

スケーラブルデータベースサーバ

# HiRDB Version 8 バッチ高速化機能

解説・手引・操作書

3020-6-368-20

## マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，  
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参  
照ください。

## 対象製品

適用 OS : HP-UX 11.0 , HP-UX 11i , HP-UX 11i V2(PA-RISC)

P-1B62-1581 HiRDB/Single Server Version 8(64) 08-05

P-1B62-1781 HiRDB/Parallel Server Version 8(64) 08-05

P-F1B62-1181A HiRDB Accelerator Version 8 08-03

適用 OS : HP-UX 11i V2(IPF) , HP-UX 11i V3(IPF)

P-1J62-1581 HiRDB/Single Server Version 8(64) 08-05

P-1J62-1781 HiRDB/Parallel Server Version 8(64) 08-05

P-F1J62-1181A HiRDB Accelerator Version 8 08-03

適用 OS : Solaris 8 , Solaris 9 , Solaris 10

P-9D62-1581 HiRDB/Single Server Version 8(64) 08-05

P-9D62-1781 HiRDB/Parallel Server Version 8(64) 08-05

P-F9D62-1181A HiRDB Accelerator Version 8 08-03

適用 OS : AIX 5L V5.1 , AIX 5L V5.2 , AIX 5L V5.3 , AIX V6.1

P-1M62-1581 HiRDB/Single Server Version 8(64) 08-05

P-1M62-1781 HiRDB/Parallel Server Version 8(64) 08-05

P-F1M62-1181A HiRDB Accelerator Version 8 08-03

適用 OS: Red Hat Enterprise Linux AS 3(AMD64 & Intel EM64T) , Red Hat Enterprise Linux AS 4(AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4(AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , Red Hat Enterprise Linux 5 (AMD/Intel 64)

P-9W62-1183 HiRDB/Single Server Version 8(64) 08-05

P-9W62-1383 HiRDB/Parallel Server Version 8(64) 08-05

P-F9S62-1181A HiRDB Accelerator Version 8 08-03

注 動作環境としては、Intel EM64T にだけ対応しています。

適用 OS : Red Hat Enterprise Linux AS 3(IPF) , Red Hat Enterprise Linux AS 4(IPF) , Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium) , Red Hat Enterprise Linux 5 (Intel Itanium)

P-9V62-1183 HiRDB/Single Server Version 8(64) 08-05

P-9V62-1383 HiRDB/Parallel Server Version 8(64) 08-05

P-F9V62-1181A HiRDB Accelerator Version 8 08-03

適用 OS : Windows XP x64 Edition , Windows Server 2003 x64 Editions , Windows Server 2008 , Windows Vista Business , Windows Vista Enterprise , Windows Vista Ultimate

P-2962-7184 HiRDB/Single Server Version 8(64) 08-05

P-2962-7384 HiRDB/Parallel Server Version 8(64) 08-05

P-2462-7P84 HiRDB Accelerator Version 8 08-03

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。

これらの製品は、ISO9001 および TickIT の認証を受けた品質マネジメントシステムで開発されました。

## 輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

## 商標類

ActiveX は、米国およびその他の国における米国 Microsoft Corp. の商標です。

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

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

BEA WebLogic Server は、BEA Systems, Inc. の登録商標です。

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

DataStage, MetaBroker, MetaStage および QualityStage は、IBM Corporation の商標です。

DB2 は、米国における米国 International Business Machines Corp. の登録商標です。

日本国内においての DCE の商標使用権は、OSF よりサブライセンスされています。

DNCWARE, ClusterPerfect は、東芝ソリューション株式会社の商標です。

gzip は、米国 FSF(Free Software Foundation) が配布しているソフトウェアです。

HACMP/6000 は、米国における米国 International Business Machines Corp. の商標です。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

IBM は、米国およびその他の国における International Business Machines Corporation の商標です。

Intel は、Intel Corporation の会社名です。

Itanium は、アメリカ合衆国および他の国におけるインテル コーポレーションまたはその子会社の登録商標です。

Java 及びすべての Java 関連の商標及びロゴは、米国及びその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

JBuilder は、Borland Software Corporation の米国およびその他の国における商標です。

JDK は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

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、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Access は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Office Excel は、米国 Microsoft Corporation の商品名称です。

Motif は、Open Software Foundation, Inc. の商標です。

MS-DOS は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

NLS は、米国 Hewlett-Packard Company の商品名称です。

ODBC は、米国 Microsoft Corp. が提唱するデータベースアクセス機構です。

OLE は、米国 Microsoft Corp. が開発したソフトウェア名称です。

ORACLE は、米国 Oracle Corporation の登録商標です。

Oracle は、米国 Oracle Corporation 及びその子会社、関連会社の登録商標です。

Oracle 10g は、米国 Oracle Corporation の商標です。

Oracle8i は、米国 Oracle Corporation の商標です。

Oracle9i は、米国 Oracle Corporation の商標です。

OS/390 は、米国およびその他の国における International Business Machines Corporation の商標です。

PA-RISC は、米国 Hewlett-Packard Company の商標です。

POSIX は、the Institute of Electrical and Electronics Engineers, Inc. (IEEE) で制定された標準仕様です。

PowerBuilder は、米国法人 Sybase, Inc. の登録商標です。

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

RISC System/6000 は、米国における米国 International Business Machines Corp. の登録商標です。

Solaris は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Sun, Sun Microsystems, Java は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

TurboLinux の名称及びロゴは、TurboLinux, Inc. の商標です。

TUXEDO は、米国 UNIX System Laboratories, Inc. の商品名称です。

UNIFY2000 は、米国 Unify Corp. の商品名称です。

UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

VERITAS および VERITAS ロゴは、米国 Symantec Corporation の登録商標です。

VERITAS NetBackup は、米国及びその他の国での米国 Symantec Corporation の商標です。

VERITAS NetBackup BusinessServer は、米国及びその他の国での米国 Symantec Corporation の商品名称です。

Visual Basic は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Visual C++ は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Visual Studio は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows NT は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows Server は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。

Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。

X/Open は、X/Open Company Limited の英国ならびに他の国における登録商標です。

X Window System は、米国 X Consortium, Inc. が開発したソフトウェアです。

イーサネットは、富士ゼロックス（株）の商品名称です。

プログラムプロダクト「P-9D62-1581, P-9D62-1781, P-F9D62-1181A」には、米国 Sun Microsystems, Inc. が著作権を有している部分が含まれています。

プログラムプロダクト「P-9D62-1581, P-9D62-1781, P-F9D62-1181A」には、UNIX System Laboratories, Inc. が著作権を有している部分が含まれています。

## 発行

2008年1月（第1版）3020-6-368

2009年1月（第3版）3020-6-368-20

## 著作権

All Rights Reserved. Copyright (C) 2008, 2009, Hitachi, Ltd.

## 変更内容

変更内容 (3020-6-368-20) HiRDB Version 8 08-05

追加・変更内容	変更箇所
HiRDB が使用する共用メモリをメモリ上に固定できるようにしました。これに伴い、pd_max_resident_rdarea_no オペランドの説明を変更しました。	2.1.4

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 (3020-6-368-10) HiRDB Version 8 08-04

追加・変更内容
ユーザ LOB 用 RD エリアをインメモリ化できるようにしました。



# はじめに

---

このマニュアルは、プログラムプロダクト スケーラブルデータベースサーバ HiRDB Version 8 のバッチ高速化機能の運用方法について説明したものです。

## 対象読者

バッチ高速化機能を使って HiRDB Version 8 (以降, HiRDB と表記します) を運用する方々を対象にしています。

このマニュアルの記述は、次に示す知識があることを前提にしています。

- UNIX, または Windows のシステム管理の基礎的な知識
- HiRDB のシステム管理の基礎的な知識

## マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

### 第 1 章 バッチ処理の高速化

大量のデータを扱うバッチ処理の高速化を実現するインメモリデータ処理について説明しています。

### 第 2 章 インメモリデータ処理を実行するための環境を作る

インメモリデータ処理を実行するために必ず行う準備と、より効果的に使うために考えておくことについて説明しています。

### 第 3 章 インメモリデータ処理を適用する

業務の種類に応じたインメモリデータ処理の適用例について説明しています。

### 第 4 章 運用時に気をつけること

インメモリデータ処理を実行することで、そのほかの機能や運用で注意が必要な場合があります。この章では、そのような注意事項について説明しています。

### 第 5 章 トラブルシュート

インメモリデータ処理中に障害が発生した場合の回復方法について説明しています。

### 第 6 章 こんなときどうする

インメモリデータ処理を実行する場合に知っておくとよい操作について説明しています。

### 付録 A インメモリデータ処理の状態遷移図

インメモリデータ処理の状態遷移について説明しています。

### 付録 B インメモリデータ処理に関するコマンド一覧

インメモリデータ処理に関するコマンドと機能概要について説明しています。

### 付録 C pdmemdb コマンド実行時の前提条件

pdmemdb コマンド実行時の前提条件について説明しています。

## 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 (Windows 用マニュアル)

- HiRDB Version 8 解説 (Windows(R) 用) (3020-6-351)
- HiRDB Version 8 システム導入・設計ガイド (Windows(R) 用) (3020-6-352)
- HiRDB Version 8 システム定義 (Windows(R) 用) (3020-6-353)
- HiRDB Version 8 システム運用ガイド (Windows(R) 用) (3020-6-354)
- HiRDB Version 8 コマンドリファレンス (Windows(R) 用) (3020-6-355)
- HiRDB ファーストステップガイド (Windows(R) 用) (3020-6-054)

### HiRDB (UNIX 用マニュアル)

- HiRDB Version 8 解説 (UNIX(R) 用) (3000-6-351)
- HiRDB Version 8 システム導入・設計ガイド (UNIX(R) 用) (3000-6-352)
- HiRDB Version 8 システム定義 (UNIX(R) 用) (3000-6-353)
- HiRDB Version 8 システム運用ガイド (UNIX(R) 用) (3000-6-354)
- HiRDB Version 8 コマンドリファレンス (UNIX(R) 用) (3000-6-355)
- インナレプリカ機能 HiRDB Staticizer Option Version 8 (3000-6-363)
- HiRDB Version 8 ディザスタリカバリシステム 構築・運用ガイド (3000-6-364)
- HiRDB ファーストステップガイド (UNIX(R) 用) (3000-6-254)

### HiRDB (Windows, UNIX 共通マニュアル)

- HiRDB Version 8 UAP 開発ガイド (3020-6-356)
- HiRDB Version 8 SQL リファレンス (3020-6-357)
- HiRDB Version 8 メッセージ (3020-6-358)
- HiRDB Version 8 セキュリティガイド (3020-6-359)
- HiRDB Version 8 XDM/RD E2 接続機能 (3020-6-365)
- HiRDB データ連動機能 HiRDB Datareplicator Version 8 (3020-6-360)
- HiRDB データ連動拡張機能 HiRDB Datareplicator Extension Version 8 (3020-6-361)
- データベース抽出・反映サービス機能 HiRDB Dataextractor Version 8 (3020-6-362)
- HiRDB 全文検索プラグイン HiRDB Text Search Plug-in Version 8 (3020-6-375)
- HiRDB XML 拡張機能 HiRDB XML Extension Version 8 (3020-6-376)

なお、本文中で使用している HiRDB Version 8 のマニュアル名は、(UNIX(R) 用) または (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。

### 関連製品

- HiRDB External Data Access Version 8 (3020-6-366)

## 利用者ごとの関連マニュアル

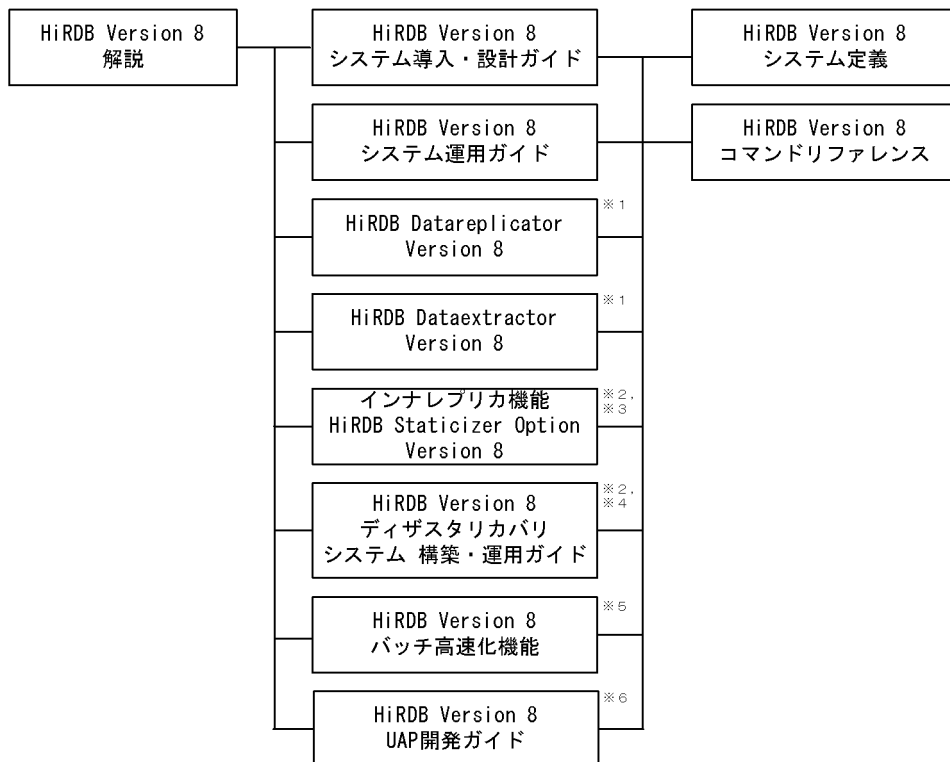
HiRDB のマニュアルをご利用になる場合、利用者ごとに次のようにお読みください。

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 OLTPシステムと連携する場合は必ずお読みください。
- 注※7 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。

## 読書手順

このマニュアルは、次に示す表に従ってお読みいただくことをお勧めします。

章タイトル	参照の目安
1. バッチ処理の高速化	
2. インメモリデータ処理を実行するための環境を作る	
3. インメモリデータ処理を適用する	
4. 運用時に気をつけること	
5. トラブルシュート	
6. こんなときどうする	
付録 A インメモリデータ処理の状態遷移図	
付録 B インメモリデータ処理に関するコマンド一覧	
付録 C pdmemdb コマンド実行時の前提条件	

(凡例)

：必ず読んでいただきたい内容です。

：必要に応じて読んでいただきたい内容です。

## このマニュアルでの表記

このマニュアルでは製品名称およ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/Single Server Version 8	HiRDB/ シングル サーバ	HiRDB また は HiRDB サーバ
HiRDB/Single Server Version 8(64)		
HiRDB/Parallel Server Version 8	HiRDB/ パラレル サーバ	
HiRDB/Parallel Server Version 8(64)		
HiRDB/Developer's Kit Version 8	HiRDB/ Developer's Kit	HiRDB クラ イアント
HiRDB/Developer's Kit Version 8(64)		
HiRDB/Run Time Version 8	HiRDB/Run Time	
HiRDB/Run Time Version 8(64)		
HiRDB Datareplicator Version 8	HiRDB Datareplicator	
HiRDB Dataextractor Version 8	HiRDB Dataextractor	
HiRDB Text Search Plug-in Version 8	HiRDB Text Search Plug-in	
HiRDB XML Extension Version 8	HiRDB XML Extension	
HiRDB Spatial Search Plug-in Version 3	HiRDB Spatial Search Plug-in	

製品名称または名称	表記
HiRDB Staticizer Option Version 8	HiRDB Staticizer Option
HiRDB LDAP Option Version 8	HiRDB LDAP Option
HiRDB Advanced Partitioning Option Version 8	HiRDB Advanced Partitioning Option
HiRDB Advanced High Availability Version 8	HiRDB Advanced High Availability
HiRDB Non Recover Front End Server Version 8	HiRDB Non Recover FES
HiRDB Disaster Recovery Light Edition Version 8	HiRDB Disaster Recovery Light Edition
HiRDB Accelerator Version 8	HiRDB Accelerator
HiRDB External Data Access Version 8	HiRDB External Data Access
HiRDB External Data Access Adapter Version 8	HiRDB External Data Access Adapter
HiRDB Adapter for XML - Standard Edition	HiRDB Adapter for XML
HiRDB Adapter for XML - Enterprise Edition	
HiRDB Control Manager	HiRDB CM
HiRDB Control Manager Agent	HiRDB CM Agent
Hitachi TrueCopy	TrueCopy
Hitachi TrueCopy basic	
TrueCopy	
TrueCopy remote replicator	
JP1/Automatic Job Management System 2	JP1/AJS2
JP1/Automatic Job Management System 2 - Scenario Operation	JP1/AJS2-SO
JP1/Cm2/Extensible SNMP Agent	JP1/ESA
JP1/Cm2/Extensible SNMP Agent for Mib Runtime	
JP1/Cm2/Network Node Manager	JP1/NNM
JP1/Integrated Management - Manager	JP1/Integrated Management または JP1/IM
JP1/Integrated Management - View	
JP1/Magnetic Tape Access	EasyMT
EasyMT	
JP1/Magnetic Tape Library	MTguide
JP1/NETM/Audit - Manager	JP1/NETM/Audit
JP1/NETM/DM	JP1/NETM/DM
JP1/NETM/DM Manager	
JP1/Performance Management	JP1/PFM
JP1/Performance Management - Agent Option for HiRDB	JP1/PFM-Agent for HiRDB

はじめに

製品名称または名称	表記	
JP1/Performance Management - Agent Option for Platform	JP1/PFM-Agent for Platform	
JP1/Performance Management/SNMP System Observer	JP1/SSO	
JP1/VERITAS NetBackup BS v4.5	NetBackup	
JP1/VERITAS NetBackup v4.5		
JP1/VERITAS NetBackup BS V4.5 Agent for HiRDB License	JP1/VERITAS NetBackup Agent for HiRDB License	
JP1/VERITAS NetBackup V4.5 Agent for HiRDB License		
JP1/VERITAS NetBackup 5 Agent for HiRDB License		
OpenTP1/Server Base Enterprise Option	TP1/EE	
Virtual-storage Operating System 3/Forefront System Product	VOS3/FS	VOS3
Virtual-storage Operating System 3/Leading System Product	VOS3/LS	
Extensible Data Manager/Base Extended Version 2 XDM 基本プログラム XDM/BASE E2	XDM/BASE E2	
XDM/Data Communication and Control Manager 3 XDM データコミュニケーションマネジメントシステム XDM/DCCM3	XDM/DCCM3	
XDM/Relational Database リレーショナルデータベースシステム XDM/RD	XDM/RD	XDM/RD
XDM/Relational Database Extended Version 2 リレーショナルデータベースシステム XDM/RD E2	XDM/RD E2	
VOS3 Database Connection Server	DB コネクションサーバ	
BEA WebLogic Server	WebLogic Server	
DB2 Universal Database for OS/390 Version 6	DB2	
DNCWARE ClusterPerfect (Linux 版)	ClusterPerfect	
Microsoft(R) Office Excel	Microsoft Excel または Excel	
Microsoft(R) Visual C++(R)	Visual C++ または C++ 言語	
Oracle8i	ORACLE	
Oracle9i		
Oracle 10g		
Sun Java™ System Directory Server	Sun Java System Directory Server またはディレクトリサーバ	
HP-UX 11i V2 (IPF)	HP-UX または HP-UX (IPF)	
HP-UX 11i V3 (IPF)		
AIX 5L V5.1	AIX 5L	AIX
AIX 5L V5.2		
AIX 5L V5.3		
AIX V6.1	AIX V6.1	
Linux(R)	Linux	

製品名称または名称	表記			
Red Hat Linux	Red Hat Linux	Linux		
Red Hat Enterprise Linux	Red Hat Enterprise Linux			
Red Hat Enterprise Linux AS 3 (IPF)	Linux (IPF)			
Red Hat Enterprise Linux AS 4 (IPF)				
Red Hat Enterprise Linux 5.1 Advanced Platform (Intel Itanium)				
Red Hat Enterprise Linux 5.1 (Intel Itanium)				
Red Hat Enterprise Linux 5.2 Advanced Platform (Intel Itanium)				
Red Hat Enterprise Linux 5.2 (Intel Itanium)				
Red Hat Enterprise Linux AS 3(AMD64 & Intel EM64T)		Linux (EM64T)		
Red Hat Enterprise Linux AS 4(AMD64 & Intel EM64T)				
Red Hat Enterprise Linux ES 4(AMD64 & Intel EM64T)				
Red Hat Enterprise Linux 5.1 Advanced Platform (AMD/Intel 64)				
Red Hat Enterprise Linux 5.1 (AMD/Intel 64)				
Red Hat Enterprise Linux 5.2 Advanced Platform (AMD/Intel 64)				
Red Hat Enterprise Linux 5.2 (AMD/Intel 64)				
Red Hat Enterprise Linux AS 4(AMD64 & Intel EM64T)	Linux AS 4			
Red Hat Enterprise Linux AS 4(x86)				
Red Hat Enterprise Linux ES 4(AMD64 & Intel EM64T)	Linux ES 4			
Red Hat Enterprise Linux ES 4(x86)				
Red Hat Enterprise Linux 5.1 Advanced Platform (x86)	Linux 5.1	Linux 5		
Red Hat Enterprise Linux 5.1 (x86)				
Red Hat Enterprise Linux 5.1 Advanced Platform (AMD/Intel 64)				
Red Hat Enterprise Linux 5.1 (AMD/Intel 64)				
Red Hat Enterprise Linux 5.1 Advanced Platform (Intel Itanium)				
Red Hat Enterprise Linux ES 4(x86)				
Red Hat Enterprise Linux 5.2 Advanced Platform (x86)			Linux 5.2	
Red Hat Enterprise Linux 5.2 (x86)				
Red Hat Enterprise Linux 5.2 Advanced Platform (AMD/Intel 64)				
Red Hat Enterprise Linux 5.2 (AMD/Intel 64)				
Red Hat Enterprise Linux 5.2 (AMD/Intel 64)				

製品名称または名称	表記		
Red Hat Enterprise Linux 5.2 Advanced Platform (Intel Itanium)			
Red Hat Enterprise Linux 5.2 (Intel Itanium)			
turbolinux 7 Server for AP8000	Linux for AP8000		
Microsoft(R) Windows NT(R) Workstation Operating System Version 4.0	Windows NT		
Microsoft(R) Windows NT(R) Server Network Operating System Version 4.0			
Microsoft(R) Windows(R) 2000 Professional Operating System	Windows 2000		
Microsoft(R) Windows(R) 2000 Server Operating System			
Microsoft(R) Windows(R) 2000 Datacenter Server Operating System			
Microsoft(R) Windows(R) 2000 Advanced Server Operating System			
Microsoft(R) Windows(R) 2000 Advanced Server Operating System	Windows 2000 Advanced Server		
Microsoft(R) Windows Server(R) 2003, Standard Edition	Windows Server 2003 Standard Edition	Windows Server 2003	
Microsoft(R) Windows Server(R) 2003, Enterprise Edition	Windows Server 2003 Enterprise Edition		
Microsoft(R) Windows Server(R) 2003, Standard x64 Edition	Windows Server 2003 Standard x64 Edition		
Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition	Windows Server 2003 Enterprise x64 Edition		
Microsoft(R) Windows Server(R) 2003 R2, Standard Edition	Windows Server 2003 R2		
Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition			
Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition			
Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition			
Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition	Windows Server 2003 R2 x64 Editions		
Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition			
64 ビットバージョン Microsoft(R) Windows Server(R) 2003, Enterprise Edition	Windows Server 2003 (IPF)		
Microsoft(R) Windows Server(R) 2008 Standard	Windows Server 2008 Standard	Windows Server 2008	

製品名称または名称	表記	
Microsoft(R) Windows Server(R) 2008 Enterprise	Windows Server 2008 Enterprise	
Microsoft(R) Windows Server(R) 2003, Standard x64 Edition	Windows Server 2003 x64 Editions	Windows (x64)
Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition		
Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition		
Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition		
Microsoft(R) Windows(R) XP Professional x64 Edition	Windows XP x64 Edition	
64 ビットバージョン Microsoft(R) Windows Server(R) 2003, Enterprise Edition	Windows Server 2003 (IPF)	Windows(IPF )
Microsoft(R) Windows(R) XP Professional x64 Edition	Windows XP x64 Edition	Windows XP
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP Professional	
Microsoft(R) Windows(R) XP Home Edition Operating System	Windows XP Home Edition	
Microsoft(R) Windows Vista(R) Home Basic	Windows Vista Home Basic	
Microsoft(R) Windows Vista(R) Home Premium	Windows Vista Home Premium	Windows Vista
Microsoft(R) Windows Vista(R) Ultimate	Windows Vista Ultimate	
Microsoft(R) Windows Vista(R) Business	Windows Vista Business	
Microsoft(R) Windows Vista(R) Enterprise	Windows Vista Enterprise	
シングルサーバ	SDS	
システムマネージャ	MGR	
フロントエンドサーバ	FES	
ディクショナリサーバ	DS	
バックエンドサーバ	BES	

- Windows Server 2003 および Windows Server 2008 を総称して Windows Server と表記します。また、Windows 2000、Windows XP、Windows Server、および Windows Vista を総称して Windows と表記します。
- TCP/IP が規定する hosts ファイル (UNIX の場合 /etc/hosts ファイルも含む) を hosts ファイルと表記します。hosts ファイルとは通常、Windows の場合は %windir%\system32\drivers\etc\hosts のことです。

## このマニュアルで使用する略語

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字の表記
ACK	<u>A</u> cknowledgement
ADM	<u>A</u> daptable <u>D</u> ata <u>M</u> anager
ADO	<u>A</u> ctiveX <u>D</u> ata <u>O</u> bjects
ADT	<u>A</u> bstract <u>D</u> ata <u>T</u> ype
AP	<u>A</u> pplication <u>P</u> rogram
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
ASN.1	<u>A</u> bstract <u>S</u> yntax <u>N</u> otation <u>O</u> ne
BES	<u>B</u> ack <u>E</u> nd <u>S</u> erver
BLOB	<u>B</u> inary <u>L</u> arge <u>O</u> bject
BMP	<u>B</u> asic <u>M</u> ultilingual <u>P</u> lane
BOM	<u>B</u> yte <u>O</u> rder <u>M</u> ark
CD-ROM	<u>C</u> ompact <u>D</u> isc - <u>R</u> ead <u>O</u> nly <u>M</u> emory
CGI	<u>C</u> ommon <u>G</u> ateway <u>I</u> nterface
CLOB	<u>C</u> haracter <u>L</u> arge <u>O</u> bject
CMT	<u>C</u> assette <u>M</u> agnetic <u>T</u> ape
COBOL	<u>C</u> ommon <u>B</u> usiness <u>O</u> riented <u>L</u> anguage
CORBA	<u>C</u> ommon <u>O</u> RB <u>A</u> rchitecture
CPU	<u>C</u> entral <u>P</u> rocessing <u>U</u> nit
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alues
DAO	<u>D</u> ata <u>A</u> ccess <u>O</u> bject
DAT	<u>D</u> igital <u>A</u> udio <u>T</u> aperecorder
DB	<u>D</u> atabase
DBM	<u>D</u> atabase <u>M</u> odule
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DDL	<u>D</u> ata <u>D</u> efinition <u>L</u> anguage
DF for Windows NT	<u>D</u> istributing <u>F</u> acility <u>f</u> or <u>W</u> indows <u>N</u> T
DF/UX	<u>D</u> istributing <u>F</u> acility / <u>f</u> or <u>U</u> NIX
DIC	<u>D</u> ictionary <u>S</u> erver
DLT	<u>D</u> igital <u>L</u> inear <u>T</u> ape
DML	<u>D</u> ata <u>M</u> anipulate <u>L</u> anguage



英略語	英字の表記
DNS	<u>D</u> omain <u>N</u> ame <u>S</u> ystem
DOM	<u>D</u> ocument <u>O</u> bject <u>M</u> odel
DS	<u>D</u> ictionary <u>S</u> erver
DTD	<u>D</u> ocument <u>T</u> ype <u>D</u> efinition
DTP	<u>D</u> istributed <u>T</u> ransaction <u>P</u> rocessing
DWH	<u>D</u> ata <u>W</u> arehouse
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
EX	<u>E</u> xclusive
FAT	<u>F</u> ile <u>A</u> llocation <u>T</u> able
FD	<u>F</u> loppy <u>D</u> isk
FES	<u>F</u> ront <u>E</u> nd <u>S</u> erver
FQDN	<u>F</u> ully <u>Q</u> ualified <u>D</u> omain <u>N</u> ame
FTP	<u>F</u> ile <u>T</u> ransfer <u>P</u> rotocol
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HBA	<u>H</u> ost <u>B</u> us <u>A</u> dapter
HD	<u>H</u> ard <u>D</u> isk
HTML	<u>H</u> yper <u>T</u> ext <u>M</u> arkup <u>L</u> anguage
ID	<u>I</u> dentification number
IP	<u>I</u> nternet <u>P</u> rotocol
IPF	<u>I</u> tanium <sub>(R)</sub> <u>P</u> rocessor <u>F</u> amily
JAR	<u>J</u> ava <u>A</u> rchive <u>F</u> ile
Java VM	<u>J</u> ava <u>V</u> irtual <u>M</u> achine
JDBC	<u>J</u> ava <u>D</u> atabase <u>C</u> onnectivity
JDK	<u>J</u> ava <u>D</u> eveloper's <u>K</u> it
JFS	<u>J</u> ournaled <u>F</u> ile <u>S</u> ystem
JFS2	<u>E</u> nhanced <u>J</u> ournaled <u>F</u> ile <u>S</u> ystem
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandard code
JP1	<u>J</u> ob <u>M</u> anagement <u>P</u> artner <u>1</u>
JRE	<u>J</u> ava <u>R</u> untime <u>E</u> nvironment
JTA	<u>J</u> ava <u>T</u> ransaction <u>A</u> PI
JTS	<u>J</u> ava <u>T</u> ransaction <u>S</u> ervice
KEIS	<u>K</u> anji processing <u>E</u> xtended <u>I</u> nformation <u>S</u> ystem
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork

## はじめに

英略語	英字の表記
LDAP	<u>L</u> ightweight <u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
LIP	<u>L</u> oop <u>I</u> nitialization <u>P</u> rocess
LOB	<u>L</u> arge <u>O</u> bject
LRU	<u>L</u> east <u>R</u> ecently <u>U</u> sed
LTO	<u>L</u> inear <u>T</u> ape- <u>O</u> pen
LU	<u>L</u> ogical <u>U</u> nit
LUN	<u>L</u> ogical <u>U</u> nit <u>N</u> umber
LVM	<u>L</u> ogical <u>V</u> olume <u>M</u> anager
MGR	System <u>M</u> anager
MIB	<u>M</u> anagement <u>I</u> nformation <u>B</u> ase
MRCF	<u>M</u> ultiple <u>R</u> AID <u>C</u> oupling <u>F</u> eature
MSCS	<u>M</u> icrosoft <u>C</u> luster <u>S</u> erver
MSFC	<u>M</u> icrosoft <u>F</u> ailover <u>C</u> luster
NAFO	<u>N</u> etwork <u>A</u> dapter <u>F</u> ail <u>O</u> ver
NAPT	<u>N</u> etwork <u>A</u> ddress <u>P</u> ort <u>T</u> ranslation
NAT	<u>N</u> etwork <u>A</u> ddress <u>T</u> ranslation
NIC	<u>N</u> etwork <u>I</u> nterface <u>C</u> ard
NIS	<u>N</u> etwork <u>I</u> nformation <u>S</u> ervice
NTFS	<u>N</u> ew <u>T</u> echnology <u>F</u> ile <u>S</u> ystem
ODBC	<u>O</u> pen <u>D</u> atabase <u>C</u> onnectivity
OLAP	<u>O</u> nline <u>A</u> nalytical <u>P</u> rocessing
OLE	<u>O</u> bject <u>L</u> inking and <u>E</u> mbedding
OLTP	<u>O</u> n- <u>L</u> ine <u>T</u> ransaction <u>P</u> rocessing
OOCOBOL	<u>O</u> bject <u>O</u> riented <u>C</u> OBOL
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
OS	<u>O</u> perating <u>S</u> ystem
OSI	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection
OTS	<u>O</u> bject <u>T</u> ransaction <u>S</u> ervice
PC	<u>P</u> ersonal <u>C</u> omputer
PDM II E2	<u>P</u> ractical <u>D</u> ata <u>M</u> anager <u>I</u> I <u>E</u> xtended <u>V</u> ersion <u>2</u>
PIC	<u>P</u> lug-in <u>C</u> ode
PNM	<u>P</u> ublic <u>N</u> etwork <u>M</u> anagement
POSIX	<u>P</u> ortable <u>O</u> perating <u>S</u> ystem <u>I</u> nterface for <u>U</u> NIX

英略語	英字の表記
PP	<u>P</u> rogram <u>P</u> roduct
PR	<u>P</u> rotected <u>R</u> etrieve
PU	<u>P</u> rotected <u>U</u> pdate
RAID	<u>R</u> edundant <u>A</u> rrays of <u>I</u> nexpensive <u>D</u> isk
RD	<u>R</u> elational <u>D</u> atabase
RDB	<u>R</u> elational <u>D</u> ataba <u>s</u> e
RDB1	<u>R</u> elational <u>D</u> ataba <u>s</u> e <u>M</u> anager <u>1</u>
RDB1 E2	<u>R</u> elational <u>D</u> ataba <u>s</u> e <u>M</u> anager <u>1</u> <u>E</u> xtended <u>V</u> ersion <u>2</u>
RDO	<u>R</u> emote <u>D</u> ata <u>O</u> bjects
RiSe	<u>R</u> eal <u>t</u> ime <u>S</u> AN <u>r</u> eplication
RM	<u>R</u> esource <u>M</u> anager
RMM	<u>R</u> esource <u>M</u> anager <u>M</u> onitor
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SAX	<u>S</u> imple <u>A</u> PI for <u>X</u> ML
SDS	<u>S</u> ingle <u>D</u> atabase <u>S</u> erver
SGML	<u>S</u> tandard <u>G</u> eneralized <u>M</u> arkup <u>L</u> anguage
SJIS	<u>S</u> hift <u>J</u> IS
SNMP	<u>S</u> imple <u>N</u> etwork <u>M</u> anagement <u>P</u> rotocol
SNTP	<u>S</u> imple <u>N</u> etwork <u>T</u> ime <u>P</u> rotocol
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
SQL/K	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage / <u>V</u> OS <u>K</u>
SR	<u>S</u> hared <u>R</u> etrieve
SU	<u>S</u> hared <u>U</u> pdate
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol / <u>I</u> nternet <u>P</u> rotocol
TM	<u>T</u> ransaction <u>M</u> anager
TMS-4V/SP	<u>T</u> ransaction <u>M</u> anagement <u>S</u> ystem - <u>4</u> V / <u>S</u> ystem <u>P</u> roduct
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
VOS1	<u>V</u> irtual-storage <u>O</u> perating <u>S</u> ystem <u>1</u>
VOS3	<u>V</u> irtual-storage <u>O</u> perating <u>S</u> ystem <u>3</u>
VOS K	<u>V</u> irtual-storage <u>O</u> perating <u>S</u> ystem <u>K</u> indness
WS	<u>W</u> orkstation
WWW	<u>W</u> orld <u>W</u> ide <u>W</u> eb

英略語	英字の表記
XDM/BASE E2	Extensible Data Manager / Base Extended Version 2
XDM/DF	Extensible Data Manager / Distributing Facility
XDM/DS	Extensible Data Manager / Data Spreader
XDM/RD E2	Extensible Data Manager / Relational Database Extended Version 2
XDM/SD E2	Extensible Data Manager / Structured Database Extended Version 2
XDM/XT	Extensible Data Manager / Data Extract
XFIT	Extended File Transmission program
XML	Extensible Markup Language

## パス名の表記

- パス名の区切りは「¥」で表記しています。UNIX 版 HiRDB 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、UNIX 版は \$PDDIR と表記します。例を次に示します。

Windows 版：%PDDIR%¥CLIENT¥UTL¥

UNIX 版：\$PDDIR/client/lib/

- Windows のインストールディレクトリのパスを %windir% と表記します。

## 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

- [ 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。
- [ 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「HiRDBSingleServer」または「HiRDBParallelServer」になっているのが HiRDB が出力したメッセージです。  
なお、セットアップ識別子を指定してインストールした場合は、「HiRDBSingleServer」または「HiRDBParallelServer」にセットアップ識別子が付いた名称となります。

### 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
Solaris	/var/adm/messages または /var/log/syslog
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。

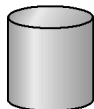
記号	意味
[ ]	ボタンやテキストボックスなど、画面に表示されている要素を示します。
[ ] - [ ]	画面に表示されるメニューやアイコンなどを選択する操作を示します。

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

## 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。

●ファイル



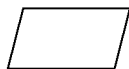
●プログラム  
またはサーバ



●作業手順



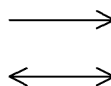
●インメモリ  
データバッファ



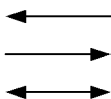
●データの流れ



●制御の流れ



●その他の流れ



## Windows のパス名に関する注意

- パス名を絶対パスで指定する場合はドライブ名を指定してください。

(例) C:\win32app\hitachi\hirdb\_s\spool\tmp

- コマンドの引数、制御文ファイル、および HiRDB システム定義ファイル中に空白または丸括弧を含むパス名を指定する場合は、前後を引用符 (") で囲んでください。

(例) pdinit -d "C:\Program Files(x86)\hitachi\hirdb\_s\conf\mkinit"

ただし、バッチファイルもしくはコマンドプロンプト上で set コマンドを使用して環境変数を設定する場合、またはインストールディレクトリを指定する場合は引用符は不要です。引用符で囲むと、引用符も環境変数の値に含まれます。

(例) set PDCLTPATH=C:\Program Files\hitachi\hirdb\_s\spool

- HiRDB はネットワークドライブのファイルを使用できないため、HiRDB のインストール、および環境構築はローカルドライブで行ってください。また、ユティリティの入出力ファイルなども、ローカルドライブ上のファイルを使用してください。
- パス名には、ショートパス名 (例えば、C:\PROGRA~1 など) は使用しないでください。

### 常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語については、常用漢字以外の漢字を使用しています。

- 改竄 (かいざん)
- 個所 (かしょ)
- 閉塞 (へいそく)

### KB (キロバイト) などの単位表記について

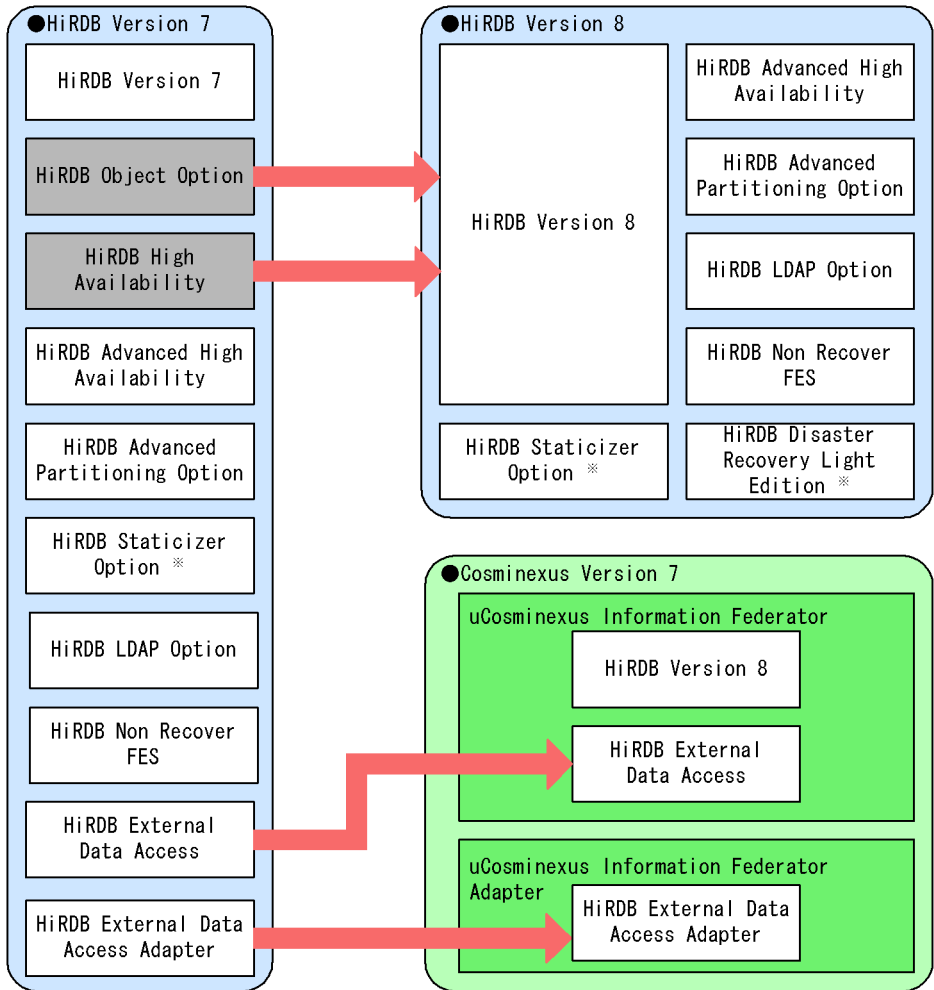
1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ 1,024 バイト, 1,024<sup>2</sup> バイト, 1,024<sup>3</sup> バイト, 1,024<sup>4</sup> バイトです。

### Version 7 と Version 8 の製品体系の違い

HiRDB Version 8 では、HiRDB Version 7 までオプション製品 (HiRDB Object Option および HiRDB High Availability) で提供していた機能を HiRDB の標準機能としました。それに伴い、オプション製品が廃止になりました。

また、Version 8 以降、HiRDB External Data Access および HiRDB External Data Access Adapter は HiRDB シリーズではなく、Cosminexus Version 7 シリーズとなりました。

HiRDB Version 7 と Version 8 の製品体系の違いを次に示します。



注※ UNIX版でだけ使用できる製品です。





# 目次

<b>1</b>	<b>バッチ処理の高速化</b>	<b>1</b>
1.1	バッチ処理の高速化が必要とされる理由	2
1.2	インメモリデータ処理の仕組み	3
1.2.1	インメモリデータ処理の概要	3
1.2.2	インメモリデータバッファの優位点	4
1.2.3	インメモリデータ処理に関する用語	5
1.3	インメモリデータ処理を適用すると効果が期待できるケース	6
1.4	インメモリデータ処理を適用しても効果が期待できないケース	7
1.5	インメモリデータ処理が適用できないケース	8
<b>2</b>	<b>インメモリデータ処理を実行するための環境を作る</b>	<b>9</b>
2.1	必ず行う準備	10
2.1.1	マシン環境の確認	10
2.1.2	HiRDB Accelerator のインストール	11
2.1.3	HiRDB Accelerator のセットアップ (UNIX 限定)	12
2.1.4	システム定義の設定	12
2.2	インメモリデータ処理をより効果的に実行するために考えておくこと	14
<b>3</b>	<b>インメモリデータ処理を適用する</b>	<b>17</b>
3.1	基本的な運用の流れ	18
3.2	バッチ業務に適用する場合	21
3.3	オンライン業務に適用する場合	27
3.4	テスト業務に適用する場合	31
3.5	複数の RD エリアを一度にインメモリ化する場合の注意事項	37
<b>4</b>	<b>運用時に気をつけること</b>	<b>41</b>
4.1	バックアップを取得するときに気をつけること	42
4.1.1	バックアップを取得するタイミング	42
4.1.2	参照可能モード (-Mr) でバックアップを取得する場合	43
4.2	HiRDB を終了するときに気をつけること	46
4.3	データベース構成変更ユーティリティを使用するときに気をつけること	47

4.4	インナレプリカ機能を使用しているときに気をつけること	48
4.5	系切り替え機能を使用しているときに気をつけること	49
4.6	その他の注意事項	50

## 5

トラブルシュート	53
----------	----

5.1	障害回復の基本的な考え方	54
5.2	障害が発生したときに最初に確認すること	56
5.3	RD エリア障害が発生した場合の回復手順	59
5.4	バッファ障害が発生した場合の回復手順	63
5.4.1	最新の状態に回復する場合	63
5.4.2	同期取得時点で回復する場合（関連 RD エリアなし）	67
5.4.3	同期取得時点で回復する場合（関連 RD エリアあり）	71

## 6

こんなときどうする	77
-----------	----

6.1	インメモリデータバッファと RD エリアの同期を取るには	78
6.2	インメモリデータバッファの利用状況を確認するには	79
6.3	インメモリデータバッファの統計情報を確認するには	81
6.4	強制的にインメモリ化を解除するには	83
6.5	インメモリデータバッファ上のデータを破棄するには	84
6.6	RD エリアのデータをインメモリデータバッファに再読み込みするには	85

## 付録

付録 A	インメモリデータ処理の状態遷移図	88
付録 B	インメモリデータ処理に関するコマンド一覧	90
付録 C	pdmemb コマンド実行時の前提条件	91

## 索引

93

## 目次

図 1-1	インメモリデータ処理の概要	3
図 2-1	必ず行う準備	10
図 2-2	RD エリア内に使用しない表が混在している場合の対処例	14
図 3-1	基本的な運用の流れ	19
図 3-2	バッチ業務に適用する場合の運用の流れ	21
図 3-3	オンライン業務に適用する場合の運用の流れ	27
図 3-4	テスト業務に適用する場合の運用の流れ	32
図 3-5	共用メモリセグメントの割り当ての仕組み	38
図 4-1	バックアップを取得するタイミング	42
図 4-2	DB 非同期状態でバックアップを取得した場合の例	44
図 5-1	pddb1s -M コマンドの実行結果	56
図 5-2	回復手順の参照先を決めるフローチャート	57
図 5-3	回復手順の参照先を決めるフローチャート（ログレスモードでバッチ業務を実行した場合）	57
図 5-4	障害発生時の状況（RD エリア障害が発生した場合）	59
図 5-5	障害回復の手順（RD エリア障害が発生した場合）	60
図 5-6	障害発生時の状況（バッファ障害を最新の状態に回復する場合）	63
図 5-7	障害回復の手順（バッファ障害を最新の状態に回復する場合）	64
図 5-8	障害発生時の状況（バッファ障害を同期取得時点で回復する場合（関連 RD エリアなし））	68
図 5-9	障害回復の手順（バッファ障害を同期取得時点で回復する場合（関連 RD エリアなし））	69
図 5-10	障害発生時の状況（バッファ障害を同期取得時点で回復する場合（関連 RD エリアあり））	72
図 5-11	障害回復の手順（バッファ障害を同期取得時点で回復する場合（関連 RD エリアあり））	73
図 A-1	インメモリデータ処理の状態遷移	89

## 表目次

表 1-1	インメモリデータ処理に関する用語	5
表 5-1	RD エリアとインメモリデータバッファの状態	56
表 5-2	障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移	61
表 5-3	障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移	67
表 5-4	障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移	70
表 5-5	障害回復時のインメモリデータバッファ, インメモリ RD エリアおよびインメモリ化していない RD エリアの状態遷移	75
表 6-1	インメモリ化を強制的に解除するときの RD エリアの状態	83
表 B-1	インメモリデータ処理に関するコマンド一覧	90
表 C-1	pdmembd コマンド実行時の前提条件 (インメモリデータバッファと対象 RD エリアの状態)	91

# 1

## バッチ処理の高速化

この章では、大量のデータを扱うバッチ処理の高速化を実現するインメモリデータ処理について説明します。

---

1.1 バッチ処理の高速化が必要とされる理由

---

1.2 インメモリデータ処理の仕組み

---

1.3 インメモリデータ処理を適用すると効果が期待できるケース

---

1.4 インメモリデータ処理を適用しても効果が期待できないケース

---

1.5 インメモリデータ処理が適用できないケース

---

## 1.1 バッチ処理の高速化が必要とされる理由

---

企業や組織が保有するデータは増加の一途をたどっています。データ量の増加に伴い、バッチ処理に掛かる時間は長くなる一方です。さらに、業務のサービス時間は延長傾向にあるため、バッチ処理を実行できる時間が短くなってきています。また、これまでメインフレームでやっていたような大量のバッチ処理を、オープンシステムでも実施するという需要が高まっています。このような理由から、オープンシステムでのバッチ処理の高速化が企業や組織の課題となっています。

HiRDB では、大量のデータを扱うバッチ処理を高速化するために、RD エリア内の全データをメモリ上に一括して読み込み、バッチ処理の実行中はメモリ上のデータだけを更新し、ディスク上のデータは更新しません。バッチ処理が完了したあとにメモリ上の更新データを一括してディスクに書き込みます。バッチ処理中はディスク入出力が発生しないため、その分バッチ処理に掛かる時間を短縮できます。

このバッチ高速化を実現するための処理方式をインメモリデータ処理といいます。

例えば、次のようなケースに当てはまる場合は、インメモリデータ処理を適用してバッチ処理の高速化を検討してください。

### ケース 1

バッチ処理を実行できる時間の範囲が午前 0 時から午前 6 時までの 6 時間と決まっていて、バッチ処理が完了するまで 4 時間掛かります。バッチ処理の途中でエラーが発生した場合、決められた時間内にバッチ処理を完了できないおそれがあります。

### ケース 2

データ量の増加によって、3 か月後のバッチ処理時間が 12 時間以上になると予想され、翌日のオンライン業務に影響が出ると考えられます。

### ケース 3

将来、メインフレームからデータを移行する予定があります。しかし、データの移行に掛かる時間を試算した結果、移行予定期間内にデータの移行が終わりそうにありません。

## 1.2 インメモリデータ処理の仕組み

ここでは、インメモリデータ処理の仕組みについて説明します。

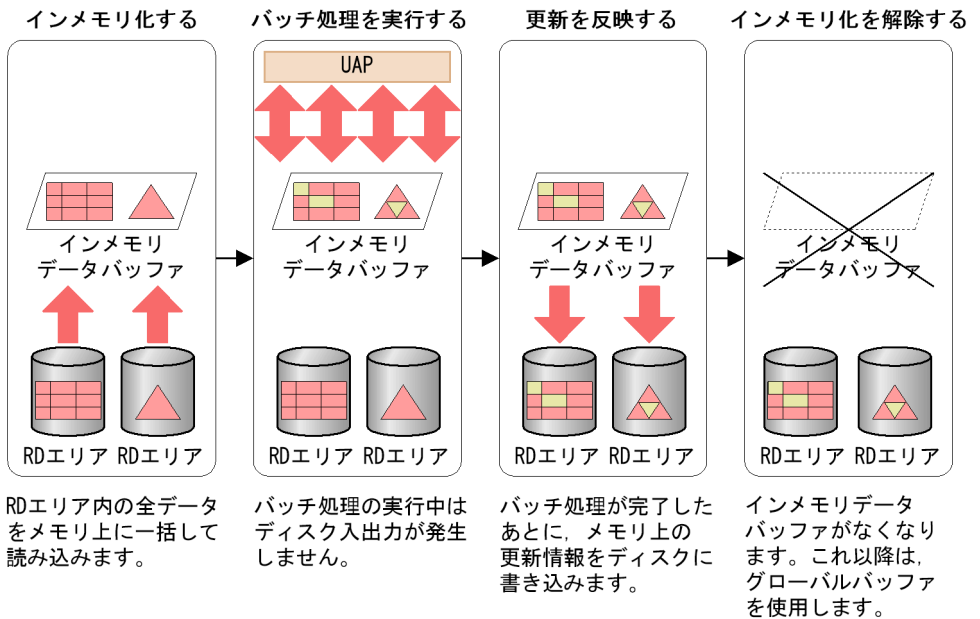
なお、インメモリデータ処理を実行するには HiRDB Accelerator が必要です。

### 1.2.1 インメモリデータ処理の概要

インメモリデータ処理では、RD エリア内の全データをメモリ上に一括して読み込み、バッチ処理の実行中はメモリ上のデータだけを更新し、ディスク上のデータは更新しません。バッチ処理が完了したあとにメモリ上の更新データを一括してディスクに書き込みます。バッチ処理中はディスク入出力が発生しません。ディスク入出力が発生するのは、RD エリア内のデータをメモリ上に読み込むときと、メモリ上の更新データをディスクに書き込むときの2回だけです。

インメモリデータ処理の概要を次に示します。

図 1-1 インメモリデータ処理の概要



#### [ 説明 ]

RD エリア内の全データをメモリ上に一括して読み込むことをインメモリ化といいます。インメモリ化はRD エリアごとに行います。インメモリ化しているRD エリアをインメモリRD エリアといいます。また、インメモリデータ処理で使用するデータバッファをインメモリデータバッファといいます。

## 1. バッチ処理の高速化

### ポイント

---

インメモリデータ処理を実行する場合は、グローバルバッファを使用しないでインメモリデータバッファを使用します。インメモリデータバッファは、HiRDB が動的に共用メモリ上に確保するため、HiRDB 管理者がインメモリデータバッファを定義する必要はありません。

---

バッチ処理の実行中はインメモリデータバッファ上のデータだけが更新されて、ディスク上のデータは更新されません。バッチ処理が完了したあとにコマンド (pdhold コマンド) を実行して、インメモリデータバッファ上の更新情報を一括してディスクに書き込みます。シンクポイント時もディスクへの書き込みが発生しません。なお、ログ取得モードであれば、インメモリデータ処理中でも、通常と同じようにデータベースの更新ログが取得されます。障害が発生した場合、そのログを使って、データベースを最新の状態に回復できます。

### 1.2.2 インメモリデータバッファの優位点

ここでは、グローバルバッファと比較したときのインメモリデータバッファの優位点について説明します。性能面ではインメモリデータバッファの方が優れていますが、障害発生時の対処方法がグローバルバッファに比べてやや複雑になります。

インメモリデータバッファとグローバルバッファのそれぞれの優位点を次に説明します。

#### (1) インメモリデータバッファが優位な点

インメモリデータバッファが優位な点を次に示します。

- ディスク入出力回数が少ない  
グローバルバッファを使用している場合はディスク入出力が定期的に発生しますが、インメモリデータバッファを使用している場合は、ディスク入出力が発生するのはインメモリ化した時点とディスクへのデータ書き込み時点の 2 回だけです。シンクポイント時にもディスクへの書き込みが発生しません。そのため、インメモリデータバッファを使用する方が、グローバルバッファを使用したときに比べてディスク入出力回数が少なくなります。
- バッファの管理処理が少ない  
グローバルバッファを使用している場合は次に示すバッファ管理処理が発生しますが、インメモリデータバッファを使用している場合はこれらのバッファ管理処理が発生しません。
  - キャッシュバッファチェーン (バッファへのポインタの集合) の管理処理
  - LRU リストの管理処理
  - 更新バッファの管理処理

インメモリデータバッファを使用した方が、これらの管理処理が不要になり、その分性能が向上します。また、トランザクションの同時実行性も向上します。

- チューニングが不要  
グローバルバッファの場合はヒット率を意識してグローバルバッファのチューニング



をする必要があります。一方、インメモリデータバッファの場合はヒット率を意識する必要がないため、チューニングが必要ありません。そのため、チューニングに掛かるコストを削減できます。

## (2) グローバルバッファが優位な点

グローバルバッファが優位な点を次に示します。

- HiRDB が異常終了したときのデータの回復方法

グローバルバッファを使用している場合は、HiRDB が異常終了しても、再開開始時に最新の状態で HiRDB が回復します。インメモリデータバッファを使用している場合は、HiRDB が異常終了して再開開始しても、HiRDB は自動回復しません。この場合、HiRDB 管理者がバックアップとログを使用して最新の状態で回復するか、またはインメモリ化時点以降の処理を再実行する必要があります。

### 1.2.3 インメモリデータ処理に関する用語

インメモリデータ処理に関する用語を次の表に示します。

表 1-1 インメモリデータ処理に関する用語

用語	説明
インメモリデータ処理	RD エリア内の全データをメモリ（インメモリデータバッファ）上に常駐させて、ディスク入出力回数を抑える処理方式のことです。
インメモリデータバッファ	インメモリデータ処理で使用するデータバッファのことです。
インメモリ化	インメモリデータ処理で、RD エリア内の全データをメモリ上一括して読み込み、常駐させることです。
インメモリ RD エリア	インメモリデータ処理の対象となっている RD エリアのことです。この RD エリア内の全データがメモリ上に常駐します。
DB 同期状態	インメモリ RD エリア内のデータとインメモリデータバッファ上のデータの同期が取れている状態のことです。
DB 非同期状態	インメモリ RD エリア内のデータとインメモリデータバッファ上のデータの同期が取れていない状態のことです。
RD エリア障害	インメモリ RD エリアに障害が発生した状態です。
バッファ障害	インメモリデータバッファに障害が発生した状態です。

## 1.3 インメモリデータ処理を適用すると効果が期待できるケース

---

次の場合にインメモリデータ処理を適用すると効果が期待できます。

- 処理時間の長いバッチ処理を実行している場合  
大量のデータ更新を行う、処理時間の長いバッチ処理にインメモリデータ処理を適用すると、効果が期待できます。また、ログレスモードを適用すると、さらに処理時間を短縮できます。  
バッチ処理に掛かる時間が長いほど、ディスク入出力回数に差が出るため、インメモリデータ処理を適用したときの効果が期待できます。
- 複数のバッチ処理を連続して実行している場合  
複数のバッチ処理を連続して実行する場合にインメモリデータ処理を適用すると効果が期待できます。  
バッチ処理を連続して行うほど、ディスク入出力回数に差が出るため、インメモリデータ処理を適用したときの効果が期待できます。
- グローバルバッファの排他競合が原因で性能が上がらない場合  
グローバルバッファの排他競合が原因で性能が上がらない場合に、インメモリデータ処理を適用すると性能向上が期待できます。  
グローバルバッファを使用する場合（インメモリデータ処理を使用しない場合）、参照または更新処理の発生時、グローバルバッファ上にデータがキャッシュされているかどうかを検索する処理（キャッシュサーチ処理）が行われます。このとき、グローバルバッファの全ページに対して排他が掛かるため、ほかの参照または更新処理は排他待ちとなります。インメモリデータ処理の場合は、インメモリデータバッファ上に全データがキャッシュされているため、キャッシュサーチ処理が発生しません。グローバルバッファの全ページに対する排他制御がなくなるため、同時実行性の向上が見込まれ、その分性能向上が期待できます。例えば、同じグローバルバッファを割り当てている複数の RD エリアを更新するバッチ処理などで排他競合が多発している場合に、インメモリデータ処理を適用すると性能向上が期待できます。  
なお、排他競合の発生頻度については、統計解析ユティリティのグローバルバッファに関する統計情報で確認できます。
- メインフレームからデータを移行する場合  
メインフレームの膨大なデータを HiRDB に移行する場合（データを HiRDB のデータベースにデータロードする場合）に、インメモリデータ処理を適用すると効果が期待できます。

---

### 参考

インメモリデータ処理はユティリティにも適用できます。データロードや表の再編成など、処理時間の長いユティリティを実行する場合にも、効果が期待できます。

---

## 1.4 インメモリデータ処理を適用しても効果が期待できないケース

---

次のような場合にインメモリデータ処理を適用しても効果が期待できません。

- 処理時間の短いバッチ処理を実行している場合  
インメモリデータ処理では、RD エリア内の全データをインメモリ化します。そのため、RD エリアのデータ量に比例して、インメモリ化に掛かる時間や、インメモリデータバッファ上のデータをディスクに書き込む時間が長くなります。これらの処理に掛かる時間とバッチ業務の処理時間を考慮した上で、インメモリデータ処理の適用を検討してください。  
例えば、処理時間が 30 分のバッチ業務にインメモリデータ処理を適用したとします。インメモリデータ処理の適用によってバッチ業務の処理時間は 20 分と短くなりましたが、インメモリ化の処理に 10 分、インメモリデータバッファ上のデータをディスクに書き込む時間が 10 分掛かると、合計 40 分掛かり、インメモリデータ処理を適用する前より処理時間が長くなってしまいます。

---

### 参考

この説明で使用している時間はあくまで例です。業務の内容によって各処理に掛かる時間は変わります。

- 
- RD エリア内の全データをグローバルバッファ上にキャッシュしている場合  
RD エリア内の全データをグローバルバッファ上にキャッシュしている場合と、インメモリデータ処理を適用した場合では、あまり性能に差がありません。

## 1.5 インメモリデータ処理が適用できないケース

---

ここでは、インメモリデータ処理が適用できないケースについて説明します。

### (1) インメモリデータ処理を適用できない RD エリア

ユーザ用 RD エリア、ユーザ LOB 用 RD エリア、およびリスト用 RD エリアに対してだけ、インメモリデータ処理を適用できます。次に示す RD エリアについては、インメモリデータ処理を適用できません。

- マスタディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ LOB 用 RD エリア
- レジストリ用 RD エリア
- レジストリ LOB 用 RD エリア

なお、ユーザ用 RD エリアおよびユーザ LOB 用 RD エリアについても、オープン契機が SCHEDULE 属性の RD エリアの場合はインメモリ化できません。また、ユーザ用 RD エリアが共用 RD エリアの場合もインメモリ化できません。

### (2) リアルタイム SAN レプリケーション（ログ同期方式を除く）を使用している場合

全同期方式、全非同期方式、またはハイブリッド方式のリアルタイム SAN レプリケーションを使用している場合は、インメモリデータ処理を適用できません。

ログ同期方式のリアルタイム SAN レプリケーションの場合は、インメモリデータ処理を適用できます。ただし、ログ適用サイトのデータベースの更新ができなくなるため、ログレスモードは使用しないでください。

# 2

## インメモリデータ処理を実行するための環境を作る

この章では、インメモリデータ処理を実行するために必ず行う準備と、処理性能を向上させるために考えておくことについて説明します。

---

2.1 必ず行う準備

---

2.2 インメモリデータ処理をより効果的に実行するために考えておくこと

---

## 2.1 必ず行う準備

ここでは、インメモリデータ処理を実行するために必ず行う準備について説明します。準備の手順を次の図に示します。

図 2-1 必ず行う準備

手順	参照先
マシン環境の確認	「2.1.1 マシン環境の確認」
HiRDB Acceleratorのインストール	「2.1.2 HiRDB Acceleratorのインストール」
HiRDB Acceleratorのセットアップ (UNIX限定)	「2.1.3 HiRDB Acceleratorのセットアップ (UNIX限定)」
システム定義の設定	「2.1.4 システム定義の設定」

なお、これ以降の説明は、HiRDBのインストールおよびセットアップは完了していて、すでにHiRDBを運用していることを前提としています。

### 2.1.1 マシン環境の確認

インメモリデータ処理を実行するためには、マシン環境がここで説明する条件を満たしている必要があります。マシン環境について確認してください。

#### (1) 前提プラットフォーム

インメモリデータ処理は次のプラットフォームで実行できます。

- HP-UX (64 ビットモード)
- Solaris (64 ビットモード)
- AIX (64 ビットモード)
- Linux (EM64T), Linux (IPF)
- Windows (Windows (x64), Windows Vista Ultimate, Windows Vista Business, Windows Vista Enterprise)

#### (2) メモリ容量の確認

インメモリデータ処理を実行するためのメモリ容量が十分にあるか確認してください。インメモリデータ処理に必要なメモリ容量は、インメモリ化する RD エリアの容量に、HiRDB がインメモリデータバッファを管理するための領域を加えた値になります。

インメモリデータ処理に必要なメモリ所要量については、マニュアル「HiRDB Version 8 システム導入・設計ガイド」を参照してください。

### (3) OS のオペレーティングシステムパラメタの確認 (UNIX 限定)

次に示す設定をするオペレーティングシステムパラメタの値を見直してください。

- 共用メモリセグメントの合計サイズ
- システム上の共用メモリセグメントの最大数
- 1 プロセス当たりの共用メモリセグメントの最大数

OS のオペレーティングシステムパラメタの見積もりについては、マニュアル「HiRDB Version 8 システム導入・設計ガイド」または OS のマニュアルを参照してください。

## 2.1.2 HiRDB Accelerator のインストール

ここでは、HiRDB Accelerator のインストールおよびアンインストールの手順について説明します。

### (1) インストール手順

HiRDB Accelerator のインストール手順を次に示します。

#### Windows 版の場合

1. pdstop コマンドで HiRDB を正常終了します。
2. [コントロールパネル] の [サービス] で HiRDB のサービスを停止します。
3. HiRDB Accelerator をインストールします。Administrators 権限を持つユーザがインストールを実行してください。手順はインストーラの指示に従ってください。  
なお、HiRDB/ パラレルサーバの場合は、すべてのサーバマシンに HiRDB Accelerator をインストールしてください。系切り替え機能を使用している場合は予備系のサーバマシンにも、HiRDB Accelerator をインストールしてください。  
また、マルチ HiRDB 構成のときは、セットアップ識別子の指定が必要です。セットアップ識別子は pdntenv コマンドで確認できます。

#### UNIX 版の場合

1. pdstop コマンドで HiRDB を正常終了します。
2. 日立 PP インストーラを実行して HiRDB Accelerator をインストールします。  
なお、HiRDB/ パラレルサーバの場合は、すべてのサーバマシンに HiRDB Accelerator をインストールしてください。  
また、系切り替え機能を使用している場合は予備系のサーバマシンに、リアルタイム SAN レプリケーションを使用している場合はリモートサイトのサーバマシンにも、HiRDB Accelerator をインストールしてください。

## 2. インメモリデータ処理を実行するための環境を作る

### ! 注意事項

HiRDB をバージョンアップした場合（例えば、バージョン 08-03 から 09-00 にバージョンアップした場合など）は、HiRDB Accelerator を再インストールする必要があります。

## (2) アンインストール手順

HiRDB Accelerator のアンインストール手順を次に示します。

### Windows 版の場合

1. pdstop コマンドで HiRDB を正常終了します。
2. [コントロールパネル] の [サービス] で HiRDB のサービスを停止します。
3. [コントロールパネル] の [アプリケーションの追加と削除] から HiRDB Accelerator をアンインストールします。

### UNIX 版の場合

1. pdstop コマンドで HiRDB を正常終了します。
2. pdsetup -d コマンドで HiRDB Accelerator を OS から削除します。
3. 日立 PP インストーラを実行して HiRDB Accelerator をアンインストールします。

## 2.1.3 HiRDB Accelerator のセットアップ (UNIX 限定)

UNIX 版の場合、HiRDB Accelerator のセットアップをしてください。セットアップは、pdopsetup コマンドで実行します。

なお、HiRDB/ パラレルサーバの場合、すべてのサーバマシンで pdopsetup コマンドを実行する必要があります。

## 2.1.4 システム定義の設定

インメモリデータ処理を実行するには、次に示すオペランドを指定してください。

- pd\_max\_resident\_rdarea\_no  
インメモリ化する RD エリアの最大数を指定します。このオペランドに 1 以上の値を指定すると、インメモリデータ処理を実行できます。  
OS がページ固定に対応している Windows で、このオペランドに 1 以上の値を指定した場合は、SHMMAX オペランドに Windows の Large Page のページサイズに切り上げた値を指定してください。  
Windows がページ固定に対応しているかどうか、および Large Page のページサイズについては、pdntenv -os コマンドで確認してください。
- pd\_max\_resident\_rdarea\_shm\_no  
インメモリデータバッファが使用する共用メモリセグメントの最大数を指定します。



**!** 注意事項

インメモリ化する RD エリアには、`pdbuffer` オペランドでグローバルバッファが割り当てられている必要があります。グローバルバッファが割り当てられていない RD エリアは、インメモリ化できません。

---

## 2.2 インメモリデータ処理をより効果的に実行するために考えておくこと

ここでは、インメモリデータ処理をより効果的に実行するために検討するとよいポイントについて説明します。あらかじめインメモリデータ処理に適したシステム構成にしておくことで、処理性能の向上が期待できます。

### (1) どの RD エリアをインメモリ化するか

インメモリデータ処理をより効果的に実行するには、表格納 RD エリアだけではなく、インデクス格納 RD エリアも一緒にインメモリ化してください。

特に、次に示す RD エリアはまとめてインメモリ化してください。

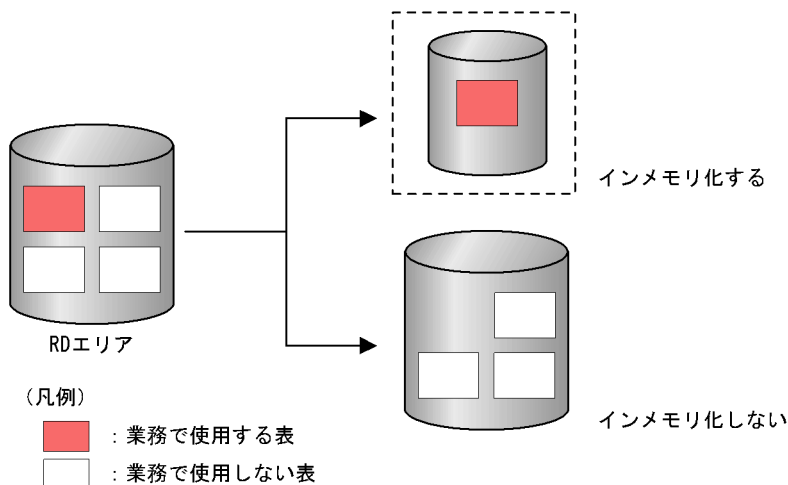
- 表に定義されているインデクスを格納している RD エリア
- 表を横分割している場合、その横分割表が格納されているすべての RD エリア

### (2) RD エリア内の表構成にむだがないか

インメモリ化する RD エリアに、業務ではアクセスしない表が混在している場合、そのような表は別の RD エリアに格納することをお勧めします。これによって、メモリ領域がむだに確保されることを防ぎます。

RD エリア内に使用しない表が混在している場合の対処例を次の図に示します。

図 2-2 RD エリア内に使用しない表が混在している場合の対処例



### (3) 一括指定ができる RD エリア名になっているか

インメモリデータ処理に使用するコマンド (pdmemdb コマンド, pdhold コマンド) は、

RD エリア名一括指定ができます。RD エリア名一括指定ができるように、RD エリア名の末尾をそろえておくなどの工夫をすると便利です。

(例)

インメモリ化する RD エリア名の末尾を次のようにそろえます。

RDDATA01\_mem, RDDATA02\_mem, RDIDX01\_mem, RDIDX02\_mem

上記四つの RD エリアをまとめてインメモリ化する場合、次のように指定できます。

```
pdmembdb -k stay -r "*_mem"
```

運用コマンドでの RD エリア名一括指定については、マニュアル「HiRDB Version 8 コマンドリファレンス」を参照してください。

#### (4) ログレスモードを適用できるか

インメモリデータ処理でバッチ業務を実行する場合、ログレスモードで実行すると、より処理時間を短縮できます。バッチ業務を実行する場合は、ログレスモードの適用を検討してください。



# 3

## インメモリデータ処理を適用する

この章では、業務の種類に応じたインメモリデータ処理の適用例について説明します。

---

3.1 基本的な運用の流れ

---

3.2 バッチ業務に適用する場合

---

3.3 オンライン業務に適用する場合

---

3.4 テスト業務に適用する場合

---

3.5 複数の RD エリアを一度にインメモリ化する場合の注意事項

---

## 3.1 基本的な運用の流れ

---

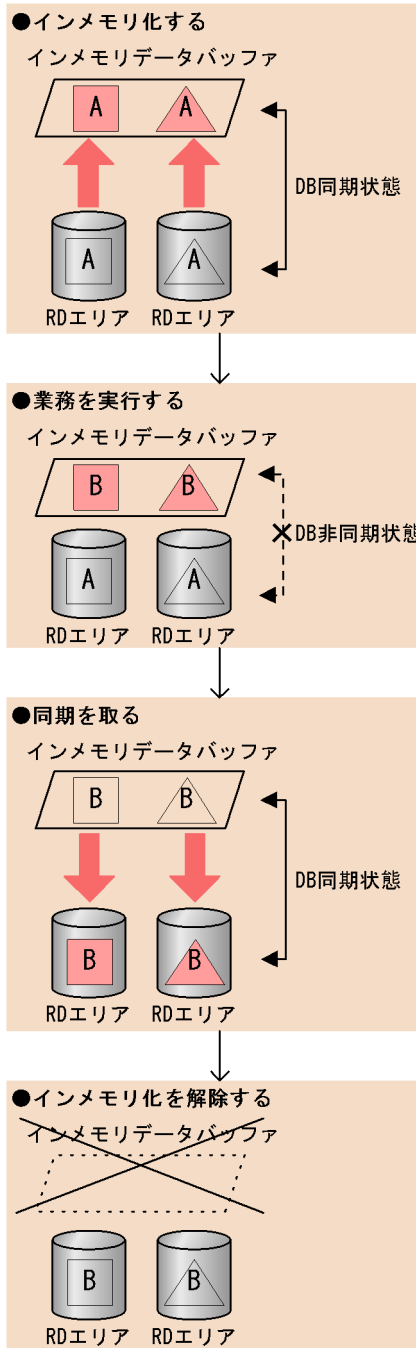
ここでは、インメモリデータ処理を実行するときの基本的な運用の流れについて説明します。

運用には、大きく分けて次の四つのステップがあります。

1. RD エリアをインメモリ化する
2. 業務を実行する
3. インメモリデータバッファと RD エリアの同期を取る
4. インメモリ化を解除する

基本的な運用の流れを次の図に示します。

図 3-1 基本的な運用の流れ



インメモリデータバッファが作成され、RDエリアのデータがインメモリデータバッファに読み込まれます。このとき、DB同期状態になります。

インメモリデータバッファのデータだけが更新され、RDエリアのデータは更新されません。このとき、DB非同期状態になります。

インメモリデータバッファのデータがRDエリアに書き込まれます。このとき、再びDB同期状態になります。

インメモリ化を解除するとインメモリデータバッファがなくなります。これ以降は、グローバルバッファを使用します。

### 3. インメモリデータ処理を適用する

#### ポイント

---

RD エリアのデータとインメモリデータバッファのデータが同じ場合を DB 同期状態といい、異なる場合を DB 非同期状態といいます。

ステップ 2. で業務を実行している間は、インメモリデータバッファ上のデータだけが更新され、RD エリアのデータは更新されないため、DB 非同期状態になります。インメモリデータバッファと RD エリアの状態は、pddb1s コマンドで確認できます。詳細は、「6.2 インメモリデータバッファの利用状況を確認するには」を参照してください。

---

#### ！ 注意事項

ステップ 1. のインメモリ化は、RD エリアのデータ量に比例して時間が掛かります。その間はシンクポイントが取得されないため、注意してください。

---

この基本的な運用の流れを基に、3.2 ではバッチ業務に適用する場合の手順、3.3 ではオンライン業務に適用する場合の手順について説明します。



## 3.2 バッチ業務に適用する場合

ここでは、インメモリデータ処理をバッチ業務に適用する場合の運用について、例題を使って説明します。

### (1) 例題の条件

この例題での条件を次に示します。

- 二つのバッチ業務を連続して実行します。
- バッチ業務でアクセスする RD エリアは、RDDATA01 (表格納 RD エリア) および RDIDX01 (インデクス格納 RD エリア) です。
- ログレスモードでバッチ業務を実行します。

運用の流れを次の図に示します。

図 3-2 バッチ業務に適用する場合の運用の流れ



### 参考

初期データロードにもインメモリデータ処理を適用できます。その際、長時間シンクポイントを取得できなくなるのを防ぐため、同期点指定のデータロードの適用を検討してください。同期点指定のデータロードについては、マニュアル「HiRDB Version 8 コマンドリファレンス」を参照してください。

初期データロードにインメモリデータ処理を適用する場合の手順は、バッチ業務に適用する場合の手順と同じです。(2)(b) および (d) のバッチ業務を pdload コマンドによる初期データロードに置き換えてお読みください。

なお、初期データロードの場合、(2)(a) のバックアップの取得は必要ありません。

### (2) コマンド実行手順

コマンドの実行手順を次に示します。

### 3. インメモリデータ処理を適用する

#### (a) RD エリアをインメモリ化する

1. RDDATA01 および RDIDX01 を閉塞クローズ状態にします。

```
pdhold -r RDDATA01,RDIDX01 -c
```

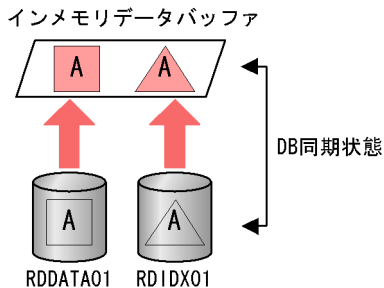
2. システムログファイルをスワップします。

```
pdlogswap -d sys -w
```

3. RDDATA01 および RDIDX01 をインメモリ化します。

```
pdmemdb -k stay -r RDDATA01,RDIDX01
```

このとき、インメモリデータバッファと RD エリアは、DB 同期状態になります。



なお、ここでは、複数の RD エリアをまとめてインメモリ化しています。この場合の注意事項については、「3.5 複数の RD エリアを一度にインメモリ化する場合の注意事項」を参照してください。

4. RDDATA01 および RDIDX01 の閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDIDX01 -o
```

5. RDDATA01 および RDIDX01 のバックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r  
-r RDDATA01,RDIDX01  
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

バックアップの取得方法については、マニュアル「HiRDB Version 8 システム運用ガイド」を参照してください。

## ポイント

ここでは、RD エリアの閉塞クローズを解除してから参照可能モード (-Mr) でバックアップを取得しています。通常と異なり、参照可能モードでバックアップを取得している間も更新ができます。このため、バックアップの取得中にバッチ業務を実行できます。インメモリ RD エリアのバックアップ取得の仕組みについては、「4.1 バックアップを取得するときの気をつけること」を参照してください。

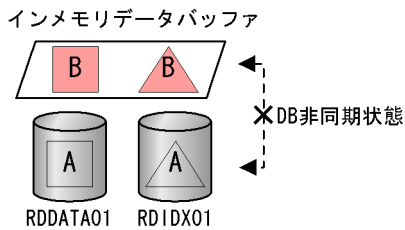
**!** 注意事項

インメモリ化するときの RD エリアの閉塞は、pdhold -c コマンドで行ってください。pdhold -c コマンドで閉塞クローズ状態にすることによって、ほかのトランザクションが対象 RD エリアにアクセスできないようにします。

## (b) 一つ目のバッチ業務を実行する

## 1. バッチ業務を実行します。

インメモリデータバッファの内容が更新され、DB 非同期状態になります。



## (c) インメモリデータバッファと RD エリアの同期を取る

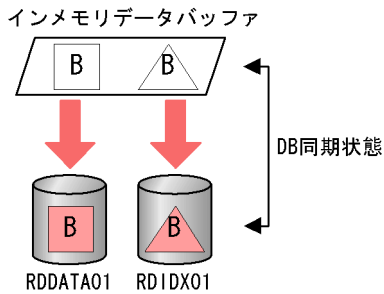
## 1. インメモリデータバッファと RD エリアの同期を取ります。

```
pdhold -r RDDATA01,RDIDX01 -b
```

一つ目のバッチ業務が終わったら、インメモリデータバッファのデータを RD エリアに書き込み、インメモリデータバッファと RD エリアの同期を取ります。これによって、DB 同期状態になります。

pdhold -b コマンドを実行すると、インメモリデータバッファのデータを RD エリアに書き込みます。

### 3. インメモリデータ処理を適用する



#### 2. システムログファイルをスワップします。

```
pdlogswap -d sys -w
```

#### 3. RDDATA01 および RDIDX01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDIDX01
```

#### 4. RDDATA01 および RDIDX01 のバックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r  
-r RDDATA01,RDIDX01  
-b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

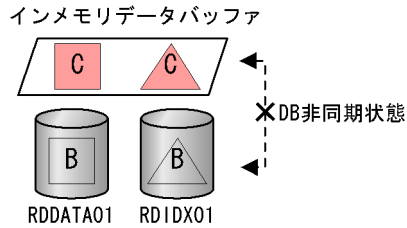
#### ポイント

ログレスモードで業務を実行しているため、一つ目のバッチ業務が終わったら、必ずインメモリデータバッファと RD エリアの同期を取ります。ここで同期を取らないと、二つ目のバッチ業務を実行中にバッファ障害が発生した場合、一つ目のバッチ業務から再実行しなくてはなりません。

#### (d) 二つ目のバッチ業務を実行する

##### 1. バッチ業務を実行します。

インメモリデータバッファの内容が更新され、DB 非同期状態になります。



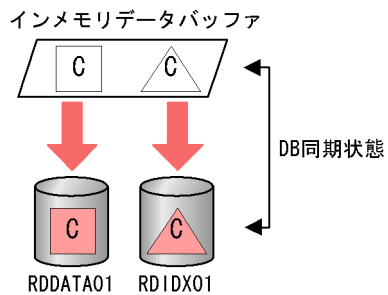
## (e) インメモリデータバッファと RD エリアの同期を取る

1. インメモリデータバッファと RD エリアの同期を取ります。

```
pdhold -r RDDATA01,RDIDX01 -c
```

二つ目のバッチ業務が終わったら、インメモリデータバッファのデータを RD エリアに書き込み、インメモリデータバッファと RD エリアの同期を取ります。これによって、DB 同期状態になります。

pdhold -c コマンドを実行すると、インメモリデータバッファのデータを RD エリアに書き込みます。



**!** 注意事項

ここで、pdhold -c コマンドを使用しているのは、(f) でインメモリ化を解除するためです。インメモリ化を解除するときの RD エリアの閉塞は、pdhold -c コマンドで行います。pdhold -c コマンドで閉塞クローズ状態にすることによって、ほかのトランザクションが対象 RD エリアにアクセスできないようにします。

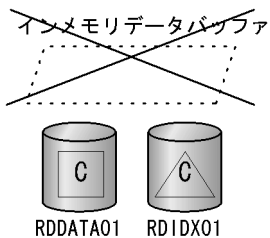
## (f) インメモリ化を解除する

1. RDDATA01 および RDIDX01 のインメモリ化を解除します。

```
pdmembdb -k rels -r RDDATA01,RDIDX01
```

### 3. インメモリデータ処理を適用する

インメモリデータバッファがなくなります。これ以降は、グローバルバッファを使用します。



#### 2. システムログファイルをスワップします。

```
pdlogswap -d sys -w
```

#### 3. RDDATA01 および RDIDX01 のバックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r  
-r RDDATA01,RDIDX01  
-b C:¥pdcopy¥backup03 -p C:¥pdcopy¥list03
```

ここでは前の手順と異なり、RD エリアの閉塞クローズを解除する前にバックアップを取得してください。バックアップを取得する前に RD エリアの閉塞クローズを解除すると、ほかの UAP からの更新が発生する場合があります。更新が発生したあとで RD エリアに障害が発生した場合、ログレスモードでバッチ業務を実行しているため、その更新内容を回復することができません。

#### 4. RDDATA01 および RDIDX01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDIDX01 -o
```

## 3.3 オンライン業務に適用する場合

ここでは、インメモリデータ処理をオンライン業務に適用する場合の運用について、例題を使って説明します。

### (1) 例題の条件

この例題での条件を次に示します。

- オンライン業務でアクセスする RD エリアは、RDDATA01（表格納 RD エリア）および RDIDX01（インデクス格納 RD エリア）です。

運用の流れを次の図に示します。

図 3-3 オンライン業務に適用する場合の運用の流れ



### (2) コマンド実行手順

コマンドの実行手順を次に示します。

#### (a) RD エリアをインメモリ化する

1. RDDATA01 および RDIDX01 を閉塞クローズ状態にします。

```
pdhold -r RDDATA01,RDIDX01 -c
```

2. システムログファイルをスワップします。

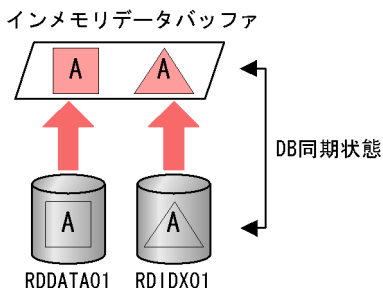
```
pdlogswap -d sys -w
```

3. RDDATA01 および RDIDX01 をインメモリ化します。

```
pdmemdb -k stay -r RDDATA01,RDIDX01
```

### 3. インメモリデータ処理を適用する

このとき、インメモリデータバッファと RD エリアは、DB 同期状態になります。



なお、ここでは、複数の RD エリアを一度にインメモリ化しています。この場合の注意事項については、「3.5 複数の RD エリアを一度にインメモリ化する場合の注意事項」を参照してください。

### 4. RDDATA01 および RDIDX01 の閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDIDX01 -o
```

### 5. RDDATA01 および RDIDX01 のバックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r  
-r RDDATA01,RDIDX01  
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

バックアップの取得方法については、マニュアル「HiRDB Version 8 システム運用ガイド」を参照してください。

### ポイント

ここでは、RD エリアの閉塞クローズを解除してから参照可能モード (-Mr) でバックアップを取得しています。通常と異なり、参照可能モードでバックアップを取得している間も更新ができます。このため、バックアップの取得中にオンライン業務を実行できます。インメモリ RD エリアのバックアップ取得の仕組みについては、「4.1 バックアップを取得するときに気をつけること」を参照してください。

### ！ 注意事項

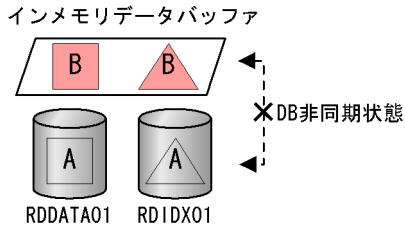
インメモリ化するときの RD エリアの閉塞は、pdhold -c コマンドで行ってください。pdhold -c コマンドで閉塞クローズ状態にすることによって、ほかのトランザクションが対象 RD エリアにアクセスできないようにします。



## (b) オンライン業務を実行する

## 1. オンライン業務を実行します。

これによって、インメモリデータバッファの内容が更新され、DB 非同期状態になります。



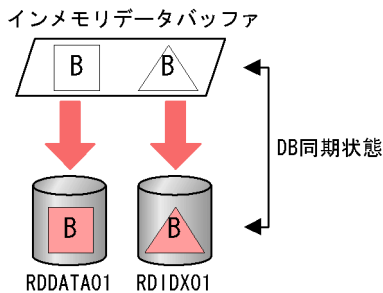
## (c) インメモリデータバッファと RD エリアの同期を取る

## 1. インメモリデータバッファと RD エリアの同期を取ります。

```
pdhold -r RDDATA01,RDIDX01 -c
```

オンライン業務が終わったら、インメモリデータバッファのデータを RD エリアに書き込み、インメモリデータバッファと RD エリアの同期を取ります。これによって、DB 同期状態になります。

pdhold -c コマンドを実行すると、インメモリデータバッファのデータを RD エリアに書き込みます。

**!** 注意事項

ここで、pdhold -c コマンドを使用しているのは、(d) でインメモリ化を解除するためです。インメモリ化を解除するときの RD エリアの閉塞は、pdhold -c コマンドで行います。pdhold -c コマンドで閉塞クローズ状態にすることによって、ほかのトランザクションが対象 RD エリアにアクセスできないようにします。

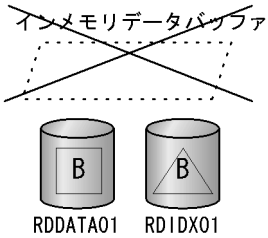
### 3. インメモリデータ処理を適用する

#### (d) インメモリ化を解除する

1. RDDATA01 および RDIDX01 のインメモリ化を解除します。

```
pdmembdb -k rels -r RDDATA01,RDIDX01
```

インメモリデータバッファがなくなります。これ以降は、グローバルバッファを使用します。



2. RDDATA01 および RDIDX01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDIDX01 -o
```

## 3.4 テスト業務に適用する場合

---

ここでは、インメモリデータバッファの更新内容を破棄したり、RD エリアのデータをインメモリデータバッファに再読み込みしたりするなどの運用方法を、インメモリデータ処理をテスト業務に適用する場合の応用例を使って説明します。

### (1) 例題の条件

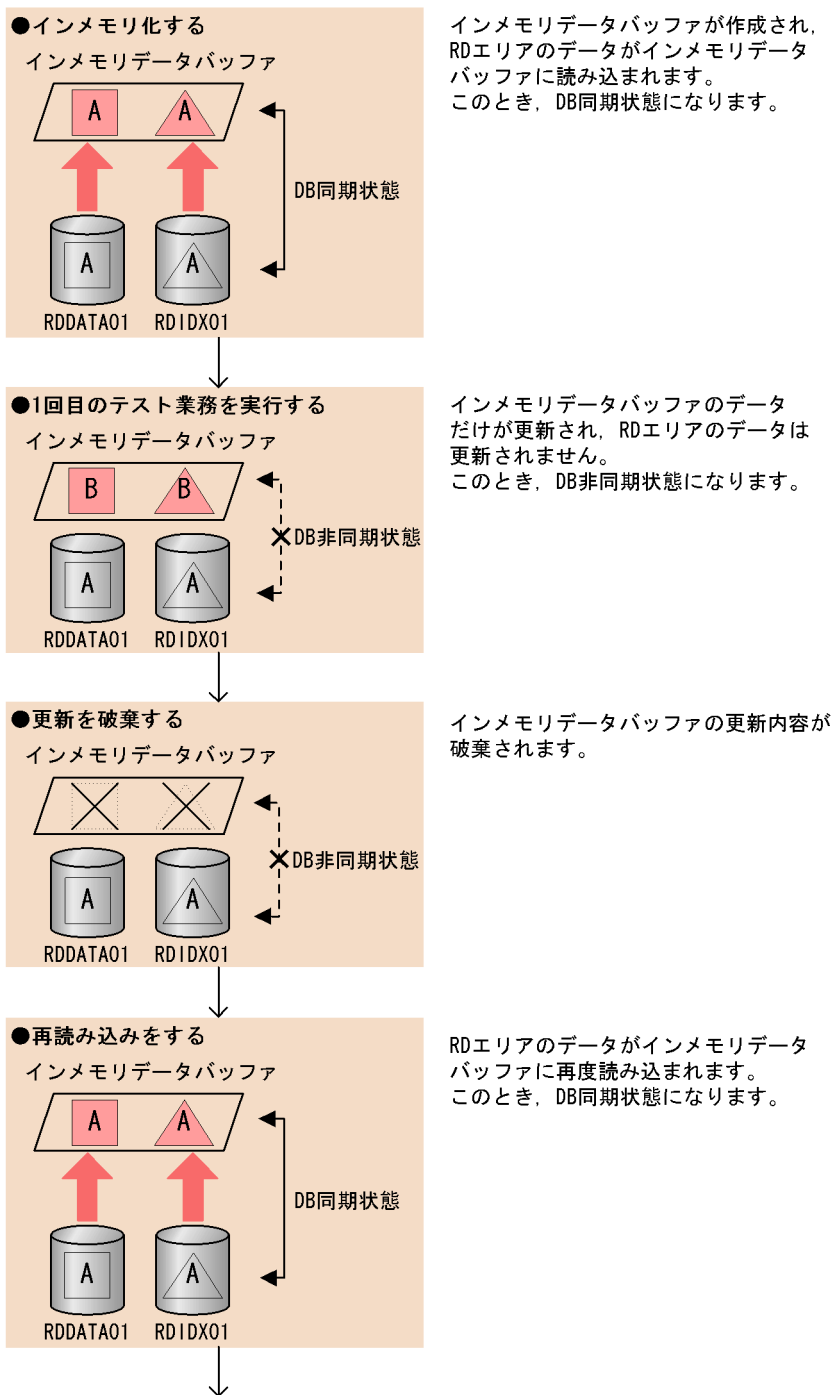
この例題での条件を次に示します。

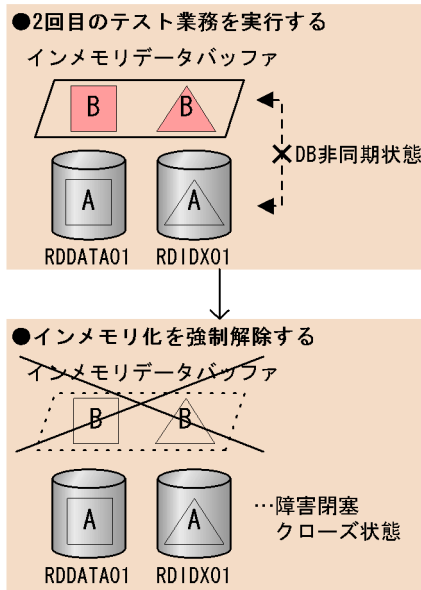
- テスト業務を連続して 2 回実行します。
- テスト業務でアクセスする RD エリアは、RDDATA01 ( 表格納 RD エリア ) および RDIDX01 ( インデクス格納 RD エリア ) です。
- テスト業務の更新内容は、RD エリアに書き込まないで破棄します。

運用の流れを次の図に示します。

### 3. インメモリデータ処理を適用する

図 3-4 テスト業務に適用する場合の運用の流れ





インメモリデータバッファのデータだけが更新され、RDエリアのデータは更新されません。  
このとき、DB非同期状態になります。

インメモリ化を解除すると、インメモリデータバッファがなくなります。  
これ以降は、グローバルバッファを使用します。

## (2) コマンド実行手順

コマンドの実行手順を次に示します。

### (a) RD エリアをインメモリ化する

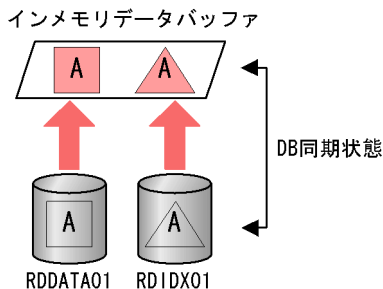
1. RDDATA01 および RDIDX01 を閉塞クローズ状態にします。

```
pdhold -r RDDATA01,RDIDX01 -c
```

2. RDDATA01 および RDIDX01 をインメモリ化します。

```
pdmemdb -k stay -r RDDATA01,RDIDX01
```

このとき、インメモリデータバッファとRDエリアは、DB同期状態になります。



### 3. インメモリデータ処理を適用する

なお、ここでは、複数の RD エリアをまとめてインメモリ化しています。この場合の注意事項については、「3.5 複数の RD エリアを一度にインメモリ化する場合の注意事項」を参照してください。

#### ! 注意事項

インメモリ化するときの RD エリアの閉塞は、pdhold -c コマンドで行ってください。  
pdhold -c コマンドで閉塞クローズ状態にすることによって、ほかの UAP が対象 RD エリアにアクセスできないようにします。

#### 3. RDDATA01 および RDIDX01 の閉塞クローズを解除します。

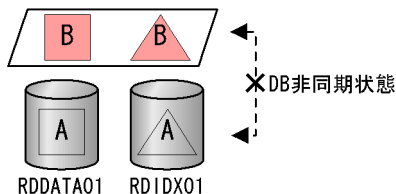
```
pdrels -r RDDATA01,RDIDX01 -o
```

#### (b) テスト業務を実行する

##### 1. 1 回目のテスト業務を実行します。

インメモリデータバッファの内容が更新され、DB 非同期状態になります。

インメモリデータバッファ



#### (c) インメモリデータバッファ上の更新内容を破棄する

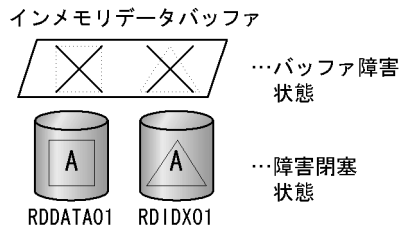
##### 1. RDDATA01 および RDIDX01 を閉塞状態にします。

```
pdhold -r RDDATA01,RDIDX01
```

##### 2. インメモリデータバッファ上のデータを破棄して更新を無効にします。

```
pdmemdb -k cancel -r RDDATA01,RDIDX01
```

このとき、インメモリデータバッファはバッファ障害状態に、RD エリアは障害閉塞状態になります。



(d) RD エリアのデータを再読み込みする

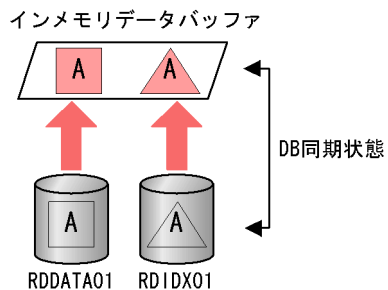
1. RDDATA01 および RDIDX01 をクローズ状態にします。

```
pdclose -r RDDATA01,RDIDX01
```

2. RD エリアのデータをインメモリデータバッファへ再読み込みします。

```
pdmemdb -k reload -r RDDATA01,RDIDX01
```

これによって、テスト業務実行前の状態に戻ります。このとき、インメモリデータバッファと RD エリアは、DB 同期状態になります。



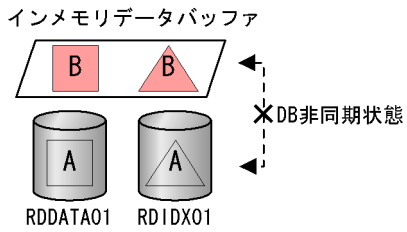
3. RDDATA01 および RDIDX01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDIDX01
```

(e) テスト業務を実行する

1. 2 回目のテスト業務を実行します。  
インメモリデータバッファの内容が更新され、DB 非同期状態になります。

### 3. インメモリデータ処理を適用する



#### (f) インメモリ化を強制解除する

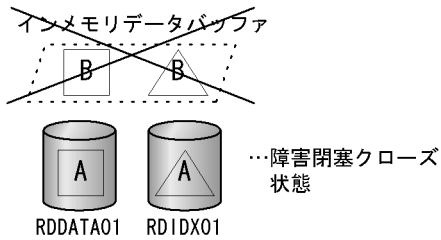
1. RDDATA01 および RDIDX01 を閉塞状態にします。

```
pdhold -r RDDATA01,RDIDX01
```

2. RDDATA01 および RDIDX01 のインメモリ化を強制解除します。

```
pdmemdb -k rels -r RDDATA01,RDIDX01 -d
```

これによって、インメモリデータバッファはなくなり、RD エリアは障害閉塞クローズ状態になります。なお、更新情報は RD エリアに反映されません。



3. RDDATA01 および RDIDX01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDIDX01 -o
```



## 3.5 複数の RD エリアを一度にインメモリ化する場合の注意事項

複数の RD エリアをインメモリ化する場合、一度にまとめてインメモリ化する方法と、個別にインメモリ化する方法があります。運用が煩雑になるのを防ぐため、関連する RD エリアはまとめてインメモリ化することをお勧めしますが、その場合は次のことに注意してください。

### (1) インメモリ化に失敗したとき

複数の RD エリアのうち、一つでもインメモリ化に失敗した場合は、一緒に指定したそのほかの RD エリアもインメモリ化しません。この場合は、`pdmembdb -k stay` コマンドを再実行してください。

なお、HiRDB/パラレルサーバの場合は、バックエンドサーバごとにこの仕組みが適用されます。

#### (例)

```
pdmembdb -k stay -r RD01,RD02,RD03,RD04
```

RD01 と RD02 はバックエンドサーバ 1 で、RD03 と RD04 はバックエンドサーバ 2 で管理されています。RD02 のインメモリ化に失敗した場合、RD01 もインメモリ化しません。RD03 と RD04 はインメモリ化します。

#### ! 注意事項

インメモリ化時だけでなく、インメモリ化の解除、再読み込み、インメモリデータバッファの破棄を行うときにも、この仕組みが適用されます。pdmembdb コマンドで複数の RD エリアを操作したときに、一部の RD エリアに対する処理でエラーが発生すると、すべての RD エリアの処理を無効化します。

### (2) セグメントの割り当てについて

インメモリデータバッファもグローバルバッファと同様に、共用メモリセグメント上に割り当てられます。

インメモリデータバッファに割り当てられる共用メモリセグメントを効率的に使用するために、複数の RD エリアをまとめてインメモリ化した場合は、インメモリ化を解除するときもまとめて解除してください。共用メモリセグメントの割り当ての仕組みを次に示す例を使って説明します。

#### (例)

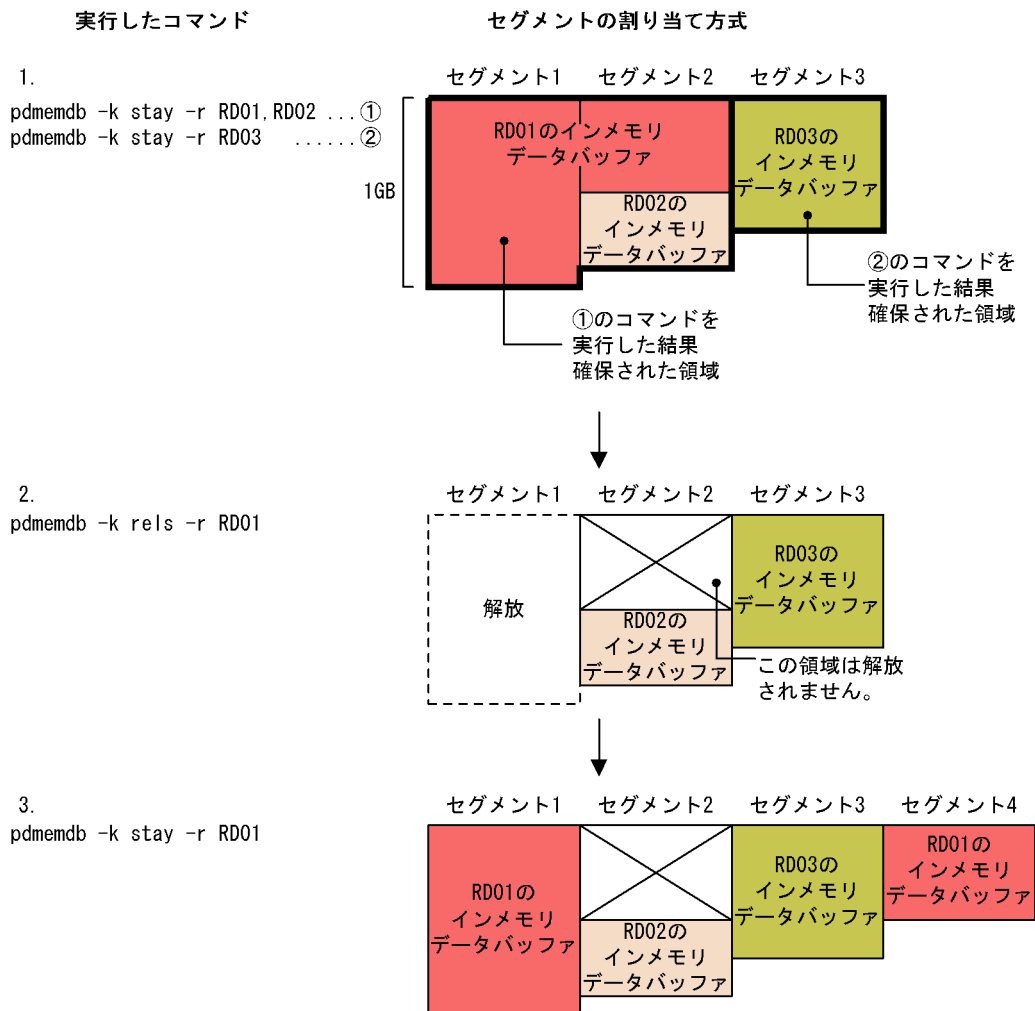
- 共用メモリセグメントの最大サイズ：1GB
- RD エリア (RD01) のサイズ：1.5GB
- RD エリア (RD02) のサイズ：0.4GB

### 3. インメモリデータ処理を適用する

- RD エリア (RD03) のサイズ : 0.7GB

この条件下での共用メモリセグメントの割り当ての仕組みを次の図に示します。

図 3-5 共用メモリセグメントの割り当ての仕組み



[ 説明 ]

- RD01 と RD02 をまとめてインメモリ化し、それとは別に RD03 をインメモリ化しました。  
RD01 のインメモリデータバッファはセグメント 1 およびセグメント 2 に割り当てられます。RD02 のインメモリデータバッファは RD01 と同じセグメント 2 に割り当てられます。  
RD03 のインメモリデータバッファは RD01 , RD02 とは別のセグメント (セグメント 3) に割り当てられます。
- RD01 のインメモリ化を解除すると、セグメント 1 は解放されて利用できます

が、セグメント2のRD01のインメモリデータバッファに割り当てられた領域は解放されません。そのため、このメモリ領域がむだになります。

3. RD01を再度インメモリ化すると、新しいセグメント（セグメント1とセグメント4）が割り当てられます。

ポイント

---

メモリ領域がむだになるのを防ぐため、複数のRDエリアをまとめてインメモリ化した場合は、インメモリ化の解除もまとめて行うようにしてください。

---



# 4

## 運用時に気をつけること

インメモリデータ処理を実行する場合に、そのほかの機能や運用で注意が必要なときがあります。この章では、そのような注意事項について説明します。

---

4.1 バックアップを取得するときに気をつけること

---

4.2 HiRDB を終了するときに気をつけること

---

4.3 データベース構成変更ユティリティを使用するときに気をつけること

---

4.4 インナレプリカ機能を使用しているときに気をつけること

---

4.5 系切り替え機能を使用しているときに気をつけること

---

4.6 その他の注意事項

---

## 4.1 バックアップを取得するときに気をつけること

ここでは、インメモリ RD エリアのバックアップを取得するときの注意点について説明します。

なお、インメモリ RD エリアのバックアップを取得する場合、バックアップ取得モードは参照・更新不可能モード (-M<sub>x</sub>)、または参照可能モード (-M<sub>r</sub>) のどちらかです。更新可能モード (-M<sub>s</sub>) でのバックアップ取得はできません。

また、インメモリ RD エリアの差分バックアップは取得できません。

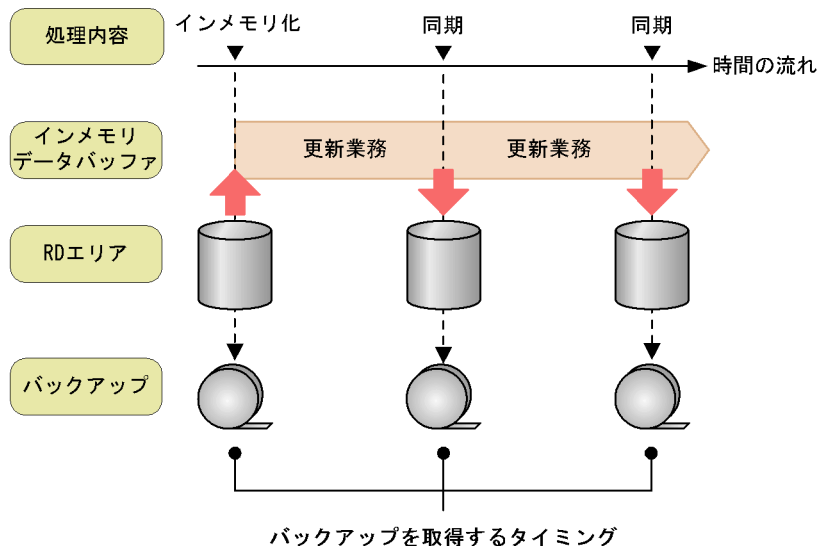
### 4.1.1 バックアップを取得するタイミング

バックアップは、次のタイミングで取得してください。

- RD エリアをインメモリ化したあと
- インメモリデータバッファと RD エリアの同期を取ったあと

バックアップを取得するタイミングを次の図に示します。

図 4-1 バックアップを取得するタイミング



RD エリアとインメモリデータバッファの両方に障害が発生した場合、pdrstr コマンドで RD エリアを回復します。前記で説明した時点でバックアップを取得しないと、インメモリ化時点または同期時点（例えば、バッチ業務の開始前の時点）に RD エリアを回復するときの手順が煩雑になります。

また、バックアップを取得するときは、インメモリ RD エリア（表格納 RD エリア）だけではなく、関連する RD エリア（インデクス格納 RD エリアや、ユーザ LOB 用 RD エリア）も一緒にバックアップを取得してください。関連する RD エリアと一緒にバックアップを取得しておかないで、インメモリ RD エリアだけをバックアップ取得時点で回復すると、関連する RD エリアとデータの不整合が発生するおそれがあります。

#### ポイント

---

バックアップを取得するときは、次のことに注意してください。

- DB 同期状態のときにバックアップを取得する
  - 関連する RD エリアはまとめてバックアップを取得する
- 

### 4.1.2 参照可能モード (-Mr) でバックアップを取得する場合

参照可能モードでバックアップを取得する場合、インメモリデータ処理を実行していないときの運用と異なる点について説明します。

#### (1) 同時に更新業務を実行できる

参照可能モードでバックアップを取得する場合、同時に更新業務を実行できます（RD エリアのバックアップを取得しながら、インメモリデータバッファの内容を更新できます）。

#### (2) バックアップ取得の仕組み

参照可能モードでバックアップを取得する場合、通常はグローバルバッファの更新情報が RD エリアに書き込まれてから、バックアップが取得されます。これによって、最新の状態のバックアップが取得できます。

しかし、インメモリ RD エリアのバックアップを参照可能モードで取得する場合、インメモリデータバッファの更新情報は RD エリアに書き込まれません。そのため、いったんインメモリデータバッファと RD エリアの同期を取り、インメモリデータバッファの更新情報を RD エリアに書き込んでからバックアップを取得してください。

#### バックアップ取得の手順

参照可能モードでバックアップを取得する場合の手順を次に示します。

1. pdhold -b コマンドでインメモリデータバッファと RD エリアの同期を取ります。
2. pdlogswap コマンドでシステムログファイルをスワップします。
3. pdrels コマンドで RD エリアの閉塞を解除します。
4. pdcopy -M r コマンドで RD エリアのバックアップを取得します。

バックアップの取得方法については、マニュアル「HiRDB Version 8 システム運用ガイド」を参照してください。

5. 更新業務を開始します。

#### 4. 運用時に気をつけること

##### 参考

「3.2(2)(c) インメモリデータバッファと RD エリアの同期を取る」に、運用の流れの中でのバックアップの取得手順が記載されています。

##### ポイント

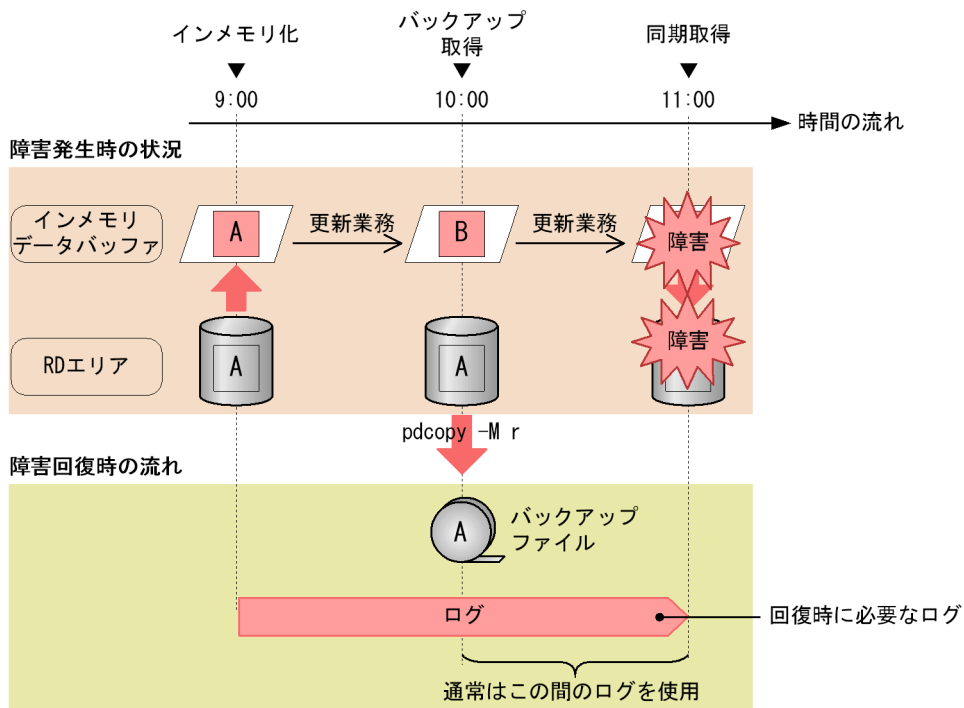
通常はバックアップとバックアップ取得時点以降のログがあれば、RD エリアを最新の状態に回復できますが、DB 非同期状態のときにバックアップを取得した場合、バックアップ取得以前のログも必要になります。

RD エリアを最新の状態に回復するには、インメモリ化した時点（または前回の同期取得時点）からのログが必要になります。DB 非同期状態でバックアップを取得した場合の例を次に示します。

##### DB 非同期状態でバックアップを取得した場合の例

DB 非同期状態でバックアップを取得した場合の例を次の図に示します。

図 4-2 DB 非同期状態でバックアップを取得した場合の例



##### [ 障害発生時の状況 ]

9:00 に RD エリアをインメモリ化し、その後インメモリデータバッファの内容を更新しました。DB 非同期状態のまま、10:00 に参照可能モードで RD エリアのバックアップを取得しました。その後、再び更新業務を行ったあと、更新内



容を RD エリアに書き込むときに、インメモリデータバッファと RD エリアに障害が発生しました。

[ 障害回復時の流れ ]

この場合、バックアップを取得した時刻は 10:00 ですが、バックアップファイルの内容は 9:00 の時点のままになっています。RD エリアを最新 ( 11:00 ) の状態に回復するには、10:00 に取得したバックアップファイルと 9:00 からのログが必要になります。

通常では、バックアップ取得時点より古いログは使用しません。通常と同じ運用ができるように、インメモリ RD エリアのバックアップを参照可能モードで取得する場合は、インメモリデータバッファと RD エリアの同期を取ったあと ( DB 同期状態のとき ) に行うことをお勧めします。

---

参考

参照・更新不可能モード ( -M x ) でバックアップを取得する場合は、必ず RD エリアを `pdhold -c` コマンドで閉塞クローズ状態にするため、インメモリデータバッファの更新情報が RD エリアに書き込まれて DB 同期状態になります。したがって、RD エリアを最新の時点で回復する場合は、バックアップとバックアップ取得時点以降のログを使うという通常と同じ考えで RD エリアを回復できます。

---

## 4.2 HiRDB を終了するときに気をつけること

---

インメモリ RD エリアがある場合は、HiRDB の正常終了、および計画停止ができません。HiRDB を停止する場合は、インメモリ化を解除してから停止するか、強制終了してください。

なお、HiRDB を強制終了した場合、インメモリデータバッファ上の更新情報は RD エリアに反映されません。また、再開後、インメモリ化は解除された状態になります。

### **!** 注意事項

RD エリアは障害閉塞状態になります。pddbls コマンドで RD エリアの状態を確認してから、pdrels コマンドで障害閉塞状態を解除してください。

---

## 4.3 データベース構成変更ユティリティを使用するとき気をつけること

---

インメモリ RD エリアに対しては、データベース構成変更ユティリティを使った次の操作ができません。次の操作をする場合は、インメモリ化を解除してから実行してください。

- RD エリアの拡張
- RD エリアの再初期化
- RD エリアの削除
- RD エリアの移動
- RD エリアの属性変更
- RD エリアのレプリカ定義（インナレプリカ機能）
- RD エリアの構成情報複写（インナレプリカ機能）
- RD エリアの統合（インナレプリカ機能）

## 4.4 インナレプリカ機能を使用しているときに気をつけること

---

インメモリ RD エリアのレプリカ RD エリアは作成できません。また、レプリカ RD エリアをインメモリ化することもできません。オリジナル RD エリアはインメモリ化できますが、オリジナル RD エリアをインメモリ化したまま、更新可能なオンライン再編成を実行することはできません。更新可能なオンライン再編成をする場合は、オリジナル RD エリアのインメモリ化を解除してから実行してください。

### ! 注意事項

インメモリ RD エリアに追いつき状態管理表を作成しないでください。インメモリ RD エリアに追いつき状態管理表を格納した場合、HiRDB が異常終了するなどしてインメモリデータバッファの更新情報がなくなると、追いつき反映ができなくなることがあります。

---

### ポイント

---

- レプリカ RD エリアは常にインメモリ化できません。
  - `pdorbegin` コマンドの実行から `pdorend` コマンドの実行の間はオリジナル RD エリアもインメモリ化できません。
  - インメモリ化している間は `pdorbegin` コマンドを実行できません。
-

## 4.5 系切り替え機能を使用しているときに気をつけること

---

インメモリ RD エリアがある場合に系切り替えが発生すると、インメモリデータバッファがなくなってしまうため、更新情報が RD エリアに反映されません。これは、計画系切り替えや、スローダウンによる系切り替えの場合も同じです。そのため、計画系切り替えは、インメモリ化を解除してから実行してください。

なお、インメモリ RD エリアがある場合に系切り替えが発生したとき、またはインメモリ化を解除しないで計画系切り替えを実行したとき、インメモリ化していた RD エリアは障害閉塞状態になります。この場合は `pdrels` コマンドで閉塞を解除してから、再度インメモリ化してください。

### ! 注意事項

- 系が切り替わると、インメモリデータバッファのデータがなくなり、更新情報が RD エリアに反映されません。また、インメモリ化していた RD エリアは障害閉塞状態になります。
- 1:1 スタンバイレス型系切り替え機能を使用している場合、代替 BES 下の RD エリアのオープン契機は常に `SCHEDULE` 属性になります。この `SCHEDULE` 属性を変えることができないため、代替 BES 下の RD エリアをインメモリ化することはできません。

## 4.6 その他の注意事項

---

ここでは、その他の注意事項について説明します。

### (1) システム用 RD エリアを更新するような操作を行う場合

インメモリ RD エリアに対して、システム用 RD エリアを更新するような操作（例えば、CREATE TABLE や、ALTER TABLE など）を行わないでください。インメモリ RD エリアに対して、システム用 RD エリアを更新するような操作を行うと、HiRDB が異常終了するなどの障害が発生したときに、システム用 RD エリアと同期を合わせて回復する必要があるため、回復作業が難しくなります。

システム用 RD エリアを更新するような操作については、マニュアル「HiRDB Version 8 システム運用ガイド」の「同時にバックアップを取得する必要がある RD エリア」を参照してください。

### (2) セキュリティ監査機能を使用している場合

監査証跡表を格納している RD エリアをインメモリ化しないでください。インメモリ化している間に HiRDB が強制終了または異常終了した場合、インメモリデータバッファ上の更新情報が RD エリア（監査証跡表）に反映されません。

監査証跡表も、ほかの表と同様にデータベース回復ユティリティを使用すれば、更新情報を反映してインメモリ RD エリアを最新の状態に回復できますが、セキュリティ上の観点から、インメモリデータ処理の適用はお勧めしません。

また、監査証跡表の自動データロード機能は更新前ログ取得モードで動作するため、自動データロード機能を使用している場合は、監査証跡表に更新情報を反映できなくなるおそれがあります。

### (3) 改竄防止表を定義している場合

改竄防止表を格納している RD エリアをインメモリ化しないでください。インメモリ化している間に HiRDB が強制終了または異常終了した場合、インメモリデータバッファ上の更新情報が RD エリア（改竄防止表）に反映されません。

改竄防止表も、ほかの表と同様にデータベース回復ユティリティを使用すれば、更新情報を反映してインメモリ RD エリアを最新の状態に回復できますが、セキュリティ上の観点から、インメモリデータ処理の適用はお勧めしません。

### (4) システム構成変更コマンドを実行する場合

インメモリ RD エリアがある場合は、システム構成変更コマンド (pdchgconf) を実行できません。システム構成変更コマンドを実行する場合は、インメモリ化を解除してから行ってください。

(5) 修正版 HiRDB の入れ替えを行う場合

インメモリ RD エリアがある場合は、修正版 HiRDB の入れ替え (pdprgrefresh) を実行できません。修正版 HiRDB の入れ替えを行う場合は、インメモリ化を解除してから行ってください。

(6) RD エリアの自動増分機能を使用する場合

インメモリ RD エリアには、RD エリアの自動増分は適用されません。ただし、インメモリ化を解除すれば、自動増分が適用されます。そのため、インメモリ化を解除したあとに再度 RD エリアの容量不足が発生した場合は、自動的に RD エリアの容量が拡張されます。

(7) アンロードレスシステムログ運用を適用している場合

バックアップ対象 RD エリアにインメモリ RD エリアがある場合は、ログポイント情報ファイルを取得できません。

(8) ローカルバッファを割り当てた場合

インメモリ RD エリアにローカルバッファを割り当てた場合でも、その RD エリアに対するアクセスはインメモリデータバッファを介して行われます。ローカルバッファは使用されません。

(9) 更新可能バックアップ閉塞を実行する場合

インメモリ RD エリアに対しては、更新可能バックアップ閉塞 (pdhold -b -u) を実行できません。





# 5

## トラブルシューティング

この章では、インメモリデータ処理中に障害が発生した場合の回復方法について説明します。

---

5.1 障害回復の基本的な考え方

---

5.2 障害が発生したときに最初に確認すること

---

5.3 RD エリア障害が発生した場合の回復手順

---

5.4 バッファ障害が発生した場合の回復手順

---

## 5.1 障害回復の基本的な考え方

ここでは、インメモリデータ処理中に発生した障害を回復するときの、基本的な考え方について説明します。

### ！ 注意事項

HiRDB が異常終了して再開始した場合、異常終了する前までの更新情報は RD エリアに反映されません (HiRDB が自動的に最新の状態に回復することはありません)。また、インメモリデータバッファはなくなり、インメモリ化が解除された状態になります。

### (1) 障害の種類

インメモリデータ処理中に障害が発生した場合、インメモリデータバッファの状態と RD エリアの状態を確認する必要があります。

インメモリデータ処理中の障害には、次の 3 種類があります。

#### RD エリア障害

RD エリアに障害が発生している状態です。

RD エリアの障害を取り除いたあと、インメモリデータバッファ上のデータを RD エリアに書き込むことで回復できます。インメモリデータバッファ上のデータは最新であるため、RD エリアも最新の状態に回復できます。

#### バッファ障害

インメモリデータバッファに障害が発生している状態です。

最新の状態に回復する場合は、いったんインメモリ化を解除してから、アンロードログファイルを使って `pdrstr` コマンドで RD エリアを最新の状態に回復し、再度インメモリ化します。または、インメモリ化したままアンロードログファイルを使って `pdrstr` コマンドで RD エリアを最新の状態に回復し、その内容をインメモリデータバッファに再読み込みします。

同期取得時点で回復する場合は、いったんインメモリ化を解除してから、RD エリアを再度インメモリ化します。または、RD エリアの内容をインメモリデータバッファに再読み込みします。

#### RD エリア障害かつバッファ障害

インメモリデータバッファと RD エリアの両方に障害が発生している状態です。

最新の状態に回復する場合は、バックアップファイルとアンロードログファイルを使って `pdrstr` コマンドで RD エリアを最新の状態に回復し、その内容を再度インメモリ化します。

同期取得時点で回復する場合は、バックアップファイルを使って `pdrstr` コマンドで RD エリアを同期取得時点で回復し、その内容を再度インメモリ化します。

これらの障害状態については、`pddbls -M` コマンドで確認できます。詳細については、

「5.2 障害が発生したときに最初に確認すること」を参照してください。

ポイント

---

バッファ障害で、かつ RD エリアが障害閉塞クローズ状態のときだけ、データベース回復ユーティリティ (pdrstr) を実行できます。RD エリア障害が発生している場合 (RD エリア障害かつバッファ障害を含む) は、一度インメモリ化を解除してから回復する必要があります。

---

## (2) 回復方法を決めるときの基準

バッファ障害が発生している場合、回復方法は次の条件によって異なります。

- どの時点で回復するか (最新の状態か、または同期取得時点か)
- 同期取得時点で回復する場合、関連 RD エリアがあるか

詳細については、「図 5-2 回復手順の参照先を決めるフローチャート」を参照してください。

## 5.2 障害が発生したときに最初に確認すること

障害が発生したときは、まず `pddbls -M` コマンドを実行して障害発生個所を確認します。  
`pddbls -M` コマンドの実行結果を次の図に示します。

図 5-1 `pddbls -M` コマンドの実行結果

STATE OF RDAREA	ID	STATUS	TYPE
RDAREA		OPNMODE	MEMORY-STATUS
⋮	⋮	⋮	⋮
RDDATA01	8	HOLD	USER
		INITIAL	Y (OBST-MEM)
RDDATA02	9	OPEN	USER
		INITIAL	Y (OBST-DB)
RDDATA03	10	CLOSE HOLD	USER
		INITIAL	Y (OBST-ALL)

(凡例)

: RDエリアとインメモリデータバッファの状態

MEMORY-STATUS に表示される RD エリアとインメモリデータバッファの状態から障害発生個所を確認します。RD エリアとインメモリデータバッファの状態を次の表に示します。

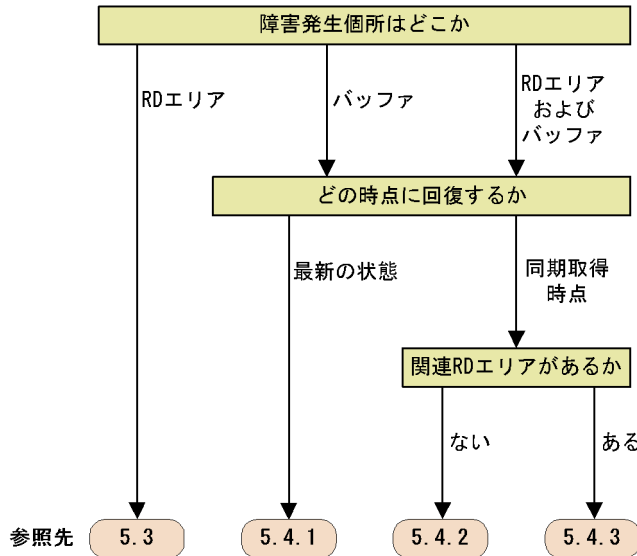
表 5-1 RD エリアとインメモリデータバッファの状態

MEMORY-STATUS の表示	状態	意味
Y(SYNC)	正常	DB 同期状態
Y(ASYNC)	正常	DB 非同期状態
Y(OBST-MEM)	障害発生	バッファ障害状態
Y(OBST-DB)	障害発生	RD エリア障害状態
Y(OBST-ALL)	障害発生	RD エリア障害状態かつバッファ障害状態
N	正常	インメモリ化していない

バッファ障害が発生している場合は、データベースをどの時点で回復するか（最新の状態か、または同期取得時点か）を決めます。また、関連 RD エリアがあるかどうかについても確認してください。回復手順の参照先を決めるフローチャートを次の図に示します。

5.3 以降では、それぞれの回復方法について説明しています。どのパターンに該当するか、次のフローチャートで確認してください。

図 5-2 回復手順の参照先を決めるフローチャート



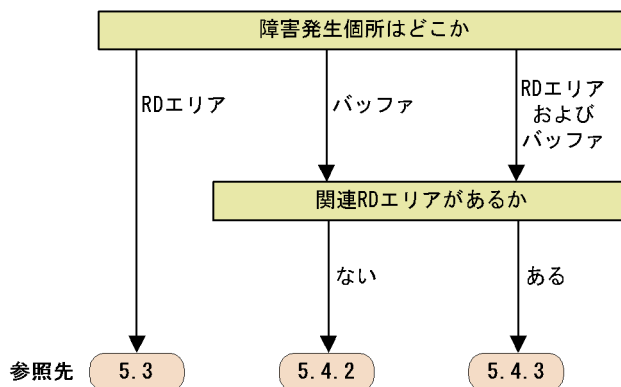
## ポイント

- RD エリア障害だけの場合は、常に最新の状態に回復できます。
- 最新の状態に回復する場合は、関連 RD エリアの有無を意識する必要はありません。
- 同期取得時点とは、DB 同期状態のときを指します（インメモリ化した時点、または RD エリアとインメモリデータバッファの同期を取った時点）。

## ログレスモードでバッチ業務を実行中に障害が発生した場合

ログレスモードでバッチ業務を実行中に障害が発生した場合は、次のフローチャートに従ってください。回復手順の参照先を決めるフローチャート（ログレスモードでバッチ業務を実行した場合）を次の図に示します。

図 5-3 回復手順の参照先を決めるフローチャート（ログレスモードでバッチ業務を実行した場合）



## 5. トラブルシュート

### ポイント

---

- RD エリア障害だけの場合は、インメモリデータバッファ上に最新のデータがあるため、再度インメモリデータバッファとインメモリ RD エリアの同期を取ることで、常に最新の状態に回復できます。
  - バッファ障害が発生している場合（RD エリア障害かつバッファ障害を含む）は、同期取得時点から業務を再実行する必要があります。
  - 同期取得時点とは、DB 同期状態のときを指します（インメモリ化した時点、または RD エリアとインメモリデータバッファの同期を取った時点）。
-

## 5.3 RD エリア障害が発生した場合の回復手順

ここでは、RD エリア障害が発生した場合の回復手順について、例題を使って説明します。

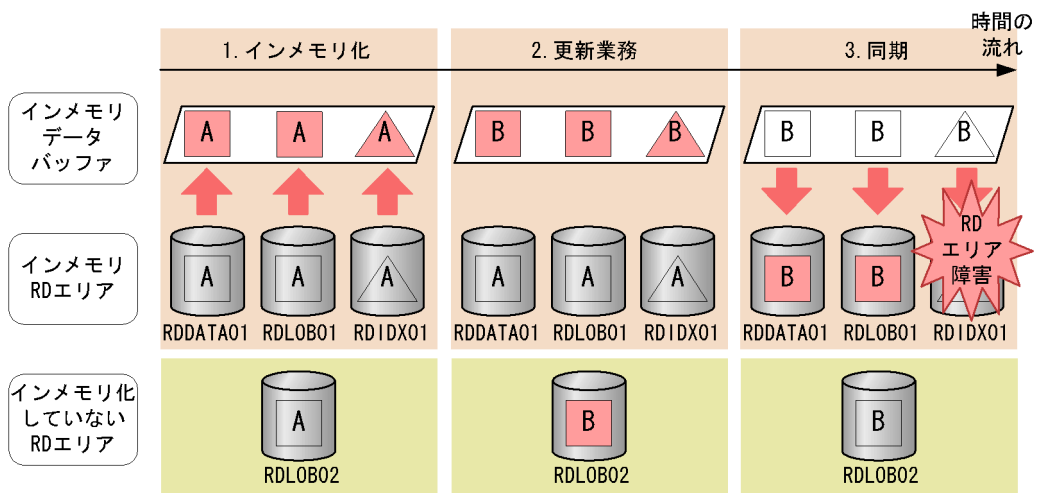
### (1) 障害発生時の状況

RDDATA01 ( 表格納 RD エリア ), RDLOB01 ( LOB データ格納 RD エリア ) および RDIDX01 ( インデクス格納 RD エリア ) をインメモリ化しました。RDLOB02 ( LOB データ格納 RD エリア ) はインメモリ化していません。

また、RDDATA01、RDLOB01 および RDIDX01 は異なる HiRDB ファイルシステム領域に作成されているとします。

障害発生時の状況を次の図に示します。

図 5-4 障害発生時の状況 ( RD エリア障害が発生した場合 )



#### [ 説明 ]

1. RDDATA01、RDLOB01 および RDIDX01 をインメモリ化しました。
2. 更新業務を実行し、インメモリデータバッファと RDLOB02 のデータを更新しました。
3. pdhold -b コマンドでインメモリデータバッファと RD エリアの同期を取りました。このとき、RDDATA01 および RDLOB01 への書き込みは正常にできましたが、RDIDX01 が書き込みエラーによって RD エリア障害になりました。

このときのインメモリデータバッファと各 RD エリアの状態を次に示します。

- インメモリデータバッファの状態：正常 ( 最新の状態 )
- RDDATA01 の状態：正常 ( 最新の状態 )

## 5. トラブルシュート

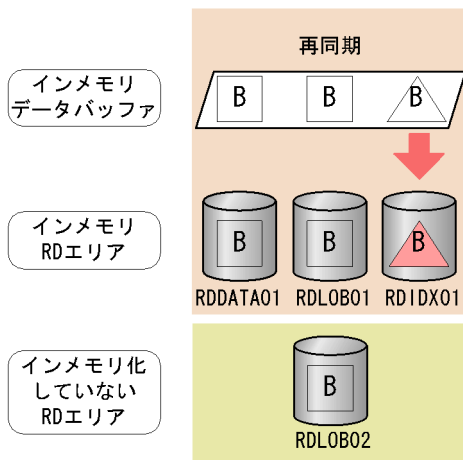
- RDLOB01 の状態：正常（最新の状態）
- RDIDX01 の状態：RD エリア障害
- RDLOB02 の状態：正常（最新の状態）

### （2）障害回復の手順

インメモリデータバッファ上に最新のデータがあるため、インメモリデータバッファと RDIDX01 の同期を再度取り、RDIDX01 を最新の状態に回復します。

障害回復の手順を次の図に示します。

図 5-5 障害回復の手順（RD エリア障害が発生した場合）



コマンドの実行手順を次に示します。

1. サーバプロセスをリフレッシュします。

```
pdpfresh -s BES1
```

ディスク交換を行う場合にこの操作を実行してください。この操作を実行しないとディスクを切り離すことができません。

2. RDDATA01 および RDLOB01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDLOB01
```

pdhold -b コマンドを実行したため、RDDATA01 および RDLOB01 が参照可能バックアップ閉塞状態になっています。閉塞を解除して、RDDATA01 および RDLOB01 に対する更新業務を続行します。

3. RDIDX01 の障害を取り除きます。



ディスク障害が発生した場合は、マニュアル「HiRDB Version 8 システム運用ガイド」の「ディスク障害が発生したときの対処方法」を参照して対処してください。

4. インメモリデータバッファと RDIDX01 の同期を取ります。

```
pdhold -r RDIDX01 -b
```

インメモリデータバッファ上のデータが RDIDX01 に書き込まれました。これによって、RDIDX01 は最新の状態に回復されました。

5. RDIDX01 の閉塞を解除します。

```
pdrels -r RDIDX01
```

#### 参考

RD エリア障害の回復を行う場合は、バックアップファイルを使用しません。

### (3) 障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移

障害回復時、インメモリデータバッファの状態およびインメモリ RD エリアの状態は次の表のように遷移します。表の項番は、「(2) 障害回復の手順」の実行手順の番号と対応しています。なお、表中の実行コマンドは、一部オプションを省略しています。

表 5-2 障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移

項番	実行コマンド	障害が発生した RD エリア		障害が発生していない RD エリア	
		RDIDX01		RDDATA01, RDLOB01	
		コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>	コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>
-	-	RD エリア障害状態	障害閉塞かつオープン状態	DB 同期状態	参照可能バックアップ閉塞状態
1	pdpfresh				
2	pdrels -r RDDATA01, RDL OB01				オープン状態
3	-				
4	pdhold -r RDIDX01 -b	DB 同期状態	参照可能バックアップ閉塞状態		

## 5. トラブルシュート

項番	実行コマンド	障害が発生した RD エリア		障害が発生していない RD エリア	
		RDIDX01		RDDATA01 , RDLOB01	
		コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>	コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>
5	pdrels -r RDIDX01		オープン状態		

(凡例)

: 状態の変化なし

- : 該当しない

注 1 バッファの状態とは、インメモリデータバッファの状態のことです。

注 2 RD エリアの状態とは、インメモリ RD エリアの状態のことです。

## 5.4 バッファ障害が発生した場合の回復手順

ここでは、バッファ障害が発生した場合の回復手順について、例題を使って説明します。また、インメモリデータバッファと RD エリアの両方に障害が発生している場合も、基本的な回復手順は同じであるため、ここで一緒に説明します。

なお、ここで説明する回復手順は、インメモリ化するときバックアップを取得していることを前提としています。バックアップを取得するタイミングについては、「4.1.1 バックアップを取得するタイミング」を参照してください。

### 5.4.1 最新の状態に回復する場合

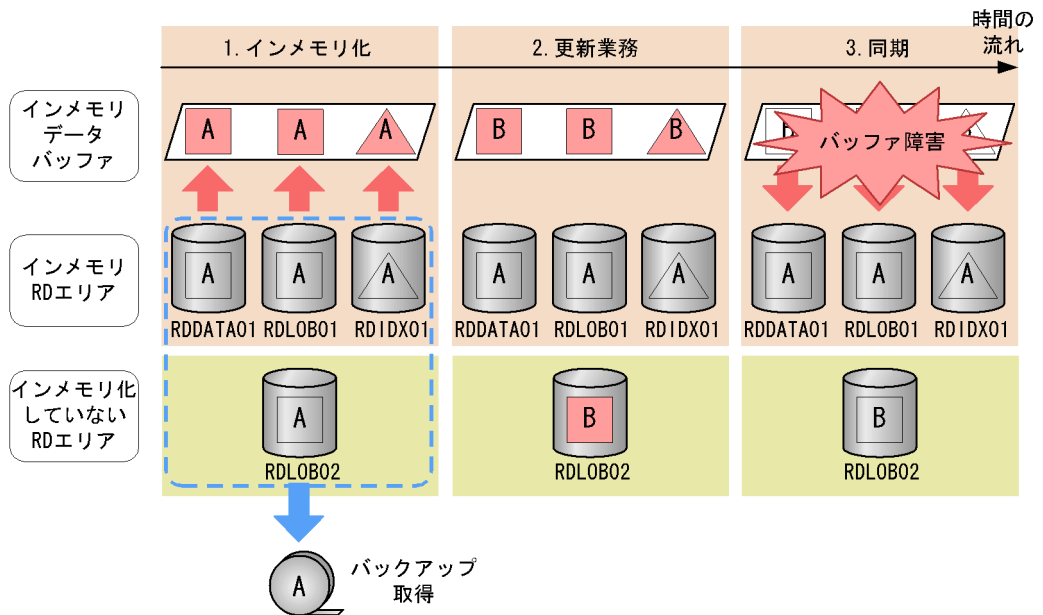
最新の状態に回復する場合について説明します。

#### (1) 障害発生時の状況

RDDATA01 ( 表格納 RD エリア ), RDLOB01 ( LOB データ格納 RD エリア ) および RDIDX01 ( インデクス格納 RD エリア ) をインメモリ化しました。RDLOB02 ( LOB データ格納 RD エリア ) はインメモリ化していません。

障害発生時の状況を次の図に示します。

図 5-6 障害発生時の状況 ( バッファ障害を最新の状態に回復する場合 )



[ 説明 ]

## 5. トラブルシュート

1. RDDATA01, RDLOB01 および RDIDX01 をインメモリ化しました。また、関連する RD エリア RDLOB02 と一緒にバックアップを取得しました。
2. 更新業務を実行し、インメモリデータバッファと RDLOB02 のデータを更新しました。
3. pdhold -b コマンドでインメモリデータバッファと RD エリアの同期を取りました。このとき、バッファ障害になりました。

このときのインメモリデータバッファと各 RD エリアの状態を次に示します。

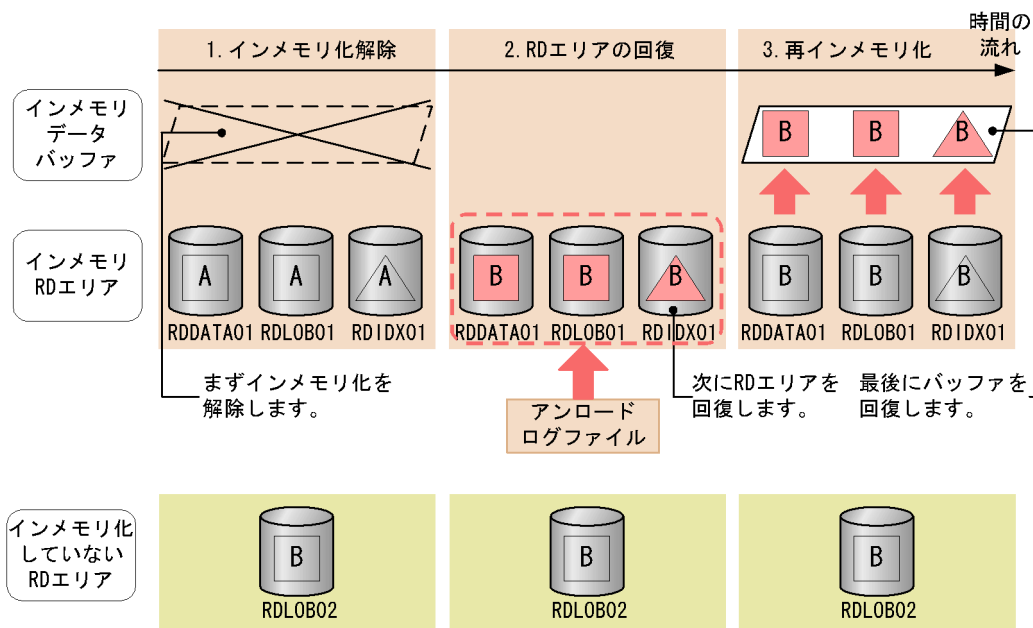
- インメモリデータバッファの状態：バッファ障害
- RDDATA01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）
- RDLOB01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）
- RDIDX01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）
- RDLOB02 の状態：正常（最新の状態）

### （2）障害回復の手順

pdrstr コマンドで RDDATA01, RDLOB01 および RDIDX01 を最新の状態に回復します。そのあとに、RDDATA01, RDLOB01 および RDIDX01 を再度インメモリ化します。

障害回復の手順を次の図に示します。

図 5-7 障害回復の手順（バッファ障害を最新の状態に回復する場合）



#### [ 説明 ]

1. RDDATA01, RDLOB01 および RDIDX01 のインメモリ化を解除します。

2. アンロードログファイルを使って RDDATA01, RDLOB01 および RDIDX01 を最新の状態に回復します (RDDATA01, RDLOB01 および RDIDX01 のデータはバックアップ取得時点と同じ状態であるため、回復時にバックアップファイルは使用しません)。ただし、RD エリア障害とバッファ障害の両方が発生している場合は、バックアップファイルとアンロードログファイルを使って RDDATA01, RDLOB01 および RDIDX01 を回復します。
3. RDDATA01 および RDIDX01 のデータを再度インメモリ化します。

コマンドの実行手順を次に示します。

1. RDDATA01, RDLOB01 および RDIDX01 をクローズ状態にします。

```
pdclose -r RDDATA01,RDLOB01,RDIDX01
```

インメモリ化を強制解除するには、RD エリアの状態が障害閉塞クローズ状態になっている必要があるため、ここで RDDATA01, RDLOB01 および RDIDX01 をクローズしています。

2. RDDATA01, RDLOB01 および RDIDX01 のインメモリ化を強制的に解除します。

```
pdmemdb -k rels -r RDDATA01,RDLOB01,RDIDX01 -d
```

3. 現用のシステムログファイルを調べます。

```
pdlogls -d sys
```

4. システムログファイルをスワップします。

```
pdlogswap -d sys -w
```

5. 現用だったファイルの内容をアンロードします。

```
pdlogunld -d sys -g log01 -o C:¥usr¥hirdb¥pdlogunld¥unldlog01
```

6. RDDATA01, RDLOB01 および RDIDX01 をアンロードログファイルから最新の状態に回復します。

## 5. トラブルシュート

```
pdrstr -m C:¥rdarea¥mast¥mast01 -d C:¥usr¥hirdb¥pdlogunld  
-p C:¥usr¥hirdb¥pdrstr¥list¥list01  
-w C:¥tmp¥sortwork -r RDDATA01,RDLOB01,RDIDX01
```

7. RDDATA01, RDLOB01 および RDIDX01 の障害閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDLOB01,RDIDX01 -o
```

再度インメモリ化するには、RD エリアの状態がコマンド閉塞クローズ状態になっている必要があるため、ここでいったん RDDATA01, RDLOB01 および RDIDX01 の障害閉塞クローズを解除しています。

8. RDDATA01, RDLOB01 および RDIDX01 をコマンド閉塞クローズ状態にします。

```
pdhold -r RDDATA01,RDLOB01,RDIDX01 -c
```

9. RDDATA01, RDLOB01 および RDIDX01 を再度インメモリ化します。

```
pdmemdb -k stay -r RDDATA01,RDLOB01,RDIDX01
```

これによって、インメモリデータバッファは最新の状態に回復されました。

10. RDDATA01, RDLOB01 および RDIDX01 の閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDLOB01,RDIDX01 -o
```

これによって、業務が再開できるようになりました。

RD エリア障害かつバッファ障害の場合

RD エリア障害とバッファ障害の両方が発生している場合は、手順 6. でバックアップファイルとアンロードログファイルから RD エリアを最新の状態に回復してください。それ以外の手順については、バッファ障害の場合と同じです。

### (3) 障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移

障害回復時、インメモリデータバッファの状態およびインメモリ RD エリアの状態は次の表のように遷移します。表の項番は、「(2) 障害回復の手順」の実行手順の番号と対応しています。なお、表中の実行コマンドは、一部オプションを省略しています。

表 5-3 障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移

項番	実行コマンド	コマンド実行後の バッファの状態 1	コマンド実行後の RD エリアの状態 2
			RDDATA01, RDLOB01, RDIDX01
-	-	バッファ障害状態	障害閉塞かつオープン状態
1	pdclose -r RDDATA01, RDLOB01, RD IDX01		障害閉塞かつクローズ状態
2	pdmemdb -k rels -r RDDATA01, RDLOB01, RD IDX01 -d	未使用状態	
3	pdlogls -d sys		
4	pdlogswap -d sys -w		
5	pdlogunld -d sys		
6	pdrstr -r RDDATA01, RDLOB01, RD IDX01		
7	pdrels -r RDDATA01, RDLOB01, RD IDX01 -o		オープン状態
8	pdhold -r RDDATA01, RDLOB01, RD IDX01 -c		コマンド閉塞かつクローズ状態
9	pdmemdb -k stay -r RDDATA01, RDLOB01, RD IDX01	DB 同期状態	
10	pdrels -r RDDATA01, RDLOB01, RD IDX01 -o		オープン状態

(凡例)

: 状態の変化なし

- : 該当しない

注 1 バッファの状態とは、インメモリデータバッファの状態のことです。

注 2 RD エリアの状態とは、インメモリ RD エリアの状態のことです。

## 5.4.2 同期取得時点に回復する場合 (関連 RD エリアなし)

同期取得時点に回復する場合について説明します。

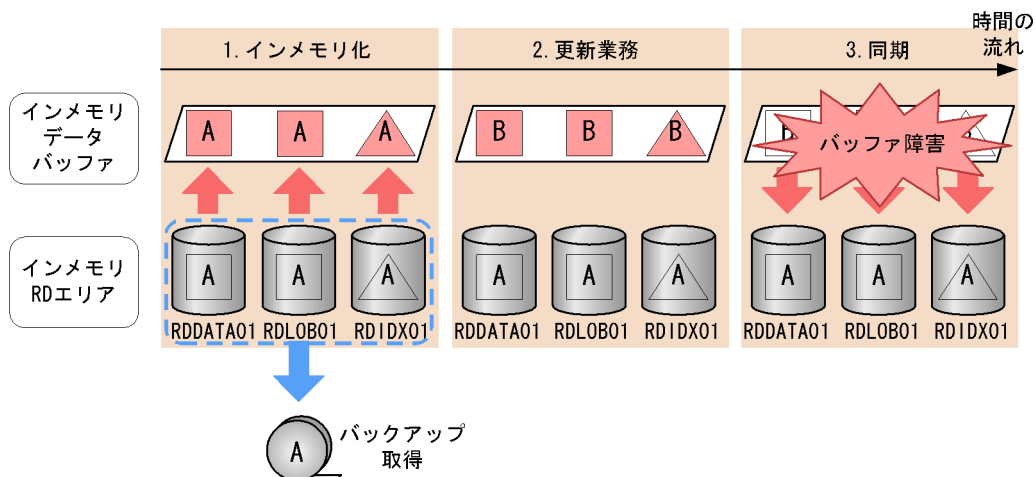
### (1) 障害発生時の状況

RDDATA01 ( 表格納 RD エリア ), RDLOB01 ( LOB データ格納 RD エリア ) および RDIDX01 ( インデクス格納 RD エリア ) をインメモリ化しました。

## 5. トラブルシュート

障害発生時の状況を次の図に示します。

図 5-8 障害発生時の状況（バッファ障害を同期取得時点で回復する場合（関連 RD エリアなし））



### [ 説明 ]

1. RDDATA01, RDLOB01 および RDIDX01 をインメモリ化しました。また、このとき、バックアップを取得しました。
2. 更新業務を実行し、インメモリデータバッファ上のデータを更新しました。
3. pdhold -b コマンドでインメモリデータバッファと RD エリアの同期を取ろうとしましたが、バッファ障害によって、RDDATA01, RDLOB01 および RDIDX01 との同期が取れませんでした。

このときのインメモリデータバッファと各 RD エリアの状態を次に示します。

- インメモリデータバッファの状態：バッファ障害
- RDDATA01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）
- RDLOB01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）
- RDIDX01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）

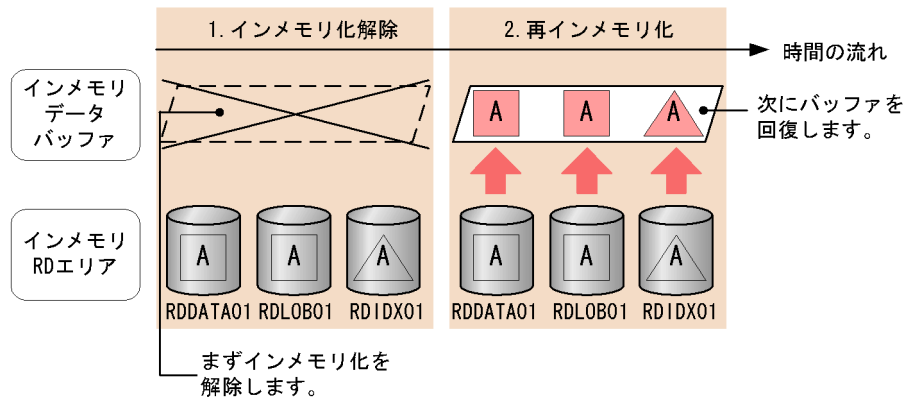
### (2) 障害回復の手順

RDDATA01, RDLOB01 および RDIDX01 中のデータは同期取得時（インメモリ化時）のままです。RDDATA01, RDLOB01 および RDIDX01 のデータを再度インメモリ化してインメモリデータバッファを同期取得時点で回復します。

障害回復の手順を次の図に示します。



図 5-9 障害回復の手順 (バッファ障害を同期取得時点に回復する場合 (関連 RD エリアなし))



[ 説明 ]

1. RDDATA01, RDLOB01 および RDIDX01 のインメモリ化を解除します。
2. RDDATA01, RDLOB01 および RDIDX01 を再度インメモリ化します。  
ただし、RD エリア障害とバッファ障害の両方が発生している場合は、バックアップファイルを使って RDDATA01, RDLOB01 および RDIDX01 を回復したあとにインメモリ化します。

コマンドの実行手順を次に示します。

1. RDDATA01, RDLOB01 および RDIDX01 をクローズ状態にします。

```
pdclose -r RDDATA01,RDLOB01,RDIDX01
```

インメモリ化を強制解除するには、RD エリアの状態が障害閉塞クローズ状態になっている必要があるため、ここで RDDATA01, RDLOB01 および RDIDX01 をクローズしています。

2. RDDATA01, RDLOB01 および RDIDX01 のインメモリ化を強制的に解除します。

```
pdmemdb -k rels -r RDDATA01,RDLOB01,RDIDX01 -d
```

3. RDDATA01, RDLOB01 および RDIDX01 の障害閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDLOB01,RDIDX01 -o
```

再度インメモリ化するには、RD エリアの状態がコマンド閉塞クローズ状態になっている必要があるため、ここでいったん RDDATA01, RDLOB01 および RDIDX01 の

## 5. トラブルシュート

障害閉塞クローズを解除しています。

4. RDDATA01, RDLOB01 および RDIDX01 をコマンド閉塞クローズ状態にします。

```
pdhold -r RDDATA01,RDLOB01,RDIDX01 -c
```

5. RDDATA01, RDLOB01 および RDIDX01 を再度インメモリ化します。

```
pdmemb -k stay -r RDDATA01,RDLOB01,RDIDX01
```

これによって、インメモリデータバッファは同期取得時点の状態に回復されました。

6. RDDATA01, RDLOB01 および RDIDX01 の閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDLOB01,RDIDX01 -o
```

これによって、業務が再開できるようになりました。

RD エリア障害かつバッファ障害の場合

RD エリア障害とバッファ障害の両方が発生している場合は、手順 2. でインメモリ化を解除したあと、バックアップファイルから RD エリアを同期取得時点に回復してください。それ以外の手順については、バッファ障害の場合と同じです。

### (3) 障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移

障害回復時、インメモリデータバッファの状態およびインメモリ RD エリアの状態は次の表のように遷移します。表の項番は、「(2) 障害回復の手順」の実行手順の番号と対応しています。なお、表中の実行コマンドは、一部オプションを省略しています。

表 5-4 障害回復時のインメモリデータバッファおよびインメモリ RD エリアの状態遷移

項番	実行コマンド	コマンド実行後の バッファの状態 1	コマンド実行後の RD エリアの状態 2
			RDDATA01, RDLOB01, RDIDX01
-	-	バッファ障害状態	障害閉塞かつオープン状態
1	pdclose -r RDDATA01,RDLOB01,RD IDX01		障害閉塞かつクローズ状態
2	pdmemb -k rels -r RDDATA01,RDLOB01,RD IDX01 -d	未使用状態	

項番	実行コマンド	コマンド実行後の バッファの状態 1	コマンド実行後の RD エリアの状態 2
			RDDATA01, RDLOB01, RDIDX01
3	pdrels -r RDDATA01, RDLOB01, RD IDX01 -o		オープン状態
4	pdhold -r RDDATA01, RDLOB01, RD IDX01 -c		コマンド閉塞かつクローズ状態
5	pdmemb -k stay -r RDDATA01, RDLOB01, RD IDX01	DB 同期状態	
6	pdrels -r RDDATA01, RDLOB01, RD IDX01 -o		オープン状態

(凡例)

- : 状態の変化なし
- : 該当しない

注 1 バッファの状態とは、インメモリデータバッファの状態のことです。

注 2 RD エリアの状態とは、インメモリ RD エリアの状態のことです。

### 5.4.3 同期取得時点で回復する場合 (関連 RD エリアあり)

同期取得時点で回復する場合について説明します。関連する RD エリアがあるため、それらも一緒に同期取得時点に戻します。

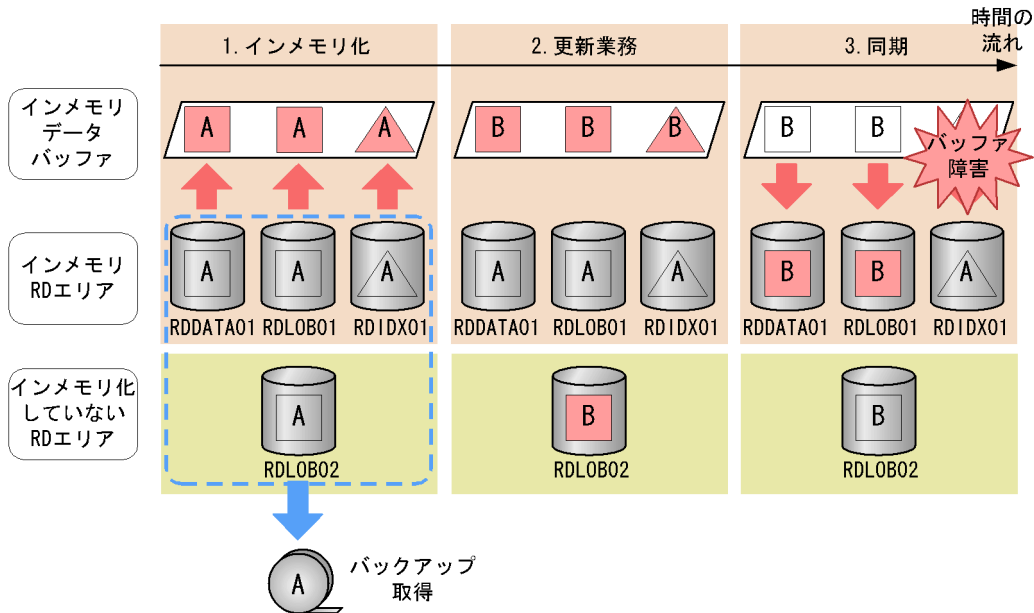
#### (1) 障害発生時の状況

RDDATA01 ( 表格納 RD エリア ), RDLOB01 ( LOB データ格納 RD エリア ) および RDIDX01 ( インデクス格納 RD エリア ) をインメモリ化しました。RDLOB02 ( LOB データ格納 RD エリア ) はインメモリ化していません。

障害発生時の状況を次の図に示します。

## 5. トラブルシュート

図 5-10 障害発生時の状況（バッファ障害を同期取得時点で回復する場合（関連 RD エリアあり））



### [ 説明 ]

1. RDDATA01, RDLOB01 および RDIDX01 をインメモリ化しました。また、このとき、関連する RD エリア RDLOB02 も一緒にバックアップを取得しました。
2. 更新業務を実行し、インメモリデータバッファと RDLOB02 のデータを更新しました。
3. `pdhold -b` コマンドでインメモリデータバッファと RD エリアの同期を取りました。このとき、RDDATA01 および RDLOB01 への書き込みは正常にできましたが、RDIDX01 への書き込み中にバッファ障害になりました。

このときのインメモリデータバッファと各 RD エリアの状態を次に示します。

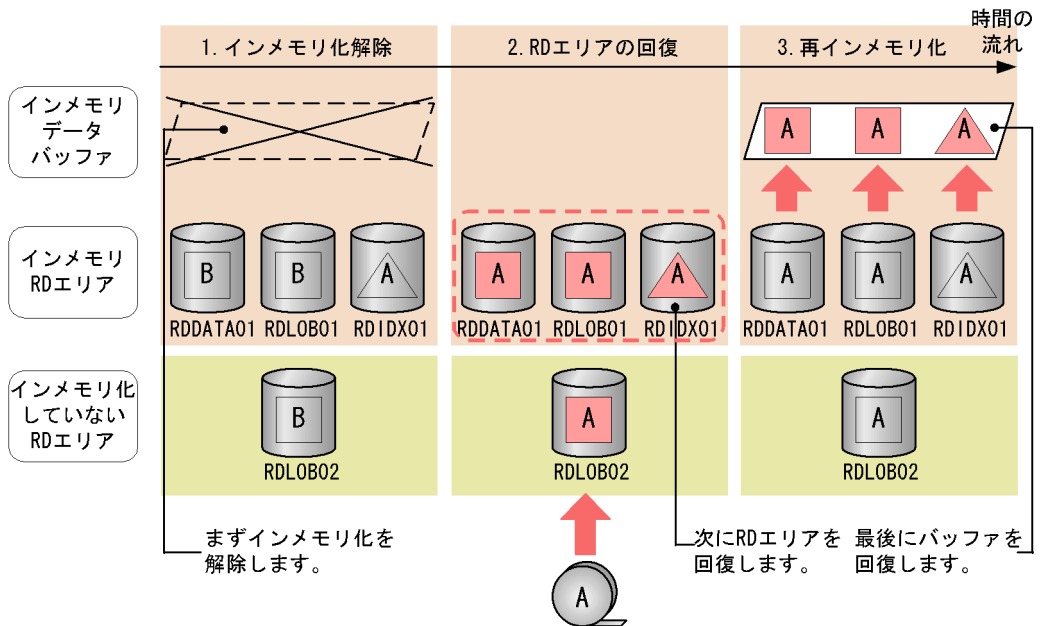
- インメモリデータバッファの状態：バッファ障害
- RDDATA01 の状態：正常（最新の状態）
- RDLOB01 の状態：正常（最新の状態）
- RDIDX01 の状態：障害閉塞状態（同期取得時（インメモリ化時）の状態）
- RDLOB02 の状態：正常（最新の状態）

### (2) 障害回復の手順

`pdrstr` コマンドで RDDATA01, RDLOB01, RDIDX01 および RDLOB02 を同期取得時点（インメモリ化時点）に回復します。そのあとに、RDDATA01, RDLOB01 および RDIDX01 のデータをインメモリデータバッファに再度インメモリ化します。

障害回復の手順を次の図に示します。

図 5-11 障害回復の手順（バッファ障害を同期取得時点で回復する場合（関連 RD エリアあり））



#### [ 説明 ]

1. RDDATA01, RDLOB01 および RDIDX01 のインメモリ化を解除します。
2. バックアップファイルを使って RDDATA01, RDLOB01, RDIDX01 および RDLOB02 を同期取得時点で回復します。
3. RDDATA01, RDLOB01 および RDIDX01 を再度インメモリ化します。

コマンドの実行手順を次に示します。

1. RDDATA01 および RDLOB01 の閉塞を解除します。

```
pdrels -r RDDATA01,RDLOB01
```

pdhold -b コマンドを実行したため、RDDATA01 および RDLOB01 が参照可能バックアップ閉塞状態になっています。インメモリ化を解除するにはコマンド閉塞クローズ状態になっている必要があるため、いったん参照可能バックアップ閉塞を解除しています。

2. RDDATA01, RDLOB01 および RDLOB02 をコマンド閉塞クローズ状態にします。

## 5. トラブルシュート

```
pdhold -r RDDATA01,RDLOB01,RDLOB02 -c
```

3. RDDATA01 および RDLOB01 のインメモリ化を解除します。

```
pdmembdb -k rels -r RDDATA01,RDLOB01
```

4. RDIDX01 をクローズ状態にします。

```
pdclose -r RDIDX01
```

インメモリ化を強制解除するには、RD エリアの状態が障害閉塞クローズ状態になっている必要があるため、ここで RDIDX01 をクローズしています。

5. RDIDX01 のインメモリ化を強制的に解除します。

```
pdmembdb -k rels -r RDIDX01 -d
```

6. RDDATA01, RDLOB01, RDIDX01, および RDLOB02 をバックアップファイルから同期取得時点に回復します。

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥usr¥hirdb¥pdcopy¥backup01  
-p C:¥usr¥hirdb¥pdrstr¥list¥list01 -r RDDATA01,RDLOB01,RDIDX01,RDLOB02
```

7. RDIDX01 の障害閉塞クローズを解除します。

```
pdrels -r RDIDX01 -o
```

再度インメモリ化するには、RD エリアの状態がコマンド閉塞クローズ状態になっている必要があるため、ここでいったん RDIDX01 の障害閉塞クローズを解除しています。

8. RDIDX01 をコマンド閉塞クローズ状態にします。

```
pdhold -r RDIDX01 -c
```

9. RDDATA01, RDLOB01 および RDIDX01 を再度インメモリ化します。

```
pdmembdb -k stay -r RDDATA01,RDLOB01,RDIDX01
```

これによって、インメモリデータバッファは同期取得時点の状態に回復されました。

10.RDDATA01, RDLOB01, RDIDX01, および RDLOB02 の閉塞クローズを解除します。

```
pdrels -r RDDATA01,RDLOB01,RDIDX01,RDLOB02 -o
```

これによって、業務が再開できるようになりました。

### (3) 障害回復時のインメモリデータバッファ, インメモリ RD エリアおよびインメモリ化していない RD エリアの状態遷移

障害回復時, 障害回復時のインメモリデータバッファ, インメモリ RD エリアおよびインメモリ化していない RD エリアの状態は次の表のように遷移します。表の項番は, 「(2) 障害回復の手順」の実行手順の番号と対応しています。なお, 表中の実行コマンドは, 一部オプションを省略しています。

表 5-5 障害回復時のインメモリデータバッファ, インメモリ RD エリアおよびインメモリ化していない RD エリアの状態遷移

項番	実行コマンド	正常に書き込みできた RD エリア		正常に書き込みできなかった RD エリア		インメモリ化していない RD エリア
		RDDATA01, RDLOB01		RDIDX01		
		コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>	コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>	
-	-	DB 同期状態	参照可能バックアップ閉塞状態	バッファ障害状態	障害閉塞かつオープン状態	オープン状態
1	pdrels -r RDDATA01,RDLOB01		オープン状態			
2	pdhold -r RDDATA01,RDLOB01,RDLOB02 -c		コマンド閉塞かつクローズ状態			コマンド閉塞かつクローズ状態

5. トラブルシュート

項番	実行コマンド	正常に書き込みできた RD エリア		正常に書き込みできなかった RD エリア		インメモリ化していない RD エリア	
		RDDATA01, RDLOB01		RDIDX01			RDLOB02
		コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>	コマンド実行後のバッファの状態 <sup>1</sup>	コマンド実行後の RD エリアの状態 <sup>2</sup>		コマンド実行後の RD エリアの状態
3	pdmembdb -k rels -r RDDATA01, RDLOB01	未使用状態					
4	pdclose -r RDIDX01				障害閉塞かつクローズ状態		
5	pdmembdb -k rels -r RDIDX01 -d			未使用状態			
6	pdrstr -r RDDATA01, RDLOB01, RDIDX01, RDLOB02						
7	pdrels -r RDIDX01 -o				オープン状態		
8	pdhold -r RDIDX01 -c				コマンド閉塞かつクローズ状態		
9	pdmembdb -k stay -r RDDATA01, RDLOB01, RDIDX01	DB 同期状態		DB 同期状態			
10	pdrels -r RDDATA01, RDLOB01, RDIDX01, RDLOB02 -o		オープン状態		オープン状態	オープン状態	

(凡例)

: 状態の変化なし

- : 該当しない

注 1 バッファの状態とは、インメモリデータバッファの状態のことです。

注 2 RD エリアの状態とは、インメモリ RD エリアの状態のことです。



# 6

## こんなときどうする

この章では、インメモリデータ処理を実行する場合に知っておくとよい操作について説明します。

- 
- 6.1 インメモリデータバッファと RD エリアの同期を取るには

---

  - 6.2 インメモリデータバッファの利用状況を確認するには

---

  - 6.3 インメモリデータバッファの統計情報を確認するには

---

  - 6.4 強制的にインメモリ化を解除するには

---

  - 6.5 インメモリデータバッファ上のデータを破棄するには

---

  - 6.6 RD エリアのデータをインメモリデータバッファに再読み込みするには
-

## 6.1 インメモリデータバッファと RD エリアの同期を取るには

---

ここでは、インメモリデータバッファと RD エリアの同期を取る方法、およびそのタイミングについて説明します。

### (1) 同期を取る方法

インメモリデータバッファと RD エリアの同期を取る場合、ほかのトランザクションが対象 RD エリアを参照できるように、通常は `pdhold -b` コマンドを使います。

また、`pdhold -b` 以外のコマンドでも、インメモリデータバッファと RD エリアの同期が取れます。インメモリデータバッファと RD エリアの同期を取るコマンドを次に示します。

- `pdhold -b`
- `pdhold -b -w`
- `pdhold -c`
- `pdclose`

なお、どのコマンドを使用しても、同期の取得に掛かる処理時間は同じです。

#### ! 注意事項

- 同期取得中の更新トランザクションに対する処理は、使用するコマンドによって異なります。このことを考慮して同期取得に使用するコマンドを決定してください。または、同期を取るときは更新業務を停止してください。
- コマンド実行後は RD エリアの状態が変わります（バックアップ閉塞、または閉塞かつクローズ状態）。そのため、業務を続行する場合は、`pdrels` コマンドで閉塞を解除する必要があります。
- `pdhold -b` コマンドの場合、リスト用 RD エリアは指定できません。

### (2) 同期を取るタイミング

バッファと RD エリアの同期を取るタイミングは、主に次の二つが考えられます。

- バックアップを取得する前  
バックアップを取得する前には、バッファと RD エリアの同期を取ることをお勧めします。
- バッチ業務を実行したあと  
バッチ業務を実行したあとには、バッチ実行後のインメモリデータバッファ障害に備えて、バッファと RD エリアの同期を取ることをお勧めします。

## 6.2 インメモリデータバッファの利用状況を確認するには

インメモリデータバッファの利用状況は、pddbIs -M コマンドを実行することで確認できます。コマンドの実行例と実行結果を次に示します。

[ コマンド実行例 ]

```
pddbIs -r ALL -M
```

[ 実行結果 ]

STATE OF RDAREA	ID	STATUS	TYPE
RDAREA		OPNMODE	MEMORY-STATUS
RDMAST	1	OPEN	MAST
		INITIAL	N
RDDIR	2	OPEN	DDIR
		INITIAL	N
RDDICT	3	OPEN	DDIC
		INITIAL	N
RDDATA01	4	HOLD (BU I)	USER
		INITIAL	Y (SYNC)
RDIDX01	5	HOLD (BU I)	USER
		INITIAL	Y (SYNC)
RDLOB01	6	HOLD (BU I)	USER
		INITIAL	N
RDDATA02	7	OPEN	USER
		INITIAL	Y (ASYNC)
RDIDX02	8	OPEN	USER
		INITIAL	Y (ASYNC)
RDDATA03	9	OPEN	USER
		INITIAL	N
RDIDX03	10	OPEN	USER
		INITIAL	N

(凡例)

: インメモリRDエリア

: RDエリアとインメモリデータバッファの状態

[ 説明 ]

インメモリデータバッファの利用状況は、MEMORY-STATUS の値で確認します。

Y : インメモリ化しています。

N : インメモリ化していません。

また、RD エリアとインメモリデータバッファの状態は、次の六つのどれかになります。

## 6. こんなときどうする

MEMORY-STATUS の表示	意味
Y(SYNC)	DB 同期状態
Y(ASYNC)	DB 非同期状態
Y(OBST-MEM)	バッファ障害状態
Y(OBST-DB)	RD エリア障害状態
Y(OBST-ALL)	RD エリア障害状態かつバッファ障害状態
N	インメモリ化していない

## 6.3 インメモリデータバッファの統計情報を確認するには

### (1) インメモリデータバッファの名称を調べる

インメモリデータバッファの統計情報を確認するには、まず、HiRDB が決定したインメモリデータバッファの名称を調べます。

インメモリデータバッファの名称は、pdbufls コマンドの実行結果で確認できます。実行結果にインメモリデータバッファの名称を表示させるには、-M オプションを指定します。コマンドの実行例と実行結果を次に示します。

[ コマンド実行例 ]

```
pdbufls -k def -M
```

[ 実行結果 ]

```
DEFINE OF GLOBAL BUFFER
EDIT TIME 2007-10-18 16:36:23
BUFFNAME      SVID      TYPE  SIZE  NUM  WRATIO  RDAREA/INDEX NAME
              PRMAX PRNUM CSIZE  MAPS
gbufsys       DB01      R     4k    20   20      "RDMAST"
              "RDDIR"
              "RDDICT"
              0 32 ***** **
gbufdata01    DB01      R     4k    256  20      "RDDATA01"
              0 32 ***** **
gbufidx01     DB01      R     4k   4925  20      "RDIDX01"
              0 32 ***** **
gbuflob01     DB01      R     4k   4925  20      "RDLOB01"
              0 32 ***** **
_pdidbuf0000001 DB01      M     4k    256  20      "RDDATA01"
              0 32 ***** **
_pdidbuf0000002 DB01      M     4k   4925  20      "RDIDX01"
              0 32 ***** **
```

「M」は、インメモリデータバッファであることを示します。

↑ インメモリデータバッファの名称が表示されます。

### (2) インメモリデータバッファの統計情報を確認する

(1) で調べたバッファの名称を pdbufls コマンドに指定して、インメモリデータバッファの統計情報を確認できます。

インメモリデータバッファの統計情報は、pdbufls コマンドに -M オプションを指定することで確認できます。コマンドの実行例を次に示します。

## 6. こんなときどうする

### [ コマンド実行例 ]

```
pdbufls -k sts -M -a _pdidbuf0000001
```

コマンドの実行結果については、マニュアル「HiRDB Version 8 コマンドリファレンス」を参照してください。

## 6.4 強制的にインメモリ化を解除するには

### (1) 前提条件

障害が発生した場合や、テスト環境の場合は、インメモリ化を強制的に解除することができます。インメモリ化を強制的に解除する場合は、最初にインメモリデータバッファの状態を確認してください。インメモリデータバッファの状態に合わせて、RD エリアの状態を次の表に示すとおりコマンドで変更してください。その後、(2) に示す方法でインメモリ化を解除します。

表 6-1 インメモリ化を強制的に解除するときの RD エリアの状態

インメモリデータバッファの状態	コマンド実行後の RD エリアの状態
DB 非同期状態	コマンド閉塞かつオープン状態
バッファ障害状態	障害閉塞かつクローズ状態
RD エリア障害状態	コマンド閉塞かつオープン状態
バッファ障害状態かつ RD エリア障害状態	障害閉塞かつクローズ状態

### (2) 実行方法

強制的にインメモリ化を解除するには、`pdmemdb -k rels -d` コマンドを使用します。コマンドの実行例を次に示します。

[ コマンド実行例 ]

```
pdmemdb -k rels -r RDDATA01 -d
```

これによって、インメモリデータバッファがなくなります。また、RD エリアは障害閉塞状態になります。なお、この場合の障害閉塞は、`pdrels` コマンドで解除できます。

#### ! 注意事項

`pdhold -s` コマンドを実行した場合にも、RD エリアのインメモリ化が強制的に解除され、インメモリデータバッファがなくなります。このとき、RD エリアは同期化閉塞状態になります。

## 6.5 インメモリデータバッファ上のデータを破棄するには

---

### (1) 前提条件

インメモリデータバッファ上のデータを破棄するには、インメモリデータバッファが DB 非同期状態で、RD エリアがコマンド閉塞かつオープン状態である必要があります。

### (2) 実行方法

インメモリデータバッファのデータを破棄するには、`pdmembdb -k cancel` コマンドを使用します。コマンドの実行例を次に示します。

[ コマンド実行例 ]

```
pdmembdb -k cancel -r RDDATA01
```

これによって、インメモリデータバッファはバッファ障害状態に、RD エリアは障害閉塞状態になります。なお、この場合の障害閉塞は、`pdmembdb -k reload` コマンドを実行してインメモリデータバッファにデータを再読み込みすることで解除できます。



## 6.6 RD エリアのデータをインメモリデータバッファに再読み込みするには

---

### (1) 前提条件

RD エリアのデータをインメモリデータバッファに再読み込みするには、インメモリデータバッファがバッファ障害状態で、RD エリアがコマンド閉塞かつクローズ状態である必要があります。

### (2) 実行方法

RD エリアのデータをインメモリデータバッファに再読み込みするには、`pdmembdb -k reload` コマンドを使用します。コマンドの実行例を次に示します。

#### [ コマンド実行例 ]

```
pdmembdb -k reload -r RDDATA01
```

RD エリアのデータを再読み込みしたため、RD エリアとインメモリデータバッファは DB 同期状態になります。



# 付録

---

付録 A インメモリデータ処理の状態遷移図

---

付録 B インメモリデータ処理に関するコマンド一覧

---

付録 C pdmemdb コマンド実行時の前提条件

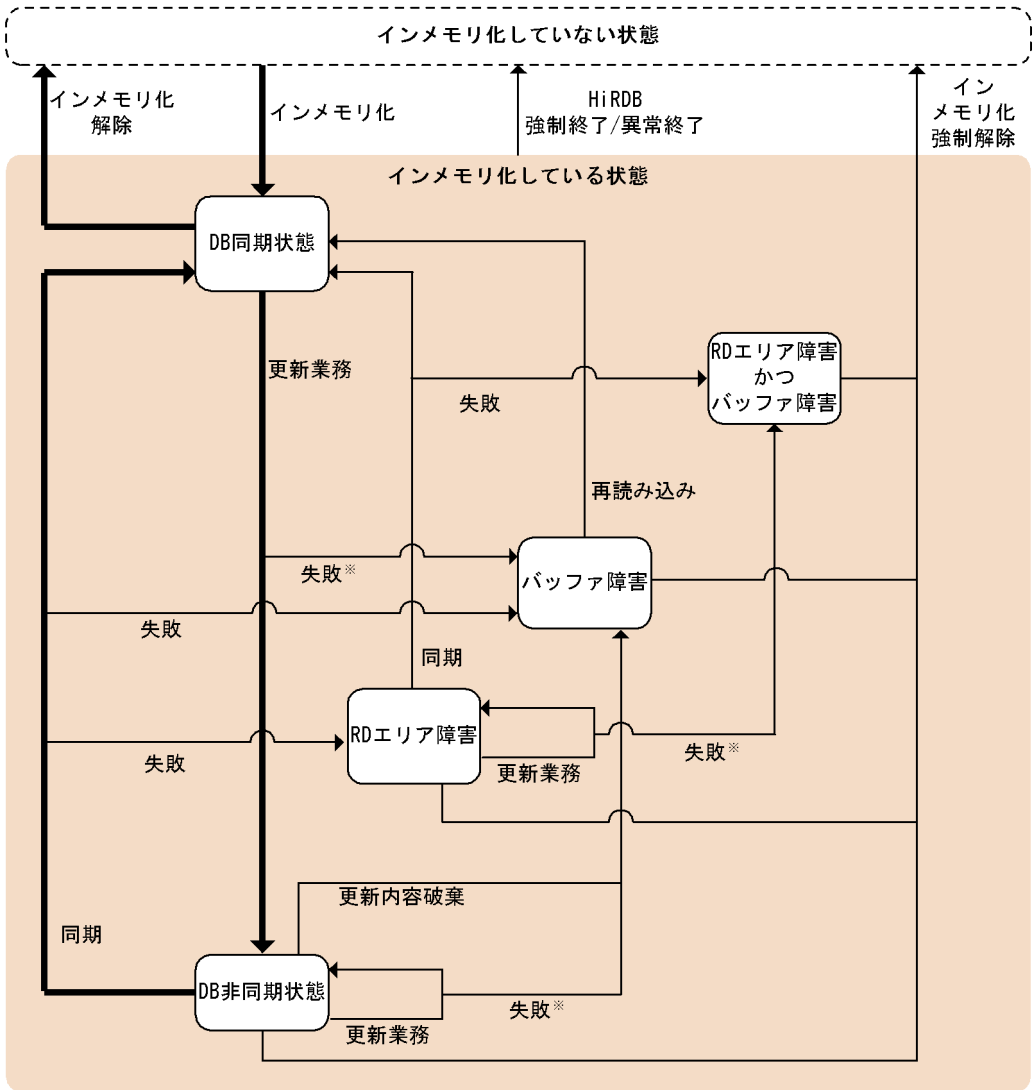
---

---




## 付録 A インメモリデータ処理の状態遷移図

インメモリデータ処理の状態遷移を次の図に示します。インメモリデータ処理の実行中は、インメモリデータバッファの状態が次に示す図のように遷移します。障害発生時には、コマンドを使って、インメモリデータバッファの状態を変更し、正常な状態に戻す必要があります。そのため、インメモリデータバッファの状態遷移を正しく理解しておく必要があります（この図は状態遷移を正しく理解するためにお使いください）。

図 A-1 インメモリデータ処理の状態遷移



(凡例)

-  : 通常運用時の状態遷移
-  : 障害発生時など、通常運用以外の状態遷移
-  : pddb1s -Mコマンドを実行することで確認できる  
インメモリデータバッファの状態 (MEMORY-STATUSの値で確認します)

注※ ログレスモードのときだけ、失敗することがあります。  
ログレスモードでない場合は、ロールバックします。

## 付録 B インメモリデータ処理に関するコマンド一覧

インメモリデータ処理に関するコマンド一覧を次の表に示します。

表 B-1 インメモリデータ処理に関するコマンド一覧

コマンド	インメモリデータ処理での操作
pdmembdb -k stay	RD エリアをインメモリ化します。
pdmembdb -k rels	インメモリ化を解除します。
pdmembdb -k rels -d	インメモリ化を強制的に解除します。
pdmembdb -k cancel	インメモリデータバッファの更新内容を破棄します。
pdmembdb -k reload	RD エリアのデータをインメモリデータバッファに再読み込みします。
pdhold -b pdhold -b -w pdhold -c pdclose	RD エリアとインメモリデータバッファの同期を取ります。
pddbpls -M	RD エリアとインメモリデータバッファの状態を表示します。
pddbpls -D	インメモリ化した時刻、および RD エリアとインメモリデータバッファの同期を取った時刻を表示します。
pdbufls -k def -M	インメモリデータバッファの情報を表示します。
pdbufls -k sts -M	インメモリデータバッファの統計情報を表示します。

## 付録 C pdmemdb コマンド実行時の前提条件

pdmemdb コマンドを実行する前に、インメモリデータバッファと対象 RD エリアの状態を確認してください。正しい状態になっていないと pdmemdb コマンドを実行できません。pdmemdb コマンド実行時の前提条件（インメモリデータバッファと対象 RD エリアの状態）を次の表に示します。

表 C-1 pdmemdb コマンド実行時の前提条件（インメモリデータバッファと対象 RD エリアの状態）

pdmemdb コマンド の -k オプションの 指定		コマンド実行時の前提条件		コマンド実行後の状態	
		インメモリデータ バッファの状態	RD エリアの状 態	インメモリデータ バッファの状態	RD エリアの状 態
stay		未使用状態	コマンド閉塞か つクローズ状態	DB 同期状態	コマンド閉塞か つクローズ状態
reload		バッファ障害状態	障害閉塞か つクローズ状態		
rels	-d オプシ ョン指 定なし	DB 同期状態	コマンド閉塞か つクローズ状態	未使用状態（イン メモリ化の解除）	コマンド閉塞か つクローズ状態
	-d オプシ ョン指 定あり	DB 非同期状態	コマンド閉塞か つオープン状態		障害閉塞か つクローズ状態
		バッファ障害状態	障害閉塞か つクローズ状態		
		RD エリア障害状 態	コマンド閉塞か つオープン状態		
	バッファ障害状態 かつ RD エリア障 害状態	障害閉塞か つクローズ状態			
cancel		DB 非同期状態	コマンド閉塞か つオープン状態	バッファ障害状態	障害閉塞か つオープン状態

### 注

オンライン再編成閉塞中からコマンド閉塞かつクローズ状態に遷移した場合は、pdmemdb -k stay コマンドを実行できません。





---

# 索引

---

## D

DB 同期状態 5  
DB 非同期状態 5

---

## O

OS のオペレーティングシステムパラメタの  
確認 11

---

## P

pd\_max\_resident\_rdarea\_no 12  
pd\_max\_resident\_rdarea\_shm\_no 12  
pdbufls -k def -M 90  
pdbufls -k sts -M 90  
pdclose 90  
pddbls -D 90  
pddbls -M 90  
pdhold -b 78,90  
pdhold -b -w 90  
pdhold -c 90  
pdmembdb -k cancel 90  
pdmembdb -k reload 90  
pdmembdb -k rels 90  
pdmembdb -k rels -d 90  
pdmembdb -k stay 90  
pdmembdb コマンド実行時の前提条件 91

---

## R

RD エリア障害 5,54  
RD エリア障害かつバッファ障害 54  
RD エリア名一括指定 15

---

## あ

アンインストール 12

---

## い

異常終了したときのデータの回復方法 5  
インストール 11

インナレプリカ機能を使用しているときに気  
をつけること 48  
インメモリ RD エリア 3,5  
インメモリ化 3,5  
インメモリ化できる RD エリア 8  
インメモリ化に失敗したとき 37  
インメモリデータ処理 2,5  
インメモリデータ処理の概要 3  
インメモリデータ処理の仕組み 3  
インメモリデータ処理の状態遷移 88  
インメモリデータバッファ 3,5  
インメモリデータバッファの統計情報を確認  
する 81  
インメモリデータバッファの名称を調べる  
81  
インメモリデータバッファの優位点 4  
インメモリデータバッファの利用状況を確認  
する 79

---

## お

オンライン業務に適用する場合 27

---

## か

回復方法を決めるときの基準 55  
必ず行う準備 10

---

## き

基本的な運用の流れ 19  
強制終了 46  
強制的にインメモリ化を解除するには 83

---

## く

グローバルバッファの割り当て 13

---

## け

計画停止 46

---

## こ

---

コマンド一覧 90

---

## さ

---

再読み込みするには 85

参照可能モード (-Mr) でバックアップを取得する場合 43

---

## し

---

システム定義の設定 12

障害回復の基本的な考え方 54

障害が発生したときに最初に確認すること  
56

障害の種類 54

初期データロード 21

---

## せ

---

正常終了 46

セグメントの割り当て 37

セットアップ 12

前提プラットフォーム 10

---

## て

---

ディスク入出力回数 4

データを破棄するには 84

テスト業務に適用する場合 31

---

## と

---

同期点指定のデータロード 21

同期を取るタイミング 78

同期を取る方法 78

---

## は

---

バックアップ取得の仕組み 43

バックアップ取得モード 42

バックアップを取得するタイミング 42

バックアップを取得するときに気をつけること 42

バッチ業務に適用する場合 21

バッファ障害 5, 54

---

## ふ

---

複数の RD エリアを一度にインメモリ化する場合 37

---

## ま

---

マシン環境の確認 10

---

## め

---

メモリ容量の確認 10

# ソフトウェアマニュアルのサービス ご案内

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

## 2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

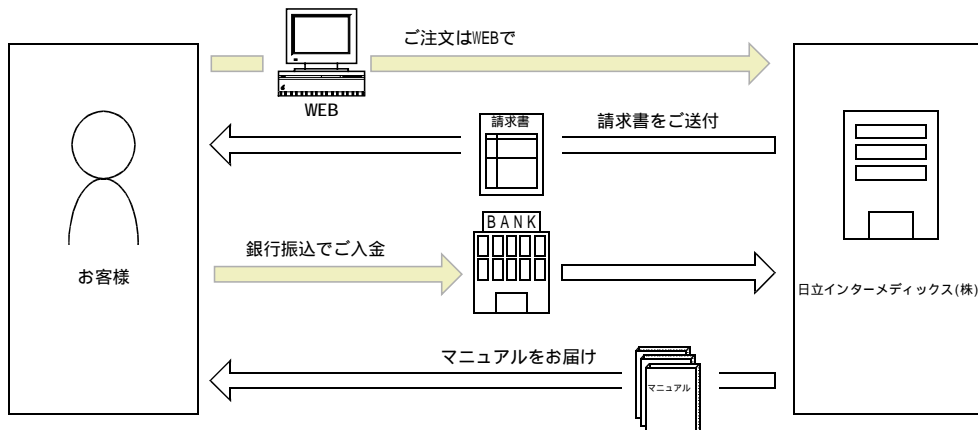
### (1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

### (2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

## 3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえWEBからご注文ください。ご注文先は日立インターメディアックス(株)となります。

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

入金確認後7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。