の抽出側 Datareplicator Extension の状態とは多少ずれが発生することがあります。

## 出力形式

```
*****
**      HiRDB Datareplicator resource status information      **
**      Wed Jun 12 14:10:40 2000                             **
*****

----- Queue table information -----

Queue status : exist .....1.
Queue table name : RPL_EXT_QUEUE_TBL .....2.
Message keep time : 86400 .....3.
Message num : 100 .....4.
Message enqueue time (first) : 2000-09-15 09:01:59 .....5.
Message enqueue time (last) : 2000-09-15 23:31:20 .....6.

----- Trigger information -----

Trigger num : 6 .....7.
RPL_EXT_TRG0001_INS:SCHTT01.TBL1 .....8.
* RPL_EXT_TRG0001_UPD:SCHTT01.TBL1
* RPL_EXT_TRG0001_DEL:SCHTT01.TBL1
* RPL_EXT_TRG0002_INS:SCHTT01.TBL2
* RPL_EXT_TRG0001_UPD:SCHTT01.TBL2
* RPL_EXT_TRG0001_DEL:SCHTT01.TBL2

----- Extraction sync management table information -----

Table status : exist .....9.
Write position : qufile001、 offset = 4096 .....10.
.

----- Message Data into Queue Table -----

Output Data num(30) .....11.
Correlation ID(0000000000000000150) .....12.
Enqueue Date(2000-09-15 09:01:59) Status(PROCESSED) .....13.
Table ID(1234) Update kind(UPDM) .....14.
Before data num(1) .....15.
Data0001 : len(8) .....16.
10001000 .....17.
After data num(3) .....18.
Data0001 : len(60) .....19.
123456789012345678901234567890 .....20.
1234567890
Data0002 : len(-1)
Data0003 : len(19)
2001-01-01 01:01:01
Correlation ID(0000000000000000151)
Enqueue Date(2000-09-15 09:01:59) Status(READY)
Table ID(1236) Update kind(INS)
Before data num(0)
After data num(3)
Data0001 : len(60)
123456789012345678901234567890
1234567890
Data0002 : len(-1)
Data0003 : len(19)
2001-01-01 01:01:01
:
```

## 説明

出力形式の説明と出力条件を次に示します。

番号	説明	出力条件
1.	キューテーブルの有無 exist : キューテーブルあり not exist : キューテーブルなし	条件はありません。
2.	キューテーブル名称	1.が exist の場合。
3.	キューテーブルからの取り出し (DEQUEUE) が完了した更新データの保存時間 (単位: 秒)	
4.	キューテーブル内の未抽出メッセージ数	
5.	先頭の未抽出メッセージがキューテーブルに格納された時間	
6.	最終の未抽出メッセージがキューテーブルに格納された時間	1.が exist で、かつ 4.が 0 以外の場合。
7.	データ連動用トリガ数	
8.	データ連動用トリガの詳細 <ul style="list-style-type: none"> <li>先頭 3 カラム目 *の場合はトリガが有効であることを示します。</li> <li>先頭 5~23 カラム目 「トリガ名+通番+トリガ種別」を表示します。 通番はトリガ種別ごとに振られる 4 けたの識別番号です。 トリガ種別には次の 3 種類があります。 <ul style="list-style-type: none"> <li>INS : トリガの投入</li> <li>UPD : トリガの更新</li> <li>DEL : トリガの削除</li> </ul> </li> <li>先頭 25 カラム目以降 トリガ定義対象スキーマ名及び表名</li> </ul>	7.が 0 以外の場合。
9	抽出データ制御表の有無 exist : 抽出データ制御表あり not exist : 抽出データ制御表なし	条件はありません。
10.	抽出データ制御表の情報 (抽出情報キューファイルの最終 WRITE 位置)	9.が exist の場合。
11.	出力メッセージ数	-d オプションを指定している場合。
12.	メッセージの連続番号	11.が 0 以外の場合。
13.	メッセージがキューテーブルに格納された時間, トランザクション ID, 及びメッセージの状態 <b>メッセージの状態</b> 'PROCESSED' : 既にキューテーブルから取り出されています。 'READY' : まだキューテーブルから取り出されていません。	
14.	表 ID 及び更新種別 <b>更新種別</b> INS : INSERT	

番号	説明	出力条件
	UPD : UPDATE (マッピングキー更新なし) UPDM : UPDATE (マッピングキー更新あり) DEL : DELETE	
15.	更新前データ数	11.が0以外で、かつ14.の更新種別がUPDM又はDELの場合。
16.	更新データ順序名(4けたの連続番号)及び更新データ長 <b>更新データ長</b> -1 : ナル値 -1以外 : データ長	15.が0以外の場合。
17.	更新前データ	15.が0以外で、かつ16.が-1以外の場合。
18.	更新後データ数	11.が0以外で、かつ14.の更新種別がINS, UPD, 又はUPDMの場合。
19.	更新データ順序名及び更新データ長 <b>更新データ長</b> -1 : ナル値 -1以外 : データ長	18.が0以外の場合。
20.	更新後データ	18.が0以外で、かつ19.が-1以外の場合。

# 8

## 障害時の対処

この章では、抽出側 Datareplicator Extension 又は反映側 Datareplicator Extension を運用中に障害が発生した場合の対処について説明します。

## 8.1 抽出側 Datareplicator Extension の障害時の対処

他社 DBMS とデータ連動する場合の、抽出側 Datareplicator Extension で障害が起こったときの対処方法について説明します。

### 8.1.1 抽出側システムでエラーが発生した場合の対処（Oracle の場合）

抽出側システムが Oracle の場合、抽出側でエラーが発生した場合の障害対策について説明します。ここでは、次の場合について説明します。

- 抽出定義プリプロセスファイルの作成時に、トリガ作成処理でエラーが発生した場合
- 抽出側 DB のトリガ実行処理でエラーが発生した場合
- 抽出側 DB がダウンした場合
- 抽出プロセスでエラーが発生した場合
- 抽出側 DB と反映側 DB に不整合が発生した場合

#### (1) 抽出定義プリプロセスファイルの作成時に、トリガ作成処理でエラーが発生した場合

hdeprepO コマンドで抽出定義プリプロセスファイルを作成するときに、トリガ作成処理でエラーが発生した場合の、対処方法を次に示します。

1. KFRB00060-E メッセージの詳細情報に出力されている表名を確認します。
2. hderesstateO コマンドを実行して、1.で確認した表に該当するトリガ名を取得します。
3. データ連動リソース管理ユーザとして Oracle に接続し、SQL\*Plus で次の SQL を実行します。

```
alter trigger トリガ名 compile ;
```

この SQL を実行すると、コンパイルエラーが発生します。

4. SQL\*Plus で次の SQL を実行します。

```
show errors ;
```

このコマンドを実行すると、トリガ作成処理で発生したエラーの詳細要因メッセージが出力されるので、このメッセージに従って対処してください。

#### (2) 抽出側 DB のトリガ実行処理でエラーが発生した場合

抽出側 DB が Oracle の場合は、更新 AP による更新処理の延長で、トリガによる更新情報の取得及びアドバンストキューへの格納が実行されます。

このため、トリガ実行処理でエラーが発生した場合は、本来の DB に対する更新処理も無効となり、データ連動だけではなく本来の業務に対しても影響を及ぼします。

このエラーに対する対処として、次の二つの方法があります。

- データ連動を打ち切って、本来の業務を優先させる方法
- データ連動の継続を優先させる方法

### (a) データ連動を打ち切って本来の業務を優先させる方法

データ連動を打ち切って本来の業務を優先させる場合の対処方法を、次に示します。

1. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を停止します。
2. DROP USER…CASCADE を実行して、データ連動リソースを削除します。
3. 更新 AP を再実行します。

なお、この方法で対処した場合は、データ連動が打ち切られて、抽出側 DB と反映側 DB に不整合が発生します。抽出側 DB と反映側 DB に不整合が発生した場合の対処については、「(5)抽出側 DB と反映側 DB に不整合が発生した場合」を参照してください。

### (b) データ連動の継続を優先させる方法

アドバンスキューが満杯であるなど、回復の見込みがあるエラーであり、かつ回復まで業務を中断できる場合は、エラーを回復した後に更新 AP を再実行してください。

## (3) 抽出側 DB がダウンした場合

抽出側 DB が Oracle の場合、抽出側 DB がダウンしたときは抽出側 Datareplicator Extension の抽出プロセスも DEQUEUE の実行時にエラー終了します。

抽出側 DB がダウンした場合の、対処方法を次に示します。

1. 抽出対象 Oracle がダウンした要因を取り除きます。
2. 抽出対象 Oracle を起動します。
3. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を起動します。

なお、抽出処理の同期を制御することで、抽出処理の整合性を保証します。

## (4) 抽出プロセスでエラーが発生した場合

抽出プロセスで発生するエラーは、次の二つに分けられます。

エラーの種類	エラーの内容
データ連動を継続できるエラー	<ul style="list-style-type: none"> <li>メモリ不足や抽出側システムが起動していないなど、ユーザがエラー要因を解消できるエラー</li> <li>例外データを検知したなど、エラーが発生しても Datareplicator Extension にデータ連動を継続するための機能があるエラー</li> </ul>
データ連動を継続できないエラー	不正なデータを検知したなど、ユーザがエラー要因を解消できないエラー

データ連動を継続できるエラーに対しては、エラー要因を解消した後に抽出側 Datareplicator 及び抽出側 Datareplicator Extension を再起動することで、データ連動を再開できます。また、抽出処理の同期を制御することで、抽出処理の整合性を保証します。

データ連動を継続できないエラーに対しては、抽出側 DB と反映側 DB に不整合が発生する場合があります。抽出側 DB と反映側 DB に不整合が発生した場合の対処については、「(5)抽出側 DB と反映側 DB に不整合が発生した場合」を参照してください。

## (5) 抽出側 DB と反映側 DB に不整合が発生した場合

抽出側システムでの障害や不正な運用などによって、抽出側 DB と反映側 DB に不整合が発生した場合の対処方法を次に示します。

1. HiRDB Dataextractor で、抽出側 DB と反映側 DB との不整合を修正します。
2. 反映側システムのデータ連動の環境を初期化します。
3. 抽出側システムのデータ連動の環境を初期化します。
4. 抽出定義プリプロセスファイルを作成します。
5. 反映側 Datareplicator 及び反映側 Datareplicator Extension を起動します。
6. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を起動します。

### 8.1.2 抽出側システムでエラーが発生した場合の対処 (SQL Server の場合)

抽出側システムが SQL Server の場合、抽出側でエラーが発生した場合の障害対策について説明します。ここでは、次の場合について説明します。

- 抽出側 DB のトリガ実行処理でエラーが発生した場合
- 抽出側 DB がダウンした場合
- 抽出プロセスでエラーが発生した場合
- ホストマシンがダウンした場合
- 抽出側 DB と反映側 DB に不整合が発生した場合

## (1) 抽出側 DB のトリガ実行処理でエラーが発生した場合

トリガ実行処理エラーによって更新 AP でエラーが発生した場合、SQL Server では最後にコミットした時点からトリガ実行処理エラーが発生した時点までのトランザクションがすべてロールバックされます。

抽出側 Datareplicator では、コミットされたデータしか抽出されないため、未保存のトランザクションを再度入力する必要があります。

トリガ実行処理エラーが発生すると、SQL Server からは更新 AP にエラーが返されます。ただし、エラーメッセージの表示と内容は更新 AP エラー処理に依存します。

このエラーへの対処には次の二つの方法があります。

- データ連動を打ち切って本来の業務を優先させる方法
- データ連動の継続を優先させる方法

### (a) データ連動を打ち切って本来の業務を優先させる方法

データ連動を打ち切って本来の業務を優先させる場合の対処方法を、次に示します。

1. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を停止します。
2. DROP TRIGGER を実行して、データ連動トリガを削除します。
3. 更新 AP を再実行します。

### (b) データ連動の継続を優先させる方法

SQL Server から更新 AP に返されるエラーに基づいて問題を解決した後、更新 AP を再実行してください。

## (2) 抽出側 DB がダウンした場合

抽出側 DB が SQL Server の場合、抽出側 DB がダウンしたときは抽出が終了します。ただし、抽出側 Datareplicator のほかの処理はそのまま実行します。

抽出側 DB がダウンした場合の、対処方法を次に示します。

1. 抽出対象 SQL Server がダウンした要因を取り除きます。
2. 抽出対象 SQL Server を起動します。
3. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を起動します。

抽出側 Datareplicator を起動すると、ダウン時に終了した抽出プロセスだけ起動されます。

## (3) 抽出プロセスでエラーが発生した場合

抽出プロセスで発生するエラーは、次の二つに分けられます。

エラーの種類	エラーの内容
データ連動を継続できるエラー	<ul style="list-style-type: none"> <li>メモリ不足や抽出側システムが起動していないなど、ユーザがエラー要因を解消できるエラー</li> <li>例外データを検知したなど、エラーが発生しても Datareplicator Extension にデータ連動を継続するための機能があるエラー</li> </ul>
データ連動を継続できないエラー	不正なデータを検知したなど、ユーザがエラー要因を解消できないエラー

データ連動を継続できるエラーに対しては、エラー要因を解消した後に抽出側 Datareplicator 及び抽出側 Datareplicator Extension を再起動することで、データ連動を再開できます。また、抽出処理の同期を制御することで、抽出処理の整合性を保証します。

データ連動を継続できないエラーに対しては、抽出側 Datareplicator 及び抽出側 Datareplicator Extension を初期起動する必要があります。

#### (4) ホストマシンがダウンした場合

抽出側システムのホストマシンがダウンした場合の対処方法を、次に示します。

1. 抽出対象 SQL Server を起動します。
2. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を起動します。

#### (5) 抽出側 DB と反映側 DB に不整合が発生した場合

抽出側システムでの障害や不正な運用などによって、抽出側 DB と反映側 DB に不整合が発生した場合の対処方法を次に示します。

1. HiRDB Dataextractor で、抽出側 DB と反映側 DB との不整合を修正します。
2. 抽出側システムのデータ連動の環境を初期化します。
3. 抽出定義プリプロセスファイルを作成します。
4. 反映側 Datareplicator 及び反映側 Datareplicator Extension を初期起動します。
5. 抽出側 Datareplicator 及び抽出側 Datareplicator Extension を起動します。

## 8.2 反映側 Datareplicator Extension の障害時の対処

他社 DBMS とデータ連動する場合の、反映側 Datareplicator Extension の障害時の対処方法は、HiRDB システム同士のデータ連動と同様です。

反映側 Datareplicator Extension の障害時の対処については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

なお、エラーメッセージについては、抽出時及び反映時共に同じメッセージが出力されます。Oracle 又は SQL Server について、特に記載がない場合、次に示すように読み替えてください。

メッセージ中の用語	読み替える用語	
	Oracle の場合	SQL Server の場合
HiRDB	Oracle	SQL Server
HiRDB システムログ	アドバンストキュー	データベースキュー
hdestart	hdestartO	hdestartS
hdestop	hdestopO	hdestopS
hdestate	hdestateO	hdestateS
hdeprep	hdeprepO	hdeprepS
hdeevent	hdeeventO	hdeeventS

## 8.3 データ連動回復機能

データ連動回復機能とは、次のどちらかの原因によって通常のデータ連動が停止した場合に、障害で消失した未反映の更新情報を再抽出し、その情報を基に抽出情報キューファイルを回復して反映側に送信することで、データ連動の整合性を回復できるようにする機能です。

- データ連動の停止 (pdrplstop コマンド) や初期開始 (hdestart -i コマンド) を誤って実行した
- Datareplicator Extension が管理するファイル (データ連動用連絡ファイル, 抽出情報キューファイル, サーバステータスファイルなど) に障害が発生した

データ連動回復機能には、次の二つの方法があります。

- システムログファイルによる回復
- アンロードログファイルによる回復

ただし、Datareplicator Extension でシステムログファイルによる回復を実行できるのは、次の前提条件を満たした場合だけです。

- 抽出側システムが Datareplicator である
- 反映側システムが Datareplicator Extension である
- 抽出側システムと反映側システムのバージョンが共に 08-04 以降である

データ連動回復機能の詳細については、マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

アンロードログファイルによるデータ連動回復を使える Datareplicator Extension のバージョンと製品の組み合わせを次の表に示します。

表 8-1 アンロードログファイルによるデータ連動回復を使える Datareplicator Extension のバージョンと製品の組み合わせ

反映側システム		抽出側システム		
		Datareplicator <sup>※1</sup>		
		06-01 以前	06-01-/A 以降 <sup>※2</sup>	07-00 以降, 及び 08-00 以降
Datareplicator Extension	06-03-/D 以降	×	○	○
	07-00 以降, 及び 08-00 以降	×	○	○
	上記以外	×	×	×

(凡例)

○：アンロードログファイルによるデータ連動回復を使えます。

×：アンロードログファイルによるデータ連動回復を使えません。

注

「以降」とは、該当形名でのバージョンを示します。

注※1

抽出側システムに Datareplicator を使用している場合だけ、データ連動回復機能を使えます。抽出側システムに Datareplicator Extension を使用している場合は、データ連動回復機能を使えません。HiRDB Dataextractor 又は XDM/XT を使って回復してください。

注※2

logmrg コマンドの入力対象となる HiRDB のファイルがラージファイルの場合、データ連動回復機能が有効な HiRDB Datareplicator のバージョンは、06-02 以降になります。

# 付録

## 付録 A Datareplicator Extension 定義の予約語

Datareplicator Extension 定義の予約語を次の表に示します。ここに示す予約語は、大文字／小文字、又は大文字と小文字の混在のすべてが予約語になります。さらに、Datareplicator 抽出定義、Datareplicator 反映定義、更新情報定義それぞれのオペランド名も予約語となります。

これらの文字列を、抽出側 Datareplicator Extension の定義、反映側 Datareplicator Extension の定義、及び更新情報定義で使うには、文字列を「"」（ダブルクォーテーション）で囲んで指定してください。

表 A-1 Datareplicator Extension の予約語の一覧

予約語	抽出側 Datareplicator Extension の定義の予約語	反映側 Datareplicator Extension の定義の予約語	更新情報定義の予約語
<A>	adt	○	○
	and	○	—
	attr	—	○
<B>	binary	—	○
	by	○	—
<C>	char* <sup>1</sup>	○	—
	comp	—	○
	const	—	○
	construct	○	○
<D>	date* <sup>2</sup>	○	—
	divide	—	○
<E>	extract	○	○
	extract_date	—	○
	extract_time	—	○
<F>	field	—	○
	filetype	—	○
	flike	○	—
	float	—	○
	format	—	○
	from	○	○
<G>	group	—	○
<H>	hash	—	○

予約語	抽出側 Datareplicator Extension の定義の予約語	反映側 Datareplicator Extension の定義の予約語	更新情報定義の予約語
	having	—	○
	hdsdefserv <sup>*3</sup>	—	○
	hdsmain <sup>*3</sup>	—	○
	hdsreflect <sup>*3</sup>	—	○
	hdssqle <sup>*3</sup>	—	○
	hdstcpmst <sup>*3</sup>	—	○
<I>	in	○	○
	ins	—	—
	into	—	○
	is	—	○
	ivl	—	—
<K>	key	○	—
<L>	lib	○	○
	load	—	○
<N>	nchar	—	—
	name	—	○
	nocodecnv	—	○
	not	—	○
	null	—	○
<O>	or	○	—
	other	—	○
	othergrp	—	○
<P>	pack	—	—
	packns	—	—
	pdm	—	—
	position	—	—
<Q>	que	—	—
<R>	reflect_date	—	○
	reflect_kind	—	○
	reflect_time	—	○

予約語		抽出側 Datareplicator Extension の定義の予約語	反映側 Datareplicator Extension の定義の予約語	更新情報定義の予約語
	reptype	○	○	○
	restruct	—	—	○
<S>	send	○	—	—
	seq_no	—	○	—
	sqlconvopt1	—	○	—
	sqlconvopt2	—	○	—
<T>	through	○	○	○
	time*2	○	—	—
	timestamp	—	○	—
	to	○	○	○
	trngroup	—	○	—
<U>	ukey	○	—	○
	unpack	—	—	○
	unpackns	—	—	○
	uocname	—	—	○
	uocxxx*4	—	○	—
	upd	—	—	○
	update	—	—	○
<W>	where	○	—	—

(凡例)

- ：予約語であることを示します。
- ：予約語に該当しないことを示します。

注

注※3 で説明している予約語以外は、大文字、小文字と、それらの組み合わせがすべて予約語です。例えば by の場合、by, bY, By, 及び BY が予約語です。

注※1

Oracle とデータ連動する場合は、抽出側 Datareplicator Extension 定義の予約語ですが、HiRDB システム同士でデータ連動する場合は、更新情報定義の予約語です。

注※2

Datareplicator Extension 固有の予約語です。

注※3

すべてが小文字の場合だけ予約語です。

注※4

*xxx* は、3けたの符号なし整数を表します。

## 付録 B プロセスファイルの作成手順及び作成例

Oracle を使用する場合のプロセスファイルの作成手順及び作成例について説明します。

### 付録 B.1 プロセスファイルの作成手順

プロセスファイルを作成する手順を次に示します。

1. サンプルメイクファイルの確認
2. 環境変数の設定
3. プロセスファイル作成シェルの実行

#### (1) サンプルメイクファイルの確認

プロセスファイルを作成する前は、Oracle が提供するサンプルメイクファイルに対する一時的制限事項や注意事項がないか、Oracle のリリースノートを確認し、修正が必要であればリリースノートに従って修正してください。

サンプルメイクファイル名、及びサンプルメイクファイルが格納されているディレクトリを次に示します。

Oracle のバージョン	サンプルメイクファイル名	サンプルメイクファイルが格納されているディレクトリ
Oracle11g	demo_rdbms64.mk	\$ORACLE_HOME/rdbms/demo
Oracle12c 以降	demo_rdbms.mk	\$ORACLE_HOME/rdbms/demo

#### (2) 環境変数の設定

プロセスファイルは、root ユーザが作成します。root ユーザがプロセスファイルを作成するために、設定が必要な環境変数を次に示します。

環境変数	設定内容
ORACLE_HOME	抽出対象 Oracle のホームディレクトリを指定します。
PATH	次に示すパスを指定します。 <ul style="list-style-type: none"><li>・ make コマンドの実行に必要なパス</li><li>・ コンパイラの実行に必要なパス※</li></ul>
LANG	「C」を設定します。

注※

コンパイラ実行に必要なパスは、Oracle の取扱説明書に従って設定してください。

### (3) プロセスファイル作成シェルの実行

次に示す手順に従って、プロセスファイル作成シェルを実行してください。

```
# cd /opt/hirdbds/make
# hdssetup0
```

なお、次に示す手順に従ってプロセスファイル作成シェルの内容を修正しておく必要があります（スーパーユーザで修正してください）。

```
(省略)
#####
# Set Default Values
#####
INSTPATH="/opt/hirdbds"
SSFIX="s1"
MAKE_DEMO="$ORACLE_HOME/rdbms/demo/demo_rdbms.mk"※
ORA_VER=8
PRODUCT_NAME="HiRDB Datareplicator Extension"
OS_KIND=`uname`
(省略)
```

注※

「(1)サンプルメイクファイルの確認」に示す、サンプルメイクファイルに変更してください。

プロセスファイル作成シェルの実行例については、「付録 B.2 プロセスファイル作成シェルの実行例」を参照してください。

プロセスファイル作成シェルの実行中にエラーが発生した場合は、エラーメッセージに従って要因を対処した後に、シェルを再実行してください。

## 付録 B.2 プロセスファイル作成シェルの実行例

プロセスファイル作成シェルの実行例を次に示します。

```
*****
HiRDB DataReplicator Extension Setup
*****
[HiRDB DataReplicator] : =====
[HiRDB DataReplicator] : It is carried out in the following option.
[HiRDB DataReplicator] : /opt/hirdbds
[HiRDB DataReplicator] : so
[HiRDB DataReplicator] : /home/oracle/Oracle/product/12.2/rdbms/demo/demo_rdbms.mk
[HiRDB DataReplicator] : =====
Choose Component (hde : 1, hds : 2) : 1※
                                     :
                                     :
[ ロード作成 (make) の実行結果 ]
                                     :
                                     :
```

```
[HiRDB DataReplicator] : =====  
[HiRDB DataReplicator] : HiRDB DataReplicator Extension Setup Completed..  
[HiRDB DataReplicator] : =====
```

注※

作成するプロセスファイルが、Oracle から更新内容を抽出するときに使うファイルか、Oracle へ更新内容を反映するときに使うファイルかを選択します。

- 1 : Oracle から更新内容を抽出する場合に使用するファイル
- 2 : Oracle へ更新内容を反映する場合に使用するファイル

# 索引

## 記号

/パスワード [抽出システム定義] 101

## 数字

1 トランザクション当たりの最大更新情報数 (Oracle) [抽出環境定義] 93

1 トランザクション当たりの最大更新情報数 (SQL Server) [抽出環境定義] 105

## C

const 初期値 [反映定義] 118

cvt\_datakind (Oracle) [抽出環境定義] 92

cvt\_datakind (SQL Server) [抽出環境定義] 104

## D

Datareplicator Extension 22

Datareplicator Extension 定義の予約語 162

Datareplicator Extension のインストール (UNIX) 33

Datareplicator Extension のインストール (Windows) 42

Datareplicator Extension の目的 22

Datareplicator 管理者の登録 (UNIX) 41

Datareplicator 管理者の登録 (Windows) 55

datasource [反映システム定義] 112

dbkind [反映システム定義] 112

decimal\_max\_precision (Oracle) [抽出システム定義] 90

decimal\_max\_precision (SQL Server) [抽出システム定義] 102

DEQUEUE 58

DEQUEUE 後のアドバンストキュー内更新情報保存時間 (Oracle) [抽出システム定義] 89

DEQUEUE 後のキューテーブル内更新情報保存時間 (SQL Server) [抽出システム定義] 102

DSN 名 (SQL Server) [抽出システム定義] 101

## E

EBCDIK/KEIS から EUC への文字コードの変換規則 81

ENQUEUE 58

exterrlevel (Oracle) [抽出環境定義] 92

exterrlevel (SQL Server) [抽出環境定義] 104

## H

hdeprepX 58

hderesstateO [コマンドの文法] 142

hderesstateS [コマンドの文法] 147

hdestartO 122

hdestartS 127

hdestartX 58

hdestopX 58

hdsstart 133

hdsstop 133

HiRDB から ODBC ドライバを使用して SQL Server へデータ連動する場合のソフトウェア構成 27

HiRDB から Oracle へデータ連動する場合のソフトウェア構成 26

## M

mapping\_key\_check [反映環境定義] 116

## N

node\_shlibpath (Oracle) [抽出システム定義] 91

## O

odbcusr [反映システム定義] 113

ODBC 環境の設定 (Windows) 50

ODBC 接続認可識別子 [反映システム定義] 113

oracle\_aq\_storage (Oracle) [抽出システム定義] 89

oracle\_maxextcol (Oracle) [抽出システム定義] 88

oracle\_msgkeepitime (Oracle) [抽出システム定義] 89

oracle\_sortidx\_storage [抽出システム定義] 89  
oracle\_sortidx [抽出システム定義] 89  
oracle\_sorttbl\_storage (Oracle) [抽出システム定義] 89  
oracleusr (Oracle) [抽出システム定義] 88  
oracleusr [反映システム定義] 112  
Oracle から HiRDB ヘデータ連動する場合のソフトウェア構成 25  
Oracle 接続認可識別子 [反映システム定義] 112  
Oracle 抽出サービス 44

## S

skip\_sqlcode [反映環境定義] 116  
sqls\_dsn (SQL Server) [抽出システム定義] 101  
sqls\_maxextcol (SQL Server) [抽出システム定義] 101  
sqls\_msgkeepime (SQL Server) [抽出システム定義] 102  
sqls\_qtbl\_filegroup (SQL Server) [抽出システム定義] 102  
sqls\_qtbl\_idx (SQL Server) [抽出システム定義] 102  
sqls\_qtblidx\_filegroup (SQL Server) [抽出システム定義] 102  
sqls\_usr (SQL Server) [抽出システム定義] 101  
SQL Server から ODBC ドライバを使用して HiRDB ヘデータ連動する場合のソフトウェア構成 26  
SQL Server 接続ユーザ ID [抽出システム定義] 101  
SQL Server 抽出サービス 44

## T

tran\_datanum (Oracle) [抽出環境定義] 93  
tran\_datanum (SQL Server) [抽出環境定義] 105

## あ

アイコン [Windows] 44  
アドバンストキュー作成時の記憶域パラメタ (Oracle) [抽出システム定義] 89  
アンインストール [UNIX] 33  
アンインストール [Windows] 45

## い

イベント機能 69  
インストール時に登録されるサービス [Windows] 44  
インストールの実行 [UNIX] 33  
インストールの実行 [Windows] 42  
インストール前の準備 [UNIX] 33  
インストール前の準備 [Windows] 42

## か

環境の設定 30  
環境変数の設定 (UNIX) 35  
環境変数の設定 (Windows) 47  
関連する製品 31

## き

キューテーブル 58  
キューテーブルインデックスの FILEGROUP 名 (SQL Server) [抽出システム定義] 102  
キューテーブルの FILEGROUP 名 (SQL Server) [抽出システム定義] 102

## こ

更新情報及びトランザクション情報の出力 61  
更新情報の取り出し 59  
更新情報フィールド定義 117  
コマンドの一覧 140

## さ

サービス [Windows] 44

## し

時系列情報の取得 67  
時系列情報表の作成 81  
時系列情報を取得する表の定義 81  
システム設計 70  
障害時の対処 152

## せ

選択条件列の属性と定数との関連 (Oracle) 97

## そ

- 送信 58
- 送信環境定義 (Oracle) [定義の文法] 93
- 送信環境定義 (SQL Server) [定義の文法] 105
- 送信滞留時間 68
- ソフトウェア構成 25

## た

- 滞留監視機能 68
- 滞留時間 68

## ち

- 遅延開始 66
- 抽出 58
- 抽出可能最大列数 (Oracle) [抽出システム定義] 88
- 抽出可能最大列数 (SQL Server) [抽出システム定義] 101
- 抽出側 Datareplicator Extension 23
- 抽出側 Datareplicator Extension の運用 (Oracle の場合) 124
- 抽出側 Datareplicator Extension の運用 (SQL Server の場合) 129
- 抽出側 Datareplicator Extension の環境変数の設定 (UNIX) 35
- 抽出側 Datareplicator Extension の環境変数の設定 (Windows) 47
- 抽出側 Datareplicator Extension の起動と終了 (Oracle の場合) 122
- 抽出側 Datareplicator Extension の起動と終了 (SQL Server の場合) 127
- 抽出側 Datareplicator Extension の障害時の対処 153
- 抽出側 Datareplicator Extension の抽出処理 57
- 抽出側 Datareplicator Extension を使用する場合の定義 (Oracle の場合) 85
- 抽出側 Datareplicator Extension を使用する場合の定義 (SQL Server の場合) 98
- 抽出側 DB 23
- 抽出側 DB と反映側 DB との対応の設計 71
- 抽出側システム 23

- 抽出側システムでエラーが発生した場合の対処 (Oracle の場合) 153
- 抽出側システムでエラーが発生した場合の対処 (SQL Server の場合) 155
- 抽出間隔の制御 62
- 抽出環境定義 (Oracle) [定義の文法] 91
- 抽出環境定義 (SQL Server) [定義の文法] 103
- 抽出システム定義 (Oracle) [定義の文法] 85
- 抽出システム定義 (SQL Server) [定義の文法] 98
- 抽出順序管理テーブルインデクス記憶域パラメタ (Oracle) [抽出システム定義] 89
- 抽出順序管理テーブル記憶域パラメタ (Oracle) [抽出システム定義] 89
- 抽出情報キューファイル 58
- 抽出処理 57
- 抽出処理の開始タイミング 58
- 抽出処理のタイミング 58
- 抽出処理の停止タイミング 58
- 抽出定義 (Oracle) [定義の文法] 94
- 抽出定義 (SQL Server) [定義の文法] 106
- 抽出データ型 (Oracle) [抽出定義] 95
- 抽出データ型 (SQL Server) [抽出定義] 107

## つ

- 通信環境の設定 (UNIX) 37
- 通信環境の設定 (Windows) 49

## て

- ディレクトリ構成 (UNIX) 34
- ディレクトリ構成 (Windows) 46
- データ型の変換 59
- データソース名 [反映システム定義] 112
- データ連動回復機能 159
- データ連動機能 56
- データ連動システムの組み合わせ 24
- データ連動システムの構築手順 29
- データ連動システムのソフトウェア構成 25
- データ連動について 24
- データ連動リソース 39, 52
- データ連動リソース管理ユーザの登録 (UNIX) 39

データ連動リソース管理ユーザの登録 (Windows) 52

データ連動リソース管理ユーザ名 (Oracle) [抽出システム定義] 88

データ連動リソースの作成 62

データ連動リソースの状態の取得 [コマンドの文法] 142, 147

## と

トランザクション単位反映方式 66

トリガ 58

## は

パスワード (Oracle) [抽出システム定義] 88

パスワード [反映システム定義] 112, 113

反映側 Datareplicator Extension 23

反映側 Datareplicator Extension の環境変数の設定 (UNIX) 36

反映側 Datareplicator Extension の環境変数の設定 (Windows) 48

反映側 Datareplicator Extension の起動と終了 133

反映側 Datareplicator Extension の障害時の対処 158

反映側 Datareplicator Extension の反映処理 66

反映側 Datareplicator Extension を使用する場合の定義 110

反映側 DB 23

反映側システム 23

反映環境定義 [定義の文法] 113

反映グループ定義 119

反映システム定義 [定義の文法] 110

反映処理 66

反映処理の開始タイミング 66

反映処理のタイミング 66

反映処理の停止タイミング 67

反映処理の方式 66

反映滞留時間 68

反映定義 [定義の文法] 117

反映表定義 118

## ひ

表単位反映方式 66

## ふ

ファイルの二重化を使うときの運用 134

プロセスファイルの作成 (UNIX) 38

プロセスファイルの作成手順及び作成例 166

## ま

マルチ FES 機能に対応した反映方式 66, 133

## も

文字コードの種類 81

文字コードの対応の設計 81

文字コードの変換 81

## ゆ

ユーザ権限の確認 [UNIX] 33

ユーザ権限の確認 [Windows] 42

## よ

予約語 162

## ら

ライブラリパス (Oracle) [抽出システム定義] 91

## れ

例外データ 59