For UNIX Systems
Scalable Database Server

HiRDB Version 8

System Operation Guide Part II

3000-6-354(F)

HITACHI

■ **Relevant program products**

List of program products:

For the HP-UX 11.0, HP-UX 11i, or HP-UX 11i V2 (PA-RISC) operating system:

P-1B62-1182 HiRDB/Single Server Version 8 08-00
P-1B62-1382 HiRDB/Parallel Server Version 8 08-00
P-1B62-1582 HiRDB/Single Server Version 8(64) 08-00
P-1B62-1782 HiRDB/Parallel Server Version 8(64) 08-00
P-1B62-1B82 HiRDB/Run Time Version 8 08-00
P-1B62-1C82 HiRDB/Developer's Kit Version 8 08-00
P-1B62-1D82 HiRDB/Run Time Version 8(64) 08-00
P-1B62-1E82 HiRDB/Developer's Kit Version 8(64) 08-00
P-F1B62-11823 HiRDB Staticizer Option Version 8 08-00
P-F1B62-11825 HiRDB Non Recover Front End Server Version 8 08-00
P-F1B62-11826 HiRDB Advanced High Availability Version 8 08-00
P-F1B62-11827 HiRDB Advanced Partitioning Option Version 8 08-00

For the HP-UX 11i V2 (IPF) operating system:

P-1J62-1582 HiRDB/Single Server Version 8(64) 08-00
P-1J62-1782 HiRDB/Parallel Server Version 8(64) 08-00
P-1J62-1D82 HiRDB/Run Time Version 8(64) 08-00
P-1J62-1E82 HiRDB/Developer's Kit Version 8(64) 08-00
P-F1J62-11823 HiRDB Staticizer Option Version 8 08-00
P-F1J62-11825 HiRDB Non Recover Front End Server Version 8 08-00
P-F1J62-11826 HiRDB Advanced High Availability Version 8 08-00
P-F1J62-11827 HiRDB Advanced Partitioning Option Version 8 08-00

For the Solaris 8, 9 or 10 operating system:

P-9D62-1182 HiRDB/Single Server Version 8 08-00
P-9D62-1382 HiRDB/Parallel Server Version 8 08-00
P-9D62-1582 HiRDB/Single Server Version 8(64) 08-00
P-9D62-1782 HiRDB/Parallel Server Version 8(64) 08-00
P-9D62-1B82 HiRDB/Run Time Version 8 08-00
P-9D62-1C82 HiRDB/Developer's Kit Version 8 08-00
P-9D62-1D82 HiRDB/Run Time Version 8(64) 08-00
P-9D62-1E82 HiRDB/Developer's Kit Version 8(64) 08-00
P-F9D62-11823 HiRDB Staticizer Option Version 8 08-00
P-F9D62-11825 HiRDB Non Recover Front End Server Version 8 08-00
P-F9D62-11826 HiRDB Advanced High Availability Version 8 08-00
P-F9D62-11827 HiRDB Advanced Partitioning Option Version 8 08-00

For the AIX(R) 5L V5.1, V5.2 or V5.3 operating system:

P-1M62-1182 HiRDB/Single Server Version 8 08-00
P-1M62-1382 HiRDB/Parallel Server Version 8 08-00
P-1M62-1582 HiRDB/Single Server Version 8(64) 08-00
P-1M62-1782 HiRDB/Parallel Server Version 8(64) 08-00
P-1M62-1B82 HiRDB/Run Time Version 8 08-00
P-1M62-1C82 HiRDB/Developer's Kit Version 8 08-00

P-1M62-1D82 HiRDB/Run Time Version 8(64) 08-00

P-1M62-1E82 HiRDB/Developer's Kit Version 8(64) 08-00

P-F1M62-11823 HiRDB Staticizer Option Version 8 08-00

P-F1M62-11825 HiRDB Non Recover Front End Server Version 8 08-00

P-F1M62-11826 HiRDB Advanced High Availability Version 8 08-00

P-F1M62-11827 HiRDB Advanced Partitioning Option Version 8 08-00

For the Red Hat Linux 7.1, Red Hat Linux 7.2, Red Hat Enterprise Linux AS 2.1, Red Hat Enterprise Linux AS 3 (x86), Red Hat Enterprise Linux ES 3 (x86), Red Hat Enterprise Linux AS 4 (x86), Red Hat Enterprise Linux ES 4 (x86), Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T),* Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T), or Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) operating system:

P-9S62-1182 HiRDB/Single Server Version 8 08-00

P-9S62-1382 HiRDB/Parallel Server Version 8 08-00

P-9S62-1B82 HiRDB/Run Time Version 8 08-00

P-9S62-1C82 HiRDB/Developer's Kit Version 8 08-00

P-F9S62-11823 HiRDB Staticizer Option Version 8 08-00

P-F9S62-11825 HiRDB Non Recover Front End Server Version 8 08-00

P-F9S62-11826 HiRDB Advanced High Availability Version 8 08-00

P-F9S62-11827 HiRDB Advanced Partitioning Option Version 8 08-00

* Only operating systems that run on the Intel EM64T are supported.

For the Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T),* Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T), or Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) operating system:

P-9W62-1182 HiRDB/Single Server Version 8 08-00

P-9W62-1382 HiRDB/Parallel Server Version 8 08-00

P-9W62-1B82 HiRDB/Run Time Version 8 08-00

P-9W62-1C82 HiRDB/Developer's Kit Version 8 08-00

* Only operating systems that run on the Intel EM64T are supported.

For the Red Hat Enterprise Linux AS 3 (IPF) or Red Hat Enterprise Linux AS 4 (IPF) operating system:

P-9V62-1182 HiRDB/Single Server Version 8 08-00

P-9V62-1382 HiRDB/Parallel Server Version 8 08-00

P-9V62-1B82 HiRDB/Run Time Version 8 08-00

P-9V62-1C82 HiRDB/Developer's Kit Version 8 08-00

P-F9V62-11823 HiRDB Staticizer Option Version 8 08-00

P-F9V62-11825 HiRDB Non Recover Front End Server Version 8 08-00

P-F9V62-11826 HiRDB Advanced High Availability Version 8 08-00

P-F9V62-11827 HiRDB Advanced Partitioning Option Version 8 08-00

This edition of the manual is released for the preceding program products, which have been developed under a quality management system that has been certified to comply with ISO9001 and TickIT. This manual may also apply to other program products; for details, see *Software Information* or *Before Installing*.

■ Trademarks

ActiveX is a trademark of Microsoft Corp. in the U.S. and other countries.

AIX is a registered trademark of the International Business Machines Corp. in the U.S.

CORBA is a registered trademark of Object Management Group, Inc. in the United States.

DataStage, MetaBroker, MetaStage and QualityStage are trademarks of International Business Machines Corporation in the United States, other countries, or both.

DB2 is a registered trademark of the International Business Machines Corp. in the U.S.

HACMP/6000 is a trademark of the International Business Machines Corp. in the U.S.

HP-UX is a product name of Hewlett-Packard Company.

IBM is a registered trademark of the International Business Machines Corp. in the U.S.

Itanium is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

JBuilder is a trademark of Borland Software Corporation in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Lotus, 1-2-3 are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft Access is a registered trademark of Microsoft Corporation in the U.S. and other countries.

Microsoft Excel is a product name of Microsoft Corp.

Microsoft is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Motif is a registered trademark of the Open Software Foundation, Inc.

MS-DOS is a registered trademark of Microsoft Corp. in the U.S. and other countries.

ODBC is Microsoft's strategic interface for accessing databases.

OLE is the name of a software product developed by Microsoft Corporation and the acronym for Object Linking and Embedding.

ORACLE is a registered trademark of Oracle Corporation.

Oracle8i is a trademark of ORACLE Corporation.

Oracle9i is a trademark of ORACLE Corporation.

Oracle 10g is a trademark of ORACLE Corporation.

OS/390 is a trademark of the International Business Machines Corp. in the U.S.

POSIX stands for Portable Operating System Interface for Computer Environment, which is a set of standard specifications published by the Institute of Electrical and Electronics Engineers, Inc.

RISC System/6000 is a registered trademark of the International Business Machines Corp. in the U.S.

Solaris is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Sun is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Sun Microsystems is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.

The right to use the trademark DCE in Japan is sub-licensed from OSF.

UNIFY2000 is a product name of Unify Corp.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VERITAS is a trademark or registered trademark of Symantec Corporation in the U.S. and other countries.

Visual Basic is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Visual C++ is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Visual Studio is a registered trademark of Microsoft Corp. in the U.S. and other countries.

WebLogic is a registered trademark of BEA Systems, Inc.

Windows is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Windows NT is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Windows Server is a registered trademark of Microsoft Corp. in the U.S. and other countries.

X/Open is a registered trademark of X/Open Company Limited in the U.K. and other countries.

X Window System is a trademark of X Consortium, Inc.

The following program products include material copyrighted by Sun Microsystems, Inc.: P-9D62-1182, P-9D62-1382, P-9D62-1582, P-9D62-1782, P-9D62-1B82, P-9D62-1C82, P-9D62-1D82, P-9D62-1E82, P-F9D62-11823, P-F9D62-11825, P-F9D62-11826, and P-F9D62-11827.

The following program products include material copyrighted by UNIX System Laboratories, Inc.: P-9D62-1182, P-9D62-1382, P-9D62-1582, P-9D62-1782, P-9D62-1B82, P-9D62-1C82, P-9D62-1D82, P-9D62-1E82, P-F9D62-11823, P-F9D62-11825,

P-F9D62-11826, and P-F9D62-11827.

Other product and company names mentioned in this document may be the trademarks of their respective owners. Throughout this document Hitachi has attempted to distinguish trademarks from descriptive terms by writing the name with the capitalization used by the manufacturer, or by writing the name with initial capital letters. Hitachi cannot attest to the accuracy of this information. Use of a trademark in this document should not be regarded as affecting the validity of the trademark.

■ Restrictions

Information in this document is subject to change without notice and does not represent a commitment on the part of Hitachi. The software described in this manual is furnished according to a license agreement with Hitachi. The license agreement contains all of the terms and conditions governing your use of the software and documentation, including all warranty rights, limitations of liability, and disclaimers of warranty.

Material contained in this document may describe Hitachi products not available or features not available in your country.

No part of this material may be reproduced in any form or by any means without permission in writing from the publisher.

Printed in Japan.

■ Edition history

Edition 1 (3000-6-354(E)): March 2007

■ Copyright

All Rights Reserved. Copyright (C) 2007, Hitachi, Ltd.

Preface

This manual describes the operating procedures for the HiRDB Version 8 scalable database server system.

Intended readers

This manual is intended for users who will be constructing or operating *HiRDB Version 8* ("HiRDB") relational database systems.

It is assumed that readers of this manual have the following:

- A basic knowledge of managing UNIX or Linux systems
- A basic knowledge of SQL

This manual is based on the *HiRDB Version 8 Description*, which should be read before reading this manual.

Organization of this manual

This manual is organized as follows:

Chapter *1. HiRDB Startup and Termination*

Explains the procedures for starting and terminating HiRDB, as well as the procedures for starting and terminating units and servers.

Chapter *2. Security Definition*

Explains the HiRDB-provided security features and their operating procedures.

Chapter *3. Handling System Log Files*

Explains the handling of system log files.

Chapter *4. Handling Synchronization Point Dump Files*

Explains the handling of synchronization point dump files.

Chapter *5. Handling Status Files*

Explains the handling of status files.

Chapter *6. Backup Procedures*

Explains the procedures for making backups.

Chapter *7. Operation Without Acquiring a Database Update Log*

Explains the procedures for executing a UAP or utility in the following modes:

- Pre-update log acquisition mode
- No-log mode

Chapter *8. Obtaining the System Operating Environment (Monitoring the System Status)*

Explains how to obtain information about the system operating environment by monitoring the system status.

Chapter *9. Modifying the System Operating Environment*

Explains how to modify the HiRDB system environment definitions.

Chapter *10. Handling HiRDB File System Areas*

Explains how to create and delete HiRDB file system areas, obtain backups, and how to restore HiRDB file system areas.

Chapter *11. Modifying the System Configuration*

Explains how to modify the unit or server configuration of a HiRDB/Parallel Server; also explains the procedures for migrating from a HiRDB/Single Server to a HiRDB/Parallel Server.

Chapter *12. Migrating Resources Between Systems*

Explains how to migrate tables and stored procedures to another HiRDB system.

Chapter *13. Handling Tables*

Explains the procedures for reorganizing tables, modifying table definitions, deleting tables, deleting abstract data types, and managing lists, as well as the space conversion facility and the facility for conversion to a decimal signed normalized number.

Chapter *14. Handling Indexes*

Explains the procedures for reorganizing and deleting indexes, as well as unbalanced index splitting and delayed batch creation of plug-in indexes.

Chapter *15. Handling RDAREAs*

Explains the procedures for adding, deleting, expanding, reinitializing, and automatically increasing RDAREAs, and how to modify the RDAREA opening trigger.

Chapter *16. Handling Stored Procedures and Stored Functions*

Explains the procedures for registering stored procedures and stored functions, and the handling of stored procedures and stored functions when they are no longer valid.

Chapter *17. Using Java Stored Procedures and Java Stored Functions*

Explains the environment setup and operating procedures for using Java stored procedures and Java stored functions.

Chapter *18. Error Handling Procedures*

Explains the handling of errors.

Chapter *19. Database Recovery Procedures*

Explains the procedures for recovering a database in the event that it is damaged by an error.

Chapter *20. Obtaining Tuning Information*

Explains the procedures for obtaining tuning information.

Chapter *21. Tuning*

Explains the tuning procedures.

Chapter *22. Using the Security Audit Facility*

Explains the environment setup and operating procedures for the security audit facility.

Chapter *23. Using the Connection Security Facility*

Explains how to use the connection security facility, which is designed to enhance the security of HiRDB systems.

Chapter *24. Using the Directory Server Linkage Facility*

Explains the environment setup and operating procedures for the Directory Server linkage facility.

Chapter *25. Using the System Switchover Facility*

Explains the environment setup and operating procedures for the system switchover facility.

Chapter *26. Using the Facility for Monitoring MIB Performance Information*

Explains how to use the facility for monitoring MIB performance information, which uses MIB to collect HiRDB operation information.

Chapter *27. Using a Distributed Database (applicable to HP-UX and AIX 5L only)*

Explains the environment setup and operating procedures for distributed databases.

Appendix *A. Q & A*

Provides in Q&A format the answers to frequently asked questions.

Appendix *B. Operations When Using a DVD-RAM Library Device*

Explains the procedures for using a DVD-RAM library unit as a storage device.

Appendix *C. Information Needed for Troubleshooting*

Explains the information needed to use problem-solving support or Q&A support services.

Appendix *D. Notes on Running HiRDB Around the Clock*

Explains the procedures for and provides notes about running HiRDB continuously around the clock.

Appendix *E. Using Performance Improvement Facilities*

Explains facilities designed to enhance system performance and how to use them.

Related publications

This manual is related to the following manuals, which should be read as required.

HiRDB (for UNIX)

- *For UNIX Systems HiRDB Version 8 Description* (3000-6-351(E))
- *For UNIX Systems HiRDB Version 8 Installation and Design Guide* (3000-6-352(E))
- *For UNIX Systems HiRDB Version 8 System Definition* (3000-6-353(E))
- *For UNIX Systems HiRDB Version 8 Command Reference* (3000-6-355(E))
- *HiRDB Staticizer Option Version 7 Description and User's Guide* (3000-6-282(E))
- *For UNIX Systems HiRDB Version 8 Disaster Recovery System Configuration and Operation Guide* (3000-6-364)*

HiRDB (for Windows)

- *For Windows Systems HiRDB Version 8 Description* (3020-6-351(E))
- *For Windows Systems HiRDB Version 8 Installation and Design Guide* (3020-6-352(E))
- *For Windows Systems HiRDB Version 8 System Definition* (3020-6-353(E))
- *For Windows Systems HiRDB Version 8 System Operation Guide* (3020-6-354(E))
- *For Windows Systems HiRDB Version 8 Command Reference* (3020-6-355(E))

HiRDB (for both Windows and UNIX)

- *HiRDB Version 8 UAP Development Guide* (3020-6-356(E))

- *HiRDB Version 8 SQL Reference* (3020-6-357(E))
- *HiRDB Version 8 Messages* (3020-6-358(E))
- *HiRDB Datareplicator Version 8 Description, User's Guide and Operator's Guide* (3020-6-360(E))
- *HiRDB Dataextractor Version 8 Description, User's Guide and Operator's Guide* (3020-6-362(E))

* This manual has been published in Japanese only; it is not available in English.

You must use the UNIX or the Windows manuals, as appropriate to the platform you are using.

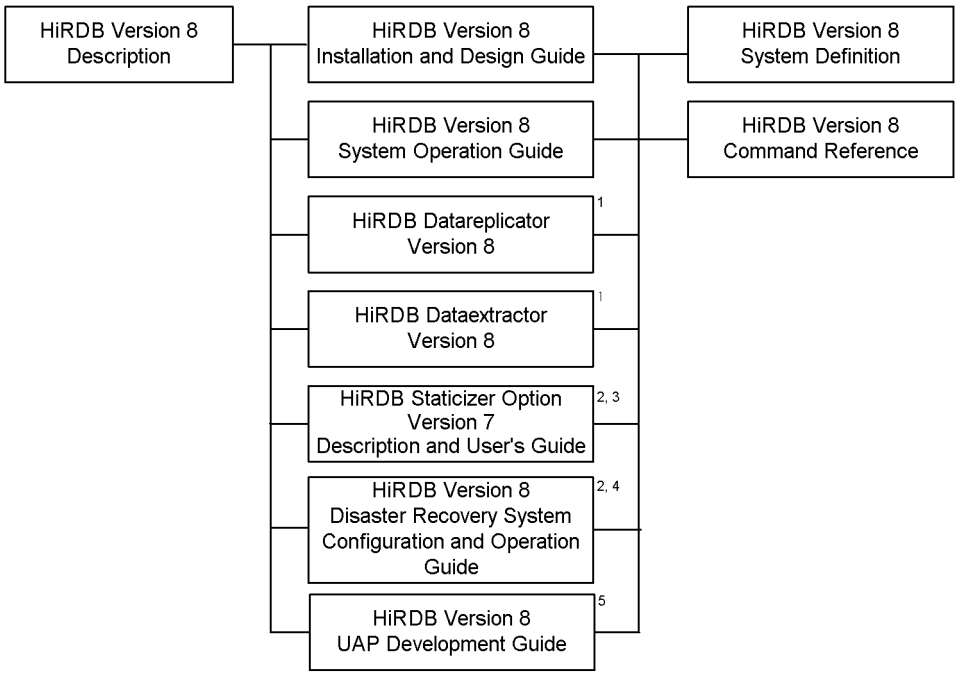
Others

- *HiRDB External Data Access Version 7 Description and User's Guide* (3000-6-284(E))
- *Distributed Database System DF/UX* (3000-3-248(E))
- *JP1 Version 6 JP1/VERITAS NetBackup v4.5 Agent for HiRDB License Description and User's Guide* (3020-3-D79(E))
- *HiCommand Tuning Manager - Agent for RAID* (3020-3-E92(E))
- *HiCommand Tuning Manager - Agent for RAID Map* (3020-3-E93(E))
- *For UNIX Systems Job Management Partner 1/Performance Management - Agent Option for Platform Description, User's Guide and Reference* (3020-3-K65(E))
- *Job Management Partner 1/Performance Management/SNMP System Observer for Extended Resource Management* (3020-3-F70(E))
- *Job Management Partner 1/Consolidated Management 2/Extensible SNMP Agent* (3020-3-F92(E))

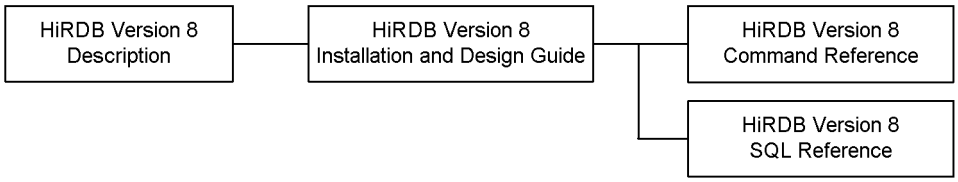
Organization of HiRDB manuals

The HiRDB manuals are organized as shown below. For the most efficient use of these manuals, it is suggested that they be read in the order they are shown, going from left to right.

Manuals for system administrators:



Manuals for users who create tables:



Manuals for users who create or execute UAPs:

[1] Read if you use the replication facility to link data.
[2] Published for UNIX only. There is no corresponding Windows manual.
[3] Read if you use the inner replica facility.
[4] Read if you are configuring a disaster recovery system.
[5] Must be read if you are linking HiRDB to an OLTP system.

Conventions: Abbreviations

Unless otherwise required, this manual uses the following abbreviations for product and other names.

Name of product or other entity	Representation	
HiRDB/Single Server Version 8	HiRDB/Single Server	HiRDB or HiRDB Server
HiRDB/Single Server Version 8(64)		
HiRDB/Parallel Server Version 8	HiRDB/Parallel Server	
HiRDB/Parallel Server Version 8(64)		
HiRDB/Developer's Kit Version 8	HiRDB/ Developer's Kit	HiRDB Client
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 7	HiRDB Text Search Plug-in	
HiRDB Spatial Search Plug-in Version 3	HiRDB Spatial Search Plug-in	
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 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	

Name of product or other entity	Representation
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 or JP1/IM
JP1/Integrated Management - View	
JP1/Magnetic Tape Access	EasyMT
EasyMT	
JP1/Magnetic Tape Library	MTguide
JP1/NETM/DM	JP1/NETM/DM
JP1/NETM/DM Manager	
JP1/Performance Management	JP1/PFM
JP1/Performance Management Agent for HiRDB	JP1/PFM-Agent for HiRDB
JP1/Performance Management - Agent 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

Name of product or other entity	Representation	
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 basic program XDM/BASE E2	XDM/BASE E2	
XDM/Data Communication and Control Manager 3 XDM Data communication control XDM/DCCM3	XDM/DCCM3	
XDM/Relational Database XDM/RD	XDM/RD	XDM/RD
XDM/Relational Database Extended Version 2 XDM/RD E2	XDM/RD E2	
VOS3 Database Connection Server	DB Connection Server	
DB2 Universal Database for OS/390 Version 6	DB2	
DNCWARE ClusterPerfect (Linux Version)	ClusterPerfect	
Microsoft(R) Excel	Microsoft Excel or Excel	
Microsoft(R) Visual C++(R)	Visual C++ or C++	
Oracle 8i	ORACLE	
Oracle 9i		
Oracle 10g		
Sun Java™ System Directory Server	Sun Java System Directory Server or Directory Server	
HP-UX 11i V2 (IPF)	HP-UX or HP-UX (IPF)	
Red Hat Linux	Linux	
Red Hat Enterprise Linux		
Red Hat Enterprise Linux AS 3 (IPF)	Linux (IPF)	Linux
Red Hat Enterprise Linux AS 4 (IPF)		
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)		
turbolinux 7 Server for AP8000	Linux for AP8000	

Name of product or other entity	Representation	
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	Windows 2000 or Windows 2000 Advanced Server	
Microsoft(R) Windows Server™ 2003, Standard Edition	Windows Server 2003	
Microsoft(R) Windows Server™ 2003, Enterprise Edition		
Microsoft(R) Windows Server™ 2003 R2, Standard Edition	Windows Server 2003 R2 or Windows Server 2003	
Microsoft(R) Windows Server™ 2003 R2, Enterprise Edition		
64 bit Version Microsoft(R) Windows Server™ 2003, Enterprise Edition (IPF)	Windows Server 2003 (IPF) or Windows Server 2003	
Microsoft(R) Windows Server™ 2003, Standard x64 Edition	Windows Server 2003 or Windows Server 2003 x64 Editions	Windows (x64)
Microsoft(R) Windows Server™ 2003, Enterprise x64 Edition		
Microsoft(R) Windows Server™ 2003 R2, Standard x64 Edition	Windows Server 2003, Windows Server 2003 R2 or Windows Server 2003 x64 Editions	
Microsoft(R) Windows Server™ 2003 R2, Enterprise x64 Edition		
Microsoft(R) Windows(R) XP Professional x64 Edition	Windows XP or Windows XP x64 Edition	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP Professional	Windows XP

Name of product or other entity	Representation
Microsoft(R) Windows(R) XP Home Edition Operating System	Windows XP Home Edition
Single server	SDS
System manager	MGR
Front-end server	FES
Dictionary server	DS
Back-end server	BES

- Windows 2000, Windows XP, and Windows Server 2003 may be referred to collectively as *Windows*.
- The HiRDB directory path is represented as `$PDDIR`.
- The hosts file means the `hosts` file stipulated by TCP/IP (including the `/etc/hosts` file).

This manual also uses the following abbreviations:

Abbreviation	Full name or meaning
ACK	Acknowledgement
ADM	Adaptable Data Manager
ADO	ActiveX Data Objects
ADT	Abstract Data Type
AP	Application Program
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
BES	Back End Server
BLOB	Binary Large Object
BOM	Byte Order Mark
CD-ROM	Compact Disc - Read Only Memory
CGI	Common Gateway Interface
CLOB	Character Large Object
CMT	Cassette Magnetic Tape

Abbreviation	Full name or meaning
COBOL	Common Business Oriented Language
CORBA(R)	Common ORB Architecture
CPU	Central Processing Unit
CSV	Comma Separated Values
DAO	Data Access Object
DAT	Digital Audio Taperecorder
DB	Database
DBM	Database Module
DBMS	Database Management System
DDL	Data Definition Language
DF for Windows NT	Distributing Facility for Windows NT
DF/UX	Distributing Facility/for UNIX
DIC	Dictionary Server
DLT	Digital Linear Tape
DML	Data Manipulate Language
DNS	Domain Name System
DOM	Document Object Model
DS	Dictionary Server
DTD	Document Type Definition
DTP	Distributed Transaction Processing
DWH	Data Warehouse
EUC	Extended UNIX Code
EX	Exclusive
FAT	File Allocation Table
FD	Floppy Disk
FES	Front End Server
FQDN	Fully Qualified Domain Name

Abbreviation	Full name or meaning
FTP	File Transfer Protocol
GUI	Graphical User Interface
HBA	Host Bus Adapter
HD	Hard Disk
HTML	Hyper Text Markup Language
ID	Identification number
IP	Internet Protocol
IPF	Itanium(R) Processor Family
JAR	Java Archive File
Java VM	Java Virtual Machine
JDBC	Java Database Connectivity
JDK	Java Developer's Kit
JFS	Journaled File System
JFS2	Enhanced Journaled File System
JIS	Japanese Industrial Standard code
JP1	Job Management Partner 1
JRE	Java Runtime Environment
JTA	Java Transaction API
JTS	Java Transaction Service
KEIS	Kanji processing Extended Information System
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LIP	loop initialization process
LOB	Large Object
LRU	Least Recently Used
LTO	Linear Tape-Open
LU	Logical Unit

Abbreviation	Full name or meaning
LUN	Logical Unit Number
LVM	Logical Volume Manager
MGR	System Manager
MIB	Management Information Base
MRCF	Multiple RAID Coupling Feature
MSCS	Microsoft Cluster Server
NAFO	Network Adapter Fail Over
NAPT	Network Address Port Translation
NAT	Network Address Translation
NIC	Network Interface Card
NIS	Network Information Service
NTFS	New Technology File System
ODBC	Open Database Connectivity
OLAP	Online Analytical Processing
OLE	Object Linking and Embedding
OLTP	On-Line Transaction Processing
OOCOBOL	Object Oriented COBOL
ORB	Object Request Broker
OS	Operating System
OSI	Open Systems Interconnection
OTS	Object Transaction Service
PC	Personal Computer
PDM II E2	Practical Data Manager II Extended Version 2
PIC	Plug-in Code
PNM	Public Network Management
POSIX	Portable Operating System Interface for UNIX
PP	Program Product

Abbreviation	Full name or meaning
PR	Protected Retrieve
PU	Protected Update
RAID	Redundant Arrays of Inexpensive Disk
RD	Relational Database
RDB	Relational Database
RDB1	Relational Database Manager 1
RDB1 E2	Relational Database Manager 1 Extended Version 2
RDO	Remote Data Objects
RiSe	Real time SAN replication
RM	Resource Manager
RMM	Resource Manager Monitor
RPC	Remote Procedure Call
SAX	Simple API for XML
SDS	Single Database Server
SGML	Standard Generalized Markup Language
SJIS	Shift JIS
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SQL/K	Structured Query Language/VOS K
SR	Shared Retrieve
SU	Shared Update
TCP/IP	Transmission Control Protocol/Internet Protocol
TM	Transaction Manager
TMS-4V/SP	Transaction Management System - 4V/System Product
UAP	User Application Program
UOC	User Own Coding
VOS K	Virtual-storage Operating System Kindness

Abbreviation	Full name or meaning
VOS1	Virtual-storage Operating System 1
VOS3	Virtual-storage Operating System 3
WS	Workstation
WWW	World Wide Web
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

Log representations

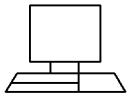
The OS log is referred to generically as *syslogfile*. syslogfile is the log output destination specified in `/etc/syslog.conf`. Typically, the following files are specified as syslogfile.

OS	File
HP-UX	`/var/adm/syslog/syslog.log`
Solaris	`/var/adm/messages` or `/var/log/syslog`
AIX 5L	`/var/adm/ras/syslog`
Linux	`/var/log/messages`

Conventions: Diagrams

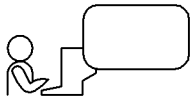
This manual uses the following conventions in diagrams:

- Workstation or personal computer



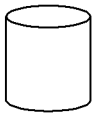
- I/O operation

- Screen display



- Program or server

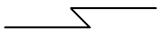
- File or magnetic disk



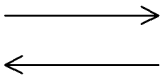
- Magnetic tape

- CMT or DAT

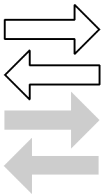
- Communication line



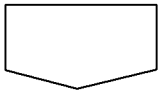
- Flow of control



- Flow of data

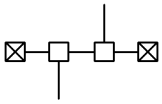


- Work procedure

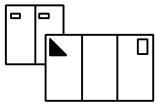


- Network

- LAN



- Mainframe



Conventions: Fonts and symbols

Font and symbol conventions are classified as:

- General font conventions
- Conventions in syntax explanations

These conventions are described below.

General font conventions

The following table lists the general font conventions:

Font	Convention
Bold	Bold type indicates text on a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example, bold is used in sentences such as the following: • From the **File** menu, choose **Open**. • Click the **Cancel** button. • In the **Enter name** entry box, type your name.
Italics	Italics are used to indicate a placeholder for some actual text provided by the user or system. Italics are also used for emphasis. For example: • Write the command as follows: `copy` *source-file target-file* • Do *not* delete the configuration file.
`Code font`	A code font indicates text that the user enters without change, or text (such as messages) output by the system. For example: • At the prompt, enter `dir`. • Use the `send` command to send mail. • The following message is displayed: `The password is incorrect.`

Examples of coding and messages appear as follows (although there may be some exceptions, such as when coding is included in a diagram):

```
MakeDatabase
...
StoreDatabase temp DB32
```

In examples of coding, an ellipsis (...) indicates that one or more lines of coding are not shown for purposes of brevity.

Conventions in syntax explanations

Syntax definitions appear as follows:

Store**D**atabase [temp|perm] (*database-name* ...)

The following table lists the conventions used in syntax explanations. The syntactical characters described below are used to provide a clear explanation of code syntax; you do not actually enter these characters.

Example font or symbol	Convention
`StoreDatabase`	Code-font characters must be entered exactly as shown.
database-name	This font style marks a placeholder that indicates where appropriate characters are to be entered in an actual command.
SD	Bold code-font characters indicate the abbreviation for a command.

Example font or symbol	Convention
perm	Underlined characters indicate the default value.
[ ]	Square brackets enclose an item or set of items whose specification is optional. Example: pdbuffer [-p] This example indicates that you can specify either pdbuffer or pdbuffer -p.
|	Only one of the options separated by a vertical bar can be specified at the same time. Example: pdlogadfg -d sys | spd This example indicates that you can specify either sys or spd for the -d option.
. . .	An ellipsis (...) indicates that the item or items enclosed in ( ) or [ ] immediately preceding the ellipsis may be specified as many times as necessary. Example: pdbuffer -r *RDAREA-name* [,*RDAREA-name*] . . . This example indicates that following -r, you can specify *RDAREA-name* as many times as necessary.
( )	Parentheses indicate the range of items to which the vertical bar (|) or ellipsis (...) is applicable.
{ }	A single pair of curly brackets encloses multiple items, one of which you must specify. Example: pdbuffer [{-r *RDAREA-name* | -i *authorization-identifier*.*index-identifier* | -o}] This example indicates that you must specify one of the three options enclosed by the curly braces: -r *RDAREA-name*, -i *authorization-identifier*.*index-identifier*, or -o.
{{ }}	A double pair of curly brackets encloses multiple items, all of which you can specify multiple times as a unit. Example: {{pdbuffer -a *option-name*}} This example indicates that you can specify the above multiple times as follows: pdbuffer -a *option-name* pdbuffer -a *option-name*
~	A swung dash precedes the attributes of a user-specified value.
<< >>	A double pair of angle brackets encloses the default value assumed by the system when the specification is omitted.
< >	A single pair of angle brackets encloses the syntax element notation for a user-specified value.
(( ))	A double pair of parentheses encloses the permitted range of values that can be specified.

Syntactical element symbols

The following syntactical element symbols are used in this manual:

Syntactical element symbol	Meaning
<*alphabetics*>	The alphabetic characters (A-Z and a-z) and the underline (_)
<*alphabetics and special characters*>	The alphabetic characters (A-Z and a-z) and the special characters #, @, and \
<*alphanumerics*>	Alphabetics and the numeric digits (0-9)
<*alphanumerics and special characters*>	Alphabetics, special characters, and numeric digits
<*unsigned integer*>	Numeric value
<*hexadecimal*>	Numeric digits and A-F (or a-f)
<*identifier*>[1]	Alphanumeric character string beginning with an alphabetic character
<*symbolic name*>	Alphanumeric character string beginning with an alphabetic character or a special character
<*character string*>	Any string of characters
<*path name*>[2]	Includes symbolic names, forward slashes (/), and periods (.).

Note

All alphabetic characters must be single-byte characters. The syntactical element symbols are case sensitive.

[1] An RDAREA name is an alphanumeric character string beginning with an alphabetic character or special character, and can include alphanumeric characters, underscores (_), and spaces. If an RDAREA name includes a space, the entire name must be enclosed in double quotation marks (").

A host name is a character string that can include alphabetic characters (A to Z, a to z), numeric characters, periods (.), hyphens (-), and underscores (_). A host name can begin with a numeric character.

[2] Path names depend on the OS being used. Do not use a backslash (\) in HiRDB file system area names.

Notations used in formulas

The following notations are used in computational expressions:

Symbol	Meaning
↑ ↑	Round up the result to the next integer. Example: The result of ↑34 ÷ 3↑ will be 12.

Symbol	Meaning
↓ ↓	Discard digits following the decimal point. Example: The result of ↓34 ÷ 3↓ will be 11.
MAX	Select the largest value as the result. Example: The result of MAX(3 × 6, 4 + 7) will be18.
MIN	Select the smallest value as the result. Example: The result of MIN(3 × 6, 4 + 7) will be 11.

Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is $1{,}024^2$ bytes.
- 1 GB (gigabyte) is $1{,}024^3$ bytes.
- 1 TB (terabyte) is $1{,}024^4$ bytes.

Conventions: Version numbers

The version numbers of Hitachi program products are usually written as two sets of two digits each, separated by a hyphen. For example:

- Version 1.00 (or 1.0) is written as 01-00.
- Version 2.05 is written as 02-05.
- Version 2.50 (or 2.5) is written as 02-50.
- Version 12.25 is written as 12-25.

The version number might be shown on the spine of a manual as *Ver. 2.00,* but the same version number would be written in the program as *02-00*.

Important notes on this manual

The following facilities are explained, but they are not supported:

- Distributed database facility
- Server mode system switchover facility
- User server hot standby
- Rapid system switchover facility
- Standby-less system switchover (1:1) facility

- Standby-less system switchover (effects distributed) facility
- HiRDB External Data Access facility
- Inner replica facility (when described for the Windows version of HiRDB)
- Updatable online reorganization (when described for the Windows version of HiRDB)
- Sun Java System Directory Server linkage facility
- Simple setup tool

The following products and option program products are explained, but they are not supported:

- HiRDB Control Manager
- HiRDB Disaster Recovery Light Edition
- HiRDB External Data Access
- HiRDB LDAP Option

Notes on printed manuals

Please note that even though the printed manuals are separated into Part I and Part II, the chapters and page numbers sequentially continue from Part I to Part II. Also, please note that the index is only included in Part II.

Contents

List of figures

List of tables

Chapter

16. Handling Stored Procedures and Stored Functions

This chapter explains the procedures for handling stored procedures and stored functions.

Stored procedure is a generic term for SQL stored procedures and Java stored procedures; *stored function* is a generic term for SQL stored functions and Java™ stored functions.

This chapter contains the following sections:

16.1 Before creating (registering) stored procedures or stored functions

Executor: HiRDB administrator

Before stored procedures or stored functions are created, the database structure modification utility (`pdmod` command) must be used to create RDAREAs in which the stored procedures or stored functions will be stored. The following RDAREAs can be created:

- Data dictionary LOB RDAREAs
- Data dictionary RDAREAs*

* Data dictionary RDAREAs are needed if data dictionary tables for the stored procedures or stored functions are to be stored in RDAREAs other than existing data dictionary RDAREAs.

16.2 Creating (registering) a stored procedure or stored function

(1) Creating a stored procedure

`CREATE PROCEDURE` is used to create a stored procedure. For details on how to use `CREATE PROCEDURE` to create a procedure, see the manual *HiRDB Version 8 UAP Development Guide*.

(2) Creating a stored function

`CREATE FUNCTION` is used to create a stored function. For details on how to use `CREATE FUNCTION` to create a function, see the manual *HiRDB Version 8 UAP Development Guide*.

(a) When a stored function is created, an existing stored function may be invalidated

When a stored function is created, an existing stored function may become invalid. An existing stored function that satisfies the following condition becomes invalid:

- A stored function that calls a stored function that has the same name (authorization identifier and routine identifier) and the same number of parameters as the stored function being created

In this case, use `ALTER ROUTINE` to re-create the invalidated stored function.

(b) When a stored function is created, an existing stored procedure may be invalidated

When a stored function is created, an existing stored procedure may become invalid. An existing stored procedure that satisfies the following condition becomes invalid:

- A stored procedure that calls a stored function that has the same name (authorization identifier and routine identifier) and the same number of parameters as the stored function being created

In this case, use the `ALTER PROCEDURE` statement or `ALTER ROUTINE` to re-create the invalidated stored procedure.

(c) When a stored function is created, an existing trigger may be invalidated

When a stored function is created, an existing trigger may become invalid. An existing trigger that satisfies the following condition becomes invalid:

- A trigger that calls a stored function that has the same name (authorization identifier and routine identifier) and the same number of parameters as the stored function being created

In this case, use the `ALTER TRIGGER` statement or `ALTER ROUTINE` to re-create the invalidated trigger.

(d) A newly created stored function may be invalidated

A stored function created using the following procedure may become invalid:

Procedure

1. Install a plug-in.
2. Create a stored function for calling the plug-in function installed in step 1.*
3. Install a different plug-in from the one installed in step 1.

* If the plug-ins installed in steps 1 and 3 have the same function name and the same number of parameters, the stored function created in step 2 is invalidated when step 3 is executed. In such a case, `ALTER ROUTINE` can used to re-create the invalidated stored function.

16.3 Re-creating an invalidated stored procedure or stored function

When the definition of a table or index is modified, all the stored procedures and stored functions that use that table or index are invalidated. In such a case, the invalidated stored procedures and stored functions can be re-created, as explained in this section.

`ALTER PROCEDURE` or `ALTER ROUTINE` is used to re-create a stored procedure. `ALTER ROUTINE` is used to re-create a stored function.

16.4 Deleting a stored procedure or stored function

(1) Deleting a stored procedure

`DROP PROCEDURE` is used to delete a stored procedure.

(2) Deleting a stored function

`DROP FUNCTION` is used to delete a stored function.

(a) When a stored function is deleted, an existing stored function may be invalidated

When a stored function is deleted, an existing stored function may become invalid. A stored function that satisfies the following condition becomes invalid:

- A stored function that calls a stored function that has the same name (authorization identifier and routine identifier) and the same number of parameters as the stored function being deleted

In this case, use `ALTER ROUTINE` to re-create the invalidated stored function.

(b) When a stored function is deleted, an existing stored procedure may be invalidated

When a stored function is deleted, an existing stored procedure may become invalid. An existing stored procedure that satisfies the following condition becomes invalid:

- A stored procedure that calls a stored function that has the same name (authorization identifier and routine identifier) and the same number of parameters as the stored function being deleted

In this case, use the `ALTER PROCEDURE` statement or `ALTER ROUTINE` to re-create the invalidated stored procedure.

(c) When a stored function is deleted, an existing trigger may be invalidated

When a stored function is deleted, an existing trigger may become invalid. An existing trigger that satisfies the following condition becomes invalid:

- A trigger that calls a stored function that has the same name (authorization identifier and routine identifier) and the same number of parameters as the stored function being deleted

In this case, use the `ALTER TRIGGER` statement or `ALTER ROUTINE` to re-create the invalidated trigger.

16.5 Creating a definition SQL from an existing stored procedure

Executor: HiRDB administrator or a user with DBA privilege

The `pddefrev` command can be used to create a definition SQL from an existing stored procedure. This command is useful for creating a stored procedure that provides a similar function as an existing stored procedure. A definition SQL created with the `pddefrev` command is used as input information to the database definition utility (`pddef` command).

Chapter

17. Using Java Stored Procedures and Java Stored Functions

This chapter explains the environment setup and operating procedures for using Java stored procedures and Java stored functions.

This chapter contains the following sections:

17.1 Overview of Java stored procedures and Java stored functions

This section provides an overview of Java stored procedures and Java stored functions.

(1) Environments in which Java stored procedures and Java stored functions can be used

Table 17-1 shows the environments in which Java stored procedures and Java stored functions can be used.

Table 17-1: Environments in which Java stored procedures and Java stored functions can be used

OS	Usable	Remarks
HP-UX	Y	Can be used in either of the following environments: • 32-bit-mode POSIX library version • HP-UX (IPF) version
Solaris	Y	Can be used in the following environment: • 32-bit-mode POSIX library version
AIX 5L	Y	
Linux	Y	Can be used in the following environments • 32-bit-mode version • Linux (EM64T) version

Legend:

Y: Can be used.

(2) Java stored procedures and Java stored functions

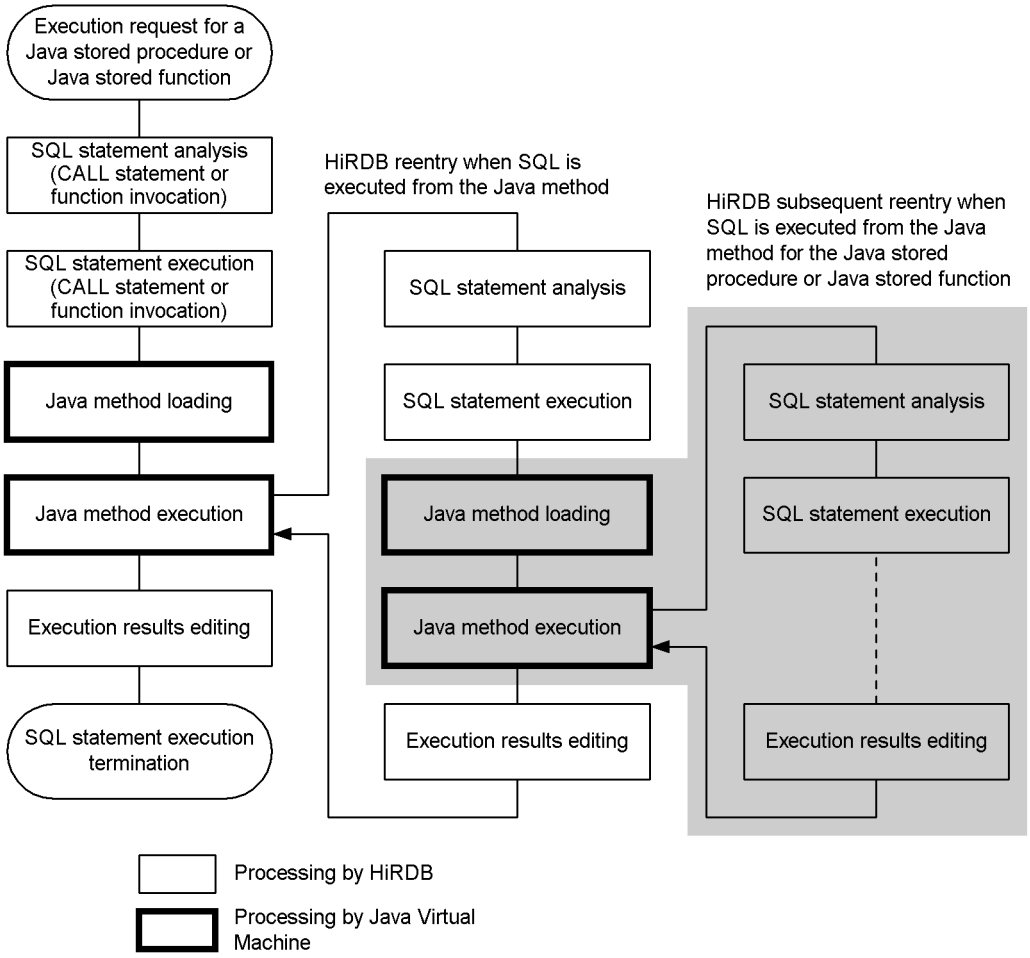
Java stored procedures and Java stored functions are stored procedures and stored functions in which the type of routine control statements that can be described in SQL are instead written in Java. Java stored procedures and Java stored functions use Java methods created externally to HiRDB. Because these methods are registered into HiRDB as routine control statements, it is possible to perform platform-independent processing, development, and debugging.

As with stored procedures and stored functions described in SQL, Java stored procedures and Java stored functions can be called from an SQL (`CALL` statement or function invocation). Thus, there are no restrictions on the control statement description language.

(3) Actions of Java stored procedures and Java stored functions

Figure 17-1 shows the actions (invocation procedures) of Java stored procedures and Java stored functions.

Figure 17-1: Actions (invocation procedures) of Java stored procedures and Java stored functions



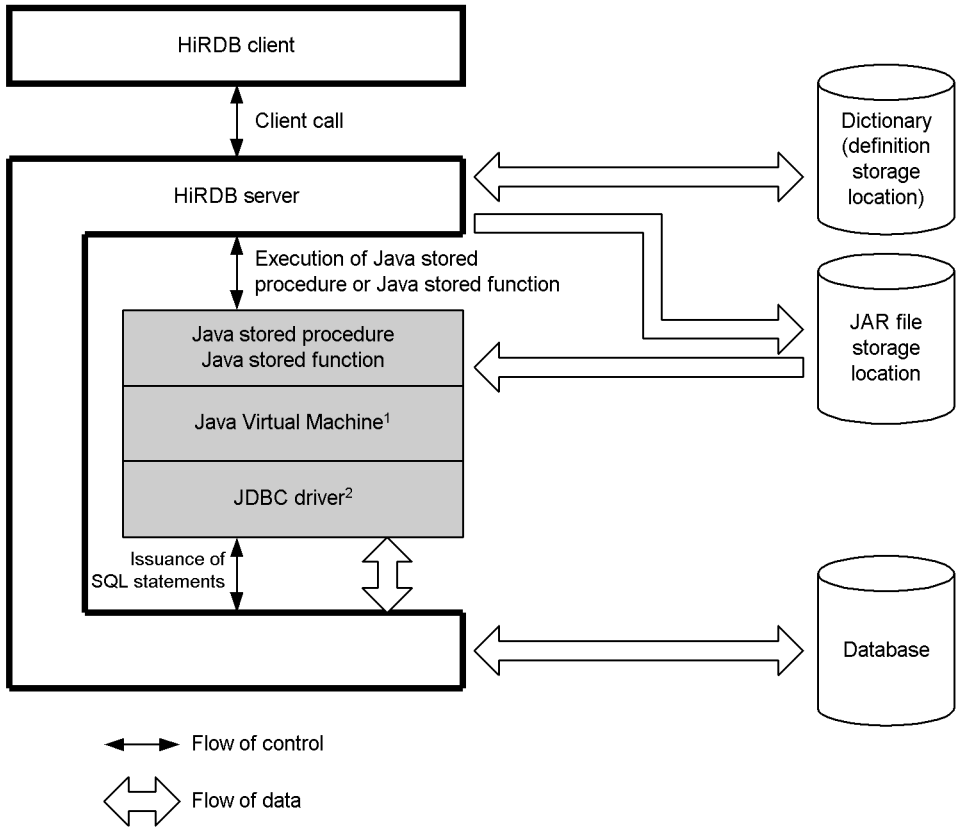
17.2 System configuration for using Java stored procedures and Java stored functions

This section explains the system configuration (positions of Java Virtual Machine, JAR file storage, etc.) for using Java stored procedures and Java stored functions.

(1) Java Virtual Machine position

Figure 17-2 shows the position of a Java Virtual Machine in a HiRDB system.

Figure 17-2: Position of Java Virtual Machine in a HiRDB system



[1] The Java virtual machine is included in a JRE (Java Runtime Environment). For details about JREs, see *(2) JRE (Java Runtime Environment)*.

[2] HiRDB provides the JDBC driver as a standard.

(2) JRE (Java Runtime Environment)

To use Java stored procedures and Java stored functions, a Java Runtime Environment (JRE) is required at the HiRDB server.

(a) How to obtain a JRE

You should be able to acquire necessary information about JREs and obtain the appropriate JRE by visiting the platform vendor's home page. The JRE version depends on the platform. Table 17-2 lists the JRE version for each platform.

Table 17-2: JRE versions

Platform		Version
HP-UX	Other than below	1.2.2.04 or later
	HP-UX (IPF)	1.4.2.02 or later
Solaris		1.3 or later
AIX 5L		1.3 or later
Linux	Other than below	1.3 or later
	Linux (IPF)	1.4.2 or later

(b) Handling of the JRE included in HiRDB version 07-02 and earlier

A JRE was included in HiRDB version 07-02 and earlier. This JRE will be deleted from the installation directory and the HiRDB directory at the following times:

- When the included JRE is deleted from the installation directory

  The JRE is deleted when the following events occur:

  - The HiRDB version that provided the JRE (07-02 or earlier) is uninstalled.
  - HiRDB version 07-03 or later is installed by overwriting the existing HiRDB.

- When the included JRE is deleted from the HiRDB directory

  When the file required for HiRDB execution is deleted in order to delete HiRDB from the OS (when `Y` is entered as the response to the `pdsetup -d` command's execution query message).

(c) Notes about upgrading to HiRDB version 07-03 or later

The following notes apply to upgrading to HiRDB version 07-03 or later.

- When you upgrade your HiRDB to version 07-03 or later, you must specify the root directory of the Java Runtime Environment (JRE) in the `pd_java_runtimepath` operand. If necessary, also specify in the

pd_java_libpath operand the directory that contains the JRE libraries.

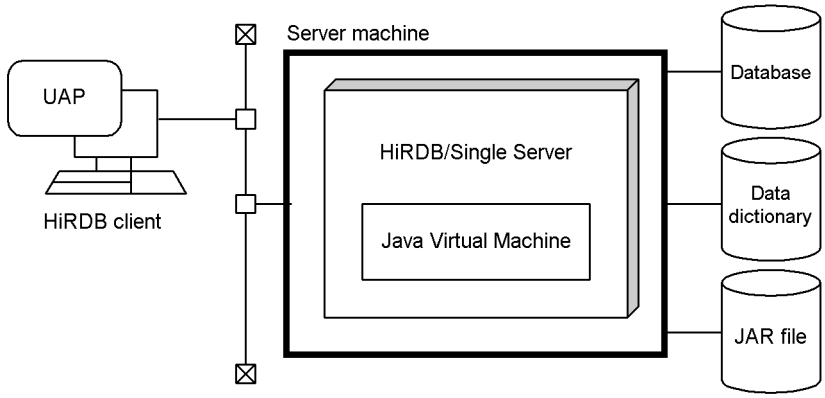
- The JRE that was expanded when HiRDB version 07-02 or earlier was installed will be deleted as explained in (b) above. If you want to continue using that JRE, do the following:

  1. Before the JRE is deleted, back it up to a directory other than the installation directory or HiRDB directory.
  2. In the pd_java_runtimepath operand, specify the directory containing the JRE that was backed up.

(3) System configuration in a HiRDB/Single Server

Figure 17-3 shows the system configuration for using Java stored procedures and Java stored functions in a HiRDB/Single Server.

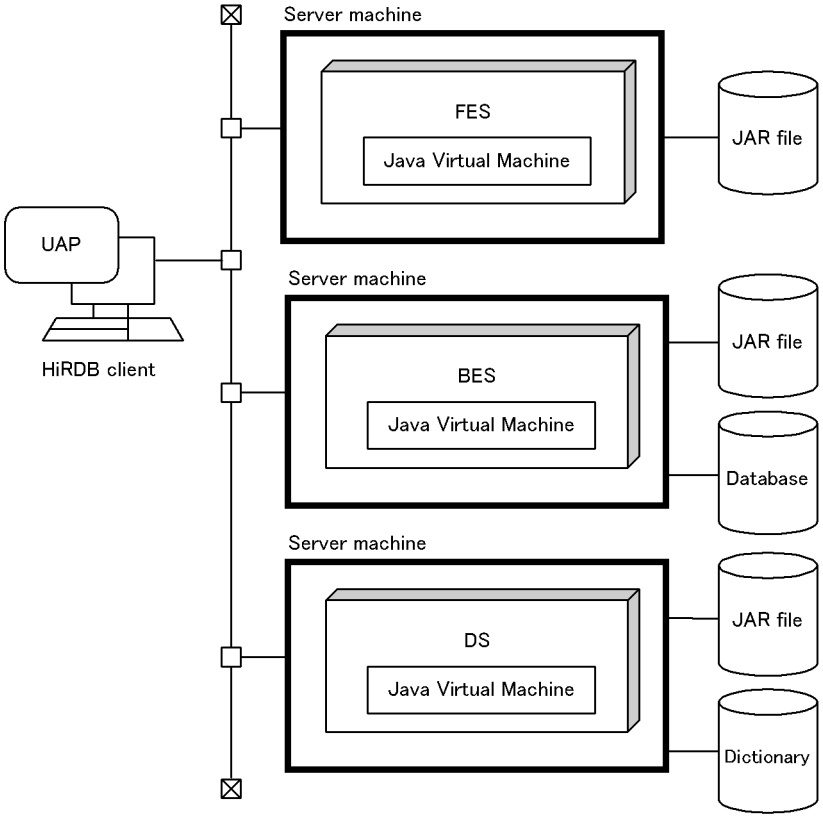
Figure 17-3: System configuration for using Java stored procedures and Java stored functions in a HiRDB/Single Server



(4) System configuration in a HiRDB/Parallel Server

Figure 17-4 shows the system configuration for using Java stored procedures and Java stored functions in a HiRDB/Parallel Server.

Figure 17-4: System configuration for using Java stored procedures and Java stored functions in a HiRDB/Parallel Server



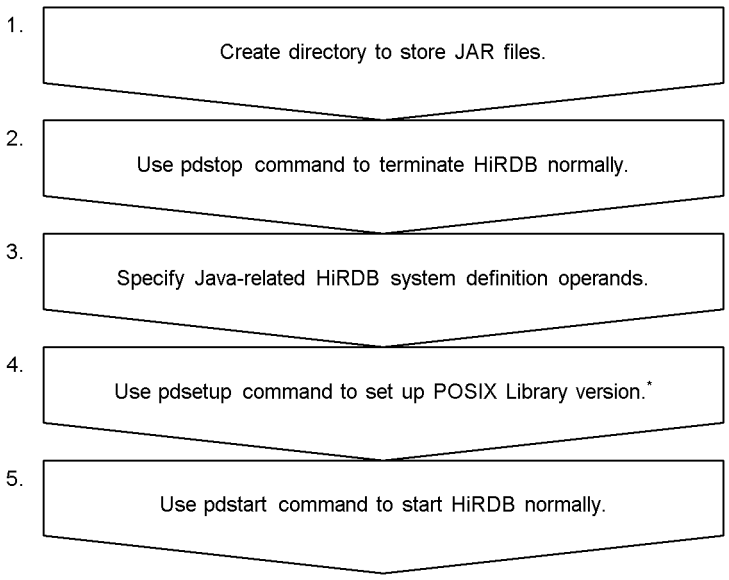
17.3 Environment setup

Executor: Superuser and HiRDB administrator

Figure 17-5 shows the procedure for setting up an environment for using Java stored procedures and Java stored functions. The explanation here assumes that the environment setup for the Java Virtual Machine and JDBC Driver has been completed.

When the environment setup explained here is completed, Java stored procedures and Java stored functions can be created and executed; for details on the creation and execution procedures, see the manual *HiRDB Version 8 UAP Development Guide*.

Figure 17-5: Environment setup procedure for use of Java stored procedures and Java stored functions



Note

The numbers to the left of the processing boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 3 is explained in paragraph (3) below.

* This step is applicable to the HP-UX, Solaris, and AIX 5L versions only; it is not required for the Linux version. This step is also not required if the POSIX Library version is already being used.

(1) Create a directory for storing JAR files

Create a directory for storing the JAR files. The name of the directory created here will

be specified in the pd_java_archive_directory operand in step (3).

Hint:

- Create a special directory in which to store JAR files.
- Do not store files other than installed JAR files in the JAR directory.

(2) Use the pdstop command to terminate HiRDB normally

```
pdstop
```

If you use the system reconfiguration command (pdchgconf command), you do not need to restart HiRDB normally, because this command enables you to modify HiRDB system definitions while HiRDB is running. Note that HiRDB Advanced High Availability must be installed in order to use this command. For details about modifying HiRDB system definitions while HiRDB is running, see *9.2 Modifying HiRDB system definitions while HiRDB is running (system reconfiguration command)*.

(3) Specify Java-related HiRDB system definition operands

Specify the following Java-related operands in the HiRDB system definition:

pd_java_option

Specifies the Java option.

pd_java_routine_stack_size

Specifies in bytes the size of the stack to be used by Java routines.

pd_java_archive_directory

Specifies the directory for storing the JAR files.

pd_java_classpath

Specifies a Java class path.

pd_java_runtimepath

Specifies the root directory of the Java Runtime Environment.

pd_java_libpath

Specifies the library directory of the Java Virtual Machine.

pd_java_stdout_file

Specifies the output destination file for the Java Virtual Machine's standard output and standard error output.

(4) Use pdsetup command to set up POSIX Library version (applicable to HP-UX, Solaris, and AIX 5L versions only)

To use a Java stored procedure or Java stored function, the POSIX library version must be used. To use the POSIX library version, specify the `-l` option in the `pdsetup` command that is executed at the time of environment setup for HiRDB. The `-l` option should be specified in the `pdsetup` command at the time a new HiRDB is installed.

If HiRDB is already running but the POSIX library version is not being used, apply the procedure described below to change HiRDB to the POSIX library version.

Procedure

1. Use the `pdsetup -d` command to delete HiRDB from the OS. Choose `y` in response to the message.

   In the case of a HiRDB/Parallel Server, execute the `pdsetup -d` command at all server machines.

2. Execute the `pdsetup` command with POSIX specified in the `-l` option.

   In the case of a HiRDB/Parallel Server, execute the `pdsetup -l` command at all server machines.

Note:

- Because it supports POSIX, you do not need to perform the above procedure for the HP-UX (IPF) version of HiRDB.
- HiRDB Version 5.0 and older versions do not support the POSIX library version. When HiRDB is being upgraded from Version 5.0 or older, use this procedure to change HiRDB to the POSIX library version.
- If the `-l` option was not specified in the `pdsetup` command when a new HiRDB was being installed, use this procedure to change the HiRDB to the POSIX library version.

(5) Use the pdstart command to start HiRDB normally

```
pdstart
```

17.4 JAR file operations

Executor: HiRDB administrator

The following operations can be performed with the `pdjarsync` command:

- JAR file registration
- JAR file re-registration
- JAR file deletion
- JAR file listing display

17.4.1 When an error occurs in a JAR file

When an error occurs in a JAR file, either each programmer must re-register the JAR file with an `INSTALL JAR` statement or the HiRDB administrator must re-register the JAR file with the `pdjarsync` command.

17.4.2 When the server configuration is modified (HiRDB/Parallel Server only)

When the HiRDB server configuration is modified, use the `pdjarsync -S` command to re-register the JAR files.

Chapter
18. Error Handling Procedures

This chapter explains the procedures for handling HiRDB errors.

It contains the following sections:

18.1 HiRDB processing and the HiRDB administrator's action in the event of an error

This section explains the following aspects of the processing by HiRDB and the actions that should be taken by the HiRDB administrator when an error occurs:

- Actions to be taken by the HiRDB administrator when an error occurs
- Information collected by HiRDB when an error occurs
- HiRDB processing when an error occurs
- Handling HiRDB processing errors
- Information inherited during HiRDB restart
- Facility for changing the process-down message when a transaction is cancelled

18.1.1 Actions to be taken by the HiRDB administrator when an error occurs

This section explains the actions to be taken by the HiRDB administrator when an error occurs.

(1) Actions to be taken by the HiRDB administrator

When an error occurs, the HiRDB administrator must take the following actions:

1. Check the output messages and the manual *HiRDB Version 8 Messages* to determine the causes of the error.
2. Use the `pdgeter` command to make a backup copy of the files under `$PDDIR/spool` and `$PDDIR/tmp` to which troubleshooting information has been output

Remarks

- Information needed to use problem-solving support and Q&A support services to resolve errors can be found in Appendix *C. Information Needed for Troubleshooting*. In the event of a problem, we recommend strongly that the HiRDB administrator always refer to Appendix *C. Information Needed for Troubleshooting*.
- Frequently asked questions about the error handling procedures are answered in Q&A format in Section *A.4 Errors*.

(2) Executing the pdgeter command

When the `pdgeter` command is executed, a backup copy of the troubleshooting information is created under a directory in the server machine where the `pdgeter` command was entered. Any directory can be specified in an option of the `pdgeter` command. The backup copy can also be created at any device, not just in the server

machine used to enter the pdgeter command (not supported in the AIX 5L version). The following is the procedure for executing the pdgeter command:

Procedure

1. Check that the server machine where the pdgeter command is entered has sufficient disk space to store the troubleshooting information.
2. Create a directory* for storing the troubleshooting information at the server machine where the pdgeter command is entered. This step is not necessary if the troubleshooting information is to be output to a device.
3. If troubleshooting information for a machine other than the server machine where the pdgeter command is entered is to be output to a device, a work directory is required. In such a case, create a work directory* in the server machine where the pdgeter command is entered.
4. Copy the pdinit control statement to $PDDIR/conf, using INITCONT as the file name. If all troubleshooting information is to be obtained (by specifying the -a option), copy the information under the file name INITCONT into $PDDIR/conf at the server machine where the pdgeter command is entered.
5. Copy the pdmod control statement into $PDDIR/conf. In this case, any file name can be used.
6. If shared memory is to be dumped (by omitting the -m option from the pdgeter command), check that there is a directory for storing a shared memory dump ($PDDIR/spool/pdshmdump) at the server machine subject to collection of troubleshooting information (the server machine with the host name specified in the -x option of the pdgeter command). If there is no such directory, create it.
7. Execute the pdgeter command. Specify the directories provided in steps 2 and 3 in the pdgeter command's options.
8. When the troubleshooting information is no longer needed, delete it with the pdcspool command (retaining it may result in a shortage of disk space).

* If this directory already exists, check whether or not it contains any of the directories or files listed below; if it contains these directories or files, the pdgeter command will result in an error.

- PDDIR
- lib
- usr
- HiRDB

(3) Deleting unneeded troubleshooting information

When a server process or client is terminated forcibly, HiRDB outputs troubleshooting information to the `$PDDIR/spool` directory. In addition, whenever the **Ctrl** + **C** keys are pressed to terminate a command or a utility while it is executing, the command or utility outputs a temporary work file to the `$PDDIR/tmp` directory, where it remains resident. If these troubleshooting and temporary work files are left on the disk, they may stress the capacity of the disk on which the HiRDB directory resides. A shortage of free space on the disk containing the HiRDB directory can cause HiRDB to terminate abnormally. To avoid such a problem, HiRDB deletes the following files periodically:

- Troubleshooting information files (files in the `$PDDIR/spool` directory)
- Temporary work files (files in the `$PDDIR/tmp` directory)

Normally, HiRDB deletes these files every 24 hours. You can change this deletion interval in the `pd_spool_cleanup_interval` operand. You can also specify that only files that were output a specified number of days previous to the current date are to be deleted; you make this specification in the `pd_spool_cleanup_interval_level` operand.

You can also use the following methods to all delete troubleshooting information (files in the `$PDDIR/spool` directory):

- You can use the `pdcspool` command to delete troubleshooting information files. You can also use this command to delete temporary work files (files in the `$PDDIR/tmp` directory).
- You can configure HiRDB to delete troubleshooting information files automatically when it starts. You use the `pd_spool_cleanup` operand to specify whether or not troubleshooting information files are to be deleted automatically. The default value for this operand is that these files are deleted automatically. You can also use the `pd_spool_cleanup_level` operand to specify that only troubleshooting information files that were output a specified number of days previous to the current date are to be deleted.

Reference note:

To select the troubleshooting information that is to be deleted, you can specify a `pdcspool` command option, or you can specify a value in the `pd_spool_cleanup_level` or `pd_spool_cleanup_interval_level` operand.

Note:

In some cases, troubleshooting information files that are output by commands or utilities executed by users other than the HiRDB administrator are not deleted. In these cases, a user with the privilege of deleting troubleshooting information files must delete the files with a command such as the OS's `rm` command.

(4) Reducing the amount of troubleshooting information that is output

You can specify the following operands to reduce the amount of troubleshooting information that is output. Specify these operands as needed.

- `pd_cancel_dump`: Specifies whether or not troubleshooting information is to be output.
- `pd_client_waittime_over_abort`: Specifies whether or not troubleshooting information is to be output when a client that is executing a transaction exceeds its maximum wait time (value specified in the `PDCWAITTIME` operand of the client environment definitions).
- `pd_debug_info_netstat`: Specifies whether or not network information is to be output in the troubleshooting information collected when a HiRDB process or HiRDB (unit) terminates abnormally.
- `pd_dump_suppress_watch_time`: Specifies a period of time during which re-output of troubleshooting information is to be suppressed.

18.1.2 Information collected by HiRDB when an error occurs

Table 18-1 shows the troubleshooting information that is collected by HiRDB when an error occurs.

Table 18-1: Troubleshooting information collected by HiRDB when an error occurs

Information collected	Explanation
Message log file (`$PDDIR/spool/pdlog1`, `pdlog2`)	This file consists of messages output by HiRDB; it can be viewed with the `pdcat` command. If an error occurs, make a backup of this file.
Standard output and standard error output when executing commands	These screens are output information of the operating command and error messages, which are displayed in the screen that the operating command is input. Save them as redirect if you needed.
Syslogfile	This file contains messages output by HiRDB.This file consists of messages output by HiRDB; it can be referenced using an OS editor. If there are too many accesses to syslogfile at the same time, some messages may not be output.

Information collected	Explanation
Save core file* ($PDDIR/spool/save/*file-name*)	This is HiRDB-related process data and stack information. Because only up to three core files are saved in each server, save core files that need to be saved should be backed up. The file name format is *server-name-n*, where n is the serial number (1-3) of the save core file. Note that a serial number might not be assigned in some cases.
Abort information file* (`$PDDIR/spool/save/`*file-name*)	This is a file of abort information. If this information is output, back it up. The abort codes only can be referenced using an OS editor.The file name format is `abcode.`*server-process-ID*.
Snap during an error* (`$PDDIR/spool/save/`*file-name*)	This is a file of snap information generated during an error. If this information is output, back it up. The file name format is *server-name-n*`.deb`, where *n* is the serial number (1 to 3) of the save core file. Note that a serial number may not be assigned in some cases.
Shared memory dump file* (`$PDDIR/spool/pdshmdump/`*file-name*)	This file consists of data maintained by HiRDB in the shared memory. If this information is output, back it up. The file name format is *server-name*`.rmb.`*server-process-ID*.
Simple dump file* (`$PDDIR/spool/`*server-directory/file-name*)	This file consists of data maintained by HiRDB in the shared memory and process private memory. If this information is output, back it up. The file name format is a combination of the date and process ID.
RPC trace file	This is a file of the message sent and received by HiRDB using RPC. If this information is output, back it up. The file name is specified in the `pd_rpc_trace_name` operand.
Command trace file (`$PDDIR/spool/cmdlog1`, `cmdlog2`)	This file consists of statistical information on executed commands (including commands generated internally by HiRDB). Use an OS editor to reference this information.If an error occurs, back up this statistical information.
Error log file (`$PDDIR/spool/errlog/errlog1`, `errlog2`)	This is a file of internal information output by HiRDB. If this information is output, back it up.
Connected users data file (`$PDDIR/spool/cnctusrinf`)	This file contains information on users who were connected when HiRDB terminated. Use an OS-provided text editor to view this information.
Connected users details file (`$PDDIR/spool/cnctusrdtl`)	
Data file for locked resources management table (`$PDDIR/spool/pdlckinf/`*output-date-and-time*`.mem`)	This is a file of user information when deadlock, a lock-release timeout, or an error due to a space shortage in the table for managing locked resources occurs during locking by HiRDB. Use an OS-provided text editor to view this information. If this information is output, back it up.

* This is troubleshooting information that is output when a HiRDB server process terminates abnormally. Abnormal termination of a HiRDB server process can be confirmed when the KFPS01820-E message is output. The server name, process ID, and termination status can be determined from this message. Note, however, that troubleshooting information is not output in the following cases:

1. Termination status begins with *c* or *d* (no troubleshooting information is output)
2. Termination status is 0009 (the abort information, save core file, and shared memory dump file are not output)

18.1.3 HiRDB processing in the event of an error

This section explains the processing that HiRDB performs when an error occurs.

(1) Range of applicability of an error

When a hardware or software error occurs in a HiRDB/Parallel Server, HiRDB isolates the affected units and terminates them abnormally, rather than shutting down all the units comprising the HiRDB (although there are some exceptions to this). When an error occurs in a HiRDB/Single Server, HiRDB terminates the unit, which results in complete shutdown of HiRDB, because a HiRDB/Single Server consists of only one unit.

(2) System recovery in the event of the abnormal termination of HiRDB

When HiRDB has terminated abnormally and it is restarted after the cause of the error has been eliminated, HiRDB restores the system to its status immediately before the error occurred.

(3) Consecutive occurrences of abnormal termination

When HiRDB terminates abnormally, the unit will usually restart automatically, depending on the specification of the pd_mode_conf operand in the system common definition. However, if HiRDB terminates abnormally three times* in succession, the unit will not restart again automatically.

In such a case, the HiRDB administrator must eliminate the cause of the error and use the pdstart command to restart HiRDB.

* The pd_term_watch_count operand can be used to specify a maximum number of consecutive abnormal terminations that can occur. For example, if pd_term_watch_count=2 is specified, HiRDB restart will not be attempted again after two consecutive abnormal terminations.

18.1.4 Handling of HiRDB process errors

If an error occurs during an HiRDB process, HiRDB terminates the process abnormally, then rolls back the process for each server by activating a recovery process. There is no need for the HiRDB administrator to take any action, because the

corresponding process is restarted automatically.

Depending on the severity of the error, HiRDB may terminate abnormally the unit that is executing the erroneous process. In such a case, the HiRDB administrator must restart the abnormally terminated unit (although the unit may be restarted automatically, depending on the specification of the `pd_mode_conf` operand).

Described below are the actions to be taken by the HiRDB administrator when a unit terminates abnormally due to abnormal termination of a process.

(1) HiRDB/Single Server

Table 18-2 shows the action to be taken by the HiRDB administrator in the event a HiRDB/Single Server terminates abnormally.

Table 18-2: HiRDB administrator's action in the event of an abnormal termination of a HiRDB/Single Server

HiRDB processing	HiRDB administrator's action
Terminates the HiRDB/ Single Server abnormally	Eliminate the cause of the error and restart the HiRDB/Single Server (the HiRDB/ Single Server may be restarted automatically, depending on the `pd_mode_conf` operand specification), then re-execute the previous processing.

(2) HiRDB/Parallel Server

Table 18-3 shows the actions to be taken by the HiRDB administrator in the event a HiRDB/Parallel Server terminates abnormally.

Table 18-3: HiRDB administrator's actions in the event of an abnormal termination of a HiRDB/Parallel Server

Error status	HiRDB processing	HiRDB administrator's action
Process error at unit controller	Terminates abnormally the unit where the corresponding process is located.	Restart the corresponding unit (the unit may be restarted automatically, depending on the `pd_mode_conf` operand specification).
Process error at system manager	Terminates abnormally the unit where the system manager is located. HiRDB does not accept any operation command or UAP execution request while the system manager is shut down. No message log can be collected during that time.	Eliminate the cause of the error and restart the corresponding unit (the unit may be restarted automatically, depending on the `pd_mode_conf` operand specification), then re-execute the previous processing.

Error status	HiRDB processing	HiRDB administrator's action
Process error at front-end server	Terminates abnormally the unit where the front-end server is located. HiRDB does not accept any processing request while the front-end server is shut down. It treats a processing request as a timeout-level communication error. If a front-end server process terminates abnormally, HiRDB recovers the process. If it is necessary to shut down the unit in such a case, HiRDB terminates abnormally the unit where the corresponding process is located.	Eliminate the cause of the error and restart the corresponding unit (the unit may be restarted automatically, depending on the `pd_mode_conf` operand specification), then re-execute the previous processing.
Process error at dictionary server	Terminates abnormally the unit where the dictionary server is located.	
Process error at back-end server	Terminates abnormally the unit where the back-end server is located.	

18.1.5 Information inherited during a HiRDB restart

Table 18-4 shows the information that is inherited during a HiRDB restart.

Table 18-4: Information inherited during a HiRDB restart

Classification		Information that is inherited
System files	System log files	System recovery information
	Synchronization point dump files	Table recovery information
	Status files	1. Information used to determine the system startup mode: ● Server configuration 2. Status information about each system server: ● Open/close status ● Primary/standby (alternate) status ● File shutdown status ● Service group shutdown status 3. RDAREA-related information
User data (database)	RDAREAs	• Open/close status • Shutdown status*

Classification		Information that is inherited
Operation commands (status changes made by the commands shown at the right are inherited)	Database manipulation commands	• pdclose (close RDAREA) • pdopen (open RDAREA) • pdhold (shut down RDAREA)* • pdrels (release RDAREA from shutdown status)
	Commands for manipulating system log files or synchronization point dump files	• pdlogadpf (allocate file) • pdlogchg (change file status) • pdlogcls (close file) • pdlogopen (open file) • pdlogswap (swap files) • pdlogunld (unload file)
	Commands for manipulating status files	• pdstscls (close file) • pdstsopen (open file) • pdstsswap (swap files)
	Commands for output of audit trails	• pdaudbegin command (begin collection of audit trail) • pdaudend command (end collection of audit trail)

Note

The following information is not inherited (these commands must be entered again after HiRDB is restarted):

- pdstbegin (start collection of statistical information)
- pdstend (terminate collection of statistical information)

* The following hold statuses are not inherited:

- Referencing-permitted backup-hold (update WAIT mode)
- Updatable backup-hold
- Updatable backup-hold (WAIT mode)

18.1.6 Facility for changing the process-down message when a transaction is cancelled

(1) Overview

If a server process is terminated by a forced termination request due to an interrupt at the client during transaction execution, or if a server process connected to a UAP is terminated by the `pdcancel` command, HiRDB displays the KFPS01820-E and KFPO00105-E messages that indicate termination of the server process. These messages are also displayed when a server process is terminated due to some abnormality in the server's processing. To determine the cause of such a termination, you must also check other messages, such as `KFPS00993-I`, that are displayed by HiRDB.

Use of the facility for changing the process-down message when a transaction is cancelled enables you to change the IDs of the `KFPS01820-E` and `KFPO00105-E` messages that are displayed in the case of client-related operations and errors. When you do this, you can easily identify a termination of server processing that was caused by a client-related operation or error simply by checking the message IDs (eliminating the need to check other messages).

(2) Function

This facility changes only the message IDs; it does not change the message text or output information (the original message text and output information are still displayed as is even after the message ID has been changed).

(a) Messages subject to change

Table 18-5 lists the messages that can be changed by this facility.

Table 18-5: Messages whose message IDs can be changed

Message ID that can be changed	Description of the message	Message ID after change
`KFPS01820-E`	Server process was terminated.	`KFPS01852-W`
`KFPO00105-E`	Server process was terminated because an error occurred (abort message).	`KFPO00115-W`

For details about these messages, see the manual *HiRDB Version 8 Messages*.

(b) Whether or not message IDs are changed

Table 18-6 shows whether or not message IDs are changed, depending on the process-down event.

Table 18-6: Whether or not message IDs are changed depending on the process-down event

No.	Proc-down event	Cause of process-down						Msg ID chgd	Remarks
		User op	Env set	Client error	Client comm err	Serv error	S-to-S comm error		
1	Forced termination request due to an interrupt at the client	C	--	--	--	--	--	Yes	HiRDB server process was terminated during execution of a transaction due to a forced termination request caused by an interrupt at the client.#1
2	HiRDB server process killed due to a client process-down	--	--	C	C	--	--	Yes	For a client process using XA, see No. 11.
3	HiRDB server process killed because PDCWAITTIME was exceeded	--	C#2	--	C	C	--	No	HiRDB cannot identify the cause.#3
4	Process-down (exit) at HiRDB server because PDSWAITTIME was exceeded	--	C	C	C	--	--	Yes	--
5	Process-down (exit) at HiRDB server because PDSWATCHTIME was exceeded	--	C	C	C	--	--	Yes	--

No.	Proc-down event	Cause of process-down						Msg ID chgd	Remarks
		User op	Env set	Client error	Client comm err	Serv error	S-to-S comm error		
6	HiRDB server process killed due to forced termination of unit, unit down, or transaction recovery during system switchover	C	--	--	--	--[#4]	--[#4]	Yes	Applicable to another unit's branch recovery. This is applicable only to a HiRDB/ Parallel Server.
7	HiRDB server process killed due to forced termination of unit, unit down, or utility branch recovery during system switchover	C	--	--	--	--[#4]	--[#4]	Yes	Applicable not only to recovery in the local unit but also to another unit's branch recovery.
8	HiRDB server killed by `pdcancel` command (including a killed utility)	C	--	--[#4]	--	--[#4]	--[#4]	Yes	--
9	HiRDB server process killed due to forced termination of transaction by `pdfgt` command (transaction branch recovery)	C	--	--[#4]	--	--[#4]	--[#4]	Yes	--
10	HiRDB server process killed by the facility for monitoring free area for system log file	C[#5]	C[#6]	--	--	--	--	Yes	`KFPS01160-E` message is displayed.

No.	Proc-down event	Cause of process-down						Msg ID chgd	Remarks
		User op	Env set	Client error	Client comm err	Serv error	S-to-S comm error		
11	HiRDB server process killed by XA completion instruction issued by the transaction manager in an extension of transaction recovery	C	--	--	--	C	C	No	HiRDB cannot identify the cause.#7
12	HiRDB server process killed due to recovery when the synchronization skip count exceeded the boundary value	C#8	C#6	--	--	C	--	No	HiRDB cannot determine the cause.
13	Process-down (`exit`) at HiRDB server due to timeout during communication between HiRDB server processes	--	--	--	--	C	C	No	--
14	Communication request processing from a shutdown unit when recovery-unnecessary front-end server is used	--	--	--	--	C	C	No	--

No.	Proc-down event	Cause of process-down						Msg ID chgd	Remarks
		User op	Env set	Client error	Client comm err	Serv error	S-to-S comm error		
15	Process-down (`exit`) at server due to transaction completion error when recovery-unnecessary front-end server is used	--	--	--	--	C	C	No	--
16	Process-down at HiRDB server when automatic log unloading facility was terminated due to a server failure	--	--	--	--	C	C	No	--
17	HiRDB server process failure shutdown (such as internal failure, process failure, or transaction start error)	--	--	--	--	C	C	No	--
18	HiRDB server process killed during transaction and utility branch recovery due to HiRDB server process failure shutdown	--	--	--	--	--[#4]	--[#4]	Yes	Recovery processing caused by No. 17.

Legend:

Proc-down event: Process-down event

User op: User operation

Env set: Environment settings

Client comm err: Client communication error

Serv error: Server error

S-to-S comm error: Server-to-server communication error

Msg ID chgd: Message ID changed

C: Considered to be the cause of the process-down.

--: Not the cause of the process-down.

Yes: Message is changed.

No: Message is not changed.

#1

This is applicable to UAPs that use DBPARTNER or DABroker to access HiRDB, and to UAPs that access HiRDB via ODBC. This also includes cases such as when the applicable UAP is terminated by pressing the **Ctrl** + **C** keys on the keyboard.

#2

Applicable when the value is less than the transaction processing time.

#3

This requires user judgment because HiRDB cannot determine whether this is a timeout caused by environment settings or by a server failure. For this reason, message output is required.

#4

Although this is not an error event, it can cause an error event.

#5

Applicable when the volume of transactions exceeds the expected value.

#6

Applicable when a boundary value is invalid.

#7

This is caused by a completion instruction from the transaction manager. If a transaction is cancelled due to a client failure, by the user, or due to a server failure, the transaction manager issues a completion instruction. In this case, HiRDB cannot identify the cause of the completion instruction.

#8

Applicable when the issued SQL statement is invalid.

(c) Unneeded operands

To use this facility, either omit the `pd_cancel_down_msgchange` operand or specify `Y`.

If you have specified `v6compatible` or `v7compatible` in the `pd_sysdef_default_option` operand, the default value of the `pd_cancel_down_msgchange` operand is set to `N`; therefore, you must specify `Y` in the `pd_cancel_down_msgchange` operand.

18.2 When a UAP does not execute correctly

Executor: HiRDB administrator

This section explains the actions to be taken when a UAP does not execute correctly.

(1) Actions to be taken when a UAP will not execute

Table 18-7 lists the possible reasons for UAP execution errors.

Table 18-7: Causes of UAP execution errors and actions to be taken

Possible cause	Action to be taken
An HiRDB system definition is invalid	A message is displayed indicating the HiRDB system definition that is invalid. Correct the definition on the basis of this message.
Memory is insufficient	A message is issued indicating the memory where the shortage occurred. For shared memory, reevaluate the HiRDB system definitions. For process private memory, terminate any unneeded processes.
Too many users are executing concurrently	• Increase the value of the `pd_max_users` operand. • If the maximum number of server processes for front-end server or single server has been changed with the `pdchprc` command, increase the maximum value.
An environment variable is invalid in the client environment definition	See the manual *HiRDB Version 8 UAP Development Guide* and correct the environment variable.
HiRDB has not been started	Start HiRDB. For a HiRDB/Parallel Server, a specific unit or server may be shut down; in such a case, start the unit or server.

(2) Actions to be taken when a UAP does not terminate itself

Check the UAP's execution status (for the procedure, see *8.2 When a UAP or utility execution takes too long*). If necessary, use the `pdcancel` command to terminate the UAP forcibly.

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(3) Actions to be taken when a UAP terminates abnormally

If a UAP terminates abnormally, use the `pdls -d prc` command to check for any remaining UAP processes. If there are any remaining UAP processes, terminate them with the `pdcancel` command.

It is recommended that after the command has executed you check whether or not the

execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

18.3 When operation commands do not execute correctly

This section explains the actions to be taken when operation commands do not execute correctly.

18.3.1 Actions to be taken when operation commands will not execute

Executor: Operation command executor or HiRDB administrator

Table 18-8 lists the possible causes of operation command execution errors.

Table 18-8: Possible causes of operation command execution errors and actions to be taken

Possible cause	Action to be taken
An invalid option or argument was specified in the operation command.	Correct the option or argument and re-execute the operation command.
The user does not have the required privilege to execute the operation command.	Refer to the *HiRDB Version 8 Command Reference* For details on user execution privileges for operation commands. If the user does not have the required execution privilege, either ask an authorized user to execute the command or obtain the required execution privilege.
A command available only while HiRDB is active was executed while HiRDB was inactive.	Check the *HiRDB Version 8 Command Reference* for whether or not the command can be executed while HiRDB is inactive. If the command is available only while HiRDB is active, start HiRDB and execute the command.
An operation command execution environment has not been set up.	Check that the following environment setup is correct: • Environment variables • Remote shell execution environment For details on these environment setup procedures, see the manual *HiRDB Version 8 Installation and Design Guide*.
An HiRDB system definition is invalid.	See the manual *HiRDB Version 8 System Definition* and correct the HiRDB system definition.

18.3.2 Actions to be taken when an operation command results in a timeout while waiting for a response

Executor: Operation command executor

An operation command may result in a timeout in the following cases:

- Processing was not completed within the response wait time because the OS workload was too high.
- An operation command was executed while HiRDB was inactive.

If the OS workload is high, re-execute the operation command. If the response wait timeout recurs, terminate any unneeded processes before re-executing the operation command.

18.4 When HiRDB does not start

This section explains the actions to be taken when HiRDB does not start. The following items are explained:

- When HiRDB does not start normally
- When HiRDB does not restart
- Actions when an error occurs in the master directory RDAREA
- Actions when other errors occur

18.4.1 When HiRDB does not start normally

Executor: HiRDB administrator

Table 18-9 lists the possible causes of errors during HiRDB normal startup.

Table 18-9: Possible causes of errors during HiRDB normal startup and actions to be taken

Possible cause	Action to be taken
HiRDB has not been installed or was not set up correctly.	Install HiRDB or set up HiRDB correctly.
An HiRDB system definition is invalid.	A message is displayed indicating the HiRDB system definition that is invalid. Correct the HiRDB system definition on the basis of this message.
An environment variable definition is invalid.	Correct the environment variable. For details on the environment variable setup procedures, see the manual *HiRDB Version 8 Installation and Design Guide*.
A remote shell execution environment has not been set up.	Set up the remote shell execution environment; for details, see the manual *HiRDB Version 8 Installation and Design Guide*.
A memory shortage or file space shortage occurred.	A message reporting the memory or file space shortage is displayed. Either terminate any unneeded processes or delete unneeded files. In the case of shared memory, reevaluate the HiRDB system definitions. In the case of process private memory, terminate any unneeded processes. Also, reevaluate the OS's shared memory-related operating system parameters, as required. For details on the operating system parameters, see the manual *HiRDB Version 8 Installation and Design Guide*.
A file required for HiRDB startup is missing.	A message is displayed indicating the file required for HiRDB startup that is missing. Create the required file on the basis of this message.

Possible cause	Action to be taken
An error occurred in a file required for HiRDB startup.	Check the erroneous file on the basis of the message, then take appropriate action by referencing Sections *18.6 Handling of system log file errors* to *18.9 Handling of errors in files other than system files*.
The OS configuration is not appropriate to the HiRDB execution environment.	Reconfigure the OS and correct the HiRDB system definitions.
The current system log file cannot be allocated.	Place the system log file waiting to be unloaded in unload completed status with the `pdlogunld` or `pdlogchg` command.
An error has occurred in a particular unit (HiRDB/Parallel Server only).	A HiRDB/Parallel Server will not start if any of its units cannot be started. If an error occurs in a unit, HiRDB cannot start until the unit's error is corrected. However, HiRDB can be started using only the normal units if *reduced activation* is in effect. For details on reduced activation, see *18.15 Handling of reduced activation (HiRDB/Parallel Server only)*.
A library is not shared correctly (applicable to a multi-HiRDB).	Share the library correctly; for details, see the manual *HiRDB Version 8 Installation and Design Guide*.

Remarks

Frequently asked questions about errors during HiRDB startup are answered in Q&A format in Section *A.6 HiRDB startup*.

18.4.2 When HiRDB does not restart

Executor: HiRDB administrator

When a HiRDB restart fails, see the message that is displayed during restart processing. Table 18-10 lists the possible causes.

Table 18-10: Possible causes of HiRDB restart errors and actions to be taken

Possible cause	Action to be taken
HiRDB cannot be restarted due to an error in both versions of the current system log file.	See *18.6.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file*.
HiRDB cannot be restarted because there is no system log file in swappable target status.	See *18.17 When a HiRDB (unit) terminates due to a system log file space shortage*.
HiRDB cannot be restarted because the system log file corresponding to a synchronization point dump file has been overwritten.	See *18.7(2) Actions to be taken when HiRDB cannot be restarted because the system log file corresponding to a synchronization point dump file has been overwritten*.

Possible cause	Action to be taken
HiRDB cannot be restarted because there are not enough synchronization point dump files.	See *18.7(3) Actions to be taken when HiRDB cannot be restarted due to an insufficient number of synchronization point dump files*.
HiRDB cannot be restarted due to an error on both versions of the current status file.	See *18.6.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file*.
An error has occurred in a particular unit (HiRDB/ Parallel Server only).	A HiRDB/Parallel Server will not start if any of its units cannot be started. If an error occurs in a unit, HiRDB cannot start until the unit's error is corrected. However, HiRDB can be started using only the normal units *if reduced activation* is in effect. For details on reduced activation, see *18.15 Handling of reduced activation (HiRDB/Parallel Server only)*.
HiRDB cannot be restarted due to an error in the master directory RDAREA.	See *18.4.3 Actions to be taken in the event of an error in the master directory RDAREA*.
HiRDB cannot be restarted due to some other error.	See *18.4.4 Actions to be taken in the event of other errors*.

18.4.3 Actions to be taken in the event of an error in the master directory RDAREA

If an error occurs in the master directory RDAREA, HiRDB cannot be restarted. In such a case, the following procedure must be used to restore the RDAREA:

Procedure

To restore the RDAREA:

1. Enter the `pdstart -r` command to start HiRDB.
2. Use the `pdrstr` command to restore the master directory RDAREA:

   ```
   pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
   -l /unld/unldlog01,/unld/unldlog02 -w /tmp/sortwork/
   -r PDBMAST
   ```

3. Enter the `pdstop` command to terminate HiRDB.
4. Enter the `pdstart` command to start HiRDB.
5. Use the `pdclose` command to close the erroneous RDAREA (master directory RDAREA):

   ```
   pdclose -r rdarea01,rdarea02
   ```

6. Use the database recovery utility (`pdrstr` command) to restore the

RDAREA:

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-l /unld/unldlog01,/unld/unldlog02 -w /tmp/sortwork/
-r rdarea01,rdarea02
```

7. Use the `pdrels -o` command to release the restored RDAREA from shutdown status and then open it:

```
pdrels rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

18.4.4 Actions to be taken in the event of other errors

(1) When the HiRDB process registered in the OS is not active

The HiRDB process registered with the `pdsetup` command may not be active for some reason.

The OS attempts to activate the process. If this attempt fails the specified number of times, the process will no longer start and it will not be possible to start HiRDB by entering the `pdstart` command. The following procedure should be executed in such a case:

Procedure

1. Use the OS's `ps` command to check in the HiRDB directory for the process that is being executed.
2. If the process is not active, enter the `pdsetup -d` command.
3. Enter the `pdsetup` command.

If the server machine is shut down due to a power failure during HiRDB operation, the same event may occur immediately after power is restored and the OS is started due to a disk error in the HiRDB directory. In such a case, the same action should be taken.

(2) When the HiRDB process registered in the OS is active

Procedure

1. A message is displayed when the `pdstart` command is entered. Eliminate the cause of the error on the basis of this message.
2. Use the `pdsetup -d` command to delete the HiRDB process from the OS.
3. Enter the `pdsetup` command.

18.5 When HiRDB does not terminate

Executor: HiRDB administrator

This section explains the actions to be taken when HiRDB will not terminate. Table 18-11 lists the possible causes that prevent HiRDB from terminating.

Table 18-11: Possible causes that prevent HiRDB from terminating and actions to be taken

Possible cause	Actions
A user is still connected.	As long as a user is still connected, HiRDB cannot terminate. For details on the actions to be taken in such a case, see *18.13 When HiRDB cannot be terminated because a user is still connected*.
There is an undetermined transaction	If there is an undetermined transaction, HiRDB cannot be terminated. For details on the actions to be taken in such a case, see *18.14 Actions when there is an undetermined transaction*.
Communication error occurred	When a utility special unit is used or in the case of a HiRDB/Parallel Server, it may not be possible to terminate HiRDB because of a communication error between servers. For details on the actions to be taken in such a case, see *18.12.1 Handling of communication errors*.

18.6 Handling of system log file errors

This section discusses the following topics:

- Actions to be taken in the event of an error in the current file
- Actions to be taken when HiRDB Datareplicator is being used
- Actions to be taken when HiRDB (unit) cannot be restarted due to an error in both versions of the current file

18.6.1 Actions to be taken in the event of an error in the current file

(1) Error during HiRDB operation

Table 18-12 shows the actions to be taken in the event of an error in the current file during HiRDB operation.

Table 18-12: Actions to be taken in the event of an error in the current file during HiRDB operation

Condition at the time of an error		HiRDB processing	HiRDB administrator's action
Write operation	There are files in swappable target status.	Swaps the system log files. Places the erroneous system log file in reserved status, and places in current status one of the files in swappable target status, then resumes processing.	• If application history is required, unload the reserved system log file with the `pdlogunld` command. • Place the erroneous file in swappable target status with the method shown in (3).
	There are no files in swappable target status.	Terminates abnormally the unit containing the erroneous system log file.	See *18.17 When a HiRDB (unit) terminates due to a system log file space shortage*. • Create a file that can be placed in swappable target status, then restart HiRDB. • Place the erroneous file in swappable target status with the method shown in (3).

Condition at the time of an error		HiRDB processing	HiRDB administrator's action
Read operation	Dual system log files are used.	Switches to the normal file version and resumes processing.	Place the erroneous file in swappable target status with the method shown in (3).
	Dual system log files are not used.	Places the erroneous system log file in reserved status and cancels processing.	• If the erroneous system log file has not been unloaded, unload it with the `pdlogunld` command.* • Recover the database from a backup copy of the unload log file and the created unload log file using the database recovery utility. For details on the database recovery utility, see *19. Database Recovery Procedures*. • Place the erroneous file in swappable target status with the method shown in (3).

* When checking of the unload operation has been released (`pd_log_unload_check=N` is specified), this action is not required.

(2) Error during HiRDB restart processing

Table 18-13 shows the actions to be taken in the event of an error in the current file during HiRDB restart processing.

Table 18-13: Actions to be taken in the event of an error in the current file during HiRDB restart processing

Condition at the time of an error	HiRDB processing	HiRDB administrator's action
Dual system log files are used	Switches to the normal file version and resumes processing.	After restarting HiRDB, place the erroneous file in swappable target status with the method shown in (3).
Dual system log files are not used	Places the erroneous system log file in reserved status and terminates restart processing of the unit containing the erroneous system log file.	• If the erroneous system log file has not been unloaded, unload it with the `pdlogunld` command, then start the unit forcibly. • Recover the database from a backup copy of the unload log file and created unload log file using the database recovery utility. For details on the database recovery utility, see *19. Database Recovery Procedures*. • Place the erroneous system log file in swappable target status with the method shown in (3).

(3) Placing an erroneous file in swappable target status

Procedure

To place an erroneous file in swappable target status:

1. Use the `pdlogls` command to determine the system log file that was placed in reserved status due to an error:

   ```
   pdlogls -d sys -s b001
   ```

2. Use the `pdlogrm` command to delete the system log files in reserved status:

   ```
   pdlogrm -d sys -s b001 -f /sysfile/syslog1a
   pdlogrm -d sys -s b001 -f /sysfile/syslog1b
   ```

3. Use the `pdloginit` command to re-create the system log files that were deleted in step 2:

   ```
   pdloginit -d sys -s b001 -f /sysfile/syslog1a -n 5000
   pdloginit -d sys -s b001 -f /sysfile/syslog1b -n 5000
   ```

4. Use the `pdlogopen` command to place in swappable target status the system log files that were re-created in step 3:

   ```
   pdlogopen -d sys -s b001 -g syslog01
   ```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(4) Actions to be taken in the event of a disk error

The actions to be taken in the event of a disk error are explained below by way of an example.

Example

An error occurred in one of the current file versions due to a disk error. Because a swappable target file was available, HiRDB swapped the system log files and continued operation.

Action

If online operation is continued as is and there is not enough space in the system log file, the unit may terminate abnormally due to a shortage of space in the system log file. In this case, immediately swap disks and take the action shown in Procedure 1 below.

If disks cannot be swapped immediately, take the action shown in Procedure 2. If neither Procedure 1 nor Procedure 2 can be executed immediately, take the action shown in Procedure 3 below.

Procedure 1: When disks can be swapped immediately

To continue the operation:

1. Enter the `pdstop` command to terminate HiRDB normally.
2. Swap disks and use the `pdfmkfs` command to create a HiRDB file system area.
3. Use the `pdloginit` command to create a system log file.
4. Enter the `pdstart` command to start HiRDB normally.

Procedure 2: When disks cannot be swapped immediately

To correct the shortage of disk space:

1. Enter the `pdstop` command to terminate HiRDB normally.
2. Use the `pdloginit` command to create a system log file in a HiRDB file system area that has sufficient space.
3. Modify the following server definition operands (i.e., add operands for the system log file that was added):
   - `pdlogadfg`
   - `pdlogadpf`
4. Enter the `pdstart` command to HiRDB start normally.

Procedure 3: Neither Procedure 1 nor Procedure 2 can be executed immediately

To continue the operation temporarily:

1. Enter the `pdstop` command to terminate HiRDB normally.
2. Specify `pd_log_singleoperation=Y` (to implement the single operation mode for the system log file) in the server definition for the erroneous system log file.
3. Enter the `pdstart` command to start HiRDB normally.

Note:

If Procedure 3 is used and an error occurs in the system log file during single operation, the database cannot be recovered from the unload log (system log). Therefore, Procedure 1 or Procedure 2 should be used as an immediate measure when possible and Procedure 3 should be used as a temporary measure only.

18.6.2 Actions to be taken when HiRDB Datareplicator is being used

If an input error occurs in the system log file (or in both versions of the system log file when dual system log files are being used) and the system log file is required for data

linkage while HiRDB Datareplicator is being used, data linkage becomes unavailable. In such a case, the following action must be taken:

Procedure

To restore data linkage availability:

1. Enter the `pdrplstop` command to terminate HiRDB Datareplicator linkage.
2. To restart HiRDB Datareplicator linkage after correcting the error in the system log file, enter the `pdrplstart` command. Once HiRDB Datareplicator linkage is cancelled, conformity may be lost between the extracted database and the target database subject to data linkage. Therefore, before entering the `pdrplstart` command, it is important to re-create the target database on the basis of the extracted database.

18.6.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file

If an error occurs on both versions of the current file, information needed to restart HiRDB (unit) is lost and HiRDB can no longer be restarted. In such a case, the `pdstart dbdestroy` command must be used to start HiRDB forcibly.

When HiRDB is restarted in this manner, it does not inherit information that was in effect during the previous session. Therefore, the HiRDB administrator must recover the database by executing the database recovery utility using a backup copy and the system log (unload log) as the input information. For details on the database recovery utility, see *19. Database Recovery Procedures*.

Before executing forced startup of HiRDB, see *1.6.2 Notes on forced startup of HiRDB (or a unit)*.

Note:

- The database should be placed in inaccessible status (by placing the corresponding RDAREAs in command shutdown and closed status with the `pdhold -c` command) until the database has been recovered.
- When HiRDB is started forcibly, all RDAREAs (including system RDAREAs) that were updated during the previous HiRDB session are destroyed. These RDAREAs must be recovered with the database recovery utility. If they are not recovered, subsequent HiRDB operations cannot be guaranteed.

18.7 Handling of synchronization point dump file errors

Table 18-14 shows the actions to be taken in the event of an error in a synchronization point dump file.

Table 18-14: Actions to be taken in the event of an error in a synchronization point dump file

Condition at the time of an error		HiRDB processing	HiRDB administrator's action
Write operation	There are files in overwrite enabled status	Places the erroneous synchronization point dump file in reserved status and resumes processing using one of the files in overwrite-enabled status.	Place the erroneous synchronization point dump file in overwrite enabled status with the procedure shown in (1).
	There are no files in overwrite enabled status	Terminates abnormally the unit containing the erroneous synchronization point dump file.	• Create a new synchronization point dump file and then restart the unit. • After restarting the unit, place the erroneous synchronization point dump file in overwrite enabled status with the procedure shown in (1).
Read operation	If HiRDB cannot read the file of the most recent generation, it attempts to read the file of the immediately preceding generation. If HiRDB cannot read that file either, it attempts to read the file immediately preceding that one. In this manner, HiRDB continues attempting to read a valid synchronization point dump file by tracking back through the file generations. However, if all of these attempts are unsuccessful (if the number of guaranteed valid generations is exceeded), it may mean that the system is not recoverable because the system logs needed to recover the system have been overwritten. ■ If dual synchronization point dump files are maintained If HiRDB cannot read file A, it attempts to read file B. If HiRDB cannot read file B, it attempts to read file A of the immediately preceding generation, and so on.		Place the erroneous synchronization point dump file in overwrite enabled status with the procedure shown in (1).

(1) Procedure for placing an erroneous file in overwrite enabled status

Procedure

To place an erroneous file in overwrite enabled status:

1. Use the pdlogls command to determine the synchronization point dump file that has been reserved due to an error:

   ```
   pdlogls -d spd -s b001
   ```

2. If the erroneous file is not reserved, use the pdlogcls command to place it in reserved status:

   ```
   pdlogcls -d spd -s b001 -g spdfile1
   ```

3. Use the pdlogrm command to delete the reserved synchronization point dump file:

   ```
   pdlogrm -d spd -s b001 -f /sysfile/sync01
   ```

4. Use the pdloginit command to re-create the synchronization point dump file that was deleted in step 3:

   ```
   pdloginit -d spd -s b001 -f /sysfile/sync01 -n 5000
   ```

5. Use the pdlogopen command to place the synchronization point dump file re-created in 4 in overwrite enabled status:

   ```
   pdlogopen -d spd -s b001 -g spdfile1
   ```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(2) Actions to be taken when HiRDB cannot be restarted because the system log file corresponding to a synchronization point dump file has been overwritten

If an error occurs in a synchronization point dump file, HiRDB attempts system recovery using the preceding generation of the synchronization point dump file. If data is written over the system log file corresponding to the synchronization point dump file (the system log file containing the information needed for system recovery), HiRDB cannot be restarted. In such a case, the pdstart dbdestroy command must be used to start HiRDB forcibly.

In such a case, HiRDB does not inherit information that was in effect during the previous HiRDB session. Therefore, the HiRDB administrator must recover the database. To recover the database, the database recovery utility is executed using the backup copy and system log (unload log) as the input information. For details on the database recovery procedure, see *19. Database Recovery Procedures*.

Before executing forced startup of HiRDB, see *1.6.2 Notes on forced startup of HiRDB (or a unit)*.

Note:

- Until the database has been recovered, the database must be kept inaccessible (place its RDAREAs in command shutdown and closed status with the `pdhold -c` command).
- When HiRDB is started forcibly, all RDAREAs that were updated during the previous HiRDB session (including system RDAREAs) are destroyed. The destroyed RDAREAs must be recovered by the database recovery utility; otherwise, subsequent HiRDB operations cannot be guaranteed.
- RDAREAs are recovered from the system log only. See the `KFPS01262-I` message that was output when the previous `pdstart` command failed, and use as the input information to the database recovery utility the file group whose name is displayed as the log reading start file group and the subsequent system log.

(3) Actions to be taken when HiRDB cannot be restarted due to an insufficient number of synchronization point dump files

If the number of synchronization point dump files is less than the number of guaranteed valid generations, HiRDB cannot be restarted. In such a case, take one of the following actions and then restart HiRDB:

- Re-create the synchronization point dump file resulting in an error.
- Specify `ONL` in the `pdlogadfg -d spd` operand for all synchronization point dump files (if not specified already).

The length of time HiRDB is shut down can be reduced by specifying the following operands:

- `pd_spd_reduce_mode=1` or `pd_spd_reduce_mode=2`
- `pd_spd_reserved_file_auto_open=Y`

18.8 Handling of status file errors

This section explains the actions to be taken when status file errors occur. It discusses the following topics:

- Actions to be taken in the event of an error in the current file
- Procedures for starting HiRDB (unit) while there is an erroneous status file
- Actions to be taken when HiRDB (unit) cannot be restarted due to an error in both versions of the current file

18.8.1 Actions to be taken in the event of an error in the current file

Table 18-15 shows the actions to be taken in the event of an error in the current file.

Table 18-15: Actions to be taken in the event of an error in the current file

Item	Condition at the time of an error		HiRDB processing	HiRDB administrator's action
1	There are spare files.		Swaps the status files. Also issues the `KFPS01062-I` is output. Shuts down the erroneous file and resumes processing using one of the spare files as the current file.	Place the shutdown status file in spare status with the procedure shown in (1).
2	There are no spare files.	Single operation of status file specified[1]	Resumes processing, placing the status file in single operation mode. Also issues the `KFPS01044-I` message.	Create a spare file immediately and return the status file operation mode to double operation. Use the procedure shown in (1) or (2) to create a spare file. Then use the procedure shown in (3) to set a spare file as the current file.
3		Single operation of status file not specified[2]	Abnormally terminates the unit containing the erroneous status file.	Create a new status file and then restart the unit. For details about restarting the unit, see *18.8.2 Procedure for starting a HiRDB (unit) while there is an erroneous status file*. Then place the shutdown status file in spare status with the procedure shown in (1).

Item	Condition at the time of an error	HiRDB processing	HiRDB administrator's action
4	Error during single operation or error in both file versions during double operation	Abnormally terminates the unit containing the erroneous status file.	The unit cannot be restarted because needed information has been lost. See *18.8.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file*. If there is no spare status file, create a new status file before starting the unit forcibly.

[1] Applicable when `pd_syssts_singleoperation=continue` was specified in the unit control information definition or `pd_sts_singleoperation=continue` was specified in the server definition.

[2] Applicable when `pd_syssts_singleoperation=stop` (default value) was specified in the unit control information definition or `pd_sts_singleoperation=stop` (default value) was specified in the server definition.

(1) Placing shutdown files in spare status

Procedure

1. Use the `pdls` command to check for status files that have been shut down.

   For unit status files:

   ```
   pdls -d sts -u UNT1
   ```

   For server status files:

   ```
   pdls -d sts -s b001
   ```

2. Check if an error has occurred at the disk storing status files that have been shut down. If an error has occurred, correct the error. Also check for a physical error (such as physical damage or a power outage), and for an OS or disk driver error; also check that the disk is enabled.

3. Use the `pdstsrm` command to delete the status files in shutdown status.

   For unit status files:

   ```
   pdstsrm -u UNT1 -f /sysfile/usts1a
   pdstsrm -u UNT1 -f /sysfile/usts1b
   ```

   For server status files:

   ```
   pdstsrm -s b001 -f /sysfile/sstsb1a
   pdstsrm -s b001 -f /sysfile/sstsb1b
   ```

4. Use the pdstsinit command to re-create the status files that were deleted in step 3.

   For unit status files:

   ```
   pdstsinit -u UNT1 -f /sysfile/usts1a -l 4096 -c 256
   pdstsinit -u UNT1 -f /sysfile/usts1b -l 4096 -c 256
   ```

   For server status files:

   ```
   pdstsinit -s b001 -f /sysfile/sstsb1a -l 4096 -c 256
   pdstsinit -s b001 -f /sysfile/sstsb1b -l 4096 -c 256
   ```

5. Use the pdstsopen command to place in spare status the status files that were re-created in step 4.

   For unit status files:

   ```
   pdstsopen -u UNT1 -f /sysfile/usts1a
   pdstsopen -u UNT1 -f /sysfile/usts1b
   ```

   For server status files:

   ```
   pdstsopen -s b001 -f /sysfile/sstsb1a
   pdstsopen -s b001 -f /sysfile/sstsb1b
   ```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(2) Placing reserved files in spare status

Procedure

1. Use the pdls command to check for reserved status files.

   For unit status files:

   ```
   pdls -d sts -u UNT1
   ```

   For server status files:

   ```
   pdls -d sts -s b001
   ```

2. Use the pdstsopen command to place the reserved status files in spare status.

   For unit status files:

   ```
   pdstsopen -u UNT1 -n usts1a
   ```

   For server status files:

```
pdstsopen -s b001 -n sstsb01
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(3) Setting a reserved file as the current file

Procedure

1. Use the `pdstsswap` command to set a reserved file as the current file.

   For a unit status file:

   ```
   pdstsswap -u UNT1
   ```

   For a server status file:

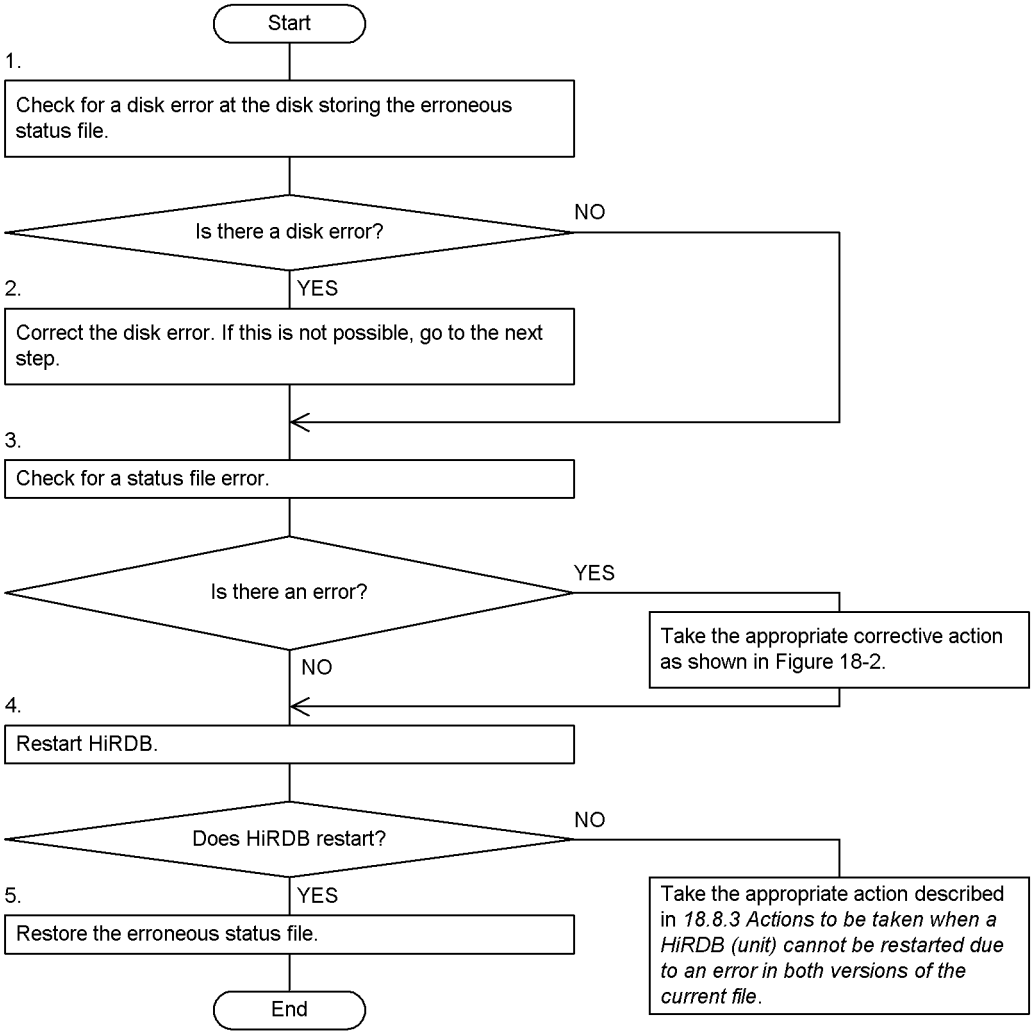
   ```
   pdstsswap -s b001
   ```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

18.8.2 Procedure for starting a HiRDB (unit) while there is an erroneous status file

Figure 18-1 shows the procedure for starting a HiRDB (unit) while there is an erroneous status file.

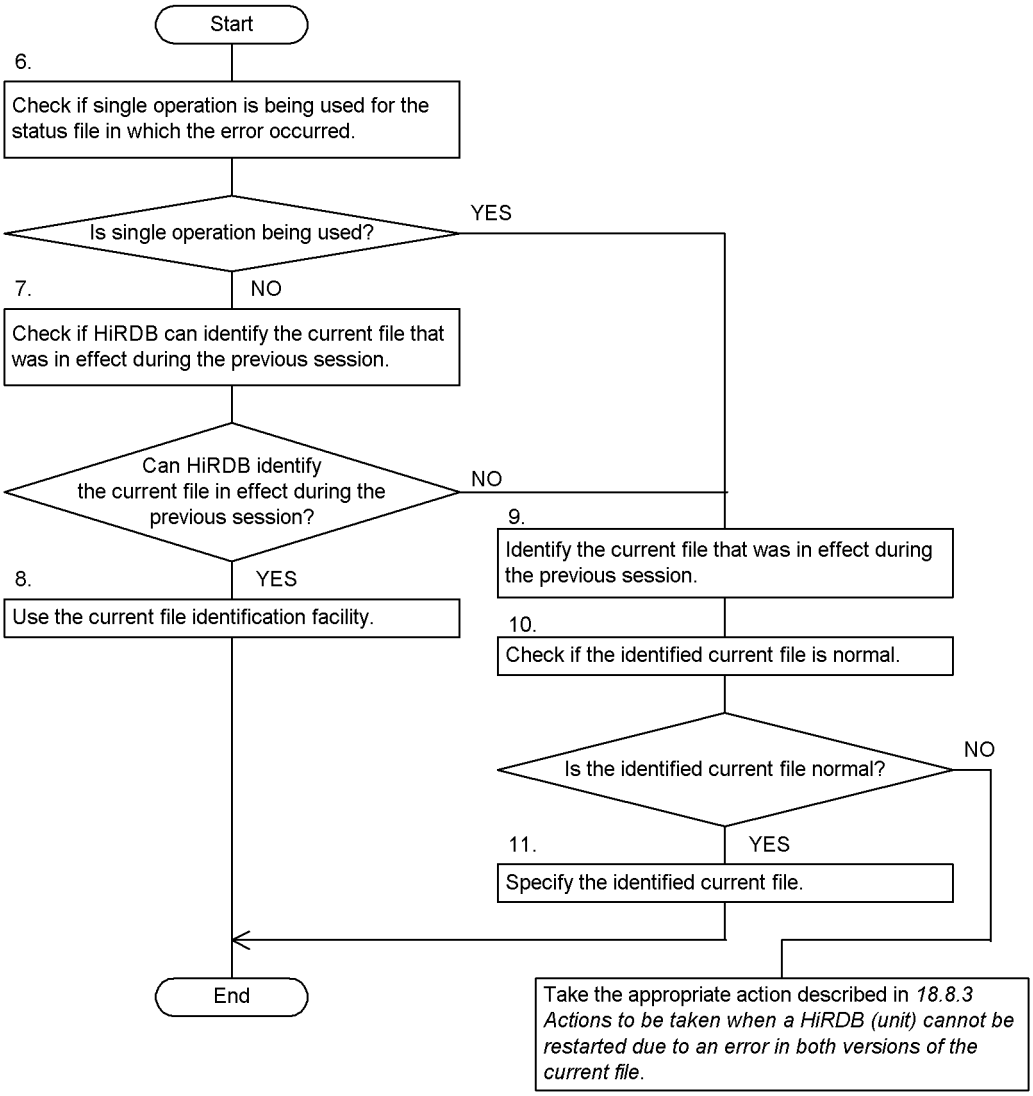
Figure 18-1: Procedure for starting a HiRDB (unit) while there is an erroneous status file



Note

The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 5 is explained in paragraph (5) below.

Figure 18-2: Actions to be taken when an error occurs in a status file



Note

The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 8 is explained in paragraph (8) below.

(1) Checking for a disk error

Check if a disk error has occurred at the disk that stores the status file in which the error

occurred. Check for a physical error (such as physical damage or a power outage), as well as for an OS or disk driver error; also check that the disk is enabled.

Table 18-16 shows how to determine if a physical error has occurred at the disk.

Table 18-16: Determining if a physical error has occurred at the disk (physical error check)

Has a disk error occurred?	Is the physical error recoverable?	Status file data	Determination result
No	—	—	No physical error
Yes	Yes	Data remains.	
		Data has been lost.	Physical error occurred (no entity)
	No	—	

Legend:

—: Not applicable

Note:

Regardless of whether or not a disk error has occurred, unless otherwise indicated, do not use the `pdstsinit`, `pdstsrm`, or `pdfmkfs` command until error recovery is completed.

(2) Correcting a disk error

If the check identifies a disk error, correct it. If it is not possible to correct the error, start HiRDB with the remaining normal disks only.

(3) Checking for a status file error

Check for an error in a status file. Table 18-17 shows how to determine if a logical error has occurred.

Table 18-17: Determining if a logical error has occurred (logical error check)

Command execution	Results displayed by the command (compared to the values specified at the time of file creation)	Determination result
Terminated normally	No inconsistency	No logical error
	Inconsistency found	Logical error occurred
Terminated abnormally (error message is output)	—	Logical error occurred

Legend:

—: Not applicable

Execute the pdcat command for a status file in which no physical error has occurred and check for an error in its contents. The status file is normal if both the following conditions are satisfied:

- The record size displayed in the execution results of the pdcat command matches the record size specified when the status file was created.
- No error message was output during execution of the pdcat command.

An example of execution of the pdcat command follows:

```
pdcat -d sts -u UNT1 -f /sysfile/usts1a -v        ...1
pdcat -d sts -s b001 -f /sysfile/sstsb1a -v       ...2
```

Explanation

1. Command execution example for a unit status file.
2. Command execution example for a server status file.

If neither a physical error nor a logical error has occurred, proceed to the next step. If a physical error or a logical error has occurred, take the appropriate actions described in Figure *18-2 Actions to be taken when an error occurs in a status file*.

(4) Restarting HiRDB

Use the pdstart command to restart HiRDB. If HiRDB cannot be restarted, take the appropriate actions described in *18.8.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file*.

(5) Restoring the status file in which the error occurred

If an error has occurred in one of the current files, the HiRDB administrator must immediately take the appropriate action described in Table *18-15 Actions to be taken in the event of an error in the current file*.

If a file has been shut down by an error, place the shutdown file in spare status by following the procedure described in *18.8.1(1) Placing shutdown files in spare status*.

After you have restored all status files, terminate HiRDB if necessary, restore the specification values of the following operands to their original values, then start HiRDB:

- pd_syssts_initial_error (for a unit status file)
- pd_syssts_singleoperation (for a unit status file)
- pd_sts_initial_error (for a server status file)

- pd_sts_singleoperation (for a server status file)

(6) Check if single operation is being used

Check if single operation is being used for the status file in which the error occurred. If the following operand is specified, as applicable, for the status file in which the error occurred, single operation is being used:

- For a unit status file: pd_syssts_singleoperation=continue
- For a server status file: pd_sts_singleoperation=continue

(7) Checking if HiRDB can identify the current file that was in effect during the previous session

Based on the results of steps (1) through (3), for each logical file of the status file in which the error occurred, determine whether file versions A and B were in the statuses shown in Table 18-18. If the status file was in any of the statuses shown in Table 18-18, HiRDB cannot identify the current file that was in effect during the previous session.

Table 18-18: Cases in which HiRDB cannot identify the current file that was in effect during the previous session

Status of file version A	Status of file version B
Logical error present	Logical error present
Logical error present	Physical error present (no entity)
Physical error present (no entity)	Logical error present
Physical error present (no entity)	Physical error present (no entity)

(8) Using the current file identification facility

Use HiRDB's current file identification facility. Specify the following operand, as applicable, for the status file:

- For a unit status file: pd_syssts_initial_error=excontinue
- For a server status file: pd_sts_initial_error=excontinue

(9) Identifying the current file that was in effect during the previous session

Identify the current (most recent) file that was in effect during the previous session. You can identify this file from the messages listed below. Retrieve these messages from the message log file or syslogfile (retrieve the messages for the unit or server for which the current file cannot be identified).

- KFPS01001-I (message output when the current file is allocated)
- KFPS01044-I (message output when single operation goes into effect)

- KFPS01063-I (message output when the current file is swapped by status file swapping)

Check these messages for the one output most recently; the current file is indicated in that message.

(10) Checking if the identified current file is normal

Check that the current file identified in step (9) is normal. You can determine this from the results of steps (1) through (3).

If status file single operation was in effect during the previous session (the last message that was output in step (9) was KFPS01044-I), check whether the status file for the active file system shown in the KFPS01044-I message is normal.

If status file single operation was not in effect during the previous session (the last message that was output in step (9) was KFPS01001-I or KFPS01063-I), check whether either of the status files shown in the KFPS01001-I or KFPS01063-I message is normal.

If the current file is normal (or if one of the files is normal if status file single operation was not in effect), proceed to the next step.

If an error had occurred in the current file, the current file that was in effect during the previous session has been lost, which means that HiRDB cannot be restarted. In such a case, take the actions described in *18.8.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file*.

(11) Specifying the identified current file

Specify the identified current file that was in effect during the previous session in the following operands:

■ When the error occurred in a unit status file

Specify the following operands for the applicable unit:

- pd_syssts_initial_error=continue or excontinue
- pd_syssts_last_active_file=*name-of-current-status-file-that-was-in-effect-during-the-previous-session**
- pd_syssts_last_active_side=*file-system-that-was-normal-during-the-previous-session**

■ When the error occurred in a server status file

Specify the following operands for the applicable server:

- pd_syssts_initial_error=continue or excontinue
- pd_syssts_last_active_file=*name-of-current-status-file-that-was-in-e*

*ffect-during-the-previous-session**

- `pd_syssts_last_active_side=`*file-system-that-was-normal-during-the-previous-session**

* Specify the current file name and normal file system identified in steps (9) and (10).

18.8.3 Actions to be taken when a HiRDB (unit) cannot be restarted due to an error in both versions of the current file

If an error occurs in both versions of the current status file, HiRDB (unit) cannot be restarted. In this case, use the `pdstart dbdestroy` command to start HiRDB forcibly.

Note:

When HiRDB is started forcibly, it does not inherit information that was in effect during the previous HiRDB session. Therefore, the HiRDB administrator must recover the database. The database is recovered by executing the database recovery utility (`pdrstr` command) using a backup copy and the system log (unload log) as the input information.

The following is the procedure for starting HiRDB in the event of an error on both versions of the current file:

Procedure

1. Check for a disk error (physical error check).

   Check if a disk error has occurred at the disk that stores the status file. Check for a physical error (such as physical damage or a power outage) as well as for an OS or disk driver error; also check that the disk is enabled.

   If the check identifies a disk error, correct it. If it is not possible to correct the error, start HiRDB with the remaining normal disks only. Proceed directly to the next step.

   To determine if a physical error has occurred, refer to Table *18-16 Determining if a physical error has occurred at the disk (physical error check)*.

2. Check for a status file error (logical error check).

   Execute the `pdcat -d sts` command for a status file in which no physical error has occurred and check for an error in its contents.

   For a unit status file:

   ```
   pdcat -d sts -u UNT1 -f /sysfile/usts1a -v
   ```

   For a server status file:

```
pdcat -d sts -s b001 -f /sysfile/sstsb1a -v
```

The status file is normal if both the following conditions are satisfied:

- The record size displayed in the execution results of the `pdcat` command matches the record size specified when the status file was created.
- No error message was output during execution of the `pdcat` command.

3. Use the `pdstsrm` command to delete the erroneous status files.

   For unit status files:

   ```
   pdstsrm -u UNT1 -f /sysfile/usts1a
   pdstsrm -u UNT1 -f /sysfile/usts1b
   ```

   For server status files:

   ```
   pdstsrm -s b001 -f /sysfile/sstsb1a
   pdstsrm -s b001 -f /sysfile/sstsb1b
   ```

4. Use the `pdstsinit` command to re-create the status files deleted in step 3.

   For unit status files:

   ```
   pdstsinit -u UNT1 -f /sysfile/usts1a -l 4096 -c 256
   pdstsinit -u UNT1 -f /sysfile/usts1b -l 4096 -c 256
   ```

   For server status files:

   ```
   pdstsinit -s b001 -f /sysfile/sstsb1a -l 4096 -c 256
   pdstsinit -s b001 -f /sysfile/sstsb1b -l 4096 -c 256
   ```

5. Use the `pdstart dbdestroy` command to start HiRDB forcibly.
6. Use the database recovery utility (`pdrstr` command) to recover the RDAREAs.

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

Note:

Starting HiRDB forcibly destroys all RDAREAs that were updated during the previous HiRDB session (including system RDAREAs). The destroyed RDAREAs must be recovered by the database recovery utility; if you fail to do this, subsequent HiRDB operations cannot be guaranteed.

18.9 Handling of errors in files other than system files

This section explains the actions to be taken in the event of a file error, such as an I/O error, in the following types of files:

- HiRDB system definitions file
- Message log file
- Statistics log file
- Data linkage file (HiRDB Datareplicator)

18.9.1 Errors in the HiRDB system definitions file

HiRDB processing

HiRDB does not start.

HiRDB administrator's action

Create a new HiRDB system definition file, specify the HiRDB system definitions, then start HiRDB normally.

18.9.2 Errors in the message log file

HiRDB processing

HiRDB messages are also output to syslogfile. However, some messages may be lost during periods of heavy traffic, because messages from other programs are also output to this log.

HiRDB administrator's action

Take the following action:

Procedure

To correct the message log file error:

1. Enter the `pdstop` command to terminate HiRDB.
2. Eliminate the cause of the message log file error.
3. Enter the `pdstart` command to start HiRDB.

18.9.3 Errors in the statistics log file

HiRDB processing

Resumes processing without collecting the statistics log. The `KFPS05360-E` message is output.

HiRDB administrator's action

Take the following action:

Procedure

To correct the statistics log file error:

1. Eliminate the cause of the error in the statistics log file.
2. Use the `pdstjswap` command to swap the output destination for the statistics log file.
3. Use the `pdstbegin` command to resume collection of the statistics log file.
4. Use the `pdls -d stj` command to confirm that the statistics log file has been collected.

18.9.4 Errors in the data linkage file (HiRDB Datareplicator)

HiRDB processing

When HiRDB Datareplicator linkage is being used, data linkage is disabled if the data linkage file that is being used to report the extraction status is initialized or if an open error or input/output error occurs in this file. When HiRDB detects an error in the data linkage file, it cancels HiRDB Datareplicator linkage and resumes operation using HiRDB only.

HiRDB administrator's action

Take the following action:

Procedure

To correct the data linkage file error:

1. Enter the `pdls -d rpl` command to determine whether or not the HiRDB unit is executing HiRDB Datareplicator linkage.
2. If the unit is executing HiRDB Datareplicator linkage, enter the `pdrplstop` command to terminate HiRDB Datareplicator linkage.
3. Correct the data linkage file.
4. To restart HiRDB Datareplicator linkage, enter the `pdrplstart` command.*

* When HiRDB Datareplicator is stopped in this manner, mismatches may arise between the extracted database and the target database being processed over the data linkage. Therefore, before entering the `pdrplstart` command, you must re-create the target database based on the extracted database. For details about how to re-create a target database, see the manual *HiRDB Datareplicator Version 8 Description, User's Guide and Operator's Guide*.

18.10 When the OS terminates abnormally

This section describes the HiRDB processing and the HiRDB administrator's action in the event of abnormal termination of the OS.

(1) HiRDB processing

If the OS terminates abnormally due to an error, HiRDB causes the unit located at the server machine where the OS terminated abnormally to also terminate abnormally.

(2) HiRDB administrator's action

After rebooting the OS, do the following:

Procedure

To restart HiRDB:

1. Use the `pdgeter` command to back up the troubleshooting information that is output under `$PDDIR/spool` and `$PDDIR/tmp`.

   For details on the `pdgeter` command execution procedure, see *18.1.1 Actions to be taken by the HiRDB administrator when an error occurs*.

   For details on the troubleshooting information that is output in the event of an error, see *18.1.2 Information collected by HiRDB when an error occurs*.

2. Enter the `pdstart` command to restart HiRDB. Depending on the `pd_mode_conf` operand specification, the unit is restarted automatically after the OS is rebooted.

18.11 Handling of errors while linked to an OLTP system

This section explains the actions to be taken if an error occurs while HiRDB is linked to an OLTP system. It covers the following topics:

- Actions to be taken when a communication error occurs while HiRDB is linked to an OLTP system
- Actions to be taken when a transaction is placed in FORGETTING status due to an error
- Actions to take when transactions remain due to inactivity of a unit with a front-end server

Note that the following discussion applies only to the X/Open XA Interface.

18.11.1 Actions to be taken when a communication error occurs while HiRDB is linked to an OLTP system

If a communication error occurs while HiRDB is linked to an OLTP system, the transaction branch can no longer be committed or rolled back. This section describes the HiRDB processing and the actions to be taken by the HiRDB administrator at this time.

(1) HiRDB processing

HiRDB retains the status of the transaction branch and resumes processing.

(2) HiRDB administrator's action

1. The OLTP system may not be active; check whether or not OLTP is active (if it is not, start it).
2. If error recovery is not possible, use the pdcmt or pdrbk command to prepare for transaction determination; for details, see *(3) Recovering a transaction that has reached secure status after an error occurrence*.

(3) Recovering a transaction that has reached secure status after an error occurrence

When the KFPS00992-E message (transaction cannot be recovered) is output, the pdls -d trn command can be used to display the status of transactions. This section explains how to recover a transaction that has reached the secure status (commit second phase wait status) in which transaction status 1 is READY and transaction status 2 is p.

(a) HiRDB/Single Server

A HiRDB/Single Server reaches secure status only when it is waiting for a commit second phase instruction from the OLTP system. The OLTP system may have

terminated abnormally for some reason or the OLTP system cannot communicate with HiRDB. Therefore, take one of the following actions.

1. When the OLTP system has abnormally terminated

   Restart the OLTP system. Once the OLTP system has restarted, the transaction is recovered automatically by establishing synchronization.

2. In any other case

   Check the OLTP system's status, the status of the network between the OLTP system and HiRDB, etc. Once communication between the OLTP system and HiRDB is reestablished, the transaction is recovered by establishing synchronization with the OLTP system.

3. Use of commands for independent determination

   If it is difficult to restart the OLTP system or to restore the network, it is possible to determine the transaction independently without establishing synchronization with the OLTP system. The commands used for this purpose are `pdcmt`, `pdrbk`, and `pdfgt`. For details, see *18.14 Actions when there is an undetermined transaction*.

   When the transaction is recovered using this method, other resource managers and transaction branches that were being processed by the OLTP system may not be synchronized; check their data contents.

(b) HiRDB/Parallel Server

In the case of a HiRDB/Parallel Server, the actions to be taken depend on the server that has reached secure status.

Front-end server

As is the case above in *1. When the OLTP system has abnormally terminated* of *(a) HiRDB/Single Server*, HiRDB must wait for a commit second phase indication from the OLTP system. Therefore, take the same action as described above in case 1 *(a) HiRDB/Single Server*.

Back-end server and dictionary server

For details about how transactions are determined, see *18.14 Actions when there is an undetermined transaction*.

Note that if `pd_trn_rerun_branch_auto_decide = Y` is specified (`Y` is the default), uncompleted transactions are determined automatically.

18.11.2 Actions to be taken when a transaction is placed in FORGETTING status due to an error

(1) Overview of transactions in FORGETTING status

A transaction that is in FORGETTING status, as explained here, is one that satisfies all the following conditions:

- The transaction's first status listing under STATUS in the execution results of the pdls -d trn command is FORGETTING (during transaction termination processing).
- The transaction's third status listing under STATUS in the execution results of the pdls -d trn command is w (the transaction is being synchronized between the transaction manager and HiRDB).

The following is an example of a FORGETTING transaction:

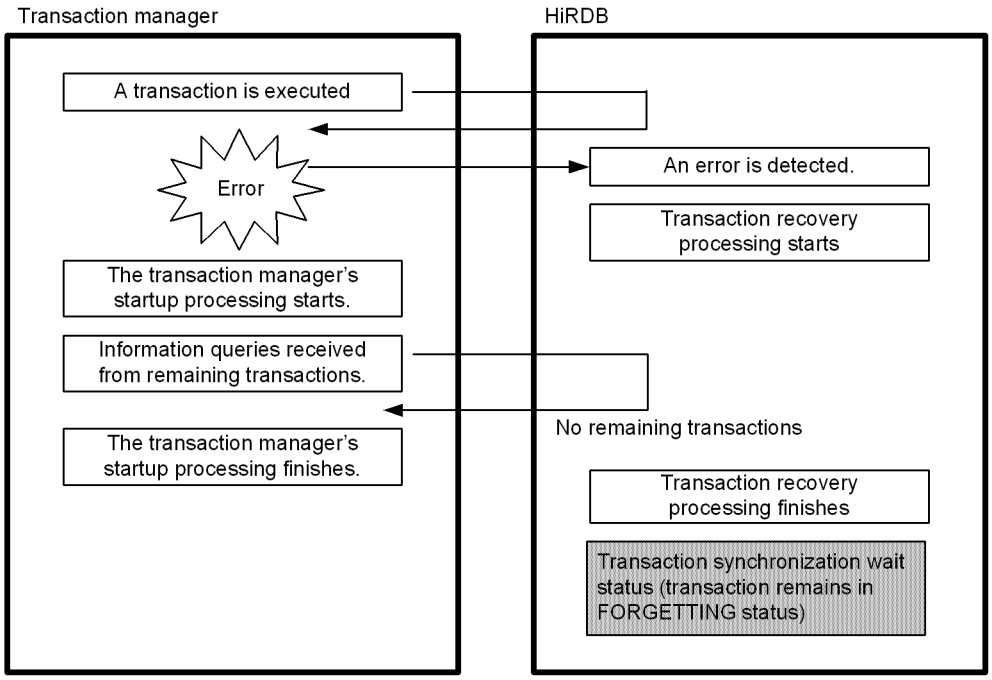
```
pdls -d trn

HOSTNAME : host1(153201)
TRNGID           TRNBID           STATUS           PID   SVID   B-SVID
HRD1unt100020b51 HRD1unt100020038 FORGETTING(r,w)    0   bes1   fes1
```

(2) When a transaction in FORGETTING status occurs

When an error that satisfies one of the following conditions occurs, the transaction is in FORGETTING status on HiRDB:

- The transaction manager is terminating abnormally.
- The first SQL (FirstSQL) to access HiRDB after a transaction being managed by the transaction manager starts ends in an error.
- The transaction manager is terminated abnormally, and restarting of the transaction manager completes before recovery of the HiRDB transaction completes (Figure 18-3 shows an example of a transaction in FORGETTING status).

Figure 18-3: Example of a transaction in FORGETTING status after restarting the transaction manager is completed



Because a transaction in `FORGETTING` status uses HiRDB memory resources that are allocated for transactions, an error may occur if too many transactions are placed in `FORGETTING` status. Such an error may prevent HiRDB from being able to perform concurrent execution of a large number of transactions from connected users.

(3) Actions to take

If a transaction in `FORGETTING` status occurs, take one of the following actions:

- Restart the transaction manager.
- Execute the `pdfgt` command to forcibly terminate the transaction in `FORGETTING` status.

18.11.3 Actions to be taken when transactions remain resident due to inactivity of a unit with a front-end server

(1) When transactions remain resident

When all the following conditions are present, transactions may remain resident on the active unit:

- A HiRDB/Parallel Server consists of multiple units.
- A unit with a front-end server is inactive.
- There are uncompleted transactions in the active units.
- The transaction manager is started after it has terminated abnormally.

Because transactions that remain resident continue to use resources, if too many transactions remain resident, HiRDB may no longer be able to perform concurrent execution of a large number of transactions from connected users.

(2) Actions to take

Use the `pdstart -u` command to start the unit with the front-end server. If you are unable to start the unit with the front-end server, check the transaction status of the transaction manager's transactions, and use the `pdcmt`, `pdrbk`, or `pdfgt` command to forcibly complete the HiRDB transactions. If you use this method to forcibly complete the transactions, the transaction completion methods recognized in the transaction manager and HiRDB may no longer match.

18.12 Handling of communication errors, CPU errors, and power failures

This section describes the handling of the following types of errors:

- Communication errors
- CPU errors
- Power failure

18.12.1 Handling of communication errors

HiRDB processing

Returns control, issuing a message indicating that a communication error occurred in a specified transaction.

HiRDB administrator's action

Check the cause of the communication error, terminate HiRDB, and take appropriate action to correct the error.

It may not be possible to terminate HiRDB normally after a communication error. If this is the case, terminate HiRDB forcibly. If HiRDB still will not terminate, recover each unit if units can be terminated or started individually. Otherwise, shut down the entire system, eliminate the cause of the error, then restart the system.

18.12.2 Handling of CPU errors

HiRDB processing

When a CPU error occurs, the unit at the server machine resulting in the CPU error terminates abnormally.

HiRDB administrator's action

After you have restarted the OS, restart HiRDB. Depending on the specification in the `pd_mode_conf` operand, HiRDB may restart the unit automatically after you restart the OS.

18.12.3 Handling of a power failure

HiRDB processing

When the power has been restored, the OS has been rebooted, and HiRDB has been restarted, the hardware power backup facility restores the system to its status immediately before the power failure occurred.

HiRDB administrator's action

After you have restarted the OS, restart HiRDB. Depending on the specification in the `pd_mode_conf` operand, HiRDB may restart the unit automatically after you restart the OS.

18.13 When HiRDB cannot be terminated because a user is still connected

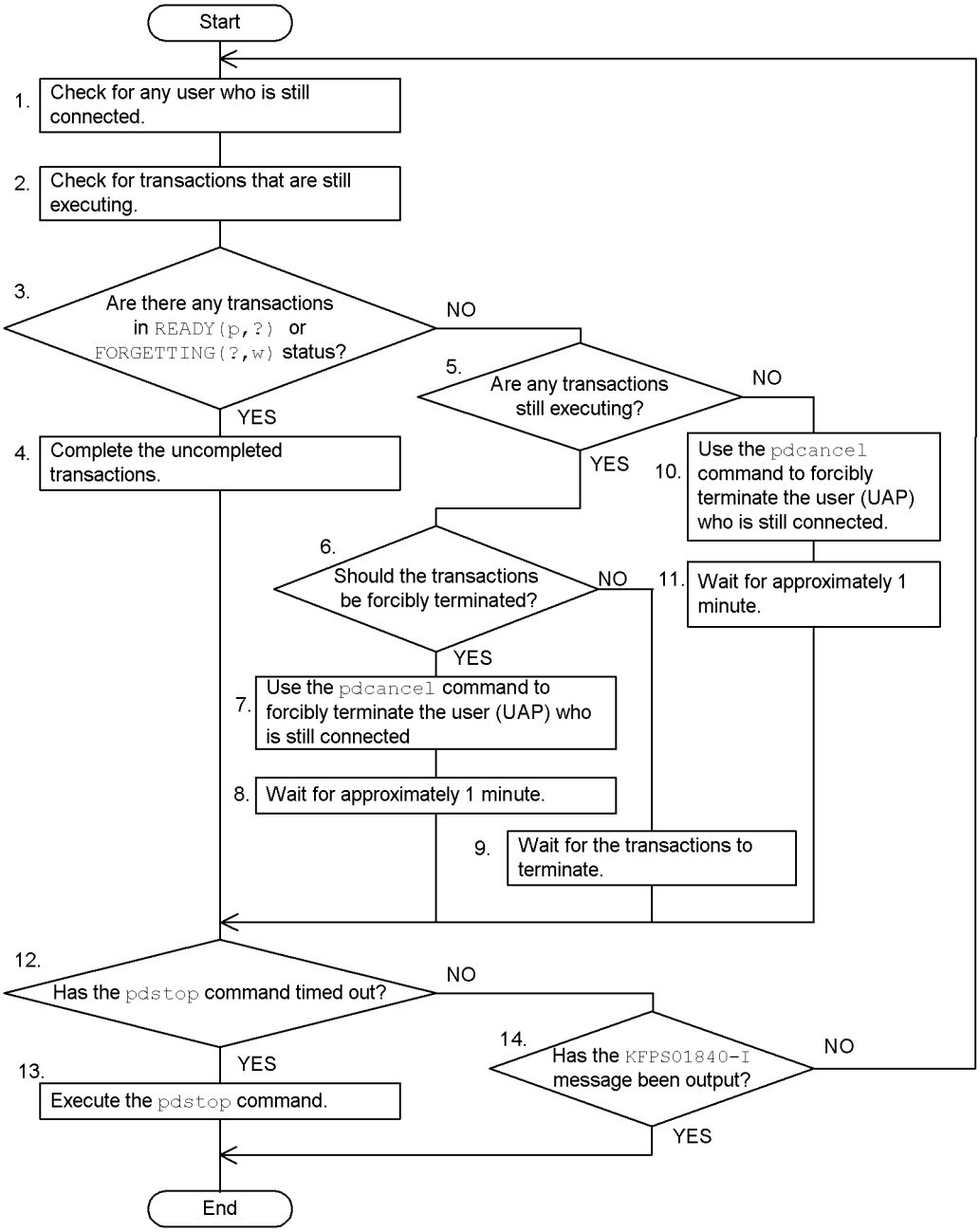
Executor: HiRDB administrator

This section explains the actions to be taken when HiRDB cannot be terminated because a user is still connected.

18.13.1 Corrective procedure

HiRDB cannot terminate if there is a UAP or utility that has not terminated (i.e., if a user is still connected). If an attempt is made to terminate HiRDB while a user is still connected, HiRDB outputs the `KFPS05120-W` message, together with the *connected user data file* and *connected user details file*. In this case, the HiRDB administrator must follow the procedure described below to disconnect the user and terminate HiRDB.

Procedure


Start
1. Check for any user who is still connected.
2. Check for transactions that are still executing.
3. Are there any transactions in READY(p,?) or FORGETTING(?,w) status?
NO
YES
4. Complete the uncompleted transactions.
5. Are any transactions still executing?
NO
YES
10. Use the pdcancel command to forcibly terminate the user (UAP) who is still connected.
11. Wait for approximately 1 minute.
6. Should the transactions be forcibly terminated?
NO
YES
7. Use the pdcancel command to forcibly terminate the user (UAP) who is still connected
8. Wait for approximately 1 minute.
9. Wait for the transactions to terminate.
12. Has the pdstop command timed out?
NO
YES
13. Execute the pdstop command.
14. Has the KFPS01840-I message been output?
NO
YES
End

Note

The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 5 is explained in paragraph (5) below.

(1) Check for any user who is still connected

(a) Using the pdls -d prc command to check if users are still connected

```
pdls -d prc

HOSTNAME : k95x620(173420)
STATUS   PID   UID   GID   SVID   TIME      PROGRAM    C-PID   C-GRP

  L      22118 334   300   fes1   173330    uap00      22205   PC
```

Explanation

Check the UAP identifier listed under `PROGRAM`. The user who is still connected is `uap00`.

(b) Checking the connected user data file for users who are still connected

For details about the output contents of the connected user data file, see *18.13.2 Connected user data file and connected user details file*.

```
UNIT ID : UNT1(173420)
 UID  PID    GID  SVID  TIME    PROGRAM  C-GRP  C-PID   C-IP
 334  22118  300  fes1  173330  uap00    PC     22205   172.17.32.37
```

Explanation

Check the UAP identifier listed under `PROGRAM`. The user who is still connected is `uap00`.

(2) Check for transactions that are still executing

Either execute the `pdls -d trn` command or use the results in the connected user details file of executing the `pdls -d trn` command to check for transactions that are still executing.

For details about the output contents of the connected user details file, see *18.13.2 Connected user data file and connected user details file*.

(3) Are there any transactions in READY(p,?) or FORGETTING(?,w) status?

Based on the execution results of the `pdls -d trn` command or the information in the connected user details file, determine whether there are any transactions in

READY(p,?) or FORGETTING(?,w) status.

(4) Complete any uncompleted transactions

Complete the uncompleted transactions; for details, see *18.14 Actions when there is an undetermined transaction*.

(5) Are any transactions still executing?

Based on the execution results of the pdls -d trn command or the information in the connected user details file, determine whether there are any transactions in READY(p,?) or FORGETTING(?,w) status.

(6) Should the transactions be terminated forcibly?

If there are transactions in READY(p,?) or FORGETTING(?,w) status, decide whether to terminate them forcibly or to wait for them to complete their processing. If the processing time of a transaction that is executing is short, wait for it to terminate; if the processing time is long, terminate it forcibly.

(7) Use the pdcancel command to terminate forcibly the user (UAP) who is still connected

Use the pdcancel command to terminate the transaction forcibly.

HiRDB/Single Server

```
pdcancel -u UAP00 -i 22118
```

HiRDB/Parallel Server (execution example 1)

```
pdcancel -x k95x620 -u UAP00 -i 22118
```

HiRDB/Parallel Server (execution example 2)

```
pdcancel -X UNT1 -u UAP00 -i 22118
```

Explanation

-u: Specifies the identifier of the UAP.

-x: Specifies the name of the host where the front-end server to which the UAP is connected is located. The specified host name is listed under HOSTNAME in the execution results of the pdls -d prc command.

-X: Specifies the unit identifier. The specified unit identifier is listed under UNIT

ID in the connected user data file.

-i: Specifies the process ID.

(8) Wait for approximately 1 minute

After executing the pdcancel command, wait for the transaction to be completed by itself.

(9) Wait for the transaction to terminate

If you do not terminate the transaction forcibly, wait for it to terminate.

(10) Use the pdcancel command to disconnect the user (UAP) who is still connected

If there is no transaction for a user who is still connected, use the pdcancel command to disconnect the user.

HiRDB/Single Server

```
pdcancel -i 22118 -d
```

HiRDB/Parallel Server (execution example 1)

```
pdcancel -x k95x620 -i 22118 -d
```

HiRDB/Parallel Server (execution example 2)

```
pdcancel -X UNT1 -i 22118 -d
```

Explanation

-x: Specifies the name of the host where the front-end server to which the UAP is connected is located. The specified host name is listed under HOSTNAME in the execution results of the pdls -d prc command.

-X: Specifies the unit identifier. The specified unit identifier is listed under UNIT ID in the connected user data file.

-i: Specifies the process ID.

-d: Specifies that the HiRDB process is to be terminated forcibly. When the -d option is specified, the core file is output. If the core file is not needed, delete it with the pdcspool command.

(11) Wait for approximately 1 minute

After executing the pdcancel command, wait for the user to be disconnected and HiRDB termination processing to be resumed.

(12) Has the pdstop command timed out?

If the KFPS05047-E message is output, the pdstop command has timed out.

(13) Execute the pdstop command

If the pdstop command has timed out, execute it again.

(14) Is the KFPS01840-I message output?

If the KFPS01840-I message is output, HiRDB termination processing has started.

18.13.2 Connected user data file and connected user details file

If an attempt is made to terminate HiRDB while a user is still connected, HiRDB outputs the KFPS05120-W message, together with the connected user data file ($PDDIR/spool/cnctusrinf) and the connected user details file ($PDDIR/spool/cnctusrdtl). For a HiRDB/Parallel Server, these files are output to the server machine where the system manager is located. The HiRDB administrator can identify the connected users from the information in these files.

(1) Information output in the connected user data file

Output example

```
UNIT ID : M350(173420)
UID  PID    GID   SVID  TIME    PROGRAM   C-GRP  C-PID  C-IP
334  22118  300   fes1  173330  uap00     PC     22205  172.17.32.37

UNIT ID : M35b(173427)
UID  PID    GID   SVID  TIME    PROGRAM   C-GRP  C-PID  C-IP
334  17524  300   fes2  173343  uap01     PC     17619  172.17.32.39
334  17533  300   fes2  173333  uap02     PC     22200  172.17.32.37
```

Explanation

UNIT ID

Displays the connected users data file unit identifier and creation time (hour minute second).

UID

Displays the user ID of each user who is connected to HiRDB.

PID

Displays the process ID of each connected user's HiRDB server process.

GID

Displays the group ID of each user who is connected to HiRDB.

SVID

Displays the server ID of the server to which each connected user is connected.

Blanks may be displayed immediately after HiRDB server process activation.

TIME

Displays the time (hour minute second) at which HiRDB received each service request. 999999 is displayed for a server where no user is connected.

PROGRAM

Displays the value specified in the PDCLTAPNAME operand of the client environment definition.

If this operand is omitted, Unknown is displayed. For details on the PDCLTAPNAME operand, see the manual *HiRDB Version 8 UAP Development Guide*.

- ****** is displayed for a utility.
- For a UAP that accesses HiRDB using the distributed database facility, RDAUSER - *authorization-identifier* is displayed.
- ****** may be displayed immediately after HiRDB server process activation.

C-GRP

Displays the user type (client group type) of each user who is connected to HiRDB.

If the connection frame guarantee facility for a client group is used and a client group selected arbitrarily by the user is defined, the user-defined client group name is displayed.

Displayed character string	User type
XA	User who connected to HiRDB using the X/Open XA interface
DF	User who connected to HiRDB from a distributed client
PC	User who connected to HiRDB from a PC client
WS	User who connected to HiRDB from a WS client
MF	User who connected to HiRDB from a mainframe client (VOS3 client, etc.)

C-PID

Displays the process ID of the client that is connected to HiRDB.

`C-IP`

Displays the IP address of the client that is connected to HiRDB.

Note

`C-GRP`, `C-PID`, and `C-IP` are not displayed in the following cases:

- The UAP or utility was not executed by a client (it was executed by HiRDB).
- The version of the client to which the UAP is linked is older than HiRDB Version 4.0 04-00.

(2) Information output in the connected user details file

The execution results of the following commands are output to the connected user details file:

- `pdls -d act`
- `pdls -d prc`
- `pdls -d trn`

For details about the execution results of each command, see the manual *HiRDB Version 8 Command Reference*.

18.14 Actions when there is an undetermined transaction

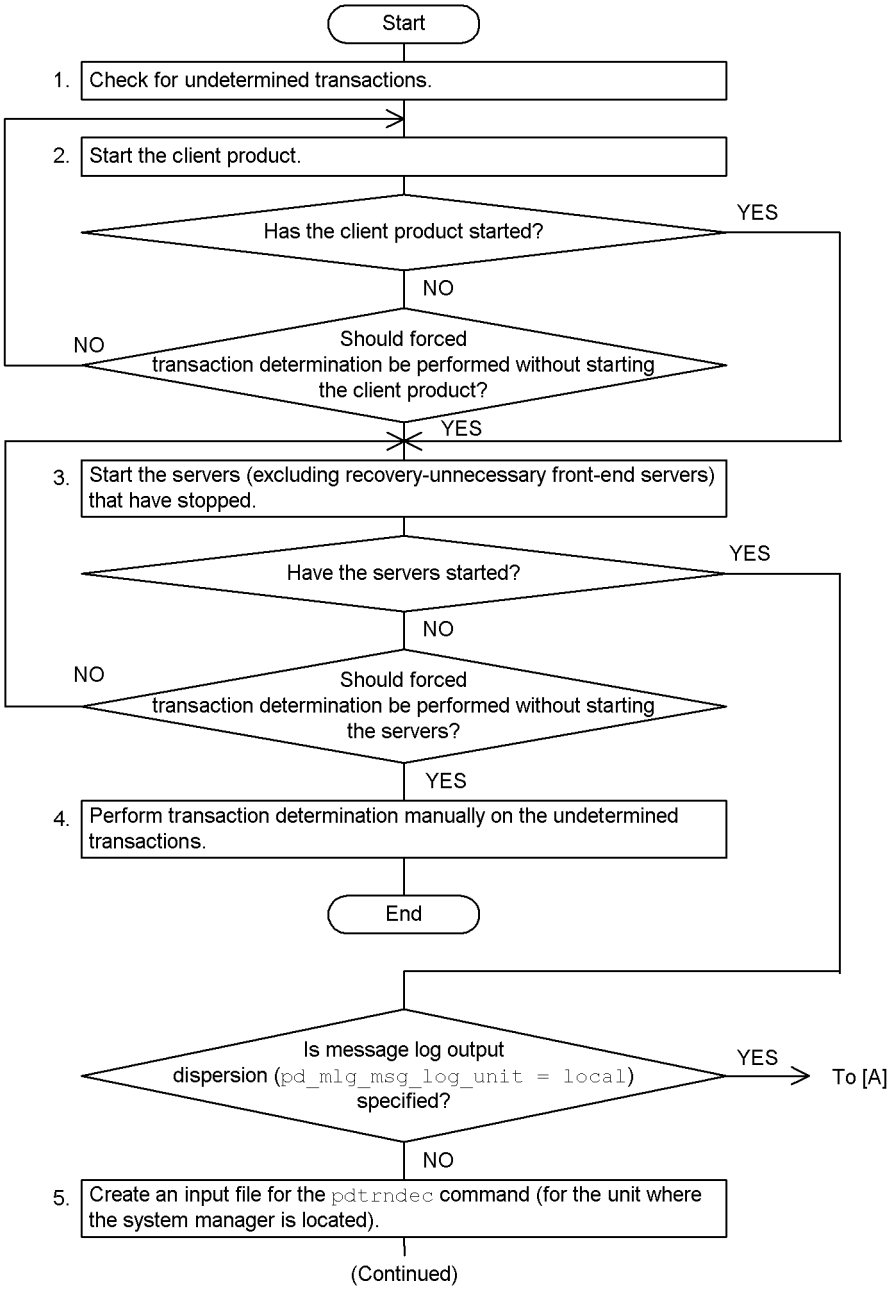
This section explains the actions to be taken when there are transactions that have not been determined (undetermined transactions).

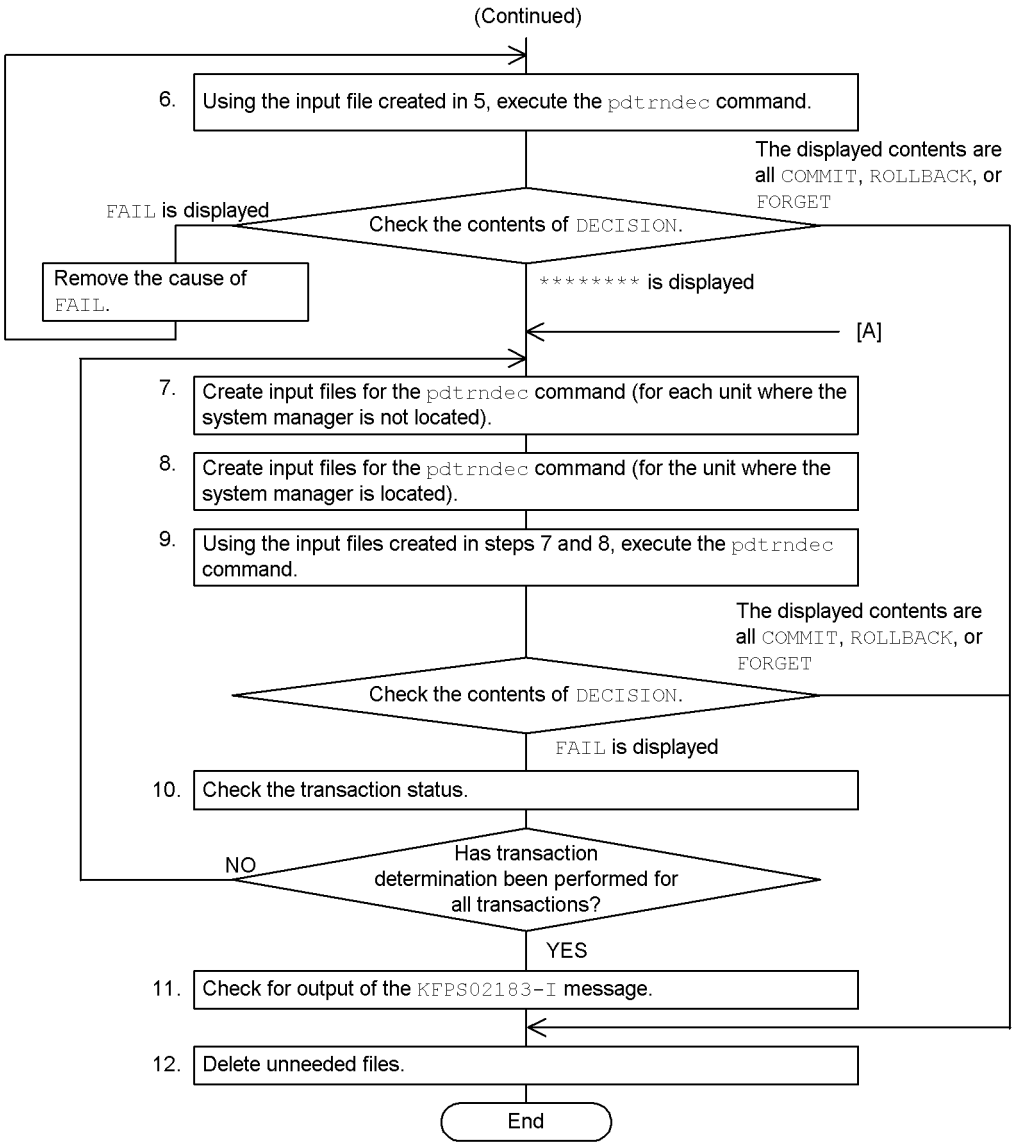
18.14.1 Forcing determination of uncompleted transactions

The KFPS00992-E message is output for each undetermined transaction (transaction branch that cannot be determined). When this happens, the HiRDB administrator must force determination of each such transaction that is uncompleted by following the procedure described below.

When pd_trn_rerun_branch_auto_decide = Y (default value) has been specified, determination of uncompleted transactions is performed automatically. If a transaction cannot be determined by this automatic determination facility or if you do not use this facility, you must perform the operation explained below.

Procedure


Start
1. Check for undetermined transactions.
2. Start the client product.
Has the client product started?
YES
NO
NO
Should forced transaction determination be performed without starting the client product?
YES
3. Start the servers (excluding recovery-unnecessary front-end servers) that have stopped.
Have the servers started?
YES
NO
NO
Should forced transaction determination be performed without starting the servers?
YES
4. Perform transaction determination manually on the undetermined transactions.
End
Is message log output dispersion (pd_mlg_msg_log_unit = local) specified?
YES
To [A]
NO
5. Create an input file for the pdtrndec command (for the unit where the system manager is located).
(Continued)



Notes

- The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 5 is explained in paragraph (5) below.
- Messages can be checked during the procedure explained below. To do so,

reference the messages in `syslogfile`.

(1) Check for undetermined transactions

Check the `KFPS00992-E` messages to determine if there are undetermined transactions.

```
KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b4d, TRNBID=HRD1unt100020034, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b22, TRNBID=HRD1unt100020035, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b5a, TRNBID=HRD1unt100020036, server=bes1, service=p_f_sqa_call
```

Explanation

A `KFPS00992-E` message displays a transaction branch that could not be committed or rolled back. In the case shown above, the transactions indicated by the following transaction identifiers may be undetermined:

- `TRNGID=HRD1unt100020b4d`
- `TRNGID=HRD1unt100020b22`
- `TRNGID=HRD1unt100020b5a`

If there are no undetermined transactions, no further action is necessary.

(2) Start the client product

Check whether the product used as the HiRDB client (OLTP system or HiRDB Datareplicator, for example) is running normally. If not, start it using the procedure appropriate to the particular product.

You should note the following about forcing determination of a transaction while there is a stopped client product:

- An undetermined transaction that was generated by a request from a client product that is now stopped may be rolled back forcibly when the steps that follow are performed. If the client product is then started subsequently, it may not be possible to synchronize the transaction between HiRDB and the client product.

To maintain transaction integrity, it is important that you do not execute any new transactions from this point on during this procedure.

(3) Start the servers that have stopped

Use the `pdls` command to check if all servers (excluding recovery-unnecessary front-end servers) are running. If any server is not running, start it.

You should note the following about forcing determination of a transaction while there is a stopped server:

- A transaction that is related to the stopped server may be rolled back forcibly when the operations that follow are performed. If the server is then started subsequently, it may not be possible to synchronize the transaction.
- There is no need to start a recovery-unnecessary front-end server that has stopped.

(4) Perform transaction determination manually on undetermined transactions

Perform transaction determination manually on the undetermined transactions. For details about the manual determination procedure, see *18.14.2 Performing transaction determination manually on undetermined transactions*.

(5) Create input files for the pdtrndec command (for the unit where the system manager is located)

Use the `grep` command to create input files for the `pdtrndec` command. This applies only to the unit where the system manager is located. The files you create with the `grep` command are for `KFPS00990-I` messages.

```
grep "KFPS00990-I" /usr/adm/OLDsyslogfile >/tmp/unit1syslog.copy
grep "KFPS00990-I" /usr/adm/syslogfile >/tmp/unit1syslog2.copy
      1                 2                  3
```

Explanation

1. Specifies the `KFPS00990-I` character string as the pattern for execution of the `grep` command.
2. Specifies the absolute path name of a `syslogfile` to be extracted (`syslogfile` of the unit where the system manager is located).
3. Specifies the absolute path name of an input file for the `pdtrndec` command.

Hint:

- To avoid adverse impacts on `syslogfile` from a direct access, extract from `syslogfile` only the row containing the character string `"KFPS00990-I"` and save it in a file of your choice. If the extracted contents are processed, the results of the `pdtrndec` command that will be executed in step (6) cannot be guaranteed. For this reason, you must not perform any processing other than execution of the `grep` command.
- Because `syslogfile` may have been swapped, all `syslogfile`s that have been created since the system started normally (time at which the `KFPS05210-I` message was output after the `KFPS01803-I start mode=S` message had been output) must be used as input targets. If there is no `syslogfile` that includes the time at which the system started normally, use all existing `syslogfile`s as input targets.
- The absolute path name of a `syslogfile` to be input depends on the platform. Specify the absolute path name specified in `/etc/syslog.conf`.

(6) Using the input files created in (5), execute the pdtrndec command

```
pdtrndec -i /tmp/unit1syslog.copy,/tmp/unit1syslog2.copy
```

Explanation

`-i`: Specifies input of the files created in (5).

Check the items listed under `DECISION` (transaction determination type) in the command execution results. Then continue the procedure by following the steps in the flowchart.

```
pdtrndec result                      START TIME:2002/09/05 17:20:08
HOSTNAME    TRNGID           TRNBID           SVID      DECISION    TIME
host1       HRD1unt100020b4d HRD1unt100020034 bes1      COMMIT      17:20:08
host1       HRD1unt100020b22 HRD1unt100020035 bes1      ********    --:--:--
host1       HRD1unt100020b51 HRD1unt100020038 bes1      FORGET      17:20:09
pdtrndec result                      START TIME:2002/09/05 17:36:47
HOSTNAME    TRNGID           TRNBID           SVID      DECISION    TIME
host1       HRD1unt100020b22 HRD1unt100020035 bes1      ********    --:--:--
```

Note:

If there are still any undetermined transactions after the `pdtrndec` command has executed, a `pdtrnrbk.bat` file is created. However, do not execute `pdtrnrbk.bat` at this point. If you execute it, you may lose transaction synchronization. Also, if the `KFPS00982-E` message is output when a `pdtrnrbk.bat` file is created, delete the `pdtrnrbk.bat` file without using it.

(7) Create input files for the pdtrndec command (for each unit where the system manager is not located)

Use the `grep` command to create input files for the `pdtrndec` command. This applies only to each unit where the system manager is not located. The files you create with the `grep` command are for the `KFPS00990-I` messages.

```
grep "KFPS00990-I" /usr/adm/OLDsyslogfile >/tmp/unit2syslog.copy
grep "KFPS00990-I" /usr/adm/syslogfile >/tmp/unit2syslog2.copy
          1                2                    3
```

Explanation

1. Specifies the `KFPS00990-I` character string as the pattern for execution of the `grep` command.
2. Specifies the absolute path name of a `syslogfile` to be extracted (`syslogfile` of a unit where the system manager is not located).
3. Specifies the absolute path name of an input file of the `pdtrndec` command.

Hint:

- To avoid adverse impacts on `syslogfile` from a direct access, extract from `syslogfile` only the row containing the character string `"KFPS00990-I"` and save it in a file of your choice. If the extracted contents are processed, the results of the `pdtrndec` command that will be executed in step (8) cannot be guaranteed. For this reason, you must not perform any processing other than execution of the `grep` command.
- Because `syslogfile` may have been swapped, all `syslogfile`s that have been created since the system started normally (time at which the `KFPS05210-I` message was output after the `KFPS01803-I start mode=S` message had been output) must be used as input targets. If there is no `syslogfile` that includes the time at which the system started normally, use all existing `syslogfile`s as input targets.
- The absolute path name of a `syslogfile` to be input depends on the platform. Specify the absolute path name specified in `/etc/syslog.conf`.

(8) Create input files for the pdtrndec command (for the unit where the system manager is located)

Use the same method as in step (5) to create input files for the `pdtrndec` command.

(9) Using the input files created in (7) and (8), execute the pdtrndec command

Using the input files created in steps (7) and (8), execute the `pdtrndec` command to force determination of the transactions. If the return code indicating the final status after the first execution of the `pdtrndec` command is `4` and `********` is listed under `DECISION` for a transaction, specify the option shown below and execute the `pdtrndec` command again.

```
pdtrndec -i /tmp/unit1syslog.copy,/tmp/unit1syslog2.copy,/tmp/unit2syslog.copy,
/tmp/unit2syslog2.copy -r pdtrnrbk.bat
```

Explanation

`-i`: Specifies input of the files created in steps (7) and (8).

`-r`: Specifies the shell script (`pdtrnrbk.bat`) file created in step (6).

If the `KFPS00982-E` message is output when a `pdtrnrbk.bat` file has been created, delete the `pdtrnrbk.bat` file without using it.

(10) Check the transaction status

Use the `pdls -d trn` command to check if all transactions have been determined. If

an undetermined transaction remains, check the output message and command execution results, take the necessary corrective action, and then repeat the steps beginning with (7).

```
pdls -d trn

HOSTNAME : host1(153415)
TRNGID           TRNBID           STATUS        PID   SVID   B-SVID
```

Explanation

No transaction information is displayed, which means that all transactions have been determined.

(11) Check for the KFPS02183-I message

Once all undetermined transactions have been determined, the `KFPS02183-I` message is output within 30 seconds. If the `KFPS02183-I` message is not output after 30 seconds, an undetermined transaction remains. In such a case, repeat the procedure from the start.

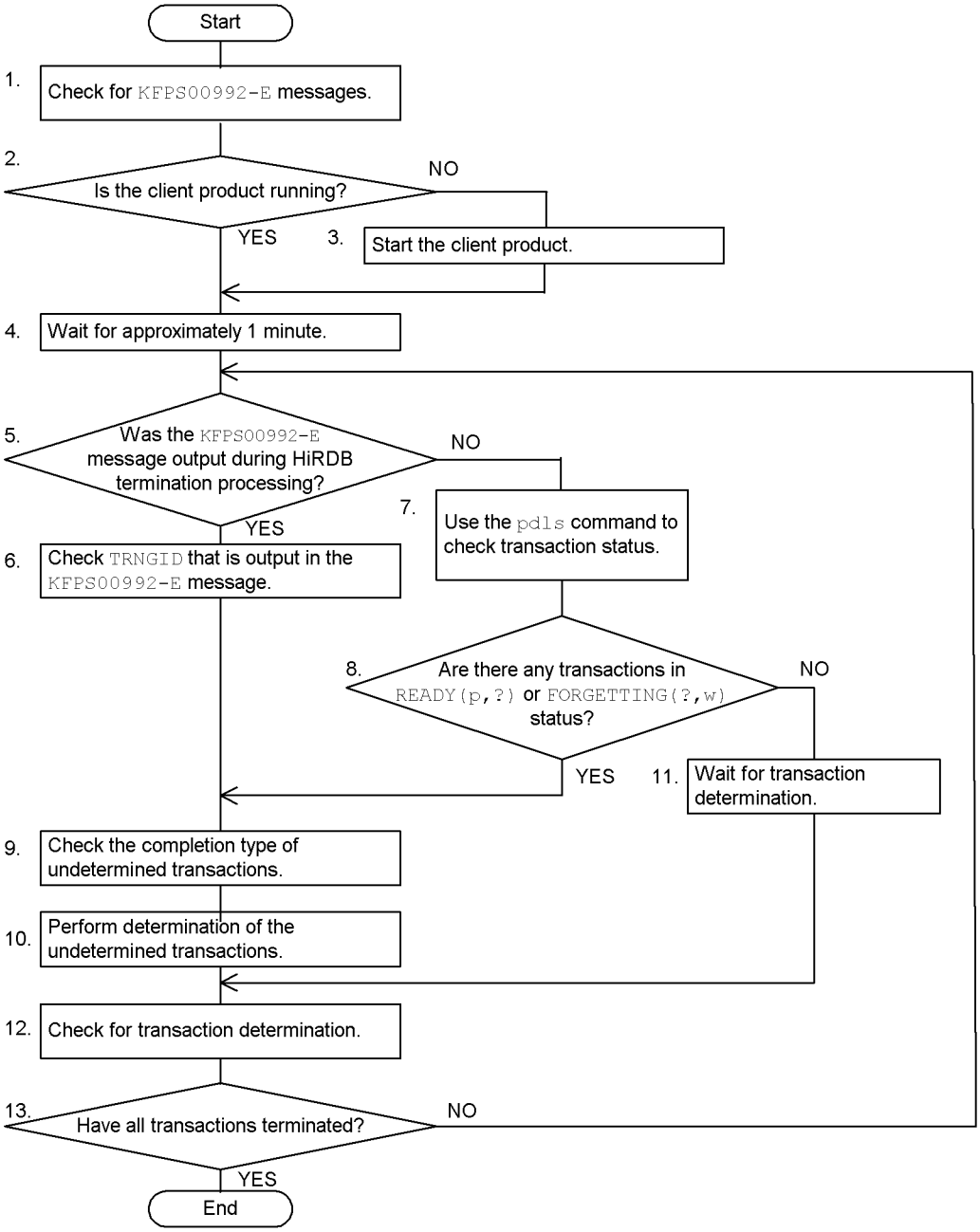
(12) Delete unneeded files

Delete the files (such as `pdtrndecout` and `pdtrnrbk.bat`) under the output destination directory specified by the `pdtrndec` command.

18.14.2 Performing transaction determination manually on undetermined transactions

This section explains how to perform transaction determination manually on undetermined transactions.

Procedure


Start
1. Check for KFPS00992-E messages.
2. Is the client product running?
NO
YES
3. Start the client product.
4. Wait for approximately 1 minute.
5. Was the KFPS00992-E message output during HiRDB termination processing?
NO
YES
6. Check TRNGID that is output in the KFPS00992-E message.
7. Use the pdls command to check transaction status.
8. Are there any transactions in READY(p,?) or FORGETTING(?,w) status?
NO
YES
11. Wait for transaction determination.
9. Check the completion type of undetermined transactions.
10. Perform determination of the undetermined transactions.
12. Check for transaction determination.
13. Have all transactions terminated?
NO
YES
End

Notes

- The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 5 is explained in paragraph (5) below.
- Messages can be checked during the procedure explained below. To do so, reference the messages in `syslogfile`.

(1) Check for KFPS00992-E messages

Search `syslogfile` for `KFPS00992-E` messages.

```
KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b4d, TRNBID=HRD1unt100020034, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b22, TRNBID=HRD1unt100020035, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b5a, TRNBID=HRD1unt100020036, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b5f, TRNBID=HRD1unt100020037, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b64, TRNBID=HRD1unt100020038, server=bes1, service=p_f_sqa_call
```

Explanation

The `KFPS00992-E` messages display transaction branches that could not be committed or rolled back. In this example, the transactions with the following transaction identifiers may be undetermined:

- `TRNGID=HRD1unt100020b4d`
- `TRNGID=HRD1unt100020b22`
- `TRNGID=HRD1unt100020b5a`
- `TRNGID=HRD1unt100020b5f`
- `TRNGID=HRD1unt100020b64`

(2) Is the client product running?

Check whether the product used as the HiRDB client is running normally.

(3) Start the client product

If the product used as the HiRDB client is not running normally, start it using the procedure appropriate to the particular product.

Do not forcibly start the client product.

(4) Wait for approximately 1 minute

Wait for approximately 1 minute, because transaction automatic determination may be underway.

(5) Was the KFPS00992-E message output during HiRDB termination processing?

The action to be taken depends on whether the KFPS00992-E message was output during HiRDB termination processing.

(6) Check TRNGID that is output in the KFPS00992-E messages

If the KFPS00992-E message was output during HiRDB termination processing, the pdls -d trn command cannot be executed. Therefore, you must check the TRNGID that is output in each KFPS00992-E message.

(7) Use the pdls command to check transaction status

Use the pdls -d trn command to check the status of transactions.

```
pdls -d trn

HOSTNAME : host1(153415)
TRNGID           TRNBID           STATUS             PID    SVID    B-SVID
HRD1unt100020b4d HRD1unt100020034  READY(p,n)        0      bes1    fes1
HRD1unt100020b22 HRD1unt100020035  READY(p,n)        0      bes1    fes1
HRD1unt100020b5a HRD1unt100020036  READY(p,n)        0      bes1    fes1
HRD1unt100020b5f HRD1unt100020037  ROLLBACK(u,n)     29799  bes1    fes1
HRD1unt100020b69 HRD1unt100020039  FORGETTING(r,w)   0      bes1    fes1
```

Explanation

Check the information listed in the STATUS column.

- The status of the three transactions listed below is READY(p,?).

  TRNGID=HRD1unt100020b4d

  TRNGID=HRD1unt100020b22

  TRNGID=HRD1unt100020b5a

- Transaction TRNGID=HRD1unt100020b5f was rolled back by determination processing because a client product was started. This transaction will be determined automatically.
- Transaction TRNGID=HRD1unt100020b64 was determined by determination processing because a client product was started. For this reason, this transaction is not listed in the execution results of the pdls -d trn command.
- The status of transaction TRNGID=HRD1unt100020b69, which was not

output in a KFPS00992-E message, is FORGETTING(?,w).

(8) Are there any transactions in READY(p,?) or FORGETTING(?,w) status?

If the execution results of the pdls -d trn command still show any transactions in READY(p,?) or FORGETTING(?,w) status (in the STATUS column), you must perform determination of those transactions.

If only transactions in statuses other than READY(p,?) or FORGETTING(?,w) remain, wait for those transactions to be determined automatically.

(9) Check the completion type of undetermined transactions

If there are any transactions in READY(p,?) or FORGETTTING(?,w) status, search the KFPS00990-I messages in syslogfile for the TRNGIDs of the transactions to be determined. Each KFPS00990-I message shows the transaction's completion type. Search only the KFPS00990-I messages that have been output since the previous normal startup.

```
KFPS00990-I Transaction branch recovery complete. TRNGID=HRD1unt100020b5a,
TRNBID=HRD1unt200020015, server=fes1, service=p_f_sqa_cauxi, completion type=c, c

KFPS00990-I Transaction branch recovery complete. TRNGID=HRD1unt100020b4d,
TRNBID=HRD1unt200020014, server=fes1, service=p_f_sqa_cauxi, completion type=r, r
```

Explanation

The search results and the actions to be taken are explained below:

- Completion type of transaction TRNGID=HRD1unt100020b5a

  Because completion type=c is shown, commit must be used to determine the transaction.

- Completion type of transaction TRNGID=HRD1unt100020b4d

  Because completion type=r is shown, rollback must be used to determine the transaction.

Transactions that were not found in the search and the actions to be taken are explained below:

- Completion type of transaction TRNGID=HRD1unt100020b22

  Because there is no KFPS00990-I message for this TRNGID, rollback must be used to determine the transaction.

- Completion type of transaction

  TRNGID=HRD1unt100020b69

  Because the STATUS is FORGETTING(?,w), the transaction must be

terminated forcibly.

You need not check the completion types of transactions identified by the following TRNGIDs:

- TRNGID=HRD1unt100020b5f
- TRNGID=HRD1unt100020b64

(10) Perform determination of the undetermined transactions

Use the following commands to perform determination of undetermined transactions:

Command name	Function and application criteria
pdcmt	Commits a transaction.
pdrbk	Rolls back a transaction.
pdfgt	Terminates a transaction forcibly. This command is executed for a transaction in FORGETTING(?,w) status.

(11) Wait for transaction determination

If the only remaining transactions are in statuses other than READY(p,?) or FORGETTING(?,w), these transactions are being determined. Therefore, wait for determination processing to be completed.

(12) Check for transaction determination

When the KFPS00992-E message was output during HiRDB termination processing, check whether or not the KFPS02183-I message has been output. When all undetermined transactions have been determined, the KFPS02183-I message is output within 30 seconds. If this message is not output within 30 seconds, an undetermined transaction remains.

If the KFPS00992-E message is output during HiRDB restart, or if the execution of the pdls -d trn command while HiRDB is running identifies an undetermined transaction, use the pdls -d trn command to check if the transaction has been determined.

(13) Have all transactions terminated?

If there is any transaction that has not terminated, return to step (5) in the procedure.

18.15 Handling of reduced activation (HiRDB/Parallel Server only)

Executor: HiRDB administrator

Normally, a HiRDB/Parallel Server cannot be started if any of its units will not start. Therefore, if an error occurs at a unit, HiRDB cannot be started until the error is corrected. In such a case, reduced activation enables HiRDB to be started using only the normal units (the units that can be started.

(1) To use reduced activation

To use reduced activation, `1` must be specified in the `pd_start_level` operand. HiRDB will start using only the available units if there are units that cannot be started.

When reduced activation takes effect, the `KFPS05217-I` message is issued. The `pdls` command can be used to determine the units that could not be started.

Remarks

During startup, HiRDB waits for up to 20 minutes to receive a startup message from all units. Therefore, there is a delay of 20 minutes before reduced activation is executed. The maximum of 20 minutes to wait to receive a startup message can be changed in the `pd_reduced_check_time` operand of the system common definition.

(2) Returning to normal operation mode

The following procedure shows how to return HiRDB to the normal mode after it has run in reduced activation.

Procedure

To return to normal operation mode:

1. Use the `pdls` command to check and eliminate the error in the unit that cannot be started.

   ```
   pdls -d svr
   ```

2. Use the `pdstart -u` command to start up a unit that is not active:

   ```
   pdstart -u unt1
   ```

   When HiRDB returns to the normal operation mode, the `KFPS05218-I` message is issued.

(3) When reduced activation is not available

In the following cases, reduced activation is not available; the cause of the error must be eliminated in order to start HiRDB:

- The unit to be started does not have either of the following servers:

- System manager
- Dictionary server

- The unit to be started has none of the following servers:
  - Front-end server
  - Back-end server
- The `pdstart -i` command is being used to start HiRDB.
- HiRDB is being started for the first time after being upgraded.

(4) Some tables may not be accessed when reduced activation is used

If an attempt is made to access a table at an inactive unit, an SQL error results. If a table is row-partitioned, accessing data stored at an inactive unit results in an SQL error.

(5) Use of reduced activation during a restart

If reduced activation is used during a restart, a transaction related to the erroneous unit may be placed in uncompleted status. The transaction will be determined automatically when reduced activation is released and normal operation is started. If there is a transaction in uncompleted status, neither normal termination nor planned termination can be executed on HiRDB. In such a case, the `pdstop -f` command must be used to terminate HiRDB forcibly.

(6) Notes on multiple front-end servers

Special care is needed when the front-end server to which a client user is to be connected has been determined; i.e., when the following operands have been specified in the client environment definition:

- `PDFESHOST`
- `PDSERVICEGRP`

If the front-end server specified in these operands is not running, no UAPs can be executed. To enable UAPs to execute, either delete these operands or specify a front-end server that is running.

(7) Notes on when HiRDB is linked to HiRDB Datareplicator

(a) HiRDB used at the extracting side

- Data extraction processing is not performed on a back-end server in an inactive unit. For such a case, once the unit is running normally again, use the `hdestart` command of HiRDB Datareplicator to restart data extraction processing. If, however, normal operations resume when HiRDB is restarted after it had been terminated while it was in reduced activation mode, you do not need to enter the `hdestart` command. Data extraction processing will restart automatically in such a case.

- If data extraction processing stops during reduced activation, execute the `pdrplstop` command. Avoid executing the `pdrplstop -f` command, if possible. When you execute the `pdrplstop` command, data execution processing will stop on completion if there are incomplete extraction status system log files in the unit that has been returned to normal operation. However, if you execute the `pdrplstop -f` command, data extraction processing stops only in conjunction with the entire system, even if there are incomplete extraction status system log files in the unit that has been returned to normal operation.

(b) HiRDB used at the target side

- If an attempt is made to reflect data in an inactive unit, an error results and the reflection processing is cancelled. In such a case, data reflection processing must be restarted with HiRDB Datareplicator's `hdsstart` command after reduced activation is released and normal operation is started.
- Re-create the target database based on the database in the unit that is inactive due to reduced activation. In such as case, however, mismatches may arise between the extracted database and the target database.

(8) When reduced activation is used for subsequent HiRDB startup

If reduced activation is to be used for a subsequent HiRDB startup because a unit error cannot be corrected, the name of the unit that cannot be started should be specified in the `pd_start_skip_unit` operand. Normally during startup, HiRDB waits for up to 20 minutes to receive a startup message from each unit. Therefore, there is a delay of 20 minutes before reduced activation is executed. If the name of a unit that cannot be started is specified in the `pd_start_skip_unit` operand, HiRDB does not wait for a unit startup message from that unit, which can save a considerable amount of time that HiRDB would otherwise spend waiting for a startup message, thereby reducing the time required for reduced activation.

Example

1. `1` is specified in the `pd_start_level` operand in the system common definition.
2. HiRDB executed reduced activation because a unit could not be started due to an error.
3. When the application terminated, the `pdstop` command was entered to terminate HiRDB. The error had not been corrected at this point.
4. The following operand was specified:

   `pd_start_skip_unit=`*name-of-unit-that-cannot-be-started*
5. To start the application, the `pdstart` command was entered to start HiRDB. The HiRDB startup time is reduced as compared to when the `pd_start_skip_unit` is not specified.

Note:

To return to normal operation, delete the `pd_start_skip_unit` operand. If you do not delete this operand, HiRDB will again be placed in reduced activation mode the next time it is started.

(9) Notes about using recovery-unnecessary front-end servers

Recovery-unnecessary front-end server units start independently in the reduced mode, regardless of the value set in the `pd_start_level` operand. Even if you specify a unit name in the `pd_start_skip_unit` operand, that specification is ignored and the recovery-unnecessary front-end server units are started. If any of the recovery-unnecessary front-end server units cannot be started during HiRDB startup, HiRDB is started without those units.

18.16 Handling of disk errors

Executor: HiRDB administrator and superuser

This sections explains the actions to take when a disk error occurs.

Procedure

1. Initialize the hard disk.
2. Set partitions.
3. Initialize the UNIX file system (if regular files are being used).
4. Change the owner and access privileges of the HiRDB file system area (if character special files are being used).
5. Create a symbolic link to the file name.
6. Initialize the HiRDB file system area.
7. Create system files.
8. Restore the RDAREAs.

The procedure step numbers correspond to the paragraph numbers in the explanation that follows. For example, step 3 above is explained in paragraph (3) below.

(1) Initialize the hard disk

Executor: Superuser

Replace the hard disk, and initialize it.

Initialize the hard disk; for the procedure, see the OS manual.

(2) Set partitions

Executor: Superuser

Set partitions on the initialized hard disk; for the procedure, see the OS manual.

(3) Initialize the UNIX file system (applicable to regular files)

Executor: Superuser

If regular files had been used in the HiRDB file system area, initialize the partitions as a UNIX file system; for the procedure, see the OS manual.

If the partitions are already initialized, skip this step.

(4) Change each character special file's owner and access privileges (applicable to character special files)

Executor: Superuser

Set the owner and access privileges for each HiRDB file system area to the previous information. Table 18-19 shows the owner and access privileges to be set for a HiRDB file system area.

Table 18-19: Owner and access privileges to be set for a HiRDB file system area

Owner, access privileges		Information to be set	Command to be executed*
Owner	User ID	HiRDB administrator	`chown` command
	Group ID	HiRDB group	`chgrp` command
Access privilege	Owner	rw- (read and write operations permitted)	`chmod` command
	Group	rw- (read and write operations permitted)	
	Other	--- (access denied)	

* These are OS commands; for details, see the OS manual.

(5) Link the file names symbolically

Executor: Superuser

If a HiRDB file system area had been linked symbolically to a name, link the same name to the HiRDB file system area; the OS's `ln` command is used for this purpose (For details on the `ln` command, see the OS manual).

(6) Initialize the HiRDB file system area

Executor: HiRDB administrator

Initialize each HiRDB file system area on the erroneous disk with the `pdfmkfs` command.

(a) System files created on the erroneous disk

Use the `pdfmkfs` command to initialize a HiRDB file system area for system files:

```
pdfmkfs -n 40 -l 5 -k SYS -i /sysfile_a
```

Explanation

`-n`: Specifies the size of the HiRDB file system area. Set the size of the HiRDB file system area to be initialized so that it does not exceed the partition size. If it exceeds the partition size, the partitions physically following the HiRDB file system area may be damaged.

`-l`: Specifies the maximum number of files that can be created in the HiRDB file system area.

`-k SYS`: Specifies that this is a HiRDB file system area for system files.

`-i`: Specifies that the entire HiRDB file system area is to be initialized.

`/sysfile_a`: Specifies a name for the HiRDB file system area.

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(b) RDAREAs created on the erroneous disk

Use the `pdfmkfs` command to initialize the HiRDB file system area for RDAREAs:

```
pdfmkfs -n 40 -l 5 -k DB -i /dbarea1
```

Explanation

`-n`: Specifies the size of the HiRDB file system area. Set the size of the HiRDB file system area to be initialized so that it does not exceed the partition size. If it exceeds the partition size, the partitions physically following the HiRDB file system area may be damaged.

`-l`: Specifies the maximum number of files that can be created in the HiRDB file system area.

`-k DB`: Specifies that this is an HiRDB file system area for RDAREAs.

`-i`: Specifies that the entire HiRDB file system area is to be initialized.

`/dbarea1`: Specifies a name for the HiRDB file system area.

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

(7) Create system files

Create system files in the HiRDB file system area that was initialized in (6)(a) above.

```
pdloginit -d sys -f /sysfile_a/log1a -n 2000          1
pdloginit -d sys -f /sysfile_a/log2a -n 2000
pdloginit -d sys -f /sysfile_a/log3a -n 2000
pdloginit -d sys -f /sysfile_a/log4a -n 2000
pdloginit -d spd -f /sysfile_a/sync1 -n 1000          2
pdloginit -d spd -f /sysfile_a/sync3 -n 1000
pdstsinit -u UNT1 -f /sysfile_a/usts1a -c 500         3
pdstsinit -u UNT1 -f /sysfile_a/usts2a -c 500
pdstsinit -s bes1 -f /sysfile_a/b1sts1a -c 500        4
pdstsinit -s bes1 -f /sysfile_a/b1sts2a -c 500
```

Explanation

1. Creates system log files.
2. Creates synchronization point dump files.
3. Creates unit status files.
4. Creates server status files.

(8) Restore RDAREAs

Restore the RDAREAs on the erroneous disk with the `pdrstr` command. For an example of restoring RDAREAs, see *19. Database Recovery Procedures*.

Note:

If the erroneous disk contains the master directory RDAREAs, start HiRDB with the `pdstart -r` command and use the `pdrstr` command to restore the master directory RDAREA. For details on how to handle an error in the master directory RDAREA, see *18.4.3 Actions to be taken in the event of an error in the master directory RDAREA*.

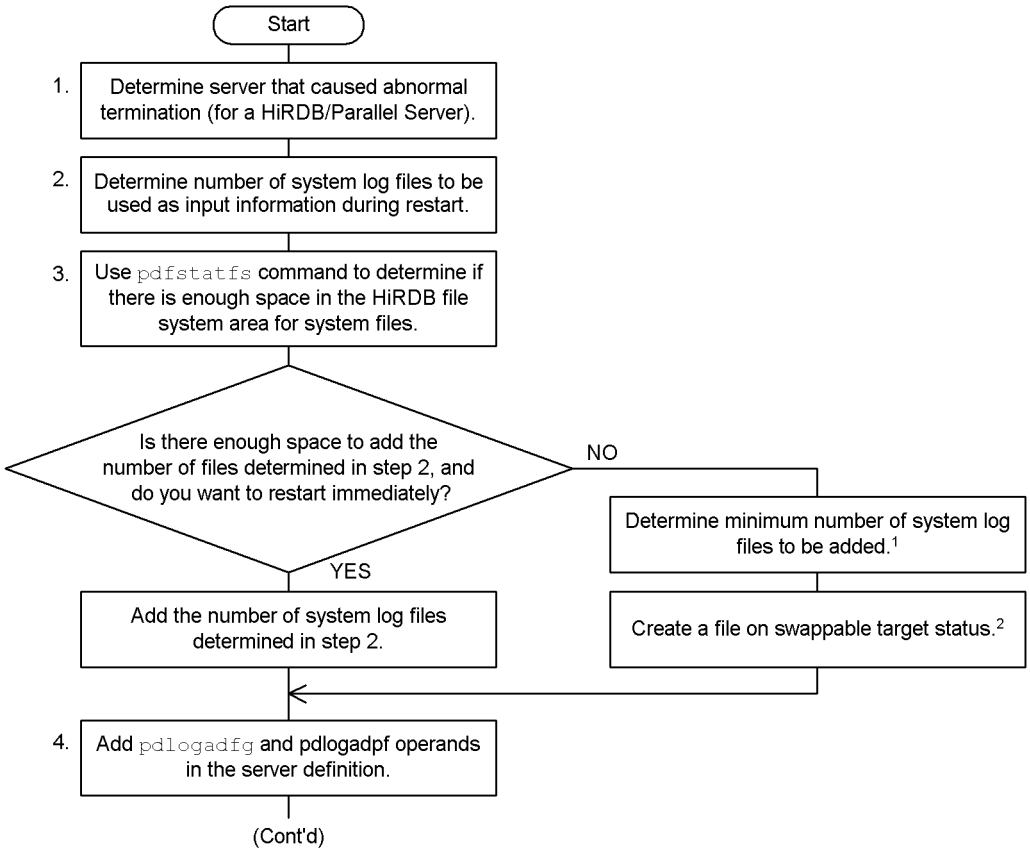
18.17 When a HiRDB (unit) terminates due to a system log file space shortage

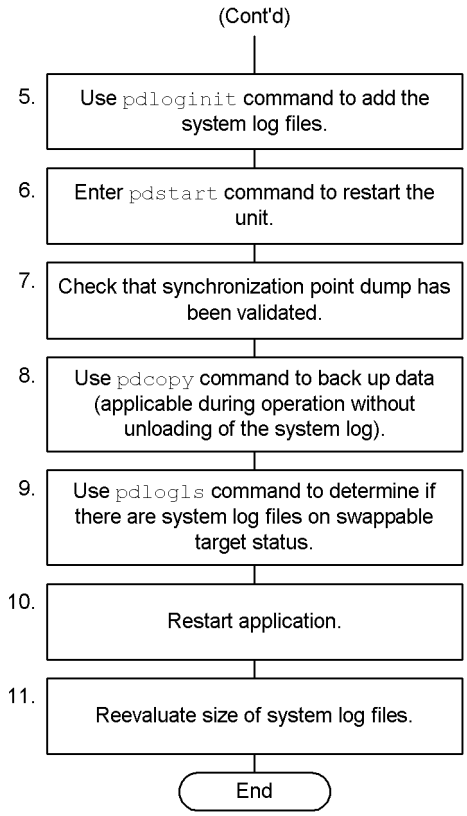
This section explains the actions to be taken when HiRDB (unit) terminates due to a system log file space shortage.

18.17.1 Restart procedure

When HiRDB runs out of current system log files due to a shortage of space, it issues the `KFPS01220-E` message and terminates itself (unit) abnormally. When this happens, abort code `Psjnf07` or `Psjn381` is output. The HiRDB administrator must restart the job using the procedure shown below:

Procedure





The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 5 is explained in paragraph (5) below.

[1] This step is explained in Section *18.17.2 Determining the minimum number of system log files to be added*.

[2] This step is explained in Section *18.17.3 Creating a file in swappable target status*.

Messages can be checked during the procedure explained below. Because messages in the HiRDB message log files (`$PDDIR/spool/pdlog1` and `pdlog2`) may have been overwritten, check the messages in syslogfile.

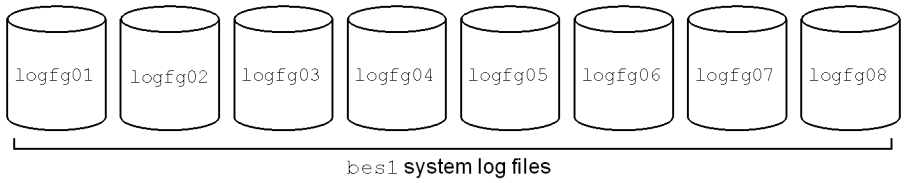
(1) Determine the server that caused the abnormal termination (for a HiRDB/ Parallel Server)

The server that caused the abnormal termination can be determined from the `KFPS01220-E` message:

Contents of the syslogfile

```
KFPS01220-E PRDT untF Request to swap sys(bes1) log file unable to be executed
because there is no standby log file group available.(13830)
```

In this example, bes1 is the cause of abnormal termination. Assume for this example that this server's system log files are organized as follows:



(2) Determine the number of system log files to be used as input information during restart

The number of system log files to be used as the input information during restart can be determined from the KFPS01229-I message and the pdlogls command.

Reference note:

- A method other than this method can also be used to determine the number of system log files to be used as input information during restart. For details on the other method, see *18.17.5 Determining the number of system log files to be used as input files during restart*.
- If pd_mode_conf=AUTO or pd_mode_conf=MANUAL1 is specified, automatic restart processing was executed several times after the unit terminated abnormally, in which case the KFPS01229-I message was output each time. The first KFPS01229-I message that was output during the first abnormal termination (during online operation) is the one that must be used.

Contents of the syslogfile

```
KFPS01220-E PRDT untF Request to swap sys(bes1) log file unable to be executed
because there is no standby log file group available.(13830)
KFPO00105-E PRDT untF Server _log1s(process ID=13830) killed by
code=Psjnf07(13830)
KFPS01821-E PRDT untF Unable to continue HiRDB unit processing because serious
error occurred; stops HiRDB unit untF (13776)
KFPS01229-I PRDT untF Next bes1 log file restart point,generation number=4,
block number=d. restart end point, generation number=6, blocknumber=11.
last acquired syncpoint dump 1998/11/15 15:54:41 (13776)
```

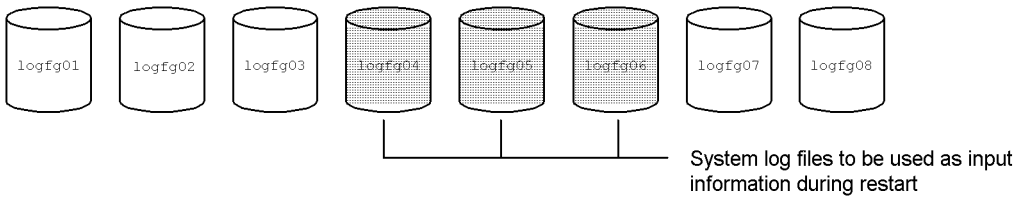
Execution results of the pdlogls command:

```
pdlogls -d sys -s bes1

HOSTNAME : dcm3500(163541)
***** Off-line Information *****
Group    Type Server   Gen No.  Status    Run ID      Block No.
logfg01  sys  bes1     1        cna---u  364a4ac2     1       6
logfg02  sys  bes1     2        cna---u  364a4ac2     7       9
logfg03  sys  bes1     3        cna---u  364a4ac2     a       c
logfg04  sys  bes1     4        cna---u  364a4ac2     d       e
logfg05  sys  bes1     5        cna---u  364a4ac2     f       10
logfg06  sys  bes1     6        cn---cu  364a4ac2     11      0
logfg07  sys  bes1     0        cn-----  00000000     0       0
logfg08  sys  bes1     0        cn-----  00000000     0       0
```

Explanation

The KFPS01229-I message displays information about the system log files that are to be used as the input information during the restart. In this example, the input start generation is 4 and the input end generation is 6 for the system log files to be used during the restart. Files with generation numbers (Gen No) 4 to 6 (logfg04, logfg05, logfg06) are the system log files to be used as the input information during the restart. Therefore, a total of 3 system log files are used as the input information during the restart.

If the number of system log files that can be added equals *number-of-system-log-files-used-as-input-during-restart* + 1 (3 + 1 = 4), HiRDB can be restarted. Otherwise, determine the minimum number of system log files to be added by referring to *18.17.2 Determining the minimum number of system log files to be added*. Then change files *in* overwrite-enabled status to swappable status and restart HiRDB.



(3) Use the pdfstatfs command to determine if there is enough space in the HiRDB file system area for system files

Use the pdfstatfs command to determine if there is enough space in the HiRDB file system area to add the four system log files determined in step (2).

```
pdfstatfs /bes1/sysfile_a
pdfstatfs /bes1/sysfile_b
```

The following procedure is used to determine the size of one system log file:

Procedure

To determine the system log file size:

1. Use the `pdfls` command to check the number of records in the existing system log file.
2. The size of one system log file can be obtained from the following formula, where *a* is the number of records obtained in step 1:

   $a \times 4096$ (bytes)

When there is not enough space in the HiRDB file system area

If there is not enough space in the HiRDB file system area to add the four system log files determined in step (2), take one of the following actions:

1. See *18.17.2 Determining the minimum number of system log files to be added* to determine the minimum number of files to be added. Then place overwrite-enabled files in swappable target status.
2. Create a HiRDB file system area for system files on the hard disk, then add the system log files there. For details on how to create a HiRDB file system area for system files, see *18.17.4 Creating a HiRDB file system area for system files*.
3. System log files cannot be added if there is not enough space on the hard disk (otherwise, the unit cannot be restarted). In such a case, add a new hard disk and create a HiRDB file system area for system files there.

(4) Add the pdlogadfg and pdlogadpf operands in the server definition

```
pdlogadfg -d sys -g logfg09 ONL
pdlogadpf -d sys -g logfg09 -a /bes1/sysfile_a/log09a\
                            -b /bes1/sysfile_b/log09b
pdlogadfg -d sys -g logfg10 ONL
pdlogadpf -d sys -g logfg10 -a /bes1/sysfile_a/log10a\
                            -b /bes1/sysfile_b/log10b
pdlogadfg -d sys -g logfg11 ONL
pdlogadpf -d sys -g logfg11 -a /bes1/sysfile_a/log11a\
                            -b /bes1/sysfile_b/log11b
pdlogadfg -d sys -g logfg12 ONL
pdlogadpf -d sys -g logfg12 -a /bes1/sysfile_a/log12a\
                            -b /bes1/sysfile_b/log12b
```

Specify the pdlogadfg and pdlogadpf operands for the system log files to be added. In this example, four files are added to each of versions A and B.

Note:

If HiRDB Datareplicator is running, it must be terminated. HiRDB Datareplicator can be started after steps (4) and (5) have been completed.

(5) Use the pdloginit command to add the system log files

```
pdloginit -d sys -s bes1 -f /bes1/sysfile_a/log09a -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_b/log09b -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_a/log10a -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_b/log10b -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_a/log11a -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_b/log11b -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_a/log12a -n 5000
pdloginit -d sys -s bes1 -f /bes1/sysfile_b/log12b -n 5000
```

In this example, four files are added to each of versions A and B.

(6) Enter the pdstart command to restart the unit

HiRDB/Single Server

```
pdstart
```

HiRDB/Parallel Server

In the case of a HiRDB/Parallel Server, restart the applicable unit.

```
pdstart -u untF
```

Although jobs can be accepted immediately after HiRDB (the unit) has been restarted, do not resume jobs yet. Immediately after HiRDB (the unit) is restarted, HiRDB is still recovering the database. If many database-updating jobs are entered at this point, a system log file space shortage may occur again. If that happens, HiRDB (the unit) will terminate abnormally again. Therefore, refrain from resuming jobs until all the steps through (9) have been completed.

(7) Check that the synchronization point dump has been validated

When HiRDB is restarted, a synchronization point dump is validated, in which case the KFPS02183-I message is output. A check should be made that the synchronization point dump has been validated before the next step is executed.

Note that the KFPS02183-I message is output only if Y was specified in the pd_spd_assurance_msg operand or if this operand was omitted. If this condition is not satisfied, check that the synchronization point dump is validated using the procedure explained in Section *18.17.6 Checking for synchronization point dump validation*.

(8) Use the pdcopy command to back up data (applicable to operation without unloading of the system log)

```
pdcopy -m dcm3500:/dbarea/area1/rdmt1 -M x -p /usr/ofile -f /usr/seifile/cfl
-z /usr/bes1/logpoint02
```

If operation without unloading of the system log was in effect, use the database copy utility (pdcopy command) to back up all RDAREAs in the server before restarting an application. In such a case, specify the -z option to determine the log point information file.

After backing up the RDAREAs, use the pdlogchg -z command to release the system log file and place it in unload completed status:

```
pdlogchg -z /usr/bes1/logpoint02 [-x host_name]
```

(9) Use the pdlogls command to determine if there is a system log file in swappable target status

```
pdlogls -d sys -s bes1
```

If there is no file in swappable target status, HiRDB may terminate abnormally again. Immediately create a file in swappable target status. For example, if a file can be placed in swappable target status by releasing it from unload wait status, use the pdlogunld command to unload the system log.

(10) Restart the application

Restart the application.

(11) Reevaluate the size of the system log files

Reevaluate the size of the system log files. For details on how to obtain the size of system log files, see the manual *HiRDB Version 8 Installation and Design Guide*. If any added system log file is not needed, delete it. The procedure for deleting a system log file is shown below.

If operation without unloading of the system log is being used, a system log file cannot

be deleted while it is in use as the current system log file. Check that the system log file to be deleted is not the current file, execute step (8) again, then delete the system log file.

Procedure

To delete a system log file:

1. Enter the `pdstop` command to terminate HiRDB normally.
2. Unload the system log file to be deleted, or change its status.
3. If HiRDB Datareplicator is being used, check that it has finished extracting the system log file that is to be deleted.
4. If HiRDB Datareplicator is running, terminate it.
5. Use the `pdlogrm` command to delete the system log file.
6. Delete the `pdlogadfg` and `pdlogadpf` operands for the deleted system log file.
7. Enter the `pdstart` command to start HiRDB normally.
8. If HiRDB Datareplicator is being used, start HiRDB Datareplicator.

18.17.2 Determining the minimum number of system log files to be added

The following formula is used to obtain the minimum number of system log files to be added:

Formula

Minimum-number-of-system-log-files-to-be-added = L - {N - (M + Z*)}

When the result of this formula is less than 1, there is no need to add system log files.

L: Number of system log files to be used as the input information during restart

This is the value derived as described in *18.17.1(2) Determine the number of system log files to be used as input information during restart*.

M: Number of system log files in overwrite disabled status

See (1) below for the procedure for determining the number of system log files in overwrite disabled status.

N: Total number of system log files in the server

Count the number of spare files also. For details about spare files, see *18.17.3(3) Use a spare file*.

Z: Number of system log files in overwrite enabled and unload wait status

* Add this value if operation without unloading of the system log is being used.

(1) Determining the number of system log files in overwrite disabled status

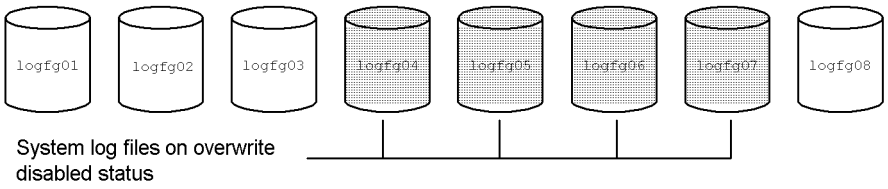
The pdlogls command cannot be used during offline operation to determine whether or not a system log file is in overwrite disabled status. The number of system log files in overwrite disabled status can be determined from messages that are output during restart processing:

Contents of the syslogfile

```
KFPS01800-I PRDT untF Now starting HiRDB unit untF (19924)
KFPS01262-I PRDT untF Log block reading started. type:sys(bes1),
read start point:logfg04,4,d (19993)
KFPS01182-I PRDT untF Generation file groups changed for further
reading of log blocks. type:sys(bes1), from:logfg04,4, to:logfg05,5,
read direction:f (19993)
KFPS01182-I PRDT untF Generation file groups changed for further
reading of log blocks. type:sys(bes1), from:logfg05,5, to:logfg06,6,
read direction:f (19993)
KFPS01182-I PRDT untF Generation file groups changed for further
reading of log blocks. type:sys(bes1), from:logfg06,6, to:logfg07,7,
read direction:f (19993)
KFPS01263-I PRDT untF Log block reading completed. type:sys(bes1),
read end point:logfg07,7,1e (19993)
KFPS01220-E PRDT untF Request to swap sys(bes1) log file unable to be
executed because there is no standby log file group available. (19993)
KFPO00105-E PRDT untF Server _log1s(process ID=19993) killed by
code=Psjnf07(19993)
KFPS01821-E PRDT untF Unable to continue HiRDB unit processing because
serious error occurred; stops HiRDB unit untF (19979)
```

Explanation

KFPS01262-I, KFPS01182-I, and KFPS01263-I indicate that system log files logfg04, logfg05, logfg06, and logfg07 are used during the restart. These are the overwrite disabled files, so the number of system log files in overwrite disabled status is 4:



Reference note:

- If `pd_mode_conf=AUTO` or `pd_mode_conf=MANUAL1` is specified, automatic restart processing was executed several times after the unit terminated abnormally and these messages were output each time. The set of messages that was output during the first abnormal termination are the ones that must be used.
- If `pd_mode_conf=MANUAL2` is specified and restart processing is not performed (the `pdstart` command is not entered), the number of system log files used as input information during restart (the value derived as described in *18.17.1(2) Determine the number of system log files to be used as input information during restart*) is the number of overwrite-disabled files.
- If the number of guaranteed valid generations of synchronization point dump files is 2 (if `pd_spd_assurance_count=2` is specified), the number of overwrite-disabled files may increase further. Check by using the method shown below in *(2) Determining the number of system log files in overwrite disabled status (number of valid generations = 2)*.

(2) Determining the number of system log files in overwrite disabled status (number of valid generations = 2)

In (1) above, the system log files in overwrite disabled status that were in existence after the most recent synchronization point dump generation was validated were identified. If the number of synchronization point dump guaranteed valid generations is 2, the files in overwrite disabled status that were in existence up to the point where the previous synchronization point dump generation was validated must also be identified. There are two ways to identify these files:

- Using the synchronization point dump validation completion message (`KFPS02183-I`)
- Using the synchronization point dump validation skip message (`KFPS02179-I`)

(a) Using the synchronization point dump validation completion message (KFPS02183-I)

The files in overwrite disabled status can be identified using the `KFPS02183-I` message and the `pdlogls` command:

Contents of the syslogfile

```
KFPS05210-I PRDT untF HiRDB system initialization process complete(13778)
KFPS01221-I PRDT untF logfg03 assigned as current file group of sys(bes1)
log file. generation number=3, first block number=a (13830)
KFPS02183-I PRDT untF Syncpoint dump for bes1 has been acquired to file
group spdfg03. log file information:logfg03, 3, a. start time=15:53:11,
end time=15:53:11 (13830)    …1

KFPS02183-I PRDT untF Syncpoint dump for bes1 has been acquired to file
group spdfg04.  log file information:logfg04, 4, d. start time=15:54:40,
end time=15:54:41 (13830)    …2

KFPS01222-I PRDT untF logfg04 released from sys(bes1) log file. generation
number=4, first block number=d, last block number=e (13830)
KFPS01224-I PRDT untF sys(bes1) log does not have standby file group
available for next swapping. (13830)
KFPS01220-E PRDT untF Request to swap sys(bes1) log file unable to be
executed because there is no standby log file group available.(13830)
KFPO00105-E PRDT untF Server _log1s(process ID=13830) killed by code=
Psjnf07(13830)
KFPS01821-E PRDT untF Unable to continue HiRDB unit processing because
serious error occurred; stops HiRDB unit untF (13776)
KFPS01229-I PRDT untF Next bes1 log file restart point, generation
number=4, block number=d. restart end point, generationnumber=7,
block number=1c. last acquired syncpoint dump 1998/11/15 15:54:41 (13776)
```

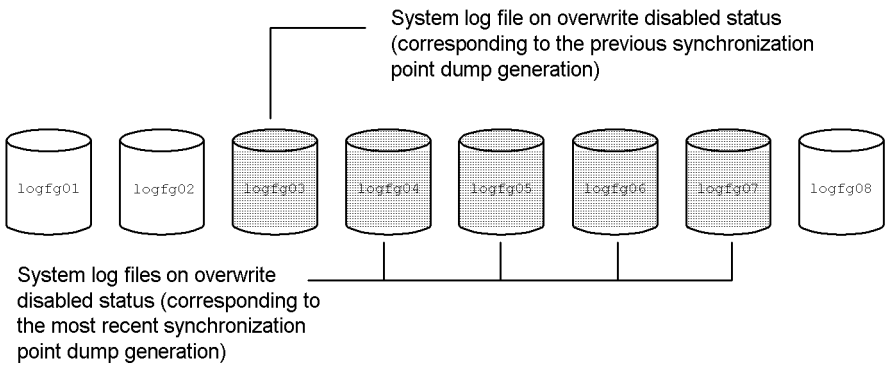
pdlogls command execution results:

```
pdlogls -d sys -s bes1

HOSTNAME : dcm3500(163541)
***** Off-line Information *****
Group    Type Server   Gen No.  Status    Run ID       Block No.
logfg01  sys  bes1     1        cna---u   364a4ac2     1        6
logfg02  sys  bes1     2        cna---u   364a4ac2     7        9
logfg03  sys  bes1     3        cna---u   364a4ac2     a        c
logfg04  sys  bes1     4        cna---u   364a4ac2     d        e
logfg05  sys  bes1     5        cna---u   364a4ac2     f        10
logfg06  sys  bes1     6        cna---u   364a4ac2     11       11
logfg07  sys  bes1     7        cn---cu   364a4ac2     12       0
logfg08  sys  bes1     0        cn-----   00000000     0        0
```

Explanation

1. This is the message regarding validation of the previous synchronization point dump generation. The indicated system log file is `logfg03` (generation number 3).
2. This is the message regarding the most recent synchronization point dump generation. The indicated system log file is `logfg04` (generation number 4).

It is clear from the first message that `logfg03` is also in overwrite disabled status. Therefore, `logfg03` through `logfg07` are all in overwrite disabled status:



(b) Using the synchronization point dump validation skip message (KFPS02179-I)

The files in overwrite disabled status can be identified from the KFPS02179-I message:

Contents of the syslogfile

```
KFPS01221-I PRDT untF logfg02 assigned as current file group of sys(bes1)
log file. generation number=b, first block number=66 (5075)
KFPS01222-I PRDT untF logfg02 released from sys(bes1) log file. generation
number=b, first block number=66, last block number=66 (5075)
KFPS01221-I PRDT untF logfg03 assigned as current file group of sys(bes1)
log file. generation number=c, first block number=67 (5075)   …3

KFPS01222-I PRDT untF logfg03 released from sys(bes1) log file.generation
number=c, first block number=67, last block number=68 (5075)
KFPS01221-I PRDT untF logfg04 assigned as current file group of sys(bes1)
log file.generation number=d, first block number=69 (5075)   …1

KFPS02179-I PRDT untF Syncpoint dump acquisition opportunities for bes1
service were skipped.number of skip=1,log generation number=d factor
code=A01-02 (5054)   …2

KFPS02179-I PRDT untF Syncpoint dump acquisition opportunities for bes1
service were skipped.number of skip=1,log generation number=d factor
code=A01-02 (5054)
KFPS01222-I PRDT untF logfg04 released from sys(bes1) log file.generation
number=d, first block number=69, last block number=73 (5075)
```

Explanation

1. Look for the KFPS01221-I message indicating that logfg04, which is the system log input start point during the restart, was allocated as the current file.
2. Look for the KFPS02179-I message indicating that the validation skips

count was set to 1 while logfg04 was being used as the current file.

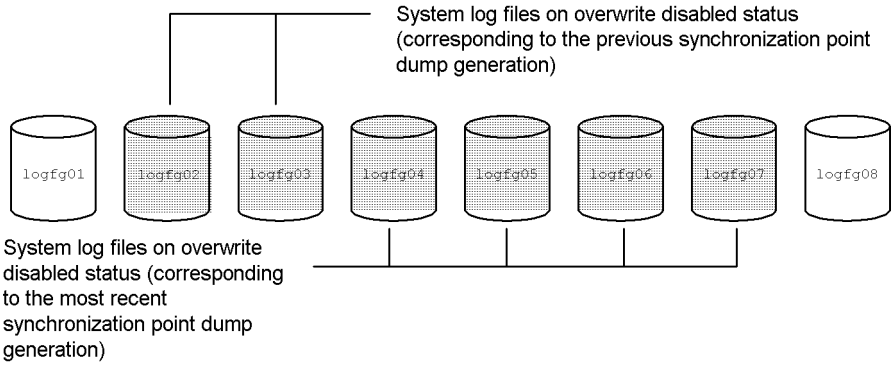
3. If the KFPS02179-I message was output while logfg04's predecessor (logfg03) was used as the current system log file, look for the KFPS02179-I message indicating the validation skips count = 1, and identify the system log file that was being used as the current file when that message was output. Then determine the additional number of overwrite-disabled system log files from the following table:

Condition		Output of KFPS02179-I message while the previous system log file generation was allocated as the current file		
		Output		Not output
		Validation skips count ≠ 1	Validation skips count = 1	
Output of KFPS02179-I message while the input start system log file was allocated as the current file at the restart	Output	(Number of files existing up to the point where the message indicating skips count = 1 was output) + 1	2	2
	Not output			1

Note

The number of files in overwrite disabled status determined in (1) plus the number of overwrite-disabled files determined above cannot exceed the total number of system log files in the corresponding HiRDB server. If the total number of system log files is exceeded, the total number of system log files in the corresponding HiRDB server must be used as the total number of overwrite-disabled system log files.

In this example, the KFPS02179-I message was output while logfg04, which is at the system log input start point during restart, was allocated as the current file. The KFPS02179-I message was not output while its predecessor, logfg03, was allocated as the current file. Therefore, logfg03 and logfg02 are identified as the system log files in overwrite disabled status, and a value of 2 is added to the result of (1).



18.17.3 Creating a file in swappable target status

This section explains the procedures for placing overwrite-enabled files and spare files in swappable target status.

(1) Change the file status from unload wait to unload completed

Use the `pdlogls` command to check for files in unload wait status. Use the `pdlogunld` or `pdlogchg` command to change their file status from unload wait to unload completed.

Note:

This procedure must not be used when operation without unloading system log is being used. In this operation mode, once a system log file in unload wait status is unloaded or its file status is changed, the database recovery method is no longer applicable. If a space shortage occurs in the system log file, add a new system log file and restart the system.

(2) Wait until the file status changes from extracting status to extraction-completed status

During offline operation, the `pdlogls` command cannot be used to determine whether or not a file is in extraction-completed status. Instead, the `pdls -d rpl -j` command can be used to determine whether the system log is being extracted by the HiRDB Datareplicator at the extracted side. If it is being extracted, there is a file in extracting status. The following is the procedure for determining whether or not the system log is being extracted:

pdls command execution results

```
pdls -d rpl -s bes1 -j

SYSTEMID         : PRDT(185014)
Data replication : *
UNITID           : untF(185014)
Data replication : Y
SERVER NAME      : bes1
Extract Database : Y
Extract Status   : C
System Log Extract Point :
Run ID    Group    Gen No.  BLock No.
364a4ac2  logfg01      1        2
System Log Sync Info     :
Run ID    Group    Gen No.  BLock No.
364a4ac2  logfg07      7        1e
```

Explanation

The generation number of the current system log file is **7**, while that of the system log file subject to extraction is **1**. This means that there is a file in extracting status. If HiRDB Datareplicator is not active, start it and extract the system log.

pdls command execution results

```
pdls -d rpl -s bes1 -j

SYSTEMID         : PRDT(185612)
Data replication : *
UNITID           : untF(185612)
Data replication : Y
SERVER NAME      : bes1
Extract Database : Y
Extract Status   : C
System Log Extract Point :
Run ID    Group    Gen No.  BLock No.
364a4ac2  logfg07      7        20
System Log Sync Info     :
Run ID    Group    Gen No.  BLock No.
364a4ac2  logfg07      7        1e
```

Explanation

The generation number of the current system log file is **7**, while that of the system log file subject to extraction is also **7**. Therefore, all files are in extraction-completed status.

(3) Use a spare file

If there is a spare file, place it in swappable target status before using it. A spare file is a system log file whose entity has been created but it is not used because `ONL` is not specified in the `pdlogadfg` operand in the server definition.

To use a spare file, specify `Y` in the `pd_log_rerun_reserved_file_open` operand in the server definition.

Hint:

Before executing this procedure, ensure that the spare file is not currently in use. The `pdlogls` command can be used to determine whether or not a specified spare file is in use.

pdlogls command execution results (spare file not in use)

```
pdlogls -d sys -s bes1

HOSTNAME : dcm3500(163541)
***** Off-line Information *****
Group    Type Server   Gen No.  Status    Run ID      Block No.
logfg01  sys  bes1     9        cna---u   364a4ac2    101      106
logfg02  sys  bes1     a        cna---u   364a4ac2    107      109
logfg03  sys  bes1     b        cna---u   364a4ac2    10a      10c
logfg04  sys  bes1     c        cna---u   364a4ac2    10d      10e
logfg05  sys  bes1     d        cna---u   364a4ac2    10f      110
logfg06  sys  bes1     e        cn---cu   364a4ac2    111      0
logfg07  sys  bes1     f        cn----u   36491223    1f0      201
logfg08  sys  bes1     0        cn-----   00000000    0        0
```

Explanation

- `logfg07` has a different Run ID than other system log files. This means that it is a spare file that is currently not in use, although it was previously used.
- Because `logfg08` has Run ID 0, it is a spare file that has never been used.

pdlogls command execution results (spare file in use)

```
pdlogls -d sys -s bes1

HOSTNAME : dcm3500(163541)
***** Off-line Information *****
Group    Type Server   Gen No.  Status    Run ID      Block No.
logfg01  sys  bes1     9        cna---u   364a4ac2    101      106
logfg02  sys  bes1     a        cna---u   364a4ac2    107      109
logfg03  sys  bes1     b        cna---u   364a4ac2    10a      10c
logfg04  sys  bes1     c        cna---u   364a4ac2    10d      10e
logfg05  sys  bes1     d        cna---u   364a4ac2    10f      110
logfg06  sys  bes1     e        cn---cu   364a4ac2    111      0
logfg07  sys  bes1     7        cnu---u   364a4ac2    e3       ef
logfg08  sys  bes1     8        cne---u   364a4ac2    f0       100
```

Explanation

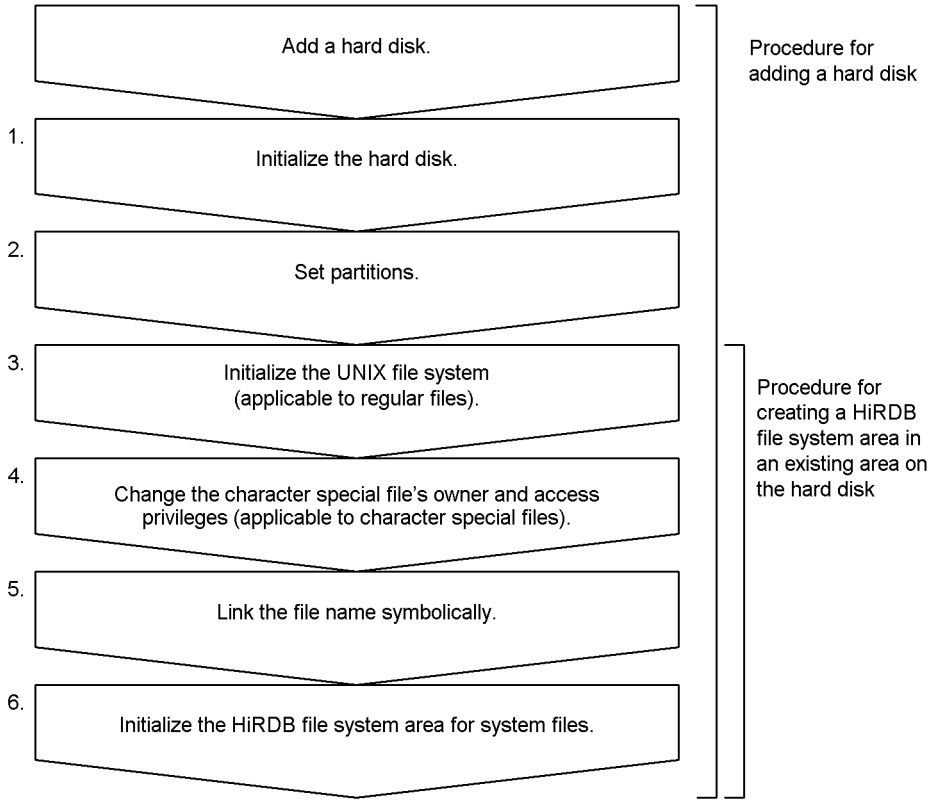
Because both `logfg07` and `logfg08` have the same Run ID as the other system log files and their generation numbers are consecutive with the other system log

files, they are currently in use.

18.17.4 Creating a HiRDB file system area for system files

Figure 18-4 shows the procedure for creating a HiRDB file system area for system files.

Figure 18-4: Procedure for creating a HiRDB file system area for system files



Note

The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 3 is explained in paragraph (3) below.

(1) Initialize the hard disk

Executor: Superuser

Initialize the hard disk.

Initialize the hard disk; for the procedure, see the OS manual.

(2) Set partitions

Executor: Superuser

Set partitions on the initialized hard disk.

Set partitions on the initialized hard disk; for the procedure, see the OS manual.

(3) Initialize the UNIX file system (applicable to regular files)

Executor: Superuser

If regular files had been used in the HiRDB file system area, initialize the partitions as a UNIX file system; for the procedure, see the OS manual.

If the partitions are already initialized, skip this step.

(4) Change each character special file's owner and access privileges (applicable to character special files)

Executor: Superuser

Change the owner and access privileges of the HiRDB file system area, so it will be protected from unauthorized accesses. Table 18-20 shows the owner and access privileges to be set for the HiRDB file system area.

Table 18-20: Owner and access privileges to be set for HiRDB file system area (HiRDB file system area for system files)

Owner, access privileges		Information to be set	Command to be executed*
Owner	User ID	HiRDB administrator	`chown` command
	Group ID	HiRDB group	`chgrp` command
Access privilege	Owner	rw- (read and write operations permitted)	`chmod` command
	Group	rw- (read and write operations permitted)	
	Other	--- (access denied)	

* These are OS commands; for details, see the OS manual.

(5) Link the file names symbolically

Executor: Superuser

For a HiRDB file system area, a name linked symbolically to its entity by the `ln` OS command should be used instead of using the name of its character special file or regular file entity as is. When a symbolically linked name is used, the HiRDB file

system area can be restored easily onto another hard disk in the event of a hard disk failure.

For details on the `ln` command, see the OS manual.

(6) Initialize the HiRDB file system area for system files

Executor: HiRDB administrator

Use the `pdfmkfs` command to initialize the HiRDB file system area for system files:

```
pdfmkfs -n 40 -l 5 -k SYS /sysfile_c
```

Explanation

`-n`: Specifies the size of the HiRDB file system area. Set the size of the HiRDB file system area to be initialized so that it does not exceed the partition size. If it exceeds the partition size, the partitions physically following the HiRDB file system area may be damaged.

`-l`: Specifies the maximum number of files that can be created in the HiRDB file system area.

`-k SYS`: Specifies that this is a HiRDB file system area for system files.

`/sysfile_c`: Specifies a name for the HiRDB file system area.

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

Duplicating the HiRDB file system area using a mirror disk

If the HiRDB file system area is to be duplicated using a mirror disk, do the following:

- **3050RX group or 3500/3xx**

  In the `pdfmkfs` command, specify the name of the character special file on the master disk (`/dev/rdsk/rdsk`*xxx*) or the name linked to it symbolically.

- **3500 series (except 3500/3xx)**

  In the `pdfmkfs` command, specify the mirror special file name (`/dev/mirror/rdsk`*xxx*) or the name linked to it symbolically.

18.17.5 Determining the number of system log files to be used as input files during restart

This section shows the procedure for determining the number of system log files to be

used as input files during a restart when this information cannot be obtained from the KFPS01229-I message. The procedure is shown below.

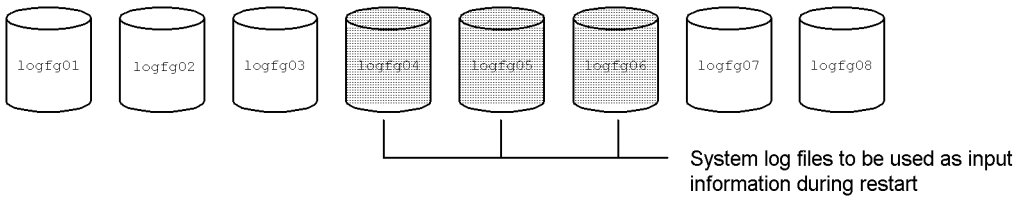
Contents of the syslogfile

```
KFPS01220-E PRDT untF Request to swap sys(bes1) log file unable to
be executed because there is no standby log file group
available.(13830)     …1

KFPO00105-E PRDT untF Server _log1s(process ID=13830) killed by code=
Psjnf07(13830)
KFPS01821-E PRDT untF Unable to continue HiRDB unit processing because
serious error occurred; stops HiRDB unit untF (13776)
KFPS01800-I PRDT untF Now starting HiRDB unit untF (18534)     …2

KFPS01262-I PRDT untF Log block reading started. type:sys(bes1), read
start point:logfg04,4,d (18641)     …3

KFPS01182-I PRDT untF Generation file groups changed for further reading
of log blocks. type:sys(bes1), from:logfg04,4, to:logfg05,5,read
direction:f (18641)
KFPS01182-I PRDT untF Generation file groups changed for further reading
of log blocks. type:sys(bes1), from:logfg05,5, to:logfg06,6,read
direction:f (18641)
KFPS01263-I PRDT untF Log block reading completed. type:sys(bes1),read
end point:logfg06,6,11c (18641)

KFPS01220-E PRDT untF Request to swap sys(bes1) log file unable to
be executed because there is no standby log file group
available.(18641)     …4

KFPO00105-E PRDT untF Server _log1s(process ID=18641) killed by code=
Psjnf07(18641)
```

Explanation

1. This message indicates the first space shortage in the system log file (during online operation).
2. This message indicates the first restart processing.

   If pd_mode_conf=AUTO or pd_mode_conf=MANUAL1 is specified, automatic restart processing was executed several times after the unit terminated abnormally, in which case the messages indicated as 3 were output each time. The messages that were output during the first restart processing are the ones that must be used.
3. The KFPS01262-I, KFPS01182-I, and KFPS01263-I messages indicate that logfg04, logfg05, and logfg06 are the system log files to be used as the input files during the restart. Therefore, the number of system log files to be used as the input information during the restart is 3.



4. This message indicates a space shortage in the system log file that occurred during restart processing.

Note:

If `pd_mode_conf=MANUAL2` is specified, restart HiRDB with the `pdstart` command in order to output these messages.

18.17.6 Checking for synchronization point dump validation

This section explains the procedure for checking for synchronization point dump validation when the `KFPS02183-I` message cannot be used for this verification due to a problem with respect to the HiRDB version, etc. This procedure uses the `pdlogls` command to check for status changes in system log files immediately after restart and after synchronization point dump validation.

pdlogls command execution results (immediately after restart)

```
pdlogls -d sys -s bes1

HOSTNAME : dcm3500(170302)
Group    Type Server   Gen No.  Status    Run ID       Block No.
logfg01  sys  bes1     11       oc-d--u   365d303d     93       97
logfg02  sys  bes1     b        os----u   365d303d     66       66
logfg03  sys  bes1     c        os----u   365d303d     67       68
logfg04  sys  bes1     d        osud--u   365d303d     69       73
logfg05  sys  bes1     e        osud--u   365d303d     74       7a
logfg06  sys  bes1     f        osud--u   365d303d     7b       82
logfg07  sys  bes1     10       osudb-u   365d303d     83       92
logfg08  sys  bes1     0        cn-----   00000000     0        0
```

Explanation

These execution results indicate that the files in overwrite disabled status in existence immediately after the restart are `logfg01` and `logfg04` to `logfg07`.

pdlogls command execution results (after synchronization point dump validation)

```
pdlogls -d sys -s bes1

HOSTNAME : dcm3500(170424)
Group    Type Server   Gen No.  Status     Run ID       Block No.
logfg01  sys  bes1     11       osu---u    365d303d     93       99
logfg02  sys  bes1     12       osu---u    365d303d     9a       a2
logfg03  sys  bes1     13       osu---u    365d303d     a3       a9
logfg04  sys  bes1     d        osu---u    365d303d     69       73
logfg05  sys  bes1     e        osu---u    365d303d     74       7a
logfg06  sys  bes1     f        osu---u    365d303d     7b       82
logfg07  sys  bes1     10       osu---u    365d303d     83       92
logfg08  sys  bes1     14       oc-d--u    365d303d     aa       af
```

Explanation

These execution results indicate that the file in overwrite disabled status in existence after synchronization point dump validation is `logfg08`. It is clear that the synchronization point dump was validated because the file in overwrite disabled status followed the system log input start point during restart.

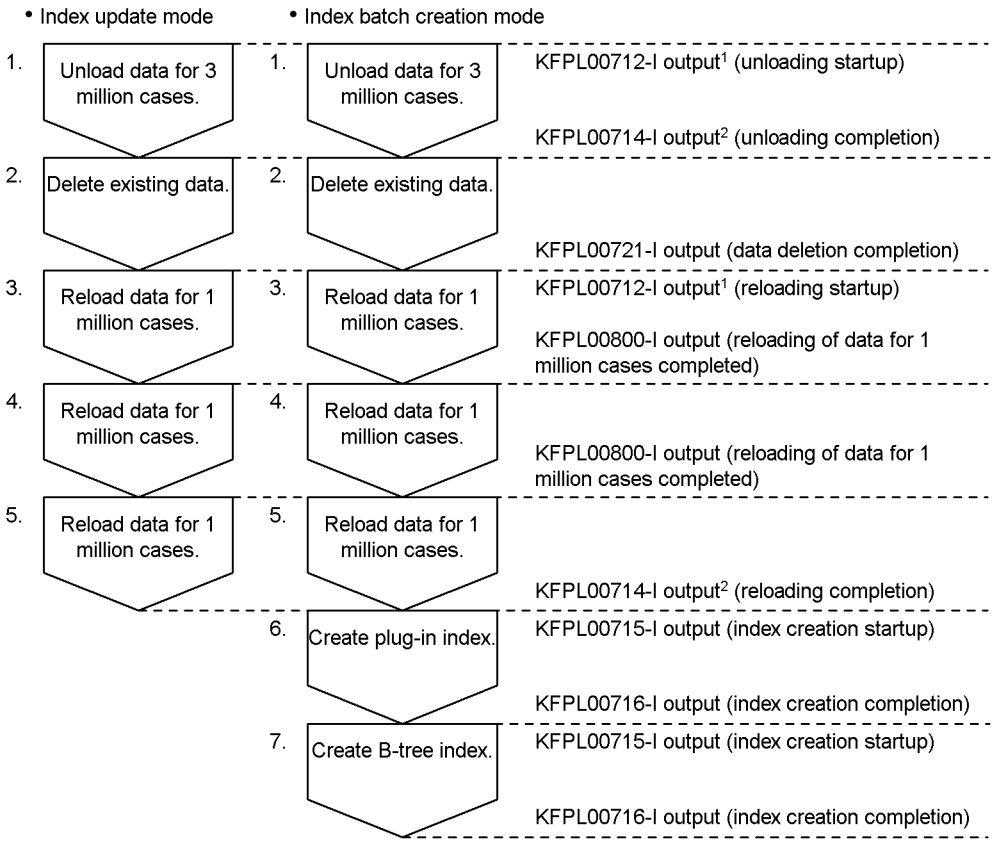
18.18 When a utility terminates abnormally during execution of a reorganization with synchronization points set

This section explains the actions to be taken when a utility terminates abnormally during execution of reorganization with synchronization points set.

18.18.1 Overview of actions

The actions to be taken depend on the timing of the abnormal termination. Figure 18-5 shows the actions to be taken when a utility terminates abnormally during execution of a reorganization with synchronization points set.

Figure 18-5: Actions to be taken when a utility terminates abnormally during execution of a reorganization with synchronization points set



Note: Steps 6 and 7 may be performed in either order.

[1] When the `-g` option is specified for execution of the database reorganization utility, `KFPL00732-I` is output.

[2] When the `-g` option is specified for execution of the database reorganization utility, `KFPL00733-I` is output.

Explanation

- The total number of data cases is 3 million; the number of synchronization points is 1 million.
- If the utility terminates abnormally at point 1, re-execute table reorganization.
- If the utility terminates abnormally at a point between 2 and 5, unloading has been completed. Therefore, use the database reorganization utility (`-k reld`) to execute reloading.
- If the utility terminates abnormally at point 6, reloading of table data has been completed. Therefore, create only the plug-in index and the B-tree index. Using the created index information file as the input information, use the database reorganization utility to create the plug-in index and B-tree index in the batch mode (`-k ixmk`).
- If the utility terminates abnormally at point 7, create only the B-tree index. Using the created index information file as the input information, use the database reorganization utility to create the B-tree index in the batch mode (`-k ixmk`).

18.18.2 Example

The database reorganization utility terminated abnormally during reorganization of a table containing 3 million data cases. The number of synchronization points is 1 million.

(1) Check for messages

The following messages have been output:

```
KFPL00714-I  unload ended, table=USR01.TABLE1, server=sds01, return code=0

KFPL00800-I Loading until 2000000th row committed
```

Explanation

- Message `KFPL00714-I` indicates that unloading has been completed.
- Message `KFPL00800-I` indicates that reloading of up to 2 million cases has been completed.

(2) Use the pdrorg command to execute reloading

Use the database reorganization utility to reload the remaining 1 million cases and create the index.

```
pdrorg -k reld -t TABLE1 /pdrorg/rorg01
```

Explanation

Change the specification of the `-k` option only; change it from `rorg` to `reld`. The specifications of other options need not be modified.

(3) Check for messages

Determine whether or not the database reorganization utility terminated normally by checking for the `KFPL00719-I` message.

```
KFPL00719-I  Pdrorg terminated, return code=0
```

(4) Back up the RDAREAs that were reloaded

Because you reloaded the RDAREAs in the pre-update log acquisition mode (default), back up the RDAREAs that were reloaded. For details about backing up RDAREAs, see *6.4.6 Example 6 (Backing up RDAREAs)*.

(5) Use the pdrels command to release RDAREAs in shutdown status

Use the `pdrels` command to release the table-storage and index-storage RDAREAs that are in shutdown status.

```
pdrels -r RDAREA1,RDAREA2,RDAREA3
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference*.

18.18.3 Actions to be taken when a utility terminates abnormally before unload data files have been consolidated (HiRDB/Parallel Server only)

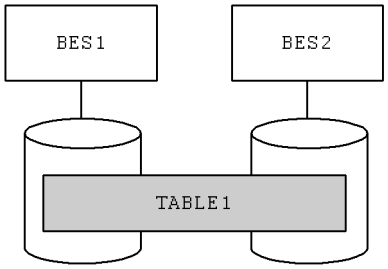
When a table is row-partitioned into multiple back-end servers, unload data file consolidation (`-g` specification for the database reorganization utility) is recommended. If the unload data files are not consolidated, the actions to be taken when the utility terminates abnormally become complicated.

This section explains the actions to be taken when a utility terminates abnormally during reorganization or reloading with synchronization points set before the unload data files have been consolidated.

(1) Overview of actions to be taken

If the unload data files have not been consolidated, it is necessary to check whether or not reorganization processing was completed at each server. Figure 18-6 shows the system configuration used for the explanation.

Figure 18-6: Case where the table is row-partitioned into multiple back-end servers



BES: Back-end server

Explanation

If the unload data files are not consolidated, reorganization at `BES1` may terminate abnormally during unloading, and reorganization in `BES2` may terminate abnormally during reloading. In such a case, it is necessary to modify the specification in the control information file for the utility. The utility must also be re-executed for each server. In this example, the utility must be executed twice.

As explained above, the database reorganization utility must be re-executed individually for each server taking into consideration the timing of abnormal termination.

(2) Example

Reorganization with synchronization points set was executed without consolidating the unload data files. The system configuration is as shown in Figure 18-6. It is assumed that reorganization in `BES1` terminated abnormally during unloading and reorganization in `BES2` terminated abnormally during reloading.

(a) Specifications for the database reorganization utility when it terminated abnormally

The control information file of the database reorganization utility and the command specification when the utility terminated abnormally are shown below:

Control information file (/pdrorg/rorg01) specification

```
option job=j01,200
unload BES1:/unload/file01
unload BES2:/unload/file02
```

pdrorg command specification

```
pdrorg -k rorg -t TABLE1 /pdrorg/rorg01
```

(b) Specifications for the database reorganization utility for re-execution

Reorganization must be re-executed for the tables in BES1; reloading must be executed for the tables in BES2. Therefore, the database reorganization utility must be executed individually for BES1 and BES2.

The control information file of the database reorganization utility and the command specification for re-executing the utility are shown below.

Control information file (/pdrorg/rorg01) specification

```
option job=j01,200
unload BES1:/unload/file01
```

Control information file (/pdrorg/rorg02) specification

```
option job=j01,200
unload BES2:/unload/file02
```

pdrorg command specification

```
pdrorg -k rorg -t TABLE1 /pdrorg/rorg01
pdrorg -k reld -t TABLE1 /pdrorg/rorg02
```

18.18.4 Notes

1. If RDAREA re-initialization or the PURGE TABLE statement is executed before re-execution of the database reorganization utility, the synchronization point information that was set may be lost. In such a case, the re-execution will not be treated as re-execution of reorganization with synchronization points set.

2. If the utility terminates abnormally during reorganization with synchronization points set for a LOB column structure base table, specify `CLR` when the utility is re-executed. If `CLR` is not specified, retrieval efficiency for the LOB column may deteriorate even after reorganization.
3. If names for the files listed below are not stated clearly, the database reorganization utility will create these files automatically:
   - Index information file
   - Error information file

   If the database reorganization utility terminates abnormally, these files will remain on the disk. Because the database reorganization utility will create files under new names during re-execution, these files remaining from before the abnormal termination may cause a space shortage on the disk. Therefore, delete these files before re-execution.

18.19 Actions when page destruction in an RDAREA is detected

Executor: HiRDB administrator

This section explains the actions to be taken when RDAREA page destruction has occurred (KFPH00308-E message is output), resulting in an RDAREA error shutdown (KFPH00306-E message is output).

18.19.1 Causes of page destruction

A page may be destroyed in the following cases:

- HiRDB was started forcibly after it had terminated abnormally or had been terminated forcibly.
- HiRDB was started by initializing the status files after HiRDB had terminated abnormally or had been terminated forcibly.
- When RDAREAs were recovered during backup acquisition, not all related RDAREAs were recovered. For example, when a table-storage RDAREA was recovered during backup acquisition, an index-storage RDAREA was missed during recovery processing.

18.19.2 Actions to be taken

The actions to be taken are described below.

Procedure

To recover from an RDAREA shutdown:

1. Back up the troubleshooting information under $PDDIR/spool.
2. Back up the contents of the entire syslogfile before the error occurrence.
3. Use the pdclose command to shut down RDAREAs that are in error shutdown status and their related RDAREAs (related table and index RDAREAs).
4. Use the pdfbkup command to back up the files comprising the RDAREAs that are in error shutdown status and their related RDAREAs.
5. After the above operations have been performed, restore the RDAREAs.

   There are two methods for restoring RDAREAs, as explained below.

(1) Restoration method 1

If a backup for the system is available, restore the entire system during backup acquisition.

(2) Restoration method 2

Use the database structure modification utility (`pdmod` command) to reinitialize the RDAREAs that went into error shutdown status. This method deletes all data in the reinitialized RDAREAs. Restore the data from unload data files, input data files, etc.

The RDAREAs related to the reinitialized RDAREAs must also be restored logically.

Note that if HiRDB has not been restarted, the same error may have occurred in other RDAREAs.

18.20 Actions to take when an RDAREA I/O error occurs

Executor: HiRDB administrator

This section explains the actions to take when an RDAREA I/O error occurs. As discussed here, *I/O error* means a failure during HiRDB file I/O operations for which HiRDB cannot determine the reason. The error code returned in response to the request to access the HiRDB file system in such a case is `-1544`.

(1) HiRDB processing when an RDAREA I/O error occurs

Table 18-21 describes the processing performed by HiRDB when an RDAREA I/O error occurs.

Table 18-21: HiRDB processing when an RDAREA I/O error occurs

Type of RDAREA	HiRDB processing	
	pd_db_io_error_action=dbhold (default)	pd_db_io_error_action=unitdown
Master directory RDAREA	HiRDB (unit for a HiRDB/Parallel Server) terminates abnormally.	
Other RDAREAs	The RDAREA is placed in error shutdown status	HiRDB (unit for a HiRDB/Parallel Server) terminates abnormally. The RDAREA is not placed in error-shutdown status, unless the I/O error recurs after the abnormal termination.

(2) When pd_db_io_error_action=unitdown (HiRDB abnormal termination) is specified

The following advantages are provided by configuring HiRDB to terminate abnormally when an RDAREA I/O error occurs:

- If the error is caused by a path error, operations can be resumed upon system switchover.
- The need to recover the RDAREA is eliminated.

(a) If the error is caused by a path error, operations can be resumed upon system switchover

Because HiRDB terminates abnormally when an RDAREA I/O error occurs, the system is switched over. If the cause of the I/O error is a path error, I/O processing can be performed after the system has been switched over, which means operations can resume from that point. As discussed here, *path error* means that HiRDB cannot access a file because the communication path between HiRDB and the file has been disrupted

for some reason.

(b) The need to recover the RDAREA is eliminated

When an RDAREA I/O error occurs, HiRDB terminates abnormally without the RDAREA being placed in error shutdown status. This allows you to eliminate the cause of the I/O error before you restart HiRDB. Once you have eliminated the cause of the error, processing resumes when HiRDB is restarted. The fact that the RDAREA is not placed in error shutdown status in this case eliminates the need for you to recover it, unless, however, the disk needs to be replaced due to a disk error. In such a case, you will need to use the database recover utility to recover the RDAREA.

(3) Environment settings

To configure HiRDB to terminate abnormally when an RDAREA I/O error occurs, you must make the environment settings explained below.

(a) pd_db_io_error_action operand specification

Specify `unitdown` in the `pd_db_io_error_action` operand.

(b) If the system switchover facility is being used

If you are using the system switchover facility, see *25.20.2 RDAREA I/O error (path error) has occurred*.

(c) pd_mode_conf operand specification (specify only if the system switchover facility is not being used)

To configure HiRDB so that it will not restart automatically after it terminates abnormally, specify `MANUAL2` in the `pd_mode_conf` operand. If you specify any value other than `MANUAL2`, HiRDB will restart automatically after it terminates abnormally. When you have specified `MANUAL2` and an I/O error occurs in the same RDAREA, that RDAREA will be placed in error shutdown status. This means that the `pd_db_io_error_action=unitdown` specification is ignored.

(d) Relationships with other functions

If you use functionality provided by the OS or the device driver* to specify the time until a physical volume or logical volume I/O error is detected, you must take into account the following operands:

- `pd_utl_exec_time` operand
- `pd_watch_time` operand
- `exectime` operand of the `option` statement of the `pdload`, `pdrorg`, and `pdreclaim` commands
- `PDCWAITTIME` and `PDCWAITTIME` operands of the client environment variables

These operands are used for monitoring execution times. If the time specified with an

OS or device driver function is longer than the time specified in these operands, the functionality provided by these operands will activate before the I/O error is detected. To avoid such an occurrence, you must increase the monitoring times specified with these operands.

* An example of such a function is the `-t` option of the `pvchange` or `lvchange` command in HP-UX. For details about such functions, see the operating system or device driver documentation.

(4) Operating method

This section explains operations when `pd_db_io_error_action=unitdown` is specified.

(a) Operations when the system switchover facility is being used

If you are using the system switchover facility, see *25.20.2 RDAREA I/O error (path error) has occurred*.

(b) Operations when the system switchover facility is not being used

If HiRDB terminates abnormally because an I/O error occurred, eliminate the error based on the message that is issued. Restart HiRDB after eliminating the error. Note that if the I/O error recurs after HiRDB is restarted, the RDAREA is placed in error shutdown status. In this case, use the database recovery utility to recover the RDAREA.

(c) Actions to take after HiRDB has terminated abnormally (for both (a) and (b))

If HiRDB terminates abnormally because an I/O error occurred, the `pd_db_io_error_action=unitdown` specification becomes invalid from that point (RDAREAs will be placed in error shutdown status). To re-enable the `pd_db_io_error_action=unitdown` specification, use one of the following methods:

- Start HiRDB normally
- Copy the HiRDB system definition files into the `$PDDIR/conf/chgconf` directory, and then execute the system reconfiguration command (`pdchgconf` command).

(5) Notes

- Because HiRDB terminates abnormally when an I/O error occurs when `pd_db_io_error_action=unitdown` is specified, if a UAP or a utility is executing in the pre-update log acquisition mode or the no-log mode, the RDAREA being processed may be placed in error shutdown status.
- If an I/O error occurs during startup or termination, HiRDB does not terminate abnormally even if `pd_db_io_error_action=unitdown` is specified.

18.21 Checking the transaction completion type when an error occurs during commit processing (HiRDB/Parallel Server)

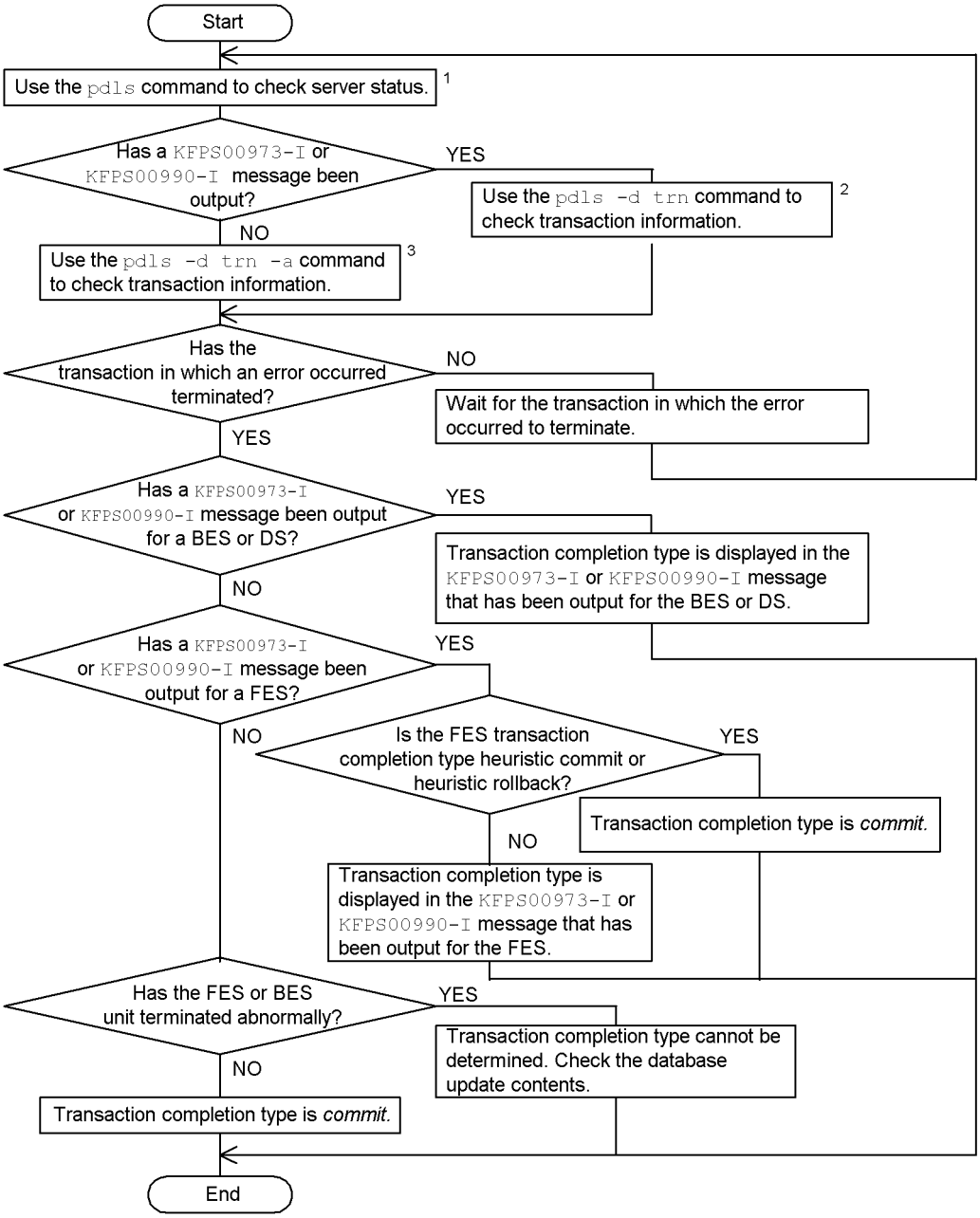
Executor: HiRDB administrator

This section explains the method of checking the transaction completion type when an error occurs during commit processing.

When an error occurs during transaction execution, HiRDB performs commit or rollback processing for each server. When this processing is completed, HiRDB sets information about each transaction and the transaction completion type in a `KFPS00990-I` message that it outputs to the transaction's server. If an error occurred that caused a transaction to roll back on its own, HiRDB sets the transaction information and transaction completion type in a `KFPS00973-I` message that it outputs to the server in which updating branching occurred.

Figure 18-7 shows how to check the transaction completion type when an error has occurred. You use this procedure also when a commit error is returned for a UAP.

Figure 18-7: Checking the transaction completion type when an error has occurred



[1] Use the `pdls` command to check if all servers (excluding recovery-unnecessary front-end servers) are running.

[2] Compare the following information:

- Transaction identifiers that are output in the execution results of the `pdls -d trn` command
- Transaction identifiers that are output in the `KFPS00973-I` or `KFPS00990-I` messages

Check if any transaction identifier is output to both. If any transaction's identifier is output to both, wait for that transaction to terminate.

[3] Compare the following information:

- Client process ID, IP address, and UAP identification name output in the execution results of the `pdls -d trn` command
- Process ID, IP address, and UAP identification name of a UAP for which a commit error occurred

Check if there is any transaction to which both of these apply.

When there is a transaction for which there is a match in all three items of information

Wait for that transaction to terminate.

When there is a transaction for which there is a match in only the UAP identification name, and neither the process ID nor the IP address is displayed

Take action by following the procedure below.

1. Check the transaction identifier of the applicable transaction.
2. Referring to the execution results of the `pdls -d trn -a` command, check if the transaction identifier found in step 1 is also displayed elsewhere.
3. If it is also displayed elsewhere, check if the client process ID and IP address of that transaction are displayed. If no transaction is displayed, wait for transactions to terminate.

Table 18-22 shows the transaction completion types when an error occurs during commit processing. Note that the transaction completion types in the table apply only in cases in which the front-end server or back-end server unit did not terminate abnormally.

Table 18-22: Transaction completion types when an error occurred during commit processing

Item	Error timing	Front-end server		Back-end or dictionary server		Transaction completion type
		KFPS00973-I or KPFS00990-I message	Completion type that is output in message	KFPS00973-I or KPFS00990-I message	Completion type that is output in message	
1	During commit processing	Yes	Commit	Yes	Commit	Commit
2					Rollback	Rollback
3				No	—	Commit
4			Rollback	Yes	Commit	Commit
5					Rollback	Rollback
6				No	—	Rollback
7		No	—	Yes	Commit	Commit
8					Rollback	Rollback
9				No	—	Commit
10	Before transaction completion	Yes	Rollback	Yes	Rollback	Rollback
11				No	—	Rollback
12		No		Yes	Rollback	Rollback
13				No	—	—

Legend:

—: Not applicable

18.22 Actions to take when an error occurs while a local buffer is being used to update a shared table (HiRDB/Parallel Server only)

Executor: HiRDB administrator

You must apply LOCK TABLE to use a local buffer to update a shared table. If the server process terminates abnormally when both of the conditions listed below are applicable, abort code Phb3008 is issued and the unit may terminate abnormally:

- A local buffer is being used.
- A shared table is being updated without applying LOCK TABLE.

When both of these conditions are applicable and an updated page exists at the time the server process terminates abnormally, recovery processing may not be possible by means of rollback processing. If this happens, recovery processing is performed when the unit is restarted. Table 18-23 shows the HiRDB processing and the actions to take if an error occurs when a local buffer is being used to update a shared table (without the LOCK TABLE specification).

Table 18-23: HiRDB processing and actions to take if an error occurs when a local buffer is being used to update a shared table (without LOCK TABLE specification)

<table>
<tr><th colspan="2">Cause of abnormal termination of server process</th><th>HiRDB processing if an updated page exists when the server process terminates abnormally</th><th>Action to be taken by the HiRDB administrator</th></tr>
<tr><td colspan="2">PDSWAITTIME timeout</td><td rowspan="5">Abort code Phb3008 is issued, and the unit may terminate abnormally.</td><td rowspan="5">Restart the unit if it terminated abnormally.</td></tr>
<tr><td colspan="2">PDCWAITTIME timeout</td></tr>
<tr><td colspan="2">pdcancel command executed</td></tr>
<tr><td>Abort</td><td>Abort that occurs due to detection of a HiRDB conflict.</td></tr>
<tr><td>Other</td><td>Any unexpected error, such as SIGSEGV, SIGBUS, reception of an external signal, exi, etc.</td></tr>
</table>

Note

HiRDB does not terminate abnormally when a server process terminates abnormally due to a PDSWATCHTIME timeout, even if an updated page exists.

18.23 Actions to take when an error occurs in the system manager unit

If the system manager unit stops because of an error, users cannot connect to HiRDB servers. If the error in the system manager unit cannot be corrected immediately, you must change the specification of the PDHOST operand in the client environment definition. When the specification of the PDHOST operand is changed as described below, users can connect to the HiRDB servers.

- When the PDFESHOST operand in the client environment definition is specified

  Specify also in the PDHOST operand the name of the host (FQDN or IP address) specified in the PDFESHOST operand.

- When the PDFESHOST operand in the client environment definition is not specified

  Specify the PDFESHOST operand and also specify in the PDHOST operand the same host name (FQDN or IP address).

For details about the PDHOST and PDFESHOST operands, see the manual *HiRDB Version 8 UAP Development Guide*.

18.24 Actions to take when a mismatch occurs between the original and the mirror duplicate

When a database is mirrored using the mirroring facility of LVM or the device driver, the duplicate volume may not match the original volume if an error occurs in the OS or server machine while the database is being copied or if the device driver shuts down input/output during a system switchover. This is called a *mismatch between the original and the mirror duplicate*. When it occurs, you use one of the following methods to eliminate the mismatch:

1. Use a facility of LVM or the device driver to eliminate the mismatch between the original and the mirror duplicate
2. Use the full recovery processing performed by HiRDB during a restart to eliminate the mismatch between the original and the mirror duplicate

Method 1 is usually used. You use method 2 when it is not possible to use the LVM or device driver's facility to eliminate the mismatch, or when the process of matching the original and duplicate volumes takes too long and, as a result, a system requirement, such as system switchover time, cannot be satisfied.

(1) Using a facility of LVM or the device driver to eliminate the mismatch between the original and the mirror duplicate

Preparation

Either specify `N` in the `pd_redo_allpage_put` operand or omit this operand.

Action to take when a mismatch occurs between the original and the mirror duplicate

Before restarting HiRDB, use the facility of LVM or the device driver to make the original and duplicate volumes match. Then restart HiRDB.

(2) Using the full recovery processing performed by HiRDB during a restart to eliminate the mismatch between the original and the mirror duplicate

Preparation

Specify `Y` in the `pd_redo_allpage_put` operand. When `Y` is specified, the full recovery processing performed by HiRDB during a restart writes into the database all pages that were updated at or subsequent to a synchronization point. This processing eliminates the mismatch between the original and the mirror duplicate.

Action to take when a mismatch occurs between the original and the mirror duplicate

There is no need to use the LVM or device driver's facility to make the original

and duplicate volumes match. Restart HiRDB immediately. The full recovery processing performed by HiRDB during the restart will make the original and duplicate volumes match.

Note:

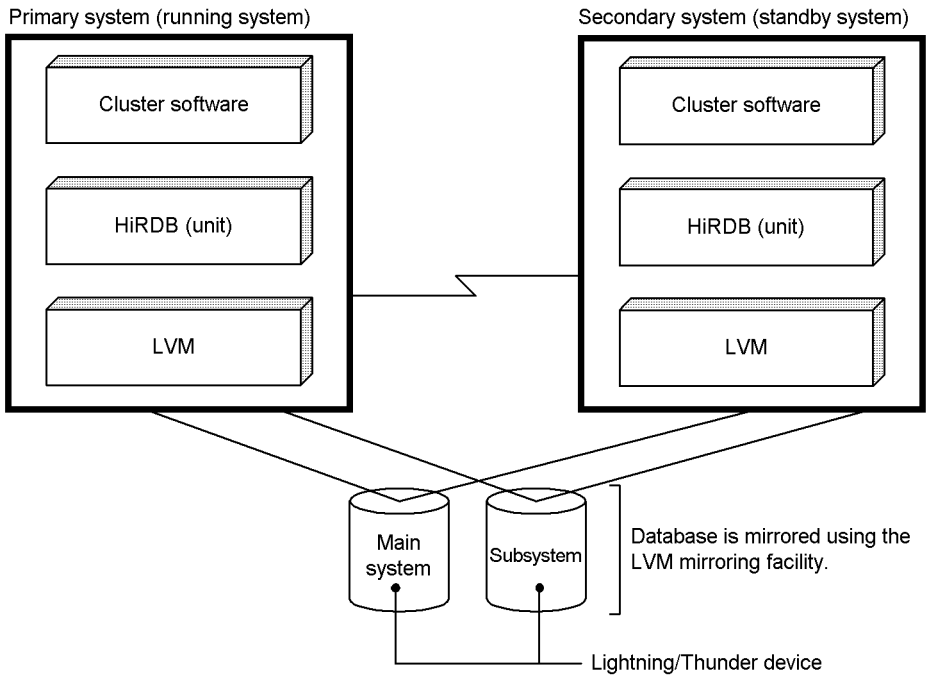
Use of this method increases the volume of data written into the database during full recovery processing. Consequently, it will take longer to restart HiRDB than when method 1 is used. To reduce the volume of data that must be written into the database, it is recommended that you shorten the synchronization point acquisition interval. Use the `pd_log_sdinterval` operand to specify the synchronization point acquisition interval.

(3) Configuration example when the rapid system switchover facility is used

Figure 18-8 shows an example of a system configuration when the rapid system switchover facility is used in an environment in which the LVM mirroring facility is used to mirror the database.

In this example, the full recovery processing performed by HiRDB during a restart is used to eliminate a mismatch between the original and the mirror duplicate.

Figure 18-8: Example of a system configuration in which the rapid system switchover facility is used



Explanation

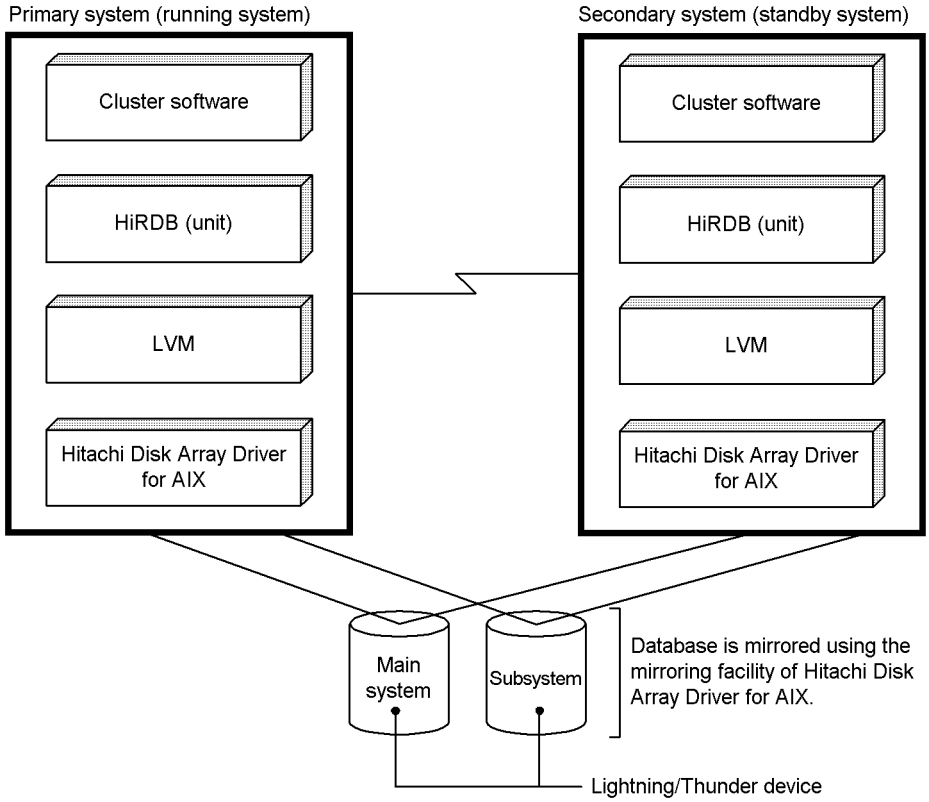
- In a system configuration in which the rapid system switchover facility is used, position a Lightning/Thunder device so that it can be accessed from both the primary and secondary systems.
- Specify `Y` in the `pd_redo_allpage_put` operand.

(4) Configuration example when the rapid system switchover facility is used (for AIX 5L V5.2 or later)

Figure 18-9 shows an example of a system configuration when the rapid system switchover facility is used in an environment in which the mirroring facility of Hitachi Disk Array Driver for AIX is used to mirror the database.

In this example, the full recovery processing performed by HiRDB during a restart is used to eliminate a mismatch between the original and the mirror duplicate.

Figure 18-9: Example of a system configuration in which the rapid system switchover facility is used (for AIX 5L V5.2 or later)

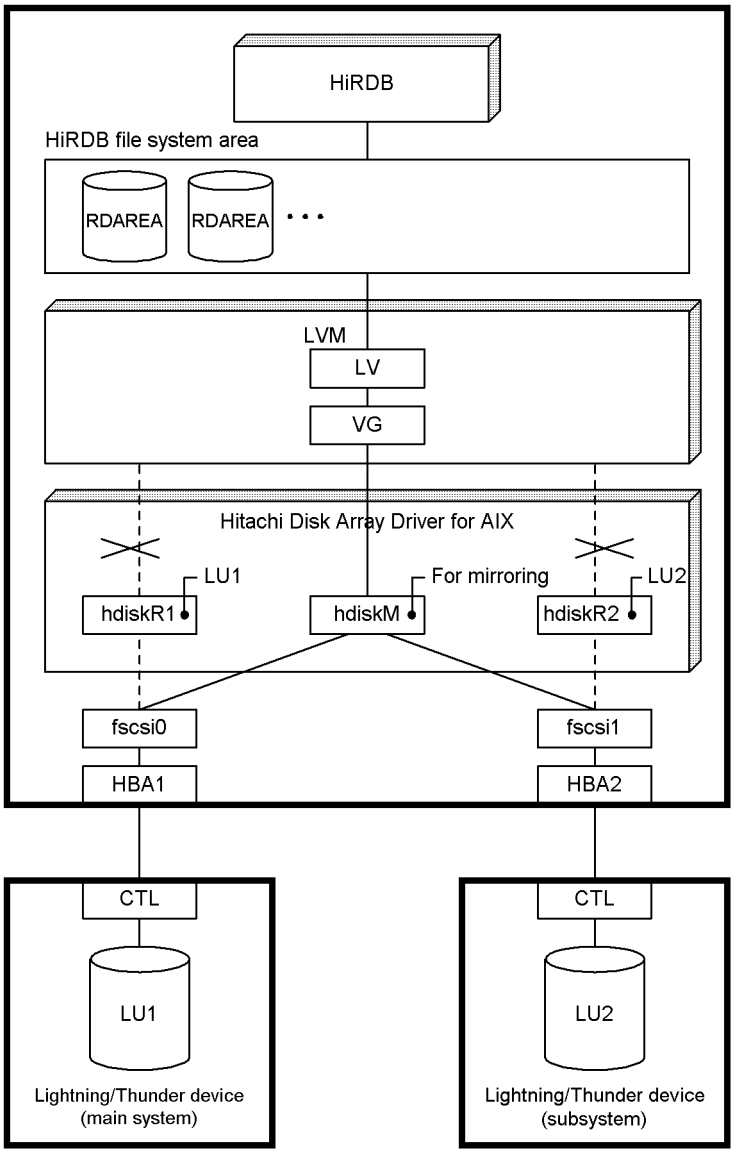


Explanation

- In a system configuration in which the rapid system switchover facility is used, position a Lightning/Thunder device so that it can be accessed from both the primary and secondary systems.
- Specify `Y` in the `pd_redo_allpage_put` operand.

Also use the mirroring facility of Hitachi Disk Array Driver for AIX to mirror the disk in which the HiRDB file system area for RDAREAs is located, as shown in Figure 18-10. Mirror both the primary and secondary systems.

Figure 18-10: Example of a disk configuration in which the system switchover facility is used (for AIX 5L V5.2 or later)



Legend:

HBA: Any of various types of physical adapter cards. A Fibre Channel Adapter is a type of HBA.

CTL: Lightning/Thunder controller.

LU: A Lightning/Thunder device is logically partitioned into LUs, each of which appears as a single physical disk for a host. The number assigned to an LU is called the LUN.

hdiskM: Pseudo device created for mirroring.

hdiskR1, hdiskR2: Real devices corresponding to LUs in Lighnting/Thunder series devices.

If an error occurs in the OS or server machine while the database is being copied or if system switchover occurs, mismatch may occur between the original and the mirror duplicate. However, the full recovery processing performed by HiRDB during a restart will make the original and the mirror duplicate match in terms of RDAREAs. Thus, there is no need to make the original and duplicate volumes match before restarting HiRDB.

18.25 Recovery of HiRDB directory

Executor: Superuser and HiRDB administrator

This section explains the actions to be taken when it is necessary to recover the HiRDB directory because of a disk error, etc. (HiRDB directory recovery procedure).

18.25.1 When installation directory is available

If a HiRDB installation directory is available, the HiRDB directory can be recovered from the installation directory. Note that the recovery procedure explained here assumes the following:

- The files under `$PDDIR/conf` have been backed up.
- If users have created files under `$PDDIR`, those files have been backed up.
- The database and system files are safe.
- In the case of a multi-HiRDB, all the HiRDB versions are identical.

Procedure

1. Reboot the server machine to clean up the remaining HiRDB processes.
2. Execute the `pdsetup -d` command to delete the HiRDB directory information registered in the OS. Respond with `Y` when prompted. Although the `pdsetup -d` command will terminate in an error, ignore this error.
3. Create a HiRDB directory.
4. Execute the `pdsetup` command. Executing this command copies the files under the installation directory to the HiRDB directory.
5. Recover the files under `$PDDIR/conf` from the backup.
6. If users created files under `$PDDIR`, recover those files.
7. Use the `pdstart` command to start HiRDB.

18.25.2 When installation directory is not available

If the HiRDB installation directory is not available because the HiRDB directory itself was used as the installation directory, HiRDB must be reinstalled. The recovery procedure explained here assumes the following:

- The files under `$PDDIR/conf` have been backed up.
- If users have created files under `$PDDIR`, those files have been backed up.
- The database and system files are safe.
- In the case of a multi-HiRDB, all the HiRDB versions are identical.

Procedure

1. Reboot the server machine to clean up the remaining HiRDB processes.
2. Execute the `pdsetup -d` command to delete the HiRDB directory information registered in the OS. Respond with `Y` when prompted. Although the `pdsetup -d` command will terminate in an error, ignore this error.
3. Reinstall HiRDB. For details on how to install HiRDB, see the manual *HiRDB Version 8 Installation and Design Guide*.
4. Execute the `pdsetup` command.
5. Recover the files under `$PDDIR/conf` from the backup.
6. If users created files under `$PDDIR`, recover those files.
7. Use the `pdstart` command to start HiRDB.

18.25.3 When a backup is available for the disk on which the HiRDB directory is located

If HiRDB cannot be reinstalled because neither the installation directory nor the HiRDB provided medium is available, recover the disk contents (including the HiRDB directory) from the disk backup.

For details on how to back up the disk in which the HiRDB directory is located, see the manual *HiRDB Version 8 Installation and Design Guide*.

(1) When the HiRDB directory is not in the root volume

Procedure

1. Reboot the server machine to clean up the remaining HiRDB processes.
2. Set the OS's Run Level (or Run Mode) to 1 (single user mode).
3. Recover the contents of the disk in which the HiRDB directory is located from the backup.
4. Return the OS's Run Level (or Run Mode) to the original mode.
5. If the `pdsetup -d` command had been executed before the disk was damaged, execute the `pdsetup` command.
6. Use the `pdstart` command to start HiRDB.

(2) When the HiRDB directory is in the root volume

Procedure

1. Reboot the server machine to clean up the remaining HiRDB processes.
2. Recover the root volume from the backup.

3. Execute the `pdsetup` command.
4. Use the `pdstart` command to start HiRDB.

18.26 Handling errors in the HiRDB file system areas

Executor: HiRDB administrator

If HiRDB processing is terminated by the OS's `kill` command or shutdown occurs while HiRDB file creation, deletion, or extension is underway, updating of the management information in the HiRDB file system areas (HiRDB file system configuration and file management information) is interrupted.

This may create files that cannot be managed and areas that cannot be referenced. Although this does not pose a problem for resuming HiRDB processing, the maximum number of files intended when the HiRDB file system area was created (`-l` operand value in the `pdfmkfs` command) or the specified maximum capacity (`-n` option value in the `pdfmkfs` command minus the size of the area management section) may no longer be available. In the case of HiRDB version 07-02 or earlier, the area management information may be corrupted.

This section describes how to respond when unmanageable files and unreferenceable areas area created and when area management information is corrupted.

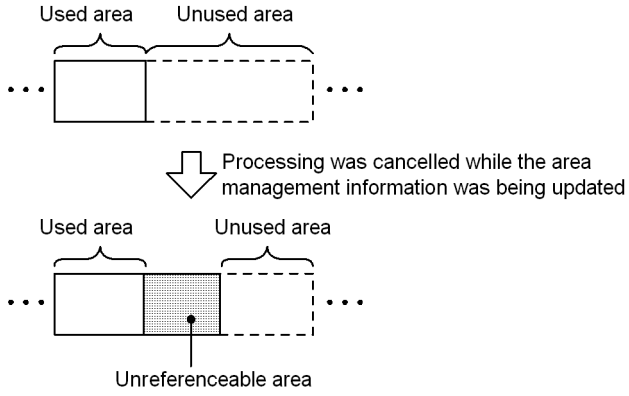
18.26.1 Unmanageable files and unreferenceable areas

(1) About unmanageable files

HiRDB manages the management information for the HiRDB file system areas in two statuses, `Used` and `Unused`, for the maximum number of files that can be created. If file creation or deletion processing is cancelled while management information for the corresponding area is being updated, the file to be processed is not referenced and may become unusable. If this occurs, the number of available files becomes fewer than the defined maximum number of files that can be created. Additionally, the area allocated to the affected file may become unreferenceable, resulting in a reduction of available capacity. If this type of cancellation occurs repeatedly, the available files and file capacity may be reduced considerably.

(2) About unreferenceable areas

Separately from the file management information, HiRDB manages the HiRDB file system areas in two statuses, used area that has been allocated to files, and unused area that has not been allocated. During file extension processing (automatic extension), HiRDB obtains the area required for file extension from the unused area and allocates it as used area. If file extension processing is cancelled while the area management information is being updated, the target area may no longer be referenced as used or unused area, and will not become available thereafter. The following illustrates the concept of unreferenceable area:



This figure shows that the file extension processing was cancelled before the unused area was allocated as used area. The shaded area becomes unreferenceable thereafter, resulting in a reduction of available capacity. If similar processing cancellations occur repeatedly, available capacity may be reduced by a considerable factor.

(3) How to check

The following describes how to check for unmanageable files and unreferenceable areas.

- If you execute the `pdfls` command and the `KFPI21586-W` message is displayed, there are unmanageable files.
- If the available capacity is less than the size specified when the HiRDB file system area was created (`-n` option value in the `pdfmkfs` command minus the size of the area management section), there are unreferenceable areas.
- Execute the `pdffsck -c` command to check the processing results.

(4) How to handle

Execute the `pdffsck` command to repair applicable files and areas.

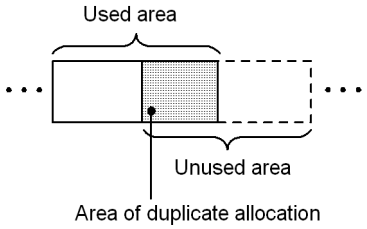
The `pdffsck` command with the `-c` option specified performs only checking and does not perform repair. If you know that there are unmanageable files or unreferenceable areas in the target HiRDB file system area, execute the command without specifying the `-c` option.

18.26.2 Corruption of the area management information (applicable to HiRDB version 07-02 and earlier)

(1) About corruption of the area management information

In HiRDB version 07-02 and earlier, if file creation, deletion, or extension processing is cancelled, some area may fall into a status that belongs neither to unmanageable files

nor to unreferenceable areas. This is the area of corrupted management information and corrupted file caused by operations such as a duplicate allocation of area; this occurence has adverse effects on system operation. The following illustrates the concept of duplicate allocation.



Because the area management information is corrupted, any attempt to recover the corrupted target file results in corruption of other files. If you upgrade the system in this status, the corrupted area will be inherited.

Note:

- When area management information becomes corrupted, the `pdfbkup`, `pdfrstr`, `pdcopy`, and `pdrstr` commands cannot recover the information.
- The `pdffsck` command can detect corruption of area management information, but cannot repair it. In such a case, the command displays the `KFPI21585-E` message and terminates abnormally.

(2) How to handle

Re-create all HiRDB file system areas. Then take the action described below for each HiRDB file system area purpose (`-k` option value in the `pdfmkfs` command).

(a) For DB and SDB

Recover the database from its backup. This backup must contain the RDAREAs that were stored in the HiRDB file system area before the area management information was corrupted.

(b) For SYS

Create the system log files, synchronization point dump files, and status files. Note that the system cannot be restarted because the system files required for restart cannot be provided. Perform a normal start.

If the security audit facility is used, create the audit trail file and then restart the HiRDB and security audit facility.

(c) For WORK

No action is needed because work tables are created automatically by HiRDB during SQL execution.

(d) For UTL

Because the backup file for the `pdcopy` command and the unload data files for the `pdrorg` command have been lost, re-execute the `pdcopy` or `pdrorg` command, if necessary.

Chapter

19. Database Recovery Procedures

This chapter explains the procedures for recovering a database (RDAREAs) when an error has resulted in damage to the database.

This chapter contains the following sections:

19.1 Overview of database recovery

Executor: HiRDB administrator

When an error occurs in a database, the HiRDB administrator uses the database recovery utility (`pdrstr` command) to recover the database. This section explains the basics of database recovery. The following items are explained in this section:

- Database recovery point
- Relationship to backup acquisition mode
- Relationship to log acquisition mode
- Notes on recovery of various types of RDAREAs
- For users of 64-bit-mode HiRDB

19.1.1 Database recovery point

A database can be recovered to its status at any of the following points:

- Backup acquisition point
- Most recent synchronization point before the error occurred
- Any synchronization point since a backup was made

Reference note:

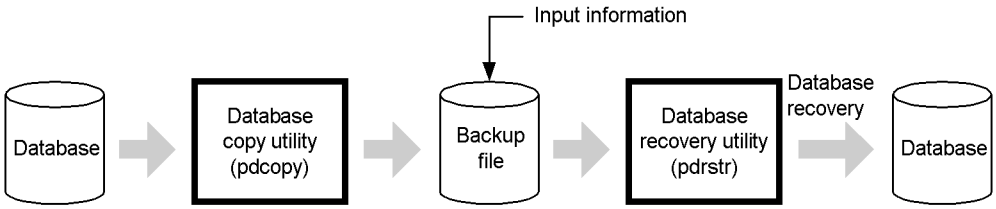
You can use the `pdbkupls` command to check the backup information collected by the database copy utility. For example, you can check the information listed below (for details about the information you can check, see the manual *HiRDB Version 8 Command Reference*):

- Backup acquisition date and time
- Names of RDAREAs that were backed up
- Value specified for the backup acquisition mode (`-M` option)

(1) Recovery to a backup acquisition point

To recover a database to the point at which a backup was made, the backup file is the only input information that is required (for input to the database recovery utility). Figure 19-1 provides an overview of database recovery to a backup acquisition point.

Figure 19-1: Overview of database recovery to a backup acquisition point



Note:

The following point must be kept in mind when a specific RDAREA is to be recovered to a backup acquisition point:

- *When only the RDAREA in which the error occurred is recovered, it will lose synchronization with other RDAREAs.*

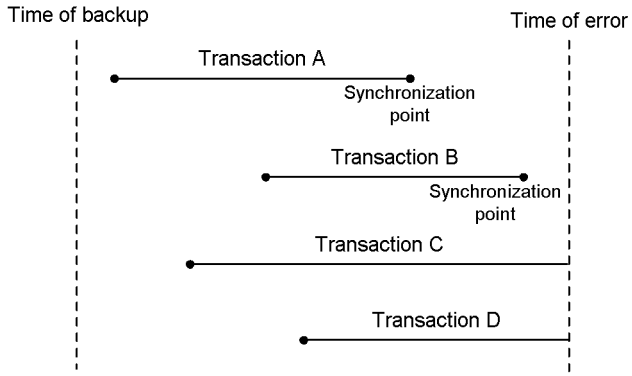
For example, if an error occurs during execution of a definition SQL statement, and then only the user RDAREA in which the error occurred is recovered to a backup acquisition point, the data dictionary RDAREA will be at the most recent synchronization point before the error occurred. Therefore, the RDAREAs described in Table *6-3* must also be recovered from backups made at the same time.

(2) Recovery to most recent synchronization point before the error occurred

(a) Transaction recovery

The point at which a transaction is completed is called a synchronization point. A synchronization point in which updates within a transaction become effective is called a commit, whereas if the updates become ineffective the synchronization point is called a rollback. Database recovery to the synchronization point of the most recently completed transaction at the time of an error is called recovery to the most recent synchronization point before an error occurred. A transaction that is being processed when an error occurs (a transaction that has not yet reached a synchronization point) is ineffective, which means that any update processing by the transaction cannot be recovered. Figure 19-2 shows recovery of a transaction.

Figure 19-2: Transaction recovery (recovery to the most recent synchronization point before an error occurred)



Explanation

Transactions A and B executed to completion and reached synchronization points; the database is recovered to these synchronization points.

Transactions C and D did not reach synchronization points, so their transaction processing is ineffective; these transactions cannot be recovered.

(b) Required input information

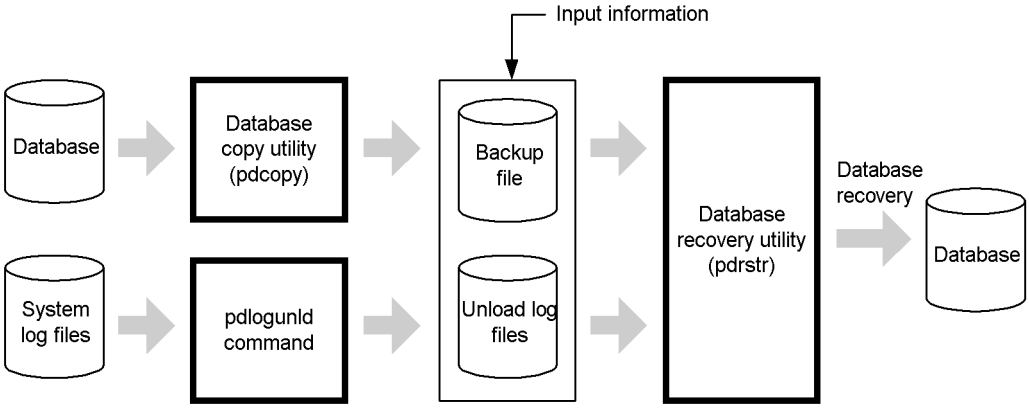
To recover a database to the most recent synchronization point before the error occurred, the following input information is required (for input to the database recovery utility):

- Backup file
- Unload log files*

* The unload log files that are required are those into which system log files subsequent to the backup acquisition point have been unloaded. If operation without unloading system log information is used, all system log files containing system log information subsequent to the backup acquisition point will be required.

Figure 19-3 provides an overview of database recovery to the most recent synchronization point before the error occurred.

Figure 19-3: Overview of database recovery to the most recent synchronization point before the error occurred



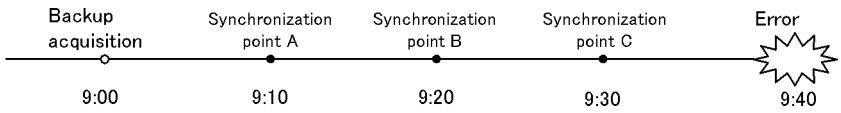
(c) Notes (important)

- The only RDAREA to be recovered is the one in which the error occurred.
- You must back up the RDAREA after it has been completely recovered. Otherwise, it will not be possible to recover the database from this synchronization point if an error occurs in this RDAREA subsequently.
- An unload log file to be used as input to the database recovery utility must be a regular file. Therefore, if the unload log files are saved on a medium such as a CMT or DAT, register them on the disk before executing the database recovery utility.
- In order to recover the master directory RDAREA using the system logs, HiRDB must be started with the `pdstart -r` command. If the system logs are used to recover an RDAREA other than the master directory RDAREA, HiRDB must be started with the `pdstart` command. Therefore, when recovering to the most recent synchronization point, the master directory RDAREA and other RDAREAs cannot be recovered simultaneously. For the procedure for recovering all RDAREAs to the most recent synchronization point, see *19.3.1 Example 1: Recovering all RDAREAs*.

(3) Recovery to any synchronization point since a backup was made (recovery with a range specification)

Database recovery to the synchronization point of a completed transaction at a time specified by the HiRDB administrator is called recovery to any synchronization point since a backup was made. It is not possible to recover the updates of a transaction that is being processed at the time specified by the HiRDB administrator (a transaction that has not yet reach a synchronization point). This is called recovery with a range specification. Figure 19-4 shows recovery with a range specification.

Figure 19-4: Recovery with a range specification



Explanation

Specify the recovery synchronization point in the `-T` option of the database recovery utility.

- To recover the database to synchronization point A, specify in the `-T` option a time after 9:10 but before 9:20 as the recovery end time.
- To recover the database to synchronization point B, specify in the `-T` option a time after 9:20 but before 9:30 as the recovery end time.
- To recover the database to synchronization point C (the synchronization point immediately before the error), the `-T` option need not be specified.

The input information (for input to the database recovery utility) that is required in order to recover the database to any synchronization point since a backup was made is the same as is needed for recovery to the most recent synchronization point before the error occurred.

19.1.2 Relationship to the backup acquisition mode

The point to which the database can be recovered depends on the backup acquisition mode that was specified during backup acquisition (specification of the `-M` option of the `pdcopy` command). Table 19-1 shows the points to which the database can be recovered depending on the backup acquisition mode.

Table 19-1: Points to which the database can be recovered depending on the backup acquisition mode

Backup acquisition mode (-M option specification)	Backup acquisition point	Most recent synchronization point	Recovery with range specification
Referencing/updating-impossible mode (`-M x` option specification)	R	R	R
Referencing-permitted mode (`-M r` option specification)	R	R	R
Updatable mode (`-M s` option specification)*	—	R	R

R: Database can be recovered to this point.

—: Database cannot be recovered to this point.

* If the backup was made with the updatable mode specified, database recovery will require what was the system log current file at the time the backup was made. The corresponding unload log file will be needed as input to the `pdrstr` command.

The name and generation number of the system log file that will be required for database recovery will be output in the `pdcopy` command's processing results output file.

19.1.3 Relationship to the log acquisition mode

(1) When a UAP or utility is executed in the pre-update log acquisition mode

If an error occurs in the database while executing a UAP or utility in the pre-update log acquisition mode, the database can be recovered only to the backup acquisition point.

(2) When a UAP or utility is executed in the no-log mode

If an error occurs in the database while executing a UAP or utility in the no-log mode, the database can be recovered only to the backup acquisition point. All RDAREAs updated by this UAP are placed in *error shutdown* status. For the recovery procedure, see *7.3 Procedure for executing a UAP or utility in the no-log mode*.

19.1.4 Notes on recovery of various types of RDAREAs

(1) Recovering the master directory RDAREA

If the RDAREAs to be recovered include the master directory RDAREA, it is necessary to start HiRDB with the `pdstart -r` command and recover the master directory RDAREA with the `pdrstr` command.

In a recovery to the most recent synchronization point or with range specification, first recover the master directory RDAREA by itself, then recover the other RDAREAs.

(2) Recovering data dictionary LOB RDAREAs

The two data dictionary LOB RDAREAs are classified as follows:

- For storage of the sources for stored routines and triggers
- For storage of the objects of stored routines and triggers

The recovery timing differs for the two data dictionary LOB RDAREAs.

(a) Recovering the data dictionary LOB RDAREA for storing sources

The data dictionary LOB RDAREA for storing sources can be recovered to the backup acquisition point or to any synchronization point since the backup acquisition point. The backup and system log (unload log) are used as the input to the database recovery utility.

(b) Recovering the data dictionary LOB RDAREA for storing objects

The data dictionary LOB RDAREA for storing objects can be recovered only to the backup acquisition point because a database update log required for rollforward is not collected. Only the backup is used as the input to the database recovery utility.

In the following cases, follow the given procedure to recover the data dictionary LOB RDAREA for storing objects.

- You are recovering the data dictionary LOB RDAREA for storing sources to a synchronization point after the backup acquisition point, but the data dictionary LOB RDAREA for storing sources cannot be synchronized with the data dictionary LOB RDAREA for storing objects
- The backup was collected by the `pdcopy` command with the `-J` option specified, but the data dictionary LOB RDAREA for storing objects was intentionally not backed up

Procedure

To recover the data dictionary LOB RDAREA for storing objects:

1. Use the `pdmod` command to reinitialize the data dictionary LOB RDAREA for storing objects:

   ```
   pdmod -a /pdmod/mod01
   ```

2. Use `ALTER ROUTINE` to re-create the SQL objects of all stored routines and triggers.

   ```
   ALTER ROUTINE ALL
   ```

(3) Recovering user LOB RDAREAs

When user LOB RDAREAs are recovered, the user RDAREAs for storing the corresponding LOB column structure base tables should also be recovered at the same time.

(4) Recovering list RDAREAs

List RDAREAs do not become the targets of the `pdrstr` command. Because a list can be re-created if its base table is available, these RDAREAs are not subject to recovery.

If a list RDAREA is placed in error shutdown status, use the following procedure to release the shutdown status:

Procedure

To release the shutdown status for a list RDAREA:

1. Use the `pdclose` command to close the RDAREA that has been placed in error shutdown status.
2. Use the `pdmod` command to re-initialize the list RDAREA.

3. Use the `pdrels -o` command to release the list RDAREA from error shutdown status and open it.

Note:

If a list RDAREA is re-initialized, all lists in that RDAREA become unusable. In such a case, you must re-create the lists with the `ASSIGN LIST` statement.

19.1.5 For users of 64-bit-mode HiRDB

There is no backup compatibility between the 32- and the 64-bit modes. Backups made in 32-bit-mode HiRDB cannot be used in 64-bit-mode HiRDB.

19.2 Recovering a database to the point at which a backup was made

Executor: HiRDB administrator

This section explains by way of examples how to recover a database to the point at which a backup was made. The following examples are provided:

- Example 1: Recovering all RDAREAs
- Example 2: Recovering specified RDAREAs
- Example 3: When JP1/OmniBack II is used for recovery

19.2.1 Example 1: Recovering all RDAREAs

This example recovers all RDAREAs, including system RDAREAs, to their status at the backup acquisition point; it is assumed that a backup copy of the entire system (backup of all RDAREAs) is available.

(1) Enter the pdstop command to terminate HiRDB normally

```
pdstop
```

If HiRDB cannot be terminated normally, take either of the following actions when starting HiRDB in step (5):

- Use the `pdstart dbdestroy` command to start HiRDB forcibly
- Start HiRDB after initializing status files

(2) Enter the pdstart -r command to start HiRDB

```
pdstart -r
```

(3) Use the pdrstr command to restore all RDAREAs

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -a
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-b`: Specifies the name of the backup file containing the backup data for all

RDAREAs.

-a: Specifies that all RDAREAs are to be recovered.

(4) Enter the pdstop command to terminate HiRDB normally

```
pdstop
```

(5) Enter the pdstart command to start HiRDB normally

```
pdstart
```

(6) Use the pdrels command to release and open RDAREAs that are in error shutdown status

```
pdrels -r rdarea01,rdarea02, ... -o
```

This step is not necessary if you start HiRDB after initializing status files.

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.2.2 Example 2: Recovering specified RDAREAs

This example recovers specified user RDAREAs (rdarea01 and rdarea02) to their status at the backup acquisition point.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use the pdrstr command to recover the RDAREAs

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -r rdarea01,rdarea02
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-b: Specifies the name of the backup file containing the backup data for rdarea01 and rdarea02.

-r: Specifies the names of the RDAREAs to be recovered (rdarea01 and rdarea02).

(3) Use the pdrels command to release the recovered RDAREAs from error shutdown status and open them

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.2.3 Example 3: When JP1/OmniBack II is used for recovery

This example recovers all RDAREAs, including the system RDAREAs, to a backup acquisition point; it is assumed that a backup of the entire system (backup of all RDAREAs) is available. Note that JP1/OmniBack II was used to make the backup.

(1) Use pdstop command to terminate HiRDB normally

```
pdstop
```

(2) Use pdstart -r command to start HiRDB

```
pdstart -r
```

(3) Check for objects to be used for recovery

Use the omnidb command of JP1/OmniBack II to check for the objects to be used for recovery. Specify the -stream option in the omnidb command.

(4) Use pdrstr command to recover all RDAREAs

```
pdrstr -m /rdarea/mast/mast01 -k o -b host01:backup01 -G DLT01 -a
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-k: Specifies the type of backup file. o is specified because a JP1/OmniBack II object is used.

-b: Specifies the name of the backup file that stores the backup of all RDAREAs.

Specifies the name of a JP1/OmniBack II object as the backup file name. The specification format is *host-name: object-name*.

`-G`: Specifies the name of the barlist file.

`-a`: Specifies that all RDAREAs are to be backed up.

(5) Use pdstop command to terminate HiRDB normally

```
pdstop
```

(6) Use pdstart command to start HiRDB normally

```
pdstart
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.3 Recovering a database to the most recent synchronization point

Executor: HiRDB administrator

This section explains by way of examples how to recover a database to the most recent synchronization point. The following examples are provided:

- Example 1: Recovering all RDAREAs
- Example 2: Recovering specified RDAREAs
- Example 3: Recovering specified RDAREAs (operation without unloading system log)
- Example 4: When JP1/OmniBack II is used for recovery

19.3.1 Example 1: Recovering all RDAREAs

This example recovers all RDAREAs, including system RDAREAs, to the most recent synchronization point before the error occurred; it is assumed that a backup of the entire system (backup of all RDAREAs) is available. The system log is unloaded.

(1) Use the pdlogls command to check for the current system log file

In the case of a HiRDB/Parallel Server, check for the system log file at the dictionary server and back-end server.

```
pdlogls -d sys
```

(2) Use the pdlogswap command to swap system log files

To unload the contents of the current file, swap the system log files. In the case of a HiRDB/Parallel Server, swap the system log files at the dictionary server and back-end server.

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

(3) Use the pdlogunld command to unload what was the current file

Unload the contents of the current system log file checked in step (1). In the case of a HiRDB/Parallel Server, unload the contents of the current system log file at the dictionary server and back-end server.

```
pdlogunld -d sys -s bes1 -g log01 -o /unld/unldlog02
```

When the automatic log unloading facility is used

In this case, this step is not necessary. Use the `pdlogatul` command to confirm that the automatic log unloading facility is functioning. Also, use the `pdlogls` command to check if unloading has been completed.

(4) Use the pdstop command to terminate HiRDB normally

If HiRDB cannot be terminated normally, initialize the status files. First, delete the status files with the `pdstsrm` command, then re-create status files with the `pdstsinit` command.

```
pdstop
```

(5) Use the pdstart -r command to start HiRDB

```
pdstart -r
```

(6) Use the pdrstr command to recover the master directory RDAREA to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-l /unld/unldlog01,/unld/unldlog02 -w /tmp/sortwork -r rdmast
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-b`: Specifies the name of the backup file that stores the backup of the master directory RDAREA (`rdmast`).

`-l`: Specifies the names of the unload log files.

`-w`: Specifies the name of the work directory for sorting.

`-r`: Specifies the name of the master directory RDAREA (`rdmast`).

(7) Use the pdrstr command to recover the data directory RDAREA, etc., to the backup acquisition point

The following RDAREAs are recovered to the backup acquisition point:

- Data directory RDAREA
- Data dictionary RDAREA
- Data dictionary LOB RDAREA (for storing sources)
- Registry RDAREA
- Registry LOB RDAREA

If HiRDB was started with the `pdstart -r` command, do not execute multiple `pdrstr` commands at the same time.

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-r rddir,rddic,diclob,regrd,reglob
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-b`: Specifies the name of the backup file that stores the backup.

`-r`: Specifies the names of the RDAREAs that are to be recovered.

(8) Use the pdstop command to terminate HiRDB normally

```
pdstop
```

(9) Use the pdstart command to start HiRDB

```
pdstart
```

(10) Use the pdhold command to close RDAREAs by shutting them down

Shut down and close all RDAREAs except the master directory RDAREA and data dictionary RDAREA.

```
pdhold -r rddir,diclob,regrd,reglob,rdarea01,rdarea02,... -c
```

(11) Use the pdhold command to close the data dictionary RDAREA by shutting it down

The data dictionary RDAREA must be shut down and closed last.

```
pdhold -r rddic -c
```

(12) Use the pdrstr command to recover RDAREAs other than the master directory RDAREA to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-l /unld/unldlog01,/unld/unldlog02 -w /tmp/sortwork
-r rddir,rddic,diclob,regrd,reglob,rdarea01,rdarea02,...
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-b`: Specifies the name of the backup file.

`-l`: Specifies the names of the unload log files.

`-w`: Specifies the name of the work directory for sorting.

`-r`: Specifies the names of the RDAREAs to be recovered.

(13) Use the pdrels command to open the data dictionary RDAREA by releasing it from shutdown status

The data dictionary RDAREA must be released from shutdown status and opened first.

```
pdrels -r rddic -o
```

(14) Use the pdrels command to open other RDAREAs by releasing them from shutdown status

Open all RDAREAs except the master directory RDAREA and data dictionary RDAREA by releasing them from shutdown status.

```
pdrels -r rddir,diclob,regrd,reglob,rdarea01,rdarea02,... -o
```

(15) Use the pdcopy command to back up all RDAREAs

```
pdcopy -m /rdarea/mast/mast01 -M r -a -b /pdcopy/backup01
-z /pdcopy/logpoint01 -p /pdcopy/list01
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-M: Specifies the updatable mode as the backup acquisition mode.

-a: Specifies that all RDAREAs are to be backed up.

-b: Specifies a name for the backup file.

-z: Specifies the name of the log point information file. This option is specified when the automatic log unloading facility is used.

-p: Specifies the output destination for the pdcopy command's processing results listing.

(16) Use the pdstop command to terminate HiRDB normally

```
pdstop
```

(17) Use the pdstart command to start HiRDB

```
pdstart
```

(18) Recover the data dictionary LOB RDAREA for storing objects

The procedure for recovering the data dictionary LOB RDAREA for storing objects is described below.

Procedure

To recover the data dictionary LOB RDAREA:

1. Use the pdmod command to re-initialize the data dictionary LOB RDAREA for storing objects:

   ```
   pdmod -a /pdmod/mod01
   ```

2. Use ALTER ROUTINE to re-create the SQL objects of all stored routines and triggers.

   ```
   ALTER ROUTINE ALL
   ```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.3.2 Example 2: Recovering specified RDAREAs

This example recovers specified user RDAREAs (rdarea01 and rdarea02) to their

status at the most recent synchronization point preceding the error. The system log file handling method is system log unloading operation.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use the pdlogls command to check for the current system log file

In the case of a HiRDB/Parallel Server, this check is for the system log file in the back-end server that contains `rdarea01` and `rdarea02`.

```
pdlogls -d sys
```

(3) Use the pdlogswap command to swap system log files

The system log files are swapped so that the current file can be unloaded. In the case of a HiRDB/Parallel Server, swap the system log files at the back-end servers in which `rdarea01` and `rdarea02` are located.

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

(4) Use the pdlogunld command to unload the current file

Unload the current system log file checked in step (2). In the case of a HiRDB/Parallel Server, unload the current system log files at the back-end servers in which `rdarea01` and `rdarea02` are located.

```
pdlogunld -d sys -g log01 -o /unld/unldlog02
```

When the automatic log unloading facility is used

In this case, this step is not necessary. Use the `pdlogatul` command to confirm that the automatic log unloading facility is functioning. Also, use the `pdlogls` command to check if unloading has been completed.

(5) Use the pdrstr command to recover the RDAREAs to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-l /unld/unldlog01,/unld/unldlog02 -w /tmp/sortwork -r rdarea01,rdarea02
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-b: Specifies the name of the backup file containing the backup data for rdarea01 and rdarea02.

-l: Specifies the names of the unload log files.

-w: Specifies the name of the work directory for sorting.

-r: Specifies the names of the RDAREAs to be recovered (rdarea01 and rdarea02).

- When range-specified recovery is to be executed, the recovery end time is specified in the -T option. If the -T option is omitted, the RDAREAs are recovered to the most recent synchronization point preceding the error.
- If there is an unload log file that cannot be read due to an error in a system log file or the unload log file, the most recent unload log file among all the unload log files that were read successfully must be analyzed to narrow the range of data that cannot be recovered. The applicable transactions can then be reexecuted, if necessary.

(6) Use the pdcopy command to back up the recovered RDAREAs

```
pdcopy -m /rdarea/mast/mast01 -M x -r rdarea01,rdarea02
-b /pdcopy/backup02 -z /pdcopy/logpoint01 -p /pdcopy/list01
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-M: Specifies the referencing/updating-impossible mode as the backup acquisition mode.

-r: Specifies the RDAREAs that are to be backed up (rdarea01 and rdarea02, the RDAREAs in which the error occurred).

-b: Specifies the name of the backup file.

-z: Specifies the name of the log point information file. This option is specified when the automatic log unloading facility is used.

-p: Specifies the output destination for the pdcopy command's processing results listing.

(7) Use the pdrels command to release the recovered RDAREAs from error shutdown status and open them

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.3.3 Example 3: Recovering specified RDAREAs (operation without unloading the system log)

This example recovers specified user RDAREAs (`rdarea01` and `rdarea02`) to their status at the most recent synchronization point preceding the error. The system log file handling method is operation without unloading the system log.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use the pdlogswap command to swap system log files

The current file cannot be used as is as the input to the database recovery utility, so the system log files are swapped. In the case of a HiRDB/Parallel Server, swap the system log files at the back-end servers in which `rdarea01` and `rdarea02` are located.

```
pdlogswap -d sys
```

(3) Use the pdrstr command to recover the RDAREAs to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -L
-w /tmp/sortwork -r rdarea01,rdarea02
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-b`: Specifies the name of the backup file containing the backup data for `rdarea01` and `rdarea02`.

`-L`: Specifies that the system log file is to be used as the input information.

-w: Specifies the name of the work directory for sorting.

-r: Specifies the names of the RDAREAs to be recovered (rdarea01 and rdarea02).

When range-specified recovery is to be executed, the recovery end time is specified in the -T option. If the -T option is omitted, the RDAREAs will be recovered to the most recent synchronization point preceding the error.

(4) Use the pdcopy command to back up the recovered RDAREAs

```
pdcopy -m /rdarea/mast/mast01 -M x -r rdarea01,rdarea02 -b /pdcopy/backup02
-z /pdcopy/logpoint01
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-M: Specifies the referencing/updating-impossible mode as the backup acquisition mode.

-r: Specifies the erroneous RDAREAs (rdarea01 and rdarea02) that are to be backed up.

-b: Specifies the name of the backup file.

-z: Specifies the name of the log point information file.

(5) Use the pdrels command to release the recovered RDAREAs from error shutdown status and open them

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.3.4 Example 4: When JP1/OmniBack II is used for recovery)

This example recovers user RDAREAs (rdarea01 and rdarea02) to the most recent synchronization point before the error occurred; it is assumed that the system log is unloaded.

Note that JP1/OmniBack II is used to make backups.

(1) Use pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use pdlogls command to check for current system log file

In the case of a HiRDB/Parallel Server, check for the system log files at the back-end servers in which `rdarea01` and `rdarea02` are located.

```
pdlogls -d sys
```

(3) Use pdlogswap command to swap system log files

The system log files are swapped so that the current file can be unloaded. In the case of a HiRDB/Parallel Server, swap the system log files at the back-end servers in which `rdarea01` and `rdarea02` are located.

```
pdlogswap -d sys
```

(4) Use pdlogunld command to unload what was the current file

Unload the current file identified in step (2).

```
pdlogunld -d sys -g log01 -o /unld/unldlog02
```

(5) Check for objects to be used for recovery

Use the `omnidb` command of JP1/OmniBack II to check for the objects to be used for recovery. Specify the `-stream` option in the `omnidb` command.

(6) Use pdrstr command to recover RDAREAs to most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -k o -b host01:backup01 -G DLT01
-l /unld/unldlog01,/unld/unldlog02 -w /tmp/sortwork -r rdarea01,rdarea02
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-k`: Specifies the type of backup file. o is specified because a JP1/OmniBack II object is used.

-b: Specifies the name of the backup file that stores the backup of rdarea01 and rdarea02. Specifies the name of a JP1/OmniBack II object as the backup file name. The specification format is host-name: object-name.

-G: Specifies the name of the barlist file.

-l: Specifies the names of the unload log files.

-w: Specifies the name of the work directory for sorting.

-r: Specifies the names of the RDAREAs (rdarea01 and rdarea02) that are to be recovered.

- For recovery with a range specification, specify a recovery termination time in the -T option. If this option is not specified, the RDAREAs will be recovered to the most recent synchronization point before the error occurred.
- If some unload log files cannot be entered because of an error in the system log file or unload log file, analyze the most recent unload log file that could be entered in order to narrow the range that cannot be recovered. Then, if necessary, re-execute the transaction.

(7) Use pdcopy command to back up recovered RDAREAs

```
pdcopy -m /rdarea/mast/mast01 -M x -r rdarea01,rdarea02 -k o
-b host01:backup002 -G DLT02
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-M: Specifies the referencing/updating-impossible mode as the backup acquisition mode.

-r: Specifies the RDAREAs to be backed up (rdarea01 and rdarea02, the RDAREAs in which the error occurred).

-k: Specifies the type of backup file. o is specified because a JP1/OmniBack II object is used.

-b: Specifies the name of a JP1/OmniBack II object as the backup file name. The specification format is host-name: object-name.

-G: Specifies the name of the barlist file.

(8) Use pdrels command to release error shutdown status of recovered RDAREAs and open them

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference*.

19.4 Database recovery using the differential backup facility

Executor: HiRDB administrator

This section explains the use of the differential backup facility to recover a database to either of the following points:

- Most recent differential backup acquisition point
- Most recent synchronization point

19.4.1 Example 1: Recover to the most recent differential backup acquisition point

This example recovers user RDAREAs (`rdarea01` and `rdarea02`) to a differential backup acquisition point.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use the pdrstr command to recover RDAREAs

```
pdrstr -m /rdarea/mast/mast01 -g backupg1 -K /pdcopy/admfile -r rdarea01,rdarea02
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-g`: Specifies the name of a differential backup group.

`-K`: Specifies the name of the HiRDB file system area for storing the differential backup management file.

`-r`: Specifies the names of the RDAREAs (`rdarea01` and `rdarea02`) that are to be recovered.

(3) Use the pdrels command to open recovered RDAREAs by releasing their error shutdown status

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results,

see the manual *HiRDB Version 8 Command Reference*.

19.4.2 Example 2: Recover to the most recent synchronization point

This example recovers user RDAREAs (`rdarea01` and `rdarea02`) to the most recent synchronization point before an error occurred; it is assumed that the system log is being unloaded.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use the pdlogls command to check for the current system log file

In the case of a HiRDB/Parallel Server, check the system log files at the back-end servers at which `rdarea01` and `rdarea02` are located.

```
pdlogls -d sys
```

(3) Use the pdlogswap command to swap system log files

In order to unload the contents of the current file, the system log files are swapped. In the case of a HiRDB/Parallel Server, swap the system log files at the back-end servers at which `rdarea01` and `rdarea02` are located.

```
pdlogswap -d sys
```

(4) Use the pdlogunld command to unload what was the current system log file

Unload the contents of the current file identified in step (2). In the case of a HiRDB/Parallel Server, unload the contents of the current system log files at the back-end servers at which `rdarea01` and `rdarea02` are located.

```
pdlogunld -d sys -g log01 -o /unld/unldlog02
```

(5) Use the pdrstr command to recover RDAREAs to the most recent synchronization

```
pdrstr -m /rdarea/mast/mast01 -g backupg1 -K /pdcopy/admfile
-l /unld/unldlog01 -w /tmp/sortwork -r rdarea01,rdarea02
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-g: Specifies the name of a differential backup group.

-K: Specifies the name of a HiRDB file system area for storing the differential backup management file.

-l: Specifies the name of the unload log file.

-w: Specifies the name of the work directory for sorting.

-r: Specifies the names of the RDAREAs (rdarea01 and rdarea02) that are to be recovered.

(6) Use the pdcopy command to back up recovered RDAREAs

```
pdcopy -m /rdarea/mast/mast01 -M r -r rdarea01,rdarea02,...
-g backupg1(S) -b /pdcopy/backup01 -d a -K /pdcopy/admfile -L 5
-o /pdcopy/rfile
```

Explanation

-m: Specifies the name of the first HiRDB file in the master directory RDAREA.

-M: Specifies the referencing-permitted mode as the backup acquisition mode.

-r: Specifies the RDAREAs that are to be backed up.

The RDAREA group specified here becomes the differential backup group. The RDAREAs to be backed up cannot be changed in the middle.

-g: Specifies the name of the differential backup group.

For a full backup, specify (S) with the differential backup group name. The differential backup group name specified here must be specified when a differential backup is made subsequently.

-b: Specifies a name for the backup file (full backup file name).

-d: Specifies a backup type:

a: Full backup

b: Accumulation-differential backup since the last full backup

c: Accumulation-differential backup since either the last accumulation-differential backup or the last full backup

d: Differential backup

For details on an accumulation-differential backup, see *6.5.4 Creating an*

accumulation-differential backup.

-K: Specifies the name of the HiRDB file system area for storing the differential backup management file.

-L: Specifies the size (in megabytes) of the differential backup management file.

-o: Specifies the name of the history file for differential backups.

(7) Use the pdrels command to open recovered RDAREAs by releasing their error shutdown status

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference*.

19.4.3 Recovery when a differential backup management file is not available

If no differential backup management file is available, the database must be recovered by executing the pdrstr command multiple times (once for each backup that was made).

When the recovery method in Example 2 above is executed, the operation at step *19.4.2(5)* will be different. This section explains the operation from this step on. It is assumed that differential backups are collected at the following times:

- A full backup is collected on Sunday.
- A differential backup is collected on Monday and Tuesday.
- The database is recovered on Wednesday.

(1) Use the pdrstr command to recover RDAREAs using the full backup as the input information

```
pdrstr -m /rdarea/mast/mast01 -g backupg1 -K /pdcopy/admfile
-b /pdcopy/backup01 -r rdarea01,rdarea02
```

Explanation

-b: Specifies the name of the full backup file.

(2) Use the pdrstr command to recover RDAREAs using a differential backup as the input information

Use the pdrstr command to recover RDAREAs using the differential backup made

on Monday as the input information.

```
pdrstr -m /rdarea/mast/mast01 -g backupg1 -K /pdcopy/admfile
-b /pdcopy/backup02 -r rdarea01,rdarea02
```

Explanation

-b: Specifies the name of the differential backup file.

(3) Use the pdrstr command to recover RDAREAs using the differential backup as the input information

Use the pdrstr command to recover RDAREAs using the differential backup made on Tuesday as the input information.

```
pdrstr -m /rdarea/mast/mast01 -g backupg1 -K /pdcopy/admfile
-b /pdcopy/backup03 -r rdarea01,rdarea02
```

Explanation

-b: Specifies the name of the differential backup file.

(4) Use the pdrstr command to recover RDAREAs using the unload log file as the input information

Use the pdrstr command to recover the RDAREAs beyond the differential backup collected on Tuesday using the unload log file as the input information. If an error occurs at this point, repeat the sequence of operations, beginning with step (1). After RDAREAs have been recovered using the unload log file as the input information, it is important to make a full backup.

```
pdrstr -m /rdarea/mast/mast01 -l /unld/unldlog01 -w /tmp/sortwork
-r rdarea01,rdarea02
```

Explanation

-l: Specifies the name of the unload log file.

19.5 Recovery procedure when the backup was not made with the pdcopy command

Executor: HiRDB administrator

This section explains database recovery when a backup was made without using the `pdcopy` command (i.e., when another product's facility was used). The database recovery procedure is outlined below:

Procedure

To recover the database:

1. Use the other product's restore facility to recover the database to the backup acquisition point.
2. Use the `pdrstr` command to recover the database to the most recent synchronization point.
   - To recover the database to the backup acquisition point only, perform step 1 only.
   - If the master directory RDAREA is included in the RDAREAs to be recovered when recovery is to the most recent synchronization point or recovery is with a range specification, first recover the master directory RDAREA only, then recover the remaining RDAREAs.

19.5.1 Example 1: Recovering all RDAREAs to the point at which a backup was made

This section explains how to recover all RDAREAs, including the system RDAREAs, to the point at which a backup was made; it is assumed that a backup of the entire system (backup of all RDAREAs) is available.

(1) Use the pdstop command to terminate HiRDB normally

```
pdstop
```

If HiRDB cannot be terminated normally, take either of the following actions when starting HiRDB in step (3):

- Use the `pdstart dbdestroy` command to start HiRDB forcibly.
- Start HiRDB after initializing status files.

(2) Use another product's facility to recover all RDAREAs

Use another product's facility to recover all RDAREAs.

(3) Use the pdstart command to start HiRDB normally

```
pdstart
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.5.2 Example 2: Recovering specified RDAREAs to the point at which a backup was made

This example recovers user RDAREAs (`rdarea01` and `rdarea02`) to the point at which a backup was made.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use another product's facility to recover all RDAREAs

Use another product's restore facility to recover the HiRDB file system area that comprises the RDAREAs (`rdarea01`, `rdarea02`).

(3) Use the pdrels command to open recovered RDAREAs by releasing their error shutdown status

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.5.3 Example 3: Recovering all RDAREAs to the most recent synchronization point

This example recovers all RDAREAs, including the system RDAREAs, to the most recent synchronization point before an error occurred; it is assumed that a backup of the entire system (backup of all RDAREAs) is available. The system log is being unloaded.

(1) Use the pdlogls command to check for the current system log file

In the case of a HiRDB/Parallel Server, check for the current system log files at the dictionary and back-end servers.

```
pdlogls -d sys
```

(2) Use the pdlogswap command to swap system log files

To unload the current system log file, swap system log files. In the case of a HiRDB/Parallel Server, swap system log files at the dictionary and back-end servers.

```
pdlogswap -d sys
```

(3) Use the pdlogunld command to unload what was the current system log file

Unload the current system log file checked in step (1).

In the case of a HiRDB/Parallel Server, unload the current system log files at the dictionary and back-end servers.

```
pdlogunld -d sys -g log01 -o /unld/unldlog02
```

(4) Use the pdstop command to terminate HiRDB normally

If HiRDB will not terminate normally, initialize the status files, use the `pdstsrm` command to delete the status files, then use the `pdstsinit` command to re-create the status files.

```
pdstop
```

(5) Recover the master directory RDAREA to the backup acquisition point

Use another product's facility to recover the master directory RDAREA to a backup acquisition point.

(6) Use the pdstart -r command to start HiRDB

```
pdstart -r
```

(7) Use the pdrstr command to recover the master directory RDAREA to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -l /unld/unldlog01,/unld/unldlog02
-w /tmp/sortwork -r rdmast
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-l`: Specifies the names of the unload log files.

`-w`: Specifies the name of the work directory for sorting.

`-r`: Specifies the name of the master directory RDAREA (`rdmast`).

(8) Recover the data directory RDAREA, etc., to the backup acquisition point

Use another product's facility to recover the following RDAREAs to a backup acquisition point:

- Data directory RDAREA
- Data dictionary RDAREA
- Data dictionary LOB RDAREA (for storing sources)
- Registry RDAREA
- Registry LOB RDAREA

(9) Use the pdstop command to terminate HiRDB normally

```
pdstop
```

(10) Use the pdstart command to start HiRDB

```
pdstart
```

(11) Use the pdhold command to close RDAREAs by shutting them down

Shut down and close all RDAREAs except the master directory RDAREA and data dictionary RDAREA.

```
pdhold -r rddir,diclob,regrd,reglob,rdarea01,rdarea02,... -c
```

(12) Use the pdhold command to close the data dictionary RDAREA by shutting it down

The data dictionary RDAREA must be shut down and closed last.

```
pdhold -r rddic -c
```

(13) Use the pdrstr command to recover RDAREAs other than the master directory RDAREA to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -l /unld/unldlog01,/unld/unldlog02
-w /tmp/sortwork -r rddir,rddic,diclob,regrd,reglob,rdarea01,rdarea02,...
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-l`: Specifies the names of the unload log files.

`-w`: Specifies the name of the work directory for sorting.

`-r`: Specifies the names of the RDAREAs to be recovered.

(14) Use the pdrels command to open the data dictionary RDAREA by releasing it from shutdown status

The data dictionary RDAREA must be released from shutdown status and opened first.

```
pdrels -r rddic -o
```

(15) Use the pdrels command to open other RDAREAs by releasing them from shutdown status

Open all RDAREAs except the master directory and data dictionary RDAREAs by releasing their shutdown status.

```
pdrels -r rddir,diclob,regrd,reglob,rdarea01,rdarea02,... -o
```

(16) Use the pdstop command to terminate HiRDB normally

```
pdstop
```

(17) Back up all RDAREAs

Use another product's backup facility to back up all RDAREAs.

(18) Use the pdstart command to start HiRDB

```
pdstart
```

(19) Recover the data dictionary LOB RDAREA for storing objects

The procedure for recovering the data dictionary LOB RDAREA for storing objects is described below.

Procedure

To recover the data dictionary LOB RDAREA:

1. Use the `pdmod` command to re-initialize the data dictionary LOB RDAREA for storing objects:

   ```
   pdmod -a /pdmod/mod01
   ```

2. Use `ALTER ROUTINE` to re-create the SQL objects of all stored routines and triggers.

   ```
   ALTER ROUTINE ALL
   ```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference*.

19.5.4 Example 4: Recovering specified RDAREAs

This example recovers user RDAREAs (`rdarea01` and `rdarea02`) to the most recent synchronization point before an error occurred. The system log is being unloaded.

(1) Use the pdclose command to close RDAREAs in error shutdown status

```
pdclose -r rdarea01,rdarea02
```

(2) Use the pdlogls command to check for the current system log file

In the case of a HiRDB/Parallel Server, check the system log files at the dictionary and back-end servers.

```
pdlogls -d sys
```

(3) Use the pdlogswap command to swap system log files

To unload the current file, swap system log files. In the case of a HiRDB/Parallel Server, swap system log files at the back-end servers where `rdarea01` and `rdarea02` are located.

```
pdlogswap -d sys
```

(4) Use the pdlogunld command to unload what was the current system log file

Unload the current system log file identified in step (2).

```
pdlogunld -d sys -g log01 -o /unld/unldlog02
```

(5) Recover RDAREAs to the backup point

Use another product's restore facility to recover `rdarea01` and `rdarea02` to a backup acquisition point.

(6) Use pdrstr command to recover RDAREAs to most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -l /unld/unldlog01,/unld/unldlog02
-w /tmp/sortwork -r rdarea01,rdarea02
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-l`: Specifies the names of the unload log files.

`-w`: Specifies the name of the work directory for sorting.

`-r`: Specifies the names of the RDAREAs (`rdarea01` and `rdarea02`) to be recovered.

(7) Back up recovered RDAREAs

Use another product's backup facility to back up `rdarea01` and `rdarea02`.

(8) Use the pdrels command to open recovered RDAREAs by releasing them from error shutdown status

```
pdrels -r rdarea01,rdarea02 -o
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

19.5.5 Example 5: Recovering the master directory RDAREA only

This example recovers only the master directory RDAREA to the most recent synchronization point before an error occurred. The system log is being unloaded.

(1) Use the pdlogls command to check for the current system log file

In the case of a HiRDB/Parallel Server, check the system log files at all servers.

```
pdlogls -d sys
```

(2) Use the pdlogunld command to unload what was the current system log file

Unload the current system log file checked in step (1).

In the case of a HiRDB/Parallel Server, unload the current system log files at all servers.

```
pdlogunld -d sys -g log01 -o /unld/unldlog02
```

(3) Recover the master directory RDAREA to the backup point

Use another product's restore facility to recover the master directory RDAREA to a backup acquisition point.

(4) Use the pdstart -r command to start HiRDB

```
pdstart -r
```

(5) Use the pdrstr command to recover the RDAREA to the most recent synchronization point

```
pdrstr -m /rdarea/mast/mast01 -l /unld/unldlog01,/unld/unldlog02
-w /tmp/sortwork -r MAST
```

Explanation

`-m`: Specifies the name of the first HiRDB file in the master directory RDAREA.

`-l`: Specifies the names of the unload log files.

-w: Specifies the name of the work directory for sorting.

-r: Specifies the name of the master directory RDAREA (MAST).

(6) Use the pdstop command to terminate HiRDB

```
pdstop
```

(7) Back up the master directory RDAREA

Use another product's backup facility to back up the HiRDB file system area that comprises the master directory RDAREA.

(8) Use the pdstart -r command to start HiRDB

```
pdstart
```

It is recommended that after the command has executed you check whether or not the execution results are correct. For details on how to check command execution results, see the manual *HiRDB Version 8 Command Reference.*

Chapter

20. Obtaining Tuning Information

This chapter explains the procedures for obtaining the tuning information that is required in order to tune the system.

This chapter contains the following sections:

20.1 Collecting tuning information from the statistics log

This section explains the procedures for collecting tuning information from the statistics log. The following topics are discussed:

- Tuning information that can be collected from the statistics log
- Preparing for collecting tuning information
- Collecting tuning information
- Shell script for creating unload statistics log files at a specified server machine
- When linked to an OLTP system

20.1.1 Tuning information that can be collected from the statistics log

Table 20-1 shows the types of tuning information that can be collected from the statistics log.

Table 20-1: Tuning information that can be collected from the statistics log

Statistics information type		Explanation
System activity statistical information	sys	Outputs activity information related to HiRDB processes, RPC, system logs, etc.
UAP statistical information	uap	Edits by edit time UAP-related information, such as the number of retrieved rows and the number of SQL executions, and outputs this information by UAP or transaction.
SQL statistical information	sql	Outputs by UAP or service information related to SQL issuances, such as the number of retrieved rows, the number of times work tables were created, and the number of back-end servers that issued intra-SQL partitioning commands.
Global buffer pool statistical information[1]	buf	Edits by edit time information related to global buffer accesses, such as the buffer hits rate and the actual I/Os counts, and outputs this information by server or global buffer.
Statistical information on HiRDB files for database manipulation	fil	Edits by edit time information related to HiRDB file accesses, such as the synchronization I/Os counts and I/O error counts, and outputs this information by server, HiRDB file, and RDAREA.
Deferred write processing statistical information	dfw	Edits by edit time information related to deferred write processing, such as the number of deferred write processes, operation factors, and the I/O parallel level, and outputs this information by server.
Index statistical information	idx	Edits by edit time index-related information, such as index key lock information and index split information, based on the statistics logs and system logs, and outputs this information by server and index.

Statistics information type		Explanation
SQL static optimization information	sop	Outputs (to a DAT file only) SQL static optimization information.
SQL dynamic optimization information	dop	Outputs (to a DAT file only) SQL dynamic optimization information.
SQL object execution information	pcd	Outputs (to a DAT file only) SQL object execution optimization information.
SQL statement statistical information[2]	sqh	Outputs (to a DAT file only) information related to data manipulation SQLs, definition SQLs, LOCK statements, and issued SQLs.
Statistics on SQL object transmission[3]	obj	Outputs (to a DAT file only) SQL object transmission information.
Statistical information on the operation of foreign servers[3]	fsv	Outputs foreign server operation information. Statistical information is output when transactions are completed. Information is output to a DAT file only.
Statistical information on the usage status of foreign servers [3]	hba	Outputs information on foreign server usage status. Statistical information is output when SQL is executed with respect to a foreign server. Information is output to a DAT file only.

[1] This statistical information is collected at synchronization points. Because it is a compilation of information between synchronization points, no statistical information is collected unless there are at least two synchronization points. To ensure that the statistical information is collected, execute the `pdlogsync` command immediately before executing the `pdstjsync` or `pdstend` command to cause a synchronization point to be set.

[2] Statistical information related to the history of SQL statements is output when statistical information related to SQL (`sql` specification) is output.

[3] This statistical information is limited to a HiRDB/Parallel Server. It is not output for a HiRDB/Single Server.

20.1.2 Preparing for collecting tuning information

Executor: HiRDB administrator

To collect tuning information, it is necessary to collect the statistical information (statistics log) output by HiRDB. The statistics log is output to a statistics log file. The HiRDB administrator must determine an appropriate size for this file.

(1) Creating the statistics log files

There are two statistics log files, named `pdstj1` and `pdstj2`, under the `$PDDIR/spool` directory. HiRDB creates these statistics log files automatically. The HiRDB

administrator must specify the size of the statistics log file in the `pd_stj_file_size` operand. The following points should be noted about determining this file size:

- The total size of the statistics log information to be collected should be specified as the file size. If multiple sets of statistical information are to be collected concurrently, the value specified in the `pd_stj_file_size` operand must be greater than the sum of the sizes of those statistics logs. For the formula for determining the size of a statistics log, see the manual *HiRDB Version 8 System Definition*.
- Whether or not the various types of statistics log information are collected depends on the type of server, as shown in Table 20-2. The HiRDB administrator must take this into account in determining the file size.

Table 20-2: Servers subject to collection of statistics log information

Type of statistics log		HiRDB/Single Server	HiRDB/Parallel Server			
		SDS	MGR	FES	DS	BES
System activity statistical information	sys	Y	Y	Y	Y	Y
UAP statistical information	uap	Y	—	Y	—	—
SQL statistical information	sql	Y	—	Y	—	—
Global buffer pool statistical information	buf	Y	—	—	Y	Y
Statistical information on HiRDB files for database manipulation	fil	Y	—	—	Y	Y
Deferred write processing statistical information	dfw	Y	—	—	Y	Y
Index statistical information*	idx	Y	—	—	Y	Y
SQL static optimization information	sop	Y	—	Y	—	—
SQL dynamic optimization information	dop	Y	—	Y	—	—
SQL object statistical information	pcd	Y	—	Y	Y	Y
SQL statement statistical information	sqh	Y	—	Y	—	—
Statistics on SQL object transfer	obj	—	—	—	Y	Y
Statistical information on the operation of foreign servers	fsv	—	—	—	—	Y
Statistical information on the usage status of foreign servers	hba	—	—	—	—	Y

Y: Statistics log information that is collected

—: Statistics log information that is not collected

Note

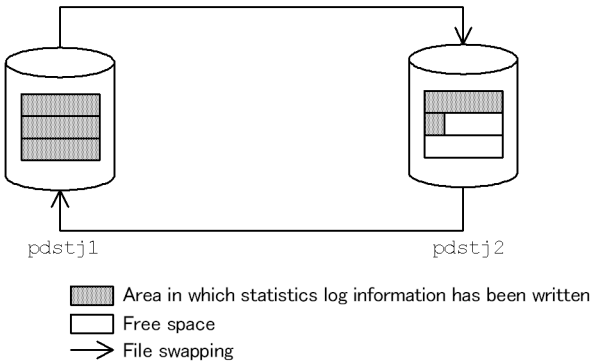
- The `CONNECT`/`DISCONNECT` statistical information collected by the `pdhibegin` operand is output to the system log file, not to the statistics log file; for details, see *20.2 Collecting tuning information from the system log*.
- In the case of a utility special unit, only system activity statistical information can be collected.

* Information on index page splits is not collected as part of the statistics log. For information on how to collect this information, see *20.2 Collecting tuning information from the system log*.

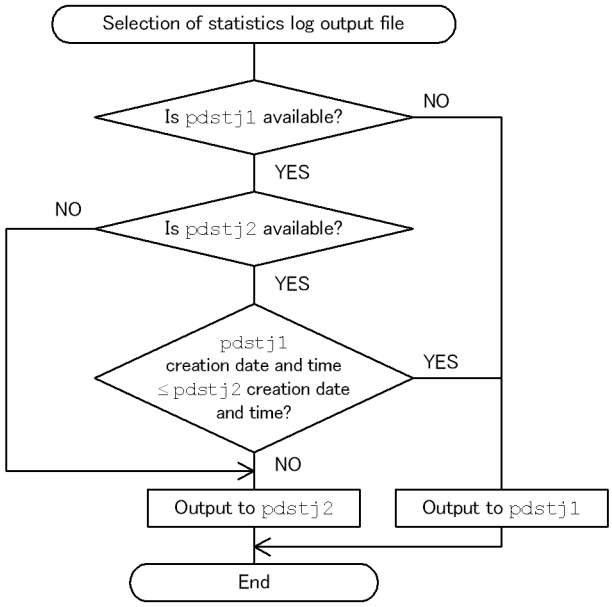
(2) Handling the statistics log file

When a statistics log file becomes full during output of statistics log information, HiRDB switches outputs of the information to the other statistics log file. This is called statistics log file swapping. When statistics log file swapping occurs, new information is written over any existing statistics log information in the swapped-in file. For this reason, as soon as a statistics log file is swapped out, the HiRDB administrator should unload it and create an unload statistics log file. Figure 20-1 shows the procedure for swapping the statistics log files.

Figure 20-1: Swapping of statistics log files



Explanation:
When the pdstj1 statistics log file becomes full, the statistics log output destination is switched to pdstj2. Any statistics log information in pdstj2 will be overwritten by new information.
At the time of startup, HiRDB uses the following procedure to select the statistics log output file to be used:



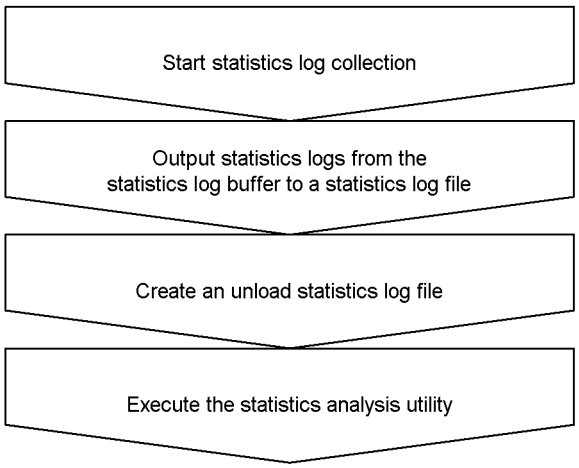
20.1.3 Collecting tuning information

Executor: HiRDB administrator

This section explains the procedure for collecting tuning information (from collection of statistics log information to execution of the statistics analysis utility that uses the collected statistics log as its input information). Figure 20-2 shows the procedure for

collecting tuning information (collecting tuning information from the statistics log).

Figure 20-2: Procedure for collecting tuning information (collecting tuning information from the statistics log)



(1) Starting statistics log collection

Use the `pdstbegin` command to start statistics log collection. To start collecting statistics logs at the time of HiRDB startup, specify the `pdstbegin` operand in the system common definition. When the `pdstbegin` operand is specified, there is no need to execute the `pdstbegin` command.

Note:

The information listed below is not collected unless at least two synchronization points occur during statistics log collection. To ensure that this information is collected, use the `pdlogsync` command to set a synchronization point immediately after entering the `pdstbegin` command, and then use the `pdlogsync` command to set a synchronization point immediately before entering the `pdstjsync` or `pdstend` command.

- Global buffer pool statistical information
- Statistical information on HiRDB files for data manipulation
- Deferred write processing statistical information
- Index statistical information

Reference note:

To end statistics log collection, execute the `pdstend` command.

(2) Outputting statistics logs from the statistics log buffer to a statistics log file

Use the pdstjsync command to output the statistics logs stored in the statistics log buffer to a statistics log file.

Reference note:

The statistics logs stored in the statistics log buffer are output to a statistics log file at the following trigger points:

- When the statistics log buffer becomes full
- When the pdstjsync command is executed
- When the pdstend command is executed
- When HiRDB is terminated (normal or planned termination)

(3) Creating an unload statistics log file

One of the following methods must be used to unload a statistics log file and create an unload statistics log file; normally, the OS's cp command is used:

- OS's cp command
- HiRDB-provided shell script (pdstjacm)

Hint:

When statistics log information is being output to multiple server machines (as in the case of a HiRDB/Parallel Server), pdstjacm should be used to create the unload statistics log files. For example, when system activity information is collected by a HiRDB/Parallel Server, the statistics log information is output to multiple server machines. In such a case, pdstjacm must be used to create the unload statistics log files at a specific server machine. For details on pdstjacm, see *20.1.4 Shell script for creating unload statistics log files at a specified server machine*.

Note:

When a statistics log file is unloaded while HiRDB is active, care must be taken not to delete the statistics log file with the OS's mv or rm command. If it is deleted, statistical information will no longer be output. If the statistics log file is deleted inadvertently, either restart HiRDB or execute the pdstjswap command.

(4) Executing the statistics analysis utility

Using the unload statistics log files created in step (3) as the input information, execute the statistics analysis utility. The HiRDB administrator can then use the execution

results to tune HiRDB or UAPs.

Reference note:

Because statistics information is collected between the time when the `pdstbegin` command is entered and the time when the `pdstend` (or `pdstjsync`) command is entered, the UAP statistics information may not match the SQL statistics information depending on the command entry timings.

20.1.4 Shell script for creating unload statistics log files at a specified server machine

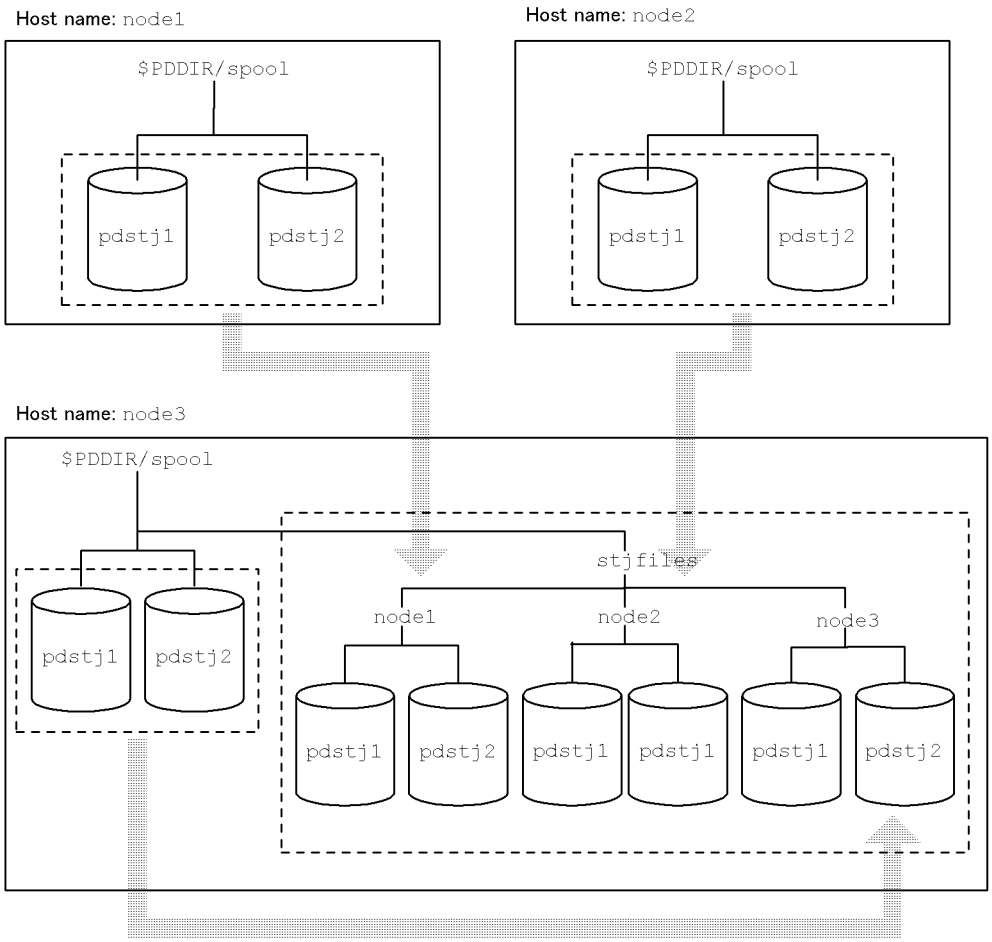
Executor: HiRDB administrator

In the case of a HiRDB/Parallel Server, a shell script (`pdstjacm`) can be used to collect all the unload statistics log files acquired at the individual server machines under a specified server machine's directory. This section describes the shell script for creating unload statistics log files at a specific server machine.

(1) Function of pdstjacm

`pdstjacm` is a HiRDB-provided shell script. Executing `pdstjacm` collects the statistics log information output from multiple server machines under the directory of a specified server machine. Unload statistics log files appropriate to the user's environment can be created by modifying the contents of `pdstjacm`. When the contents of `pdstjacm` are not modified, all the unload statistics log files are collected under `$PDDIR/spool/stjfiles` at the host used to execute `pdstjacm`. Figure 20-3 provides an overview of `pdstjacm`.

Figure 20-3: Overview of pdstjacm



Explanation

The statistics log information from node1, node2, and node3 is collected under the `$PDDIR/spool/stjfiles` directory.

(2) Contents of pdstjacm

`pdstjacm` is stored as `pdstjacm` under `$PDDIR/bin`. The shell script should be copied before it is modified. In addition to `pdstjacm`, the OS's `rcp` command (remote copy) and NFS (network file system) can also be used. The following are the contents of `pdstjacm`:

```
#!/bin/sh
# ALL RIGHTS RESERVED,COPYRIGHT (C) 1994,1995,HITACHI,LTD.
# LICENSED MATERIAL OF HITACHI,LTD.
# Name          : pdstjacm
# Function      : Collects statistics log files at host at which command is entered.
# Command Line  : pdstjacm [sysdef filename]
# Command option: sysdef filename - If name of system common definition at local
#                                   host is omitted, $PDCONFPATH/pdsys is assumed.
# Note          : (1) Change the OUTDIR as required.
#                 (2) Statistic log file is collected for each host under a separate
#                     subdirectory of the $OUTDIR directory;
#                     the name of a subdirectory under $OUTDIR is the same as the
#                     corresponding host name.
#                 (3) rcp command is used to transfer statistics log files between
#                     hosts.
#                (4) A definition of pdstart operand and pdunit operand in definition
#                    files must not be written over several lines. If pdstart operand
#                     or pdunit operand is written over several lines in definition
#                     files, this command("pdstjacm") doesn't work correctly.
#
#                    (Example)
#                      pdstart -t SDS \ |    |
#                       -s sds01 \      | -> | pdstart -t SDS -s sds01 -x host01
#                       -x host01       |    |
#
#                      pdunit -u unt1 \ |    |
#                       -x host01 \     | -> | pdunit -u unt1 -x host01 -d "/HiRDB_S"
#                       -d "/HiRDB_S"   |    |
#
#                 (5) When system switching occurs, this command("pdstjacm") doesn't
#                     work correctly. Change the hostname(s) of the executing system
#                     to the one(s) of the standby system, in order to collect the
#                     statistics log files of the standby system.
#                 (6) Pdstjacm shell is getten HiRDB operational directory name from
#                     system common definition file ($PDCONFPATH/pdsys or specified
#                    first argument). If pdunit operand in definition files is written
#                     over several lines in definition files, pdstjacm shell may be
#                     executed using $PDDIR in .cshrc on several local hosts.
#                 (7) If pdstart operands and/or pdunit operands are specified wrong
#                     in definition files, this command("pdstjacm") doesn't work
#                     correctly.
#
#*******************************************************************************
##  Definition for canceling processing when Signal is received
trap "echo processing is canceled.; exit 1" 1 2 15

##  Definition of literals
OUTDIR=$PDDIR/spool/stjfiles      .....................#1
##  OUTDIR is directory under which statistics log files are collected.
##  Modify as required.
```

```
##  Checking of number of arguments
if [ $2 ] ; then
         echo  "Usage: $0 [input_file]"
         echo  "$0 processing is canceled."
         exit 1
fi

##  Checking of input definition file
if [ $# -eq 1 ]
then DEFFILE=$1
else DEFFILE=$PDCONFPATH/pdsys
fi
if [ ! -r $DEFFILE ] ; then
         echo "$DEFFILE is not found or cannot be read."
         echo "$0 processing is canceled."
         exit 1
fi

##  Set host name in ${proc_host}.
proc_host=`cat $DEFFILE | sed -n '/^[   ]*pdunit.*[   ]-x/{s/.*[   ]-x//;p;}' \
                        | awk '{print $1}' - | sort | uniq`
set ${proc_host:=no_host_name}

if [ $1 = no_host_name ] ; then
proc_host=`cat $DEFFILE | sed -n '/^[   ]*pdstart.*[   ]-x/{s/.*[   ]-x//;p;}' \
                        | awk '{print $1}' - | sort | uniq`
set ${proc_host:=no_host_name}

##  Validity checking of host name
if [ $1 = no_host_name ] ; then
         echo "No host name to be processed."
         echo "pdstjacm processing terminated."
         exit 1
fi
fi

##  Checking of transfer destination directory
if [ ! -d $OUTDIR ] ; then
         mkdir $OUTDIR
fi

##  Checking of transfer destination directory
if [ ! -d $OUTDIR ] ; then
         mkdir $OUTDIR
fi
```

```
##  Copy local host statistics log file   ..........#3
set ${proc_host}          ..........................1
while [ $# -ne 0 ]
do
     if [ ! -d $OUTDIR/$1 ] ; then
             mkdir $OUTDIR/$1
     fi
     proc_dir=`cat $DEFFILE | /bin/grep '[   ]-x[   ]*'$1'[   ]' \
              | sed -n '/^[   ]*pdunit/{s/.*[   ]-d//;p;}' | awk '{print $1}' -`
     if [ -z "$proc_dir" ] ; then
       proc_dir=`cat $DEFFILE | /bin/grep '[   ]-x[   ]*'$1'$' \
              | sed -n '/^[   ]*pdunit/{s/.*[   ]-d//;p;}' | awk '{print $1}' -`
     fi
##  Validity checking of pddir name on pdunit operand
         case ${proc_dir} in
                  /*)
                         rcp $1:$proc_dir/spool/pdstj1 $OUTDIR/$1
                         rcp $1:$proc_dir/spool/pdstj2 $OUTDIR/$1
                ;;
                *)
                         rcp $1:'$'PDDIR/spool/pdstj1 $OUTDIR/$1   ..........#2
                         rcp $1:'$'PDDIR/spool/pdstj2 $OUTDIR/$1
                ;;
         esac
##  rcp command is used to transfer statistics log files between hosts.
##  Environment must be set up so that remote shell can be executed between hosts.
         shift
done      ..................2

##  End of processing
echo "pdstjacm : Processing completed."
exit 0
```

#1: To create unload log files on a specified server machine, the `$PDDIR/spool/stjfiles` section is modified.

#2: If statistical log files cannot be collected on a specified server machine, compile the `'$'PDDIR/spool/pdstj1` and `'$'PDDIR/spool/pdstj2` parts.

#3: If control is passed to the standby HiRDB by the system switchover facility and this shell is executed as is, statistics log files will not be collected correctly. In such a case, the shell script from 1 through 2 must be modified as follows:

```
if[ ! -d $OUTDIR/HOST1 ] ; then
 mkdir $OUTDIR/HOST1
fi          # Copying statistics log file on HOST1
rcp HOST1:HOST1_pddir/spool/pdstj1 $OUTDIR/HOST1
rcp HOST1:HOST1_pddir/spool/pdstj2 $OUTDIR/HOST1

if [ ! -d $OUTDIR/HOST2 ] ; then
 mkdir $OUTDIR/HOST2
fi          # Copying statistics log file on HOST2
rcp HOST2:HOST2_pddir/spool/pdstj1 $OUTDIR/HOST2
rcp HOST2:HOST2_pddir/spool/pdstj2 $OUTDIR/HOST2
```

Repeat above script as many times as there are units whose statistics log is output. In this case, modify the following literals appropriately:

`HOST1`, `HOST2`: Statistics-log-output-host-name

`HOST1_pddir`, `HOST2_pddir`: Statistics-log-output-HiRDB-directory

20.1.5 When linked to an OLTP system

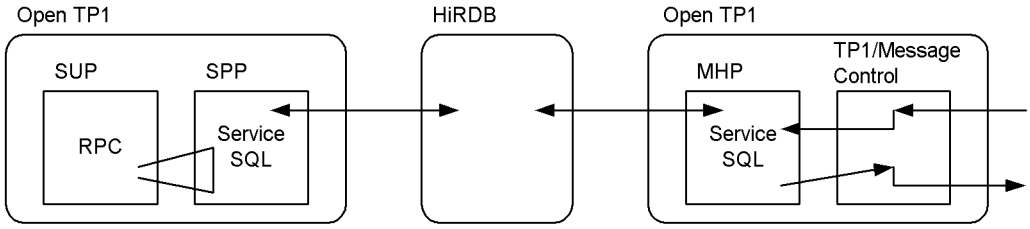
(1) When linked to OpenTP1

When HiRDB is linked to OpenTP1, the statistical information collection timing and the statistics analysis utility's editing unit change if the operation mode employed is as discussed in (a) below.

Linkage of HiRDB to OpenTP1 refers to accesses from an application operating under TP1/Server Base or TP1/Message Control (version 03-06 or later).

(a) Operation mode in which a UAP in a client/server environment accesses HiRDB as part of the server UAP's service execution

This includes an application that accesses HiRDB from one of the OpenTP1 services, such as SPP (Service Provider Program) or MHP (Message Handling Program):



In this operation mode, a UAP in OpenTP1 can form multiple services within the same UAP, and each service can access a HiRDB that is appropriate for its purposes. In such a case, the name of the service requested to SPP or MHP (the first service within the transaction accompanying HiRDB access) is added to the statistical information as

well as the UAP name of SPP or MHP (value of PDCLTAPNAME specified in the OpenTP1 user server definition for SPP or MHP). This service name is specified when SUP or TP1/Message Control issues a request using RPC; it is the name assigned to the service constituting SPP or MHP.

This service name can be used to determine the service that accessed HiRDB. It can also be used to establish correspondence with OpenTP1's transaction statistical information. Table 20-3 shows the correspondence between statistical information concerning a UAP accessing HiRDB and OpenTP1's statistical information.

Table 20-3: Statistical information for a UAP accessing HiRDB and OpenTP1's statistical information

Type of statistical information				Statistics log output timing	Statistics analysis utility's editing unit	DAT output file analysis
UAP statistical information	Other than OpenTP1's UAP			Each connection	Each UAP name • UAP executions count, average UAP execution time, executions count for each SQL statement, etc. • Total and average values for each item per UAP execution	Information can be analyzed using the transaction or UAP name as the key.
	OpenTP1's UAP	With no service name	Normal interface			
			XA interface	Each transaction		
		With service name	Normal interface	Each connection • During log output, name of the first service in the connection is added.	Each service name in UAP • Executions count for each SQL statement, etc.	Information can be analyzed using the transaction, UAP, or service name as the key.
			XA interface	Each transaction • Name of the first service in the log output period is added.		

<table>
<tr><th colspan="3">Type of statistical information</th><th>Statistics log output timing</th><th>Statistics analysis utility's editing unit</th><th>DAT output file analysis</th></tr>
<tr><td rowspan="2">SQL statistical information</td><td colspan="2">Other than OpenTP1's UAP</td><td rowspan="2">Each SQL</td><td rowspan="2">Each UAP name
• Total and average lengths of SQL objects
• Total and average number of object transmissions from front-end server to back-end server
• SQL executions count, processing time, number of rows processed</td><td rowspan="2">Information can be analyzed using the UAP name as the key.</td></tr>
<tr><td>OpenTP1's UAP</td><td>With no service name</td></tr>
<tr><td></td><td></td><td>With service name</td><td>Each SQL
• During log output, name of the first service in the transaction is added.</td><td>Each service name in UAP
• Summary of the above information for each service</td><td>Information can be analyzed using the UAP or service name as the key.</td></tr>
</table>

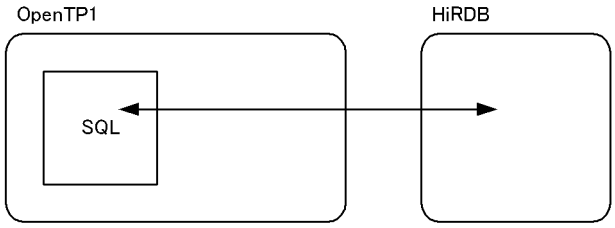
Type of statistical information			Statistics log output timing	Statistics analysis utility's editing unit	DAT output file analysis
SQL optimizing information	Other than OpenTP1's UAP		Each SQL	―	Information can be analyzed using the UAP name as the key.
	OpenTP1's UAP	With no service name			
		With service name	Each SQL • During log output, name of the first service in the transaction is added	―	Information can be analyzed using the UAP or service name as the key.
SQL object execution information	Other than OpenTP1's UAP	With no service name	Each SQL	―	Information can be analyzed using the UAP name as the key.
	OpenTP1's UAP	With service name	Each SQL • During log output, name of the first service in the transaction is added.	―	Information can be analyzed using the UAP or service name as the key.

Legend:

―: Not applicable

(b) Operation mode in which HiRDB is accessed directly from the main routine

This includes an application that accesses HiRDB directly from OpenTP1's SUP (Service Utilization Program):



In this operation mode, the statistical information is the same as for a normal application that is not connected to OpenTP1. The UAP name used in the statistical information is the value of PDCLTAPNAME specified in the OpenTP1 user server definition for SUP.

(2) When linked to TPBroker, TUXEDO or WebLogic Server

When HiRDB is linked to TPBroker, TUXEDO or WebLogic Server, the statistical information collection timing and the statistics analysis utility's editing unit change. Table 20-4 shows the statistical information collection timing when HiRDB is linked to TPBroker, TUXEDO or WebLogic Server.

Table 20-4: Statistical information collection timing when HiRDB is linked to TPBroker, TUXEDO or WebLogic Server

<table>
<tr><th colspan="3">Type of statistical information</th><th>Statistic log output timing</th><th>Statistics analysis utility's editing unit</th><th>DAT output file analysis</th></tr>
<tr><td rowspan="3">UAP statistical information</td><td colspan="2">UAP other than TPBroker, TUXEDO or WebLogic Server</td><td>Each connection</td><td rowspan="3">Each UAP name
• UAP executions count, average UAP execution time, executions count for each SQL, etc.
• Total and average values for each item per UAP execution</td><td rowspan="3">Information can be analyzed using the transaction or UAP name as the key.</td></tr>
<tr><td rowspan="2">TPBroker, TUXEDO or WebLogic Server UAP</td><td>Normal interface</td><td></td></tr>
<tr><td>XA interface</td><td>Each transaction</td></tr>
<tr><td rowspan="3">SQL statistical information</td><td colspan="2">UAP other than TPBroker, TUXEDO or WebLogic Server</td><td rowspan="3">Each SQL</td><td rowspan="3">Each UAP name
• Total and average lengths of SQL objects
• Total and average number of object transmissions from front-end server to back-end servers
• SQL executions count, processing time, number of rows processed</td><td rowspan="3">Information can be analyzed using the UAP name as the key.</td></tr>
<tr><td rowspan="2">TPBroker, TUXEDO or WebLogic Server UAP</td><td>Normal interface</td></tr>
<tr><td>XA interface</td></tr>
</table>

Type of statistical information			Statistic log output timing	Statistics analysis utility's editing unit	DAT output file analysis
SQL optimization information	UAP other than TPBroker, TUXEDO or WebLogic Server		Each SQL	Each UAP name • Average executions count and execution time	Information can be analyzed using the UAP name as the key.
	TPBroker, TUXEDO or WebLogic Server UAP	Normal interface			
		XA interface			
SQL object execution information	UAP other than TPBroker, TUXEDO or WebLogic Server		Each SQL	Each UAP name • Average executions count and execution time	Information can be analyzed using the UAP name as the key.
	TPBroker, TUXEDO or WebLogic Server UAP	Normal interface			
		XA interface			

20.2 Collecting tuning information from the system log

Executor: HiRDB administrator

(1) Tuning information that can be collected from the system log

Table 20-5 shows the tuning information that can be collected from the system log.

Table 20-5: Tuning information that can be collected from the system log

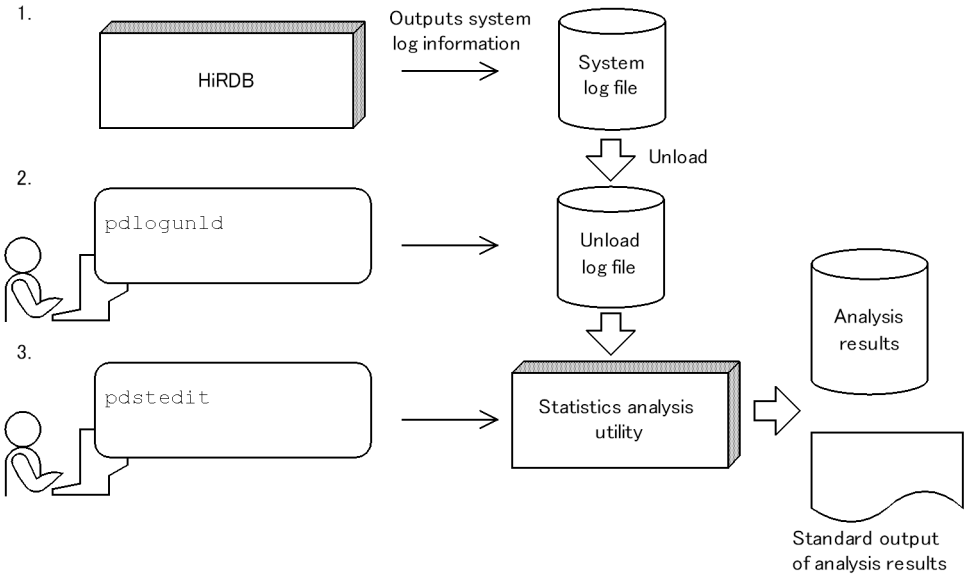
Type of statistical information		Explanation
Index statistical information	idx	Outputs information related to index page splits from within the index statistical information.
CONNECT/DISCONNECT statistical information*	cnc	Outputs (to a DAT file only) statistical information on CONNECTs and DISCONNECTs.

* In the case of a HiRDB/Parallel Server, this statistical information is output to the front-end server's system log file.

(2) Tuning information collection procedure

Figure 20-4 shows the procedure for collecting tuning information (collecting tuning information from the system log).

Figure 20-4: Procedure for collecting tuning information (collecting tuning information from the system log)



Explanation

1. HiRDB collects system log information in system log files.
2. Use the `pdlogunld` command to unload system log files and create unload log files.

   When unload log files are output to multiple server machines in a HiRDB/Parallel Server, the OS's `rcp` command (remote copy) or NFS (network file system) can be used to collect the unload log files.

   System log information can also be read directly by the statistics analysis utility. For details on the statistics analysis utility, see the manual *HiRDB Version 8 Command Reference*.
3. Execute the statistics analysis utility using the created unload log files as the input information. You can then use the execution results to tune HiRDB or UAPs.

20.3 Using the database condition analysis utility to collect tuning information

This section explains the procedures for using the database condition analysis utility to collect tuning information.

(1) Preparing for collecting the tuning information

Executor: HiRDB administrator

To use the database condition analysis utility to collect tuning information, the status of RDAREAs must be as shown in Table 20-6. For this reason, the status of RDAREAs should be checked with the `pddbls` command before the database condition analysis utility is executed.

A user with the DBA privilege only cannot check the status of RDAREAs. In such a case, the HiRDB administrator must be asked whether or not the RDAREAs are in the appropriate status for execution of the database condition analysis utility.

Table 20-6: RDAREA status for collection of tuning information by the database condition analysis utility

Type of RDAREA		Status of RDAREA
• Data dictionary RDAREA • Data dictionary LOB RDAREA • Registry RDAREA • Registry LOB RDAREA • List RDAREA		• Non-shutdown open status • Reference-enabled shutdown open status
• User RDAREA • User LOB RDAREA	RDAREA open attribute: `INITIAL`	• Non-shutdown open status • Reference-enabled shutdown open status
	RDAREA open attribute: `DEFER` or `SCHEDULE`	• Non-shutdown open status • Non-shutdown close status • Reference-enabled shutdown open status • Reference-enabled shutdown close status

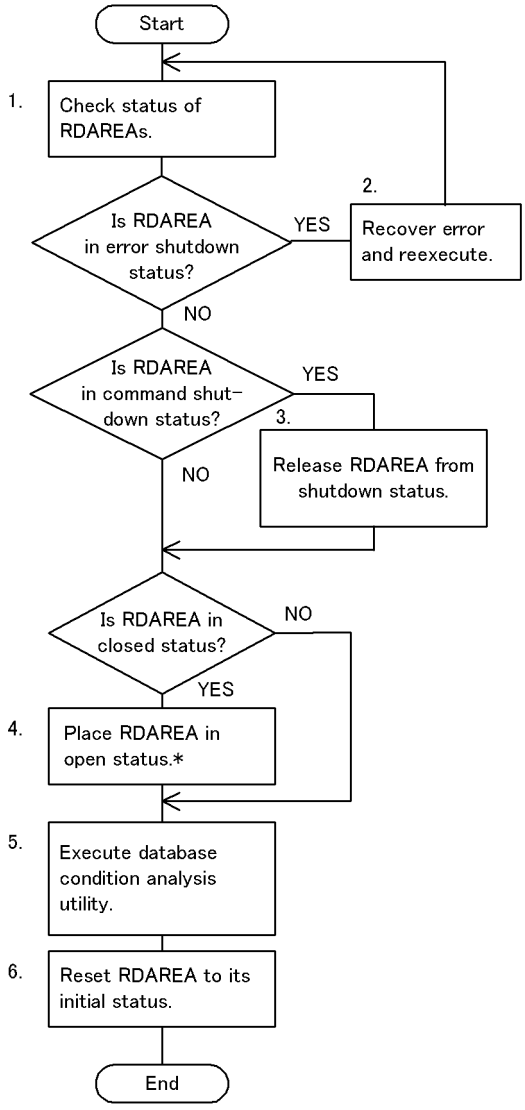
(2) Collecting the tuning information

Executor: User with the DBA privilege

This section explains the procedure for collection of tuning information (from entry of operation command to execution of the database condition analysis utility). Figure 20-5 shows the procedure for collecting tuning information (collecting tuning

information using the database condition analysis utility).

Figure 20-5: Procedure for collecting tuning information (collecting tuning information using the database condition analysis utility)



* If the RDAREA open attribute is `DEFER` or `SCHEDULE`, and the RDAREA is not in reference-possible hold status or shutdown, then it is not necessary to place it in open status.

Explanation

1. Use the `pddbls` command to check the status of the RDAREA.
2. In the case of an error shutdown, use the `pdrels` command after error recovery to release the shutdown status, then reexecute.
3. In the case of a command shutdown, use the `pdrels` command to release the RDAREA from shutdown status.
4. If in closed status, use the `pdopen` command to place in open status.
5. Execute the database condition analysis utility. Perform tuning on the basis of the results of this execution.
6. If the RDAREA was placed in open status in step 4, use the `pdclose` command to return to closed status. If command shutdown was used in step 3, use the `pdrels` command to release the command shutdown.

Chapter 21. Tuning

This chapter explains how to tune a database.

Databases should be tuned in order to maintain optimum performance. However, the available memory and disk space may not be adequate to achieve the desired level of performance. Therefore, a balance between performance on the one hand and memory and disk space requirements on the other hand must be considered when the system is tuned.

This chapter contains the following sections:

21.1 Tuning global buffer pools

The specifications for global buffer pools can have a significant effect on a disk's data I/O performance. For this reason, global buffer pools should be specified so that high hits rates are achieved. This section discusses the items to be evaluated in order to improve the buffer hits rates for the global buffer pools.

If the items described here are evaluated in conjunction with HiRDB file information related to database manipulations, performance can be improved most efficiently.

Information to be collected

Global buffer pool statistical information is collected. The following resources are provided for collecting this information:

- `pdbufls` command
- Statistics analysis utility

The `pditvtrc` command can be used to execute this command and utility periodically on a regular basis.

21.1.1 Using the pdbufls command to collect statistical information

(1) Check the global buffer pool's hits rate (HIT)

Purpose

This information is collected to determine whether or not the number of buffer sectors in a global buffer pool is appropriate. This is the most important item for tuning a global buffer pool.

Evaluating the analysis results

Check to see if the global buffer pool hits rate (HIT) is 80% or higher.

Actions to be taken

Global buffer pool hits rate is below 80%

Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).

Global buffer pool hits rate is 80% or better

There is no problem; however, to improve the hits rate even further, increase the number of global buffer sectors.

RDAREAs storing tables with no buffering effects should be combined into a single RDAREA; otherwise, more memory may be used than is necessary.

(2) Check the update requests hits rate (HIT)

Purpose

This information is collected in order to evaluate the validity of the number of global buffer sectors and the deferred write processing that is executed internally by HiRDB.

When a data insertion, updating, or deletion request is issued for a database, the corresponding data is updated in a global buffer pool. Therefore, the ratio of the number of times requested data was found in a global buffer pool to the total number of update `GET` requests has an effect on throughput.

Evaluating the analysis results

Evaluate the update requests hits rate with respect to the total number of issued `INSERT`, `UPDATE`, and `DELETE` statements, as reported in the UAP information output by the statistics analysis utility:

1. Check that the update requests hits rate is not too low
2. Check that the reference buffer hits rate is not too low even if the update requests hits rate is high.

The update requests hits rate is obtained with the following formula:

`Update requests hits rate (%) =` (*update GET requests hits count* ÷ *update GETs count*) × `100`

Actions to be taken

When the update requests hits rate is low

1. Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
2. Re-evaluate the value specified for the `pdbuffer` operand. If multiple RDAREAs are allocated to one global buffer, separate the RDAREAs for tables to be updated from the RDAREAs for tables to be referenced, and allocate a global buffer to each type of RDAREA. If one RDAREA is allocated to one global buffer, store indexes and tables in separate RDAREAs and allocate a global buffer dedicated to indexes. Or, row-partition the table.
3. When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or it is omitted), decrease the value of the `pdbuffer` operand's `-w` option (percentage of pages output in deferred write processing).

When the update requests hits rate is high

There is no problem; however, to improve the hits rate even further, increase

the number of global buffer sectors.

RDAREAs storing tables with no buffering effects should be combined into a single RDAREA; otherwise, more memory may be used than is necessary.

(3) Check the reference requests hits rate (HIT)

Purpose

This information is collected in order to evaluate the validity of the reference requests hits rate. If needed data is not found in the global buffer, the system retrieves database data from the RDAREA and sets it in the buffer. If the number of times requested data is found in the global buffer pool is small, the number of I/O operations increases, affecting adversely the transaction processing performance.

Evaluating the analysis results

Determine whether the reference requests hits rate is close to 80%. The reference requests hits rate is obtained with the following formula:

`Reference requests hits rate (%) =` (*reference requests hits count* ÷ *reference GETs count*) × `100`

Actions to be taken

When the reference requests hits rate is 80% or lower

1. Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
2. Re-evaluate the value specified for the `pdbuffer` operand. If multiple RDAREAs are allocated to one global buffer, allocate one global buffer to a frequently accessed table. If one RDAREA is allocated to one global buffer, row-partition the table.
3. If the reference requests hits rate is too low and the reference page flush count is too high for the update requests hits rate, specify `pd_dbbuff_lru_option=MIX`.
4. When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or it is omitted), decrease the value of the `pdbuffer` operand's `-w` option (percentage of pages output in deferred write processing).

When the reference requests hits rate is higher than 80%

There is no problem; however, to improve the hits rate even further, increase the number of global buffer sectors. RDAREAs storing tables with no buffering effects should be combined into a single RDAREA; otherwise, more memory may be used than is necessary.

(4) Check the update buffer flushes count (UPFLS)

Purpose

This information is collected in order to evaluate the validity of the update buffer flushes count. If the number of times a free buffer is created to read new pages after writing an updated buffer into a HiRDB file (update buffer flushes count) is high for the number of operations, the number of I/O operations increases, adversely affecting the transaction processing performance.

Evaluating the analysis results

The number of updated global buffer sectors may have increased along with the update buffer flushes count and the buffers count for reloading the same data (reference GETs count) for one of the following reasons:

- The number of global buffer pool sectors is small
- The ratio of the number of updating processes to the number of operations is high

Determine whether the update buffer flushes count is high or low for the number of operations.

Actions to be taken

When the update buffer flushes count is high

1. Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
2. Re-evaluate the value specified for the `pdbuffer` operand. If multiple RDAREAs are allocated to one global buffer, separate the RDAREAs for tables to be updated from the RDAREAs for tables to be referenced, and allocate one global buffer to each type of RDAREA. If one RDAREA is allocated to one global buffer, store indexes and tables in separate RDAREAs and allocate a global buffer dedicated to indexes. Or, row-partition the tables.
3. When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or it is omitted), decrease the value of the `pdbuffer` operand's `-w` option (percentage of pages output in deferred write processing).
4. When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or it is omitted), decrease the value of the deferred write trigger request rate (`pd_dbbuff_rate_updpage` operand).

When the update buffer flushes count is low

There is no problem; however, to improve the hits rate even further, increase

the number of global buffer sectors.

RDAREAs storing tables with no buffering effects should be combined into a single RDAREA; otherwise, more memory may be used than is necessary.

(5) Check the reference buffer flushes count (RFFLS)

Purpose

This information is collected in order to evaluate the validity of the reference buffer flushes count. If the number of times a free buffer is created to read new pages after writing a reference buffer into a HiRDB file (reference buffer flushes count) is high for the number of operations, the number of I/O operations increases, affecting adversely the transaction processing performance.

Evaluating the analysis results

When the reference buffer flushes count is too high for the number of operations

The number of reference buffers may have increased for one of the following reasons:

- The number of global buffer pool sectors is small.
- The ratio of the number of referencing processes to the number of operations is high.

As a result, buffer invalidation is occurring frequently in memory, or the same data is being fetched frequently in the buffer.

When the reference requests hits rate is low and the reference buffer flushes count is high

The number of update buffers may have increased for one of the following reasons:

- The number of global buffer pool sectors is small.
- The ratio of the number of update processes to the total number of processes is high (applicable when `pd_dbbuff_lru_option=SEPARATE` is specified or this operand is omitted).

As the amount of buffer space to be updated increases, the reference requests hits rate decreases. As a result, buffer invalidation occurs frequently in memory, or the same data is fetched frequently in the buffer.

Actions to be taken

When the reference buffer flushes count is high

If the reference requests hits rate is high, there is no problem. If it is low, the following actions can be taken:

- Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
- Specify `pd_dbbuff_lru_option=MIX`.

When the reference buffer flushes count is low

There is no problem.

(6) Check the real READs count (READ) and real WRITEs count (WRITE)

Purpose

This information is collected in order to evaluate the validity of the number of disk I/O operations. If the number of disk I/O operations (HiRDB file I/O operations) is high for the total amount of processing, transaction processing performance and throughput are affected adversely. The number of I/O operations can be reduced efficiently by evaluating this information together with the information for each RDAREA that is provided as the HiRDB file information related to database accesses.

Evaluating the analysis results

When there are many input operations

1. If many RDAREAs use the same global buffer and each RDAREA is accessed frequently, input operations may occur frequently due to a low buffer hits rate.
2. If the reference requests hits rate is low, the page input operation may occur frequently due to the following factors:
   - The number of global buffer pool sectors is small (applicable when `pd_dbbuff_lru_option=MIX` is specified).
   - The ratio of the number of updating processes to the total amount of processing is too high for the reference buffer (applicable when `pd_dbbuff_lru_option=SEPARATE` is specified or this operand is omitted).

When there are many output operations

1. If greater importance is placed on the update buffer hits rate than on the reference requests hits rate, the corresponding global buffer may be requested for update processing more frequently.
2. If the update `GETs` count is greater than the reference `GETs` count, adverse effects may be caused by update buffer flushing, as described in (4) above.

Actions to be taken

When the reference requests hits rate is low and the number of input operations

is high

1. Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
2. Re-evaluate the value specified for the `pdbuffer` operand. If multiple RDAREAs are allocated to one global buffer, allocate one global buffer to a frequently accessed table. If the same global buffer pool is allocated to an index and table, allocate a global buffer pool dedicated to the index. If one RDAREA is allocated to one global buffer, row-partition the table.
3. When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or it is omitted), increase the value of the `pdbuffer` operand's `-w` option (percentage of pages output in deferred write processing).
4. Specify `pd_dbbuff_lru_option=MIX`.

When there are many output operations

1. `pd_dbsync_point=commit` specified

   There is no problem if the update request hits rate is high. If it is low, increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).

2. `pd_dbsync_point=sync` specified or this operand omitted

- Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
- When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or it is omitted), increase the value of the `pdbuffer` operand's `-w` option (percentage of pages output in deferred write processing). Also increase the synchronization point interval.

(7) Check the buffer lock-release waits count (WAITL)

Purpose

This information is collected in order to evaluate the validity of the global buffer pool allocations to RDAREAs. If the number of concurrently executed transactions increases, contention occurs on accesses to the same global buffer pool, resulting in lock-release wait status. When the number of buffer lock-release waits increases, transaction processing time also increases, and the global buffer pool allocations should be reevaluated.

Evaluating the analysis results

1. If the buffer lock-release waits count is high for the total amount of

processing, there may be many users using the same global buffer pool, resulting in high buffer access workload.

2. If the buffer lock-release waits count is high for the total amount of processing, multiple tables may be defined for the same global buffer pool, resulting in concentration of buffer accesses.

Actions to be taken

1. If multiple tables are stored in one RDAREA, allocate the storage RDAREA to each of the applicable global buffer pools.
2. If multiple RDAREAs are allocated to one global buffer pool, reduce the number of RDAREAs allocated to that global buffer pool.
3. If a table and index are both stored in the same RDAREA, store them in separate RDAREAs, and allocate a global buffer dedicated to the index.

(8) Check the prefetch input pages count (PRRED), prefetch hit pages count (PRHIT), and prefetch hits rate (HIT)

Purpose

This information is collected in order to evaluate the validity of the prefetch facility. In the case of retrieval of all items or retrieval by range, processing time can be reduced by using the prefetch facility. When a large amount of data is to be retrieved, the prefetch facility can influence the processing time.

Evaluating the analysis results

If the prefetch hits rate is 80% or less, the data or index storage pages for the table being retrieval may be fragmented.

Actions to be taken

Reorganize the table being retrieved. If the hits rate is extremely low but the table cannot be reorganized, do not use the prefetch facility (either omit the `pdbuffer -m` operand or specify 0 in the `-m` operand).

(9) Check the prefetch buffer shortages count (PRINS)

Purpose

This information is collected in order to evaluate the validity of the prefetch facility. If a shortage occurs in the buffer for prefetch processing, a user who is unable to execute a batch input operation cannot take advantage of prefetch processing.

Evaluating the analysis results

It is no problem if the prefetch buffer shortages count is 0. If it is other than 0, it can be considered that the maximum number of simultaneous prefetches is greater than the value specified in the `pdbuffer` operand's `-m` option.

Actions to be taken

Increase the value of the `pdbuffer` operand's `-m` option (maximum number of simultaneous prefetches). Also, reevaluate the memory requirements, because if this value is increased, the shared memory for global buffer pools also increases.

21.1.2 Using the statistics analysis utility to collect statistical information

(1) Check the input waits count (WAITR) and output waits count (WAITW)

Purpose

This information is collected in order to evaluate the validity of global buffer allocations to RDAREAs. If contention results because of multiple requests to access the same page, data I/O operations are placed in wait status for the following reasons:

1. Requested data must be read from the HiRDB file, because the appropriate page cannot be found in the global buffer pool; this results in an input wait.
2. Even when the appropriate page is found in the global buffer pool, but contention occurs between a referencing request and an updating request during a HiRDB file output operation, the HiRDB file is placed in output completion wait status.

Evaluating the analysis results

When the input waits count is high

Input waits may occur frequently for the following reasons:

1. I/O operations concentrate because HiRDB files in one or more RDAREAs are allocated on the same disk.
2. If multiple tables are stored in one RDAREA, many processing requests are issued for that RDAREA.

When the output waits count is high

Output waits may occur frequently for the following reasons:

1. Deferred write processing on the appropriate page and HiRDB file output processing result in contention.
2. The number of update buffers increases in the global buffer pool, because a large amount of global buffer pool space is used for data load processing by the database load utility or for reload processing by the database reorganization utility.

Actions to be taken

1. If multiple tables are stored in one RDAREA, allocate the RDAREAs to each

of the applicable global buffer pools.

2. If a table contains a large amount of data, consider dividing the table or index into multiple partitions.
3. Distribute HiRDB files in one or more RDAREAs onto multiple disks.
4. If a large amount of data is being loaded by the database load utility or is being reloaded by the database reorganization utility, increase the number of batch output pages (`-n` option). Note that the database reorganization utility is applicable to a table whose data attribute is FIX.
5. When performing deferred write processing (when the `pd_dbsync_point` operand is specified as `sync` or is omitted) while the output waits count is high, decrease the value of the deferred write trigger request rate (`pd_dbbuff_rate_updpage` operand).

(2) Check the maximum concurrent request buffer sectors count (MAXB) and the buffer shortages count (BFINS)

Purpose

This information is collected in order to evaluate the validity of the number of buffer sectors in a global buffer pool. If the entire global buffer used for reading pages is unavailable, HiRDB cancels a transaction that requests a new page; this has an adverse effect on transaction throughput.

Evaluating the analysis results

If any of the following conditions is true, the number of buffer sectors is too low:

1. The maximum concurrent request buffer sectors count is close to or equal to the global buffer sectors count, and the buffer shortages count is 0 (this does not apply when the reference or update buffer hits rate is high; maximum performance seems to be achieved when the reference or update buffer hits rate is high).
2. The maximum concurrent request buffer sectors count is close to or equal to the global buffer sectors count, and the buffer shortages count is high.
3. The buffer shortages count is not 0.

Actions to be taken

1. Increase the value of the `pdbuffer` operand's `-n` option (number of global buffers).
2. If I/O operations concentrate on a particular RDAREA, allocate a dedicated global buffer pool instead of sharing the same global buffer pool among multiple RDAREAs.

(3) Check the number of synchronization point dump pages (SYNCW)

Purpose

This information is collected in order to evaluate the validity of the synchronization point dump output interval. Normally, all updated pages are output from a global buffer pool to the database during a synchronization point dump, which has an adverse effect on transaction performance. To prevent this, HiRDB executes internal processing (pre-sync processing and deferred write processing) and reduces the number of pages to be output during synchronization point dumps. Therefore, tuning the global buffer has an effect on the synchronization point dump output interval.

Evaluating the analysis results

Check that the number of pages output during a synchronization point dump is no greater than 50% of the global buffer sectors.

Actions to be taken

Increase the value of the `pdbuffer` operand's `-w` option (page rate of output by deferred write processing).

(4) Check the lock-release contention rate (SLEPR) in global buffer lock processing

Purpose

The purpose is to determine how much contention there is for locking of global buffers. If the contention rate is high, this will be one factor in the increased probability of global buffer lock-waiting and the resulting reduced performance.

Evaluating the analysis results

If the lock-release contention rate (`SLEPR`) in global buffer lock processing is greater than 1, the contention for global buffer lock processing can be considered to be high.

Actions to be taken

Check the value of the `pd_dbbuff_lock_release_detect` operand, and take one of the following measures:

- When `pd_dbbuff_lock_release_detect=interval`
  1. Do the following to reduce the CPU usage to below 70% and within the allowed operating range:
     - Increase the value of the `pd_dbbuff_lock_spn_count` operand.
     - Decrease the value of the `pd_dbbuff_lock_interval` operand.
  2. Review the method of allocation of global buffers.

Allocate dedicated global buffers to an RDAREA that stores tables that are accessed frequently. If dedicated global buffers are already allocated, either row partition the tables and allocate dedicated global buffers to each RDAREA or store tables with a high frequency of accesses in different RDAREAs, and allocate dedicated global buffers to those RDAREAs.

- When `pd_dbbuff_lock_release_detect=pipe` (default value)

  1. Specify `interval` for the `pd_dbbuff_lock_release_detect` operand. Then execute step 1 above.
  2. Execute step 2 above.

(5) Check the average value for the sleep processing execution count in buffer lock processing (SLEPA)

Purpose

The purpose is to determine whether or not a delay in the global buffer lock processing is causing an extremely long processing time for some jobs during concurrent job execution.

Evaluating the analysis results

If the following conditions are satisfied, it may be possible to reduce the global buffer lock processing time:

- The average value for the sleep processing execution count in buffer lock processing (SLEPA) is at least 1 but less than 2.
- There is no difference in the global buffer lock processing time between cases in which sleep processing is executed once or more, and cases in which sleep processing is never executed.

Actions to be taken

Check the value of the `pd_dbbuff_lock_release_detect` operand, and take one of the following measures:

- When no transaction stops responding during synchronization point processing

  1. Specify `interval` in the `pd_dbbuff_lock_release_detect` operand.
  2. When the CPU usage is below 70% and within the allowed operating range, do the following:

     - Specify 10 in the `pd_dbbuff_lock_interval` operand.

     - Specify 100 in the `pd_dbbuff_lock_spn_count` operand.

     If the value of SLPA remains 2 or greater after these settings have been

specified, increase the `pd_dbbuff_lock_spn_count` operand value until the value of SLPA becomes less than 2.

3. Review the method of allocating global buffers.

   Allocate dedicated global buffers to an RDAREA that stores tables that are accessed frequently. If dedicated global buffers are already allocated, either row partition the tables and allocate dedicated global buffers to each RDAREA, or store tables with a high frequency of accesses in different RDAREAs and allocate dedicated global buffers to those RDAREAs.

- When transactions stop responding during synchronization point processing

  See *(6) Check the buffer pool lock time during synchronization point processing (SYNCL)*.

(6) Check the buffer pool lock time during synchronization point processing (SYNCL)

Purpose

If a transaction stops responding during synchronization point processing, there may be contention for locking the global buffer between search processing in the buffer subject to synchronization point processing, and buffer access by the transaction. If there is contention, buffer access by a transaction is placed on hold during the buffer pool lock time during synchronization point processing (SYNCL).

Evaluating the analysis results

If a transaction stops responding during synchronization point processing, determine whether or not the buffer pool lock time during synchronization point processing (SYNCL) is too long for the lock release wait time.

Actions to be taken

Reduce the `pd_dbsync_lck_release_count` operand value, so as to adjust the buffer pool lock time during synchronization point processing (SYNCL) to be within the allowable limits of the lock release wait time.

(7) Check the take-over count of database write processing by the reference request hit during synchronization point processing (ALTRW)

Purpose

If a referencing transaction stops responding during synchronization point processing, delayed output processing may be taking place at an extension of transaction at the time of the reference request hit on the buffer that is subject to synchronization point processing.

Evaluating the analysis results

If a referencing transaction stops responding during synchronization point processing, check to see if this is caused by the take-over count of database write processing by the reference request hit during synchronization point processing (ALTRW).

Actions to be taken

By specifying `Y` in the `pd_dbsync_altwrite_skip` operand, you can suppress database write processing by the reference request hit during synchronization point processing. However, this increases the synchronization point acquisition processing time. Hitachi recommends that you use the facility for parallel writes in deferred write processing to distribute the write processing workload. For details about the facility for parallel writes in deferred write processing, see the manual *HiRDB Version 8 Installation and Design Guide*.

21.2 Tuning deferred write processing

When `sync` is specified in the `pd_dbsync_point` operand in the system common definition or the `pd_dbsync_point` operand is omitted, HiRDB executes deferred write processing to reduce the number of disk output operations. Deferred write processing improves performance by processing RDAREAs distributed among multiple disks in units of disks.

Therefore, when character special files are used for a HiRDB file system area in which RDAREAs are allocated and the RDAREAs are allocated on a single disk, the benefits of deferred write processing are not obtained.

Information to be collected

Deferred write processing statistical information is collected by the statistics analysis utility.

(1) Check the parallel level for each disk volume (MAX ,PMIN)

Purpose

This information is collected in order to evaluate the validity of table and global buffer allocations to RDAREAs on the basis of the level of parallel processing requests for multiple pages in the global buffers. When updated pages are written during deferred write processing, HiRDB executes I/O operations on each disk concurrently in order to improve performance.

Evaluating the analysis results

If multiple RDAREAs use a global buffer, it should be verified that the minimum parallel level value for each disk volume is not 1. If the RDAREAs span multiple disks, only some of the RDAREAs may be updated.

Actions to be taken

1. If update processing concentrates on a particular RDAREA, allocate a separate global buffer for that RDAREA.
2. If both index and table are stored in the same RDAREA, store them in separate RDAREAs and allocate separate global buffers to those RDAREAs.

(2) Check the average value (AVG)

Purpose

This information is collected in order to evaluate the validity of the global buffers on the basis of the average updated pages output during the following processing:

Trigger output

When the number of updated pages in the global buffers reaches a given

number of buffers, the pages are output to disk.

Note that when the deferred write trigger request rate is enabled, pages are output to disk when the specified value is reached. You specify this value in the `pd_dbbuff_rate_updpage` operand.

The number of pages output to disk is calculated by HiRDB on the basis of the output page percentage specified in the `pdbuffer` operand's `-w` option.

Pre-sync output

A pre-sync point is set internally in order to reduce the number of pages to be output to the disk at a synchronization point. Pre-sync processing involves writing updated pages from global buffer to disk prior to a synchronization point so that fewer pages need to be output during the synchronization point dump.

Synchronization point dump output

All updated pages are output from global buffer to disk at a synchronization point. Pre-sync output can greatly reduce the time required for output processing during a synchronization point dump.

Database synchronization point dump output

The contents of the update buffer are committed to the database at a synchronization point trigger at which a synchronization point dump cannot be enabled.

RDAREA synchronization point dump output

All updated pages corresponding to an RDAREA are output to the disk.

Evaluating the analysis results

When the average number of output pages is too small

Check if the buffer flushes count is greater than the page outputs count during deferred write processing. If so, check the global buffer pool whose data is not updated frequently.

1. If the update page buffer hits rate is high and the number of pages output during deferred write processing is small, there should be no problem.
2. If the update page buffer hits rate is low and the number of pages output during deferred write processing is small, the synchronization point dump interval should be reduced.

The following formula can be used to obtain the average number of output pages:

$$\text{Average (AVG)} = \frac{a - \sum_{i=1}^{n} b_i}{c - d}$$

a: Output pages count (`OUT PAGE`)

b: Number of synchronization point dump pages (`SYNCW`)

c: Deferred write processing activations count (`EXEC`)

d: Synchronization points count (`SYNC`)

n: Number of global buffer pools defined

For a HiRDB/Parallel Server

A specific back-end server may have a high output pages count for one of the following reasons:

1. A frequently updated table is stored in the back-end server.
2. Update processing is concentrated on a specific key range of a partitioned table.

Actions to be taken

When write operations are concentrated on a specific global buffer:

1. If multiple RDAREAs are allocated to the global buffer pool, allocate separate global buffers for the RDAREAs.
2. If a specific server is used to update multiple tables, each table should be stored in a different back-end server.

When multiple RDAREAs are defined on the same disk

Allocate each RDAREA on a separate disk to improve the I/O parallel level.

21.3 Tuning the synchronization point processing time when deferred write processing is used

When deferred write processing is used, it may take some time to accomplish synchronization point processing. This section describes how to reduce the synchronization point processing time.

21.3.1 Tuning procedure

When a large amount of data is updated, the number of update buffers that must be applied to the database during synchronization point processing (number of update pages in the global buffer) increases, resulting in a corresponding increase in the time required for synchronization point processing. If synchronization point processing is skipped two or more times in succession, the number of system log files that cannot be overwritten increases. If there is no swappable file, HiRDB (or the unit in the case of a HiRDB/Parallel Server) terminates abnormally.

To avoid this, it is important to complete synchronization point processing within the amount of time that was estimated at the design stage. If the following conditional expression is satisfied, there is no need to perform tuning:

- Synchronization point acquisition interval (time) > synchronization point processing time

If synchronization point processing is skipped due to a delay in deferred write processing, the `KFPS02179-I` message (`factor code = A01-01`) is displayed. In such a case, perform tuning according to the procedure described below.

(1) Acquiring the tuning information

Acquire the following tuning information:

1. Information about CPU and disk I/O operations
2. Statistical information about the global buffer
3. Statistical information about the deferred write processing

To acquire 1, use the appropriate OS function. To acquire 2 and 3, use the statistics analysis utility.

(2) Checking the statistical information about the deferred write processing

Check the statistical information about the deferred write processing whose cause (`CAUSE`) is `S` (synchronization point processing). If one of the following conditions is satisfied, proceed to (3). If neither of the conditions is satisfied, proceed to (9).

- When the facility for parallel writes in deferred write processing is not used

  Total WRITE time (DWSUM, DWSUMM) > synchronization point acquisition

interval (time)

- When the facility for parallel writes in deferred write processing is used

  Parallel WRITE time (DWPARA, DWPARAM) > synchronization point acquisition interval (time)

Obtain the synchronization point acquisition interval from the difference between the output times of the following messages:

- `KFPS02183-I` (message displayed when synchronization point processing is completed)
- `KFPS02179-I` (message displayed when synchronization point processing is skipped)

(3) Checking the average WRITE unit time

If the average WRITE unit time for the statistical information about deferred write processing (`DWAVG` and `DWAVGM`) is extremely poor compared to the disk write performance, do the following (if these do not apply, proceed to (4)):

- Check for a disk failure.
- Because a problem such as disk contention may have occurred, use OS functions to acquire tuning information about input/output operations, and perform tuning based on the results. If there is no leeway for tuning, proceed to (4).

(4) Checking the KFPS02179-I message

If synchronization point processing is skipped, the `KFPS02179-I` message is displayed. If any of the utilities listed below was executing during the period in which this message was displayed, consider changing the operating method (if this does not apply, proceed to (5)):

- When the database load utility, database reorganization utility, or rebalancing utility is executing

  Use the local buffer by specifying the `-n` option for the utility's execution. If the local buffer cannot be used, proceed to (5).

- When the free page release utility is executing

  Specify the `-p` option for the utility's execution. If the `-p` option cannot be specified, proceed to (5).

(5) Evaluating the synchronization point acquisition interval

Evaluate whether or not the synchronization point acquisition interval can be made longer. If it cannot be made longer, proceed to (6). You use the `pd_log_sdinterval` operand to change the synchronization point acquisition interval.

(6) Tuning the deferred write trigger

Perform the tuning described in *21.3.3(1) Reducing the deferred write trigger interval* and in *21.3.3(2) Increasing the update page output rate during deferred write trigger*. If the problem is not resolved after tuning, proceed to (7).

(7) Using the facility for parallel writes in deferred write processing

If there is enough room in the CPU, perform the tuning described below. If there is not enough room in the CPU, proceed to (8).

- When the facility for parallel writes in deferred write processing is not used

  Specify the `pd_dfw_awt_process` operand to use the facility for parallel writes in deferred write processing. If this does not resolve the problem, proceed to (8).

- When the facility for parallel writes in deferred write processing is used

  Perform the tuning described in *21.3.3(3) Increasing the number of parallel WRITE processes during deferred write processing*. If this does not resolve the problem, proceed to (8).

(8) Evaluating a limit on the number of update buffers

Specify the `pd_dfw_syncpoint_skip_limit` operand to limit the number of update buffers. In this case, note that when the skip count for synchronization point processing reaches the maximum value, the performance of an update transaction decreases because the update buffer is output at the extension of the update transaction.

If the `pd_dfw_syncpoint_skip_limit` operand cannot be specified, evaluate a disk enhancement.

(9) Tuning the global buffer

If any of the following events has occurred, tune the global buffer:

- Buffer lock-release wait (`WAITL`) occurs frequently.
- Output wait (`WAITW`) occurs frequently.
- The lock-release contention rate (`SLEPR`) in global buffer lock processing is high.
- The update buffer flushes count (`UPFLS`) has increased.

For details about the above information and the tuning of the global buffer, see *21.1 Tuning global buffer pools*.

21.3.2 How to interpret statistical information about deferred write processing

Information to be referenced

Use the statistics analysis utility (`pdstedit`) to acquire DAT-format files that contain statistical information about deferred write processing, and then check the

following information:

- Execution time (DWTOTAL, DWTOTALM)
- Total WRITE time (DWSUM, DWSUMM)
- Parallel WRITE time (DWPARA, DWPARAM)
- WRITE unit time

  Minimum (DWMIN, DWMINM)

  Maximum (DWMAX, DWMAXM)

  Average (DWAVG, DWAVGM)
- WRITE count (DWEXEC)

(1) Execution time (DWTOTAL, DWTOTALM)

This is the total time required for deferred write processing. DWTOTAL displays the time in seconds, and DWTOTALM displays only the microseconds portion of the time.

Purpose

This information is collected to determine whether the deferred write processing performance is appropriate.

Evaluating the analysis results

If the deferred write delay message (KFPS02179-I factor code = A01-01) is output, take the following action:

Action to be taken

For details about the action to be taken, see *21.3.1 Tuning procedure*.

(2) Total WRITE time (DWSUM DWSUMM)

This is the total time required for writing during deferred write processing. DWSUM displays the time in seconds, and DWSUMM displays only the microseconds portion of the time.

Purpose

This information is collected in order to determine whether tuning the process count when using the facility for parallel writes in deferred write processing had any effect compared to the parallel WRITE time (DWPARA DWPARAM).

Evaluating the analysis results

Check the values of WRITE unit time (maximum) (DWMAX DWMAXM), WRITE unit time (minimum) (DWMIN DWMINM), and WRITE count (DWEXEC).

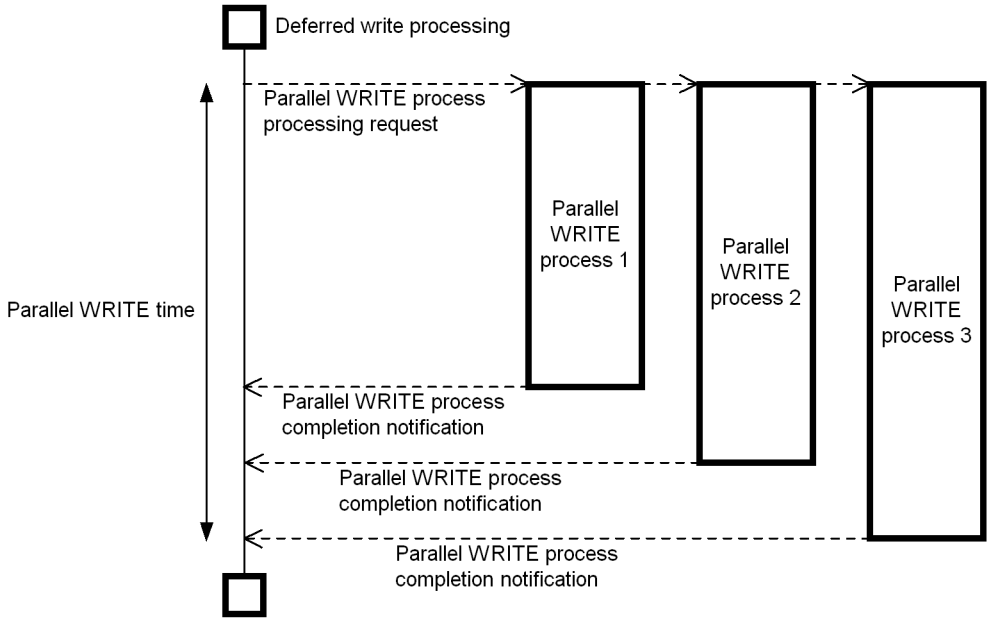
Action to be taken

Check the values of WRITE unit time (maximum) (DWMAX DWMAXM), WRITE unit time (minimum) (DWMIN DWMINM), and WRITE count (DWEXEC), and take appropriate action.

(3) Parallel WRITE time (DWPARA DWPARAM)

This is the length of time from when a request to process multiple parallel WRITE processes was issued to when completion notifications for all parallel WRITE processes have been received. DWPARA displays the time in seconds, and DWPARAM displays only the microseconds portion of the time. Figure 21-1 shows the concept of parallel WRITE time.

Figure 21-1: Concept of parallel WRITE time



If the facility for parallel writes in deferred write processing is disabled, the parallel WRITE time is 0.

Purpose

This information is collected in order to check the effects of using the facility for parallel writes in deferred write processing.

Evaluating the analysis results

If the following condition is true, use of the facility for parallel writes in deferred write processing is considered to be effective:

- Execution time (value of DWTOTAL + DWTOTALM) > parallel WRITE time

(value of DWPARA + DWPARA)

For accurate checking, measure the execution time value of DWTOTAL + DWTOTALM) at the following times:

- Before and after the facility for parallel writes in deferred write processing is used
- Before and after the pd_dfw_awt_process operand value is changed

If the execution time is reduced, use of the facility for parallel writes in deferred write processing is considered to be effective.

Actions to be taken

If use of the facility is effective, perform the tuning described in *21.3.3(3) Increasing the number of parallel WRITE processes during deferred write processing*.

If the use of the facility is not effective, do one of the following:

- Stop using the facility for parallel writes in deferred write processing.
- Restore the pd_dfw_awt_process operand to its original value

(4) WRITE unit time minimum (DWMIN DWMINM)

This is the shortest write time on a page when write operations occur on multiple pages. DWMIN displays the time in seconds, and DWMINM displays only the microseconds portion of the time.

Purpose

This information is collected in order to check the validity of disk performance.

Evaluating the analysis results

Compare with disk performance for evaluation. If you use the facility for parallel writes in deferred write processing or you change the number of parallel WRITE processes, compare disk performance before and after the facility is used, or before and after the change.

Actions to be taken

If the performance is poor compared to the disk performance, use OS functions to acquire input/output-related tuning information, and then perform tuning on the basis of that information. If the disk performance was reduced after the facility for parallel writes in deferred write processing was used or after the number of parallel WRITE processes was changed, check the parallel WRITE time. If the facility is not effective, restore the previous settings that were in use before the facility was applied or the change was made.

One of the causes of poor disk performance is disk contention. Check and, if

necessary, revise the configuration of disks, RDAREAs, and tables so that input/output processing is not concentrated on any particular volume. For the design guidelines for disks, RDAREAs, and the table configurations, see the manual *HiRDB Version 8 Installation and Design Guide*.

(5) WRITE unit time maximum (DWMAX DWMAXM)

This is the maximum write time for a page when write operations occur on multiple pages. DWMAX displays the time in seconds, and DWMAXM displays only the microseconds portion of the time.

Purpose

This information is collected in order to check the validity of disk performance.

Evaluating the analysis results

Compare with the disk performance for evaluation. If you use the facility for parallel writes in deferred write processing or if you change the number of parallel WRITE processes, compare disk performance before and after the facility is used or before and after the change.

Actions to be taken

If the performance is poor compared to disk performance, use OS functions to acquire input/output-related tuning information, and then perform tuning on the basis of that information. If disk performance was reduced after the facility for parallel writes in deferred write processing was used or after the number of parallel WRITE processes was changed, check the parallel WRITE time. If the facility is not effective, restore the previous settings that were in use before the facility was applied or the change was made.

One of the causes of poor disk performance is disk contention. Check and, if necessary, revise the configuration of disks, RDAREAs, and tables so that input/output processing is not concentrated on any particular volume. For the design guidelines for disks, RDAREAs, and the table configurations, see the manual *HiRDB Version 8 Installation and Design Guide*.

(6) WRITE unit time average (DWAVG DWAVGM)

This is the average write time when write operations occur on multiple pages. DWAVG displays the time in seconds, and DWAVGM displays only the microseconds portion of the time.

Purpose

This information is collected in order to check the validity of disk performance.

Evaluating the analysis results

Compare with the disk performance for evaluation. If you use the facility for parallel writes in deferred write processing or if you change the number of parallel

WRITE processes, compare disk performance before and after the facility is used or before and after the change.

Actions to be taken

If the performance is poor compared to the disk performance, use OS functions to acquire input/output-related tuning information and then perform tuning on the basis of that information. If the disk performance was reduced after the facility for parallel writes in deferred write processing was used or after the number of parallel WRITE processes was changed, check the parallel WRITE time. If the facility is not effective, restore the previous settings that were in use before the facility was applied or the change was made.

One of the causes of poor disk performance is disk contention. Check and, if necessary, revise the configuration of disks, RDAREAs, and tables so that input/output processing is not concentrated on any particular volume. For the design guidelines for disks, RDAREAs, and the table configurations, see the manual *HiRDB Version 8 Installation and Design Guide*.

(7) WRITE count (DWEXEC)

This is the write count for each operational cause (CAUSE).

Purpose

This information is collected to determine whether or not deferred write processing is delayed for the following reason:

- The number of update buffers that can be processed is exceeded within the synchronization point collection interval.

Evaluating the analysis results

If the operational cause (CAUSE) is S (synchronization point), check to see if the WRITE count (DWEXEC) is equal to or less than the value obtained from the formula below. If the WRITE count is greater than the value, take appropriate action, as described below.

{*synchronization point collection interval* ÷ *average WRITE unit time* (DWAVG, DWAVGN)} × 0.1

Obtain the synchronization point collection interval from the difference between the output times of the following messages:

- KFPS02183-I (message displayed when synchronization point processing is completed)
- KFPS02179-I (message displayed when synchronization point processing is skipped)

Actions to be taken

Tune by reducing the number of update buffers during synchronization point processing. For details about the tuning method, see *21.3.3(1) Reducing the deferred write trigger interval* and *21.3.3(2) Increasing the update page output rate during deferred write trigger*.

If the tuning does not help, specify the `pd_dfw_syncpoint_skip_limit` operand and limit the number of update buffers. In this case, however, if the skip count for the synchronization point processing reaches the maximum value, the update buffer is output at an extension of the update transaction, thereby adversely affecting the update transaction performance.

21.3.3 How to reduce the synchronization point processing time

There are three ways to reduce the synchronization point processing time:

- Reduce the deferred write trigger interval
- Increase the update page output rate during the deferred write trigger
- Increase the number of parallel WRITE processes during the deferred write processing

(1) Reducing the deferred write trigger interval

Reduce the deferred write trigger interval and the number of update buffers during synchronization point processing. If the number of update buffers decreases, the time required for applying the updated pages during synchronization point processing can be reduced, thereby reducing the synchronization point processing time.

The following describes how to tune the deferred write trigger interval.

Procedure

1. Acquire statistical information about the global buffer and the deferred write processing.
2. From the execution results in the global buffer statistical information, identify the global buffer whose synchronization point output pages count (`SYNCW`) is high.
3. Use the following operands to specify the deferred write trigger start conditions:

   - `pd_dbbuff_rate_updpage` operand
   - `-y` option in the `pdbuffer` operand

   Check the global buffer statistical information and change the operand values in such a manner that the following condition is satisfied:

   `TRGUP` (number of update buffers that become output triggers during

deferred write triggers) < SYNCW (number of synchronization point dump pages)/2

If the value of TRGUP is too small, the number of write operations increases and the transaction performance may be adversely affected. If this happens, restore the original operand values.

4. Check to see if the synchronization point processing time is within 10% of the synchronization point acquisition interval. If it is greater than 10%, repeat steps 1 through 3 until it is within 10%.

To determine the synchronization point processing time, check the execution time (DWTOTAL) in the statistical information about deferred write processing.

Obtain the synchronization point collection interval from the difference between the output times of the following messages:

- KFPS02183-I (message displayed when synchronization point processing is completed)
- KFPS02179-I (message displayed when synchronization point processing is skipped)

(2) Increasing the update page output rate during deferred write trigger

Increase the update page output rate during deferred write processing (increase the number of update pages to be applied during deferred write trigger processing), and reduce the number of update buffers during synchronization point processing. If the number of update buffers decreases, the time required for applying the updated pages during synchronization point processing can be reduced, thereby reducing the synchronization point processing time.

The following describes how to tune the update page output rate during deferred write trigger.

Procedure

1. Acquire statistical information about the deferred write processing.
2. Increase the -w option value in the pdbuffer operand.

   If the -w option value is too large, the number of write operations increases and the transaction performance may be affected adversely. If this happens, restore the original operand value.
3. Check to see if the synchronization point processing time is within 10% of the synchronization point acquisition interval. If it is greater than 10%, repeat steps 1 and 2 until it is within 10%.

Obtain the synchronization point collection interval from the difference between the output times of the following messages:

- KFPS02183-I (message displayed when synchronization point processing is completed)
- KFPS02179-I (message displayed when synchronization point processing is skipped)

(3) Increasing the number of parallel WRITE processes during deferred write processing

Increase the number of parallel WRITE processes during deferred write processing to reduce the deferred write processing time. The following describes how to tune the number of parallel WRITE processes during deferred write processing.

Procedure

1. Acquire statistical information about the deferred write processing.
2. Increase the pd_dfw_awt_process operand value to increase the number of parallel WRITE processes during deferred write processing.

   Note that if the pd_dfw_awt_process operand value increases, the number of processes also increases, resulting in an increase in the CPU workload. Therefore, use a facility such as an OS function to monitor the CPU workload.
3. Check to see if the synchronization point processing time is within 10% of the synchronization point acquisition interval. If it is greater than 10%, repeat steps 1 and 2 until it is within 10%.

To determine the synchronization point processing time, check the execution time (DWTOTAL) in the statistical information about deferred write processing.

Obtain the synchronization point collection interval from the difference between the output times of the following messages:

- KFPS02183-I (message displayed when synchronization point processing is completed)
- KFPS02179-I (message displayed when synchronization point processing is skipped)

If increasing the number of parallel WRITE processes during deferred write processing does not reduce the parallel WRITE time (DWPARA, DWPARAM), possible causes are as follows:

- Disk contention has occurred.

  Check and, if necessary, revise the configuration of disks, RDAREAs, and tables so that input/output processing is not concentrated on any particular volume.
- Global buffer lock-release-wait has occurred.

  You can check the global buffer statistical information referred to as lock-release

contention rate (SLEPR) in buffer lock processing to determine the occurrence rate of global buffer lock-release wait status. For details about the tuning method, see *21.1.2(4) Check the lock-release contention rate (SLEPR) in global buffer lock processing* .

- There are not enough output pages.

  The maximum number of parallel WRITE processes that can be executed can be obtained from the following formula:

  - Total number of output pages (`OUT PAGE`)/20

  Even if a value that is greater than the value obtained from this formula is specified, the value from this formula still takes effect.

  To determine the total number of output pages (`OUT PAGE`), check statistical information about the deferred write processing.

21.4 Tuning the synchronization point dump interval

Typically, the synchronization point dump interval is set taking into account the following considerations:

Considerations

- For update processing involving a large amount of data, the transaction processing time is long and there is a large amount of system log information; therefore, recovery at the time of a restart is prolonged.
- For update processing involving a small amount of data, the transaction processing time is short and there is a small amount of system log information; therefore, recovery at the time of a restart is brief.

When the synchronization point dump output interval is tuned, it is also advisable to evaluate the global buffer pool statistical information and the deferred write processing statistical information.

Information to be collected

System activity statistical information is collected by the statistics analysis utility.

Information to be referenced

- Synchronization point dump interval (`SYNC POINT GET INTERVAL`)
- Synchronization point dump collection time (`SYNC POINT GET TIME`)

Purpose

In order to evaluate the validity of the synchronization point dump interval, the time required for a restart (time required to collect the system log) and the throughput must be taken into consideration.

The time required for a restart may be shorter than usual, because HiRDB writes updated pages from global buffer to disk during deferred write processing. The performance of synchronization point dumps can be improved if deferred write processing statistical information is also collected and the number of pages written during deferred write processing (`OUT PAGE`) is evaluated.

Criteria for evaluating the analysis results

1. If many pages are written to disk during a synchronization point dump, the number of pages written to disk during deferred write processing may be too small.
2. If the update buffer hits rate is low and the number of pages output at synchronization points is large, consider reducing the value of the `pdbuffer` operand's `-w` option (the rate of pages output in deferred write processing).

3. If the system log file swapping interval is short for the synchronization point dump interval, the size of the system log file may be too small for the number of blocks output at a synchronization point dump interval. This does not apply when the amount of system log file information available for data output is less than the number of blocks to be output at a synchronization point dump. In such a case, a synchronization point dump is collected automatically when the system log file becomes full.

Actions to be taken

1. Set the time required for restart processing in a range acceptable to the user.
2. A synchronization point dump cannot become effective until a transaction that is executing has been completed; therefore, a long transaction should not be executed concurrently with other transactions. A long transaction means that the amount of log information output by one user server during execution of the transaction is greater than one-third the size of the log file for the user server (user server refers to the single server in the case of a HiRDB/Single Server and to a front-end server, back-end server, or dictionary server in the case of a HiRDB/Parallel Server).

21.5 Tuning buffer lengths

This section explains tuning of the lengths of the following types of buffers:

- Table definition information buffer
- View analysis information buffer
- User privilege information buffer
- SQL object buffer
- User-defined type information buffer
- Routine definition information buffer
- Registry information buffer

21.5.1 Tuning the buffer length for table definition information

This section explains tuning the buffer length for table definition information (`pd_table_def_cache_size` operand value).

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Table definition information buffer hits count (`#OF TBL-CACHE HIT`)
- Number of table definition information acquisition requests (`#OF TBL-DEF GET REQ`)

Tuning procedure

Obtain the table definition information buffer hits rate with the following formula, and make changes to the definitions so that the hits rate becomes greater than 80%:

`Table definition information buffer hits rate (%) =`

`(`*table definition information buffer hits count* `÷` *number of table definition information acquisition requests*`) × 100`

Actions to be taken

Increase the value of the `pd_table_def_cache_size` operand.

21.5.2 Tuning the buffer length for view analysis information

This section explains tuning of the buffer length for view analysis information (`pd_view_def_cache_size` operand value).

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- View analysis information buffer hits count (`#OF VIEW CACHE HIT`)
- Number of view analysis information acquisition requests (`#OF VIEW DEF GET REQ`)

Tuning procedure

Obtain the view analysis information buffer hits rate with the following formula, and make changes to the definitions so that the hits rate becomes greater than 80%:

`View analysis information buffer hits rate (%) =`

`(`*view analysis information buffer hits count* `÷` *number of view analysis information acquisition requests*`) × 100`

Actions to be taken

Increase the value of the `pd_view_def_cache_size` operand.

21.5.3 Tuning the buffer length for user privilege information

This section explains tuning of the buffer length for user privilege information (`pd_auth_cache_size` operand value).

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- User privilege information buffer hits count (`#OF CON/DBA CACHE HIT`)
- Number of user privilege information acquisition requests (`#OF CON/DBA DEF GET REQ`)

Tuning procedure

If the user privilege information buffer hits rate is low for the table definition information buffer hits rate, performance may be affected adversely for the following reason:

- The user privilege information buffer hits rate is low because there is too much `CONNECT` and DBA privilege information.

Obtain the user privilege information buffer hits rate with the following formula:

`User privilege information buffer hits rate (%) =`

`(`*user privilege information buffer hits count* `÷` *number of user privilege*

information acquisition requests) × 100

Actions to be taken

Increase the value of the pd_auth_cache_size operand.

21.5.4 Tuning the buffer length for SQL objects

This section explains tuning of the buffer length for SQL objects (pd_sql_objet_cache_size operand value). When SQL object buffer tuning is performed, the following must be taken into account:

Relationship between an SQL object and an SQL object buffer

HiRDB analyzes each SQL statement in a UAP and creates an execute-form SQL object when the UAP executes. If the same UAP is executed again by another user and if an SQL object corresponding to a specified SQL statement is found in an SQL object buffer, the time required for creating the SQL object is eliminated, thereby reducing SQL statement processing time.

Once an SQL object is deleted from its buffer, it must be analyzed again to re-create it during execution, resulting in a low SQL object buffer hits rate. Therefore, when an SQL statement whose database accesses count is low is executed, a long processing time is required. Furthermore, if the buffers do not contain the table definition information necessary to analyze an SQL statement, the data dictionary tables must be accessed. If many tables or columns are accessed or data is accessed many times, the amount of locked resources increases, thereby increasing the processing time.

SQL object buffers

In the case of a HiRDB/Parallel Server, multiple SQL objects are created for a single SQL statement. The characteristics of SQL objects in the back-end and front-end servers are discussed below.

- *Back-end server*

  SQL objects are created in execute form for each corresponding back-end server (including a floating server). Of all the SQL objects buffered in the front-end server, only those SQL objects for a particular back-end server are buffered in the SQL object buffers of that back-end server; therefore, fewer SQL object buffers are required for a back-end server than for the front-end server.

- *Front-end server*

  The SQL object buffers for the front-end server contain all SQL objects for the back-end servers. A large number of SQL object buffers is required in order to take advantage of the buffering effects, because cost-based optimization creates SQL objects for selectable access procedures.

The same applies to stored procedures and stored functions. It should be noted that the size of an SQL object is larger for a stored procedure or a stored function than for a normal SQL.

Performance can be improved efficiently if this information is analyzed together with the following statistical information:

- **SQL object information**

  This information is collected in order to analyze the buffer length required for each SQL object.

- **SQL information**

  This information is collected in order to determine whether or not the buffer length is too small.

(1) Analysis method (1)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- SQL object buffer hits count (`#OF CACHE HIT (SQLOBJ)`)
- Number of SQL object acquisition requests (`#OF SQLOBJ INFO GET`)
- Number of SQL objects in SQL object buffers (`#OF CACHED SQLOBJ`)
- SQL object buffer hits count for stored procedure objects (`#OF CACHED HIT (STRT)`)
- Number of stored procedure object acquisition requests (`#OF STRT INFO GET`)
- Number of stored procedure objects in SQL object buffers (`#OF CACHED STRT`)

Tuning procedure

1. If the SQL object buffer hits rate is low and the total length of SQL object information in the buffers is short, the application mode may be preventing a high hits rate from being achieved.
2. If the SQL object buffer hits rate is low and the total length of SQL object information in the buffers is long, the SQL object buffers may be too small.

Use the following formula to obtain the SQL object buffer hits rate:

`SQL object buffer hits rate (%) =`

(*SQL object buffer hits count* ÷ *number of SQL object acquisition requests*) × `100`

Actions to be taken

If the SQL object buffers are too small, increase the value of the `pd_sql_objet_cache_size` operand.

(2) Analysis method (2)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Number of SQL objects removed from SQL object buffers (`#OF SWAP OUT SQLOBJ`)
- Number of SQL object acquisition requests (`#OF SQLOBJ INFO GET`)
- Number of stored procedure objects removed from SQL object buffers (`#OF SWAP OUT STRT`)
- Number of stored procedure object acquisition requests (`#OF STRT INFO GET`)

Tuning procedure

Tune in such a manner that the number of SQL objects taken out of the SQL object buffer is small.

If the number of SQL objects removed from SQL object buffers is greater than the number of SQL object acquisition requests, too many SQL objects are having to be re-created, thereby increasing the processing time. This is probably caused by poor utilization of the SQL object buffer pool.

Actions to be taken

Increase the value of the `pd_sql_objet_cache_size` operand.

(3) Analysis method (3)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Length of SQL objects (`REQUEST SQLOBJ SIZE`)
- Length of stored procedure objects (`REQUEST STRT SIZE`)

Tuning procedure

If the maximum total of the SQL object lengths (`REQUEST SQLOBJ SIZE`) and stored procedure object lengths (`REQUEST STRT SIZE`) is greater than the specified buffer length, insufficient memory may have been allocated to execute SQL objects.

Actions to be taken

Increase the value of the pd_sql_objet_cache_size operand.

(4) Analysis method (4)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Stored procedure object recompilations count (# OF STRT RECOMPILE)

Tuning procedure

When the CALL statement is used to execute a stored procedure, recompilation usually does not occur. If an index is added to a table used within a stored procedure or an index not in use is deleted, recompilation occurs. Such recompilation occurs each time the stored procedure is executed with the CALL statement, which has an adverse effect on performance.

If recompilation has occurred, one of these changes may have been made to a table used within the stored procedure.

Actions to be taken

Use ALTER PROCEDURE or ALTER ROUTINE to re-create the stored procedure.

21.5.5 Tuning the buffer length for user-defined type information

This section explains tuning of the buffer length for user-defined type information (pd_type_def_cache_size operand value).

(1) Tuning method (1)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

Type definition cache hits count (# OF TYPE-DEF CACHE HIT)

Number of type definition information acquisition requests (# OF TYPE-DEF GET REQ)

Tuning procedure

Obtain the type definition cache hits rate with the following formula, and tune it so that the hits rate becomes 100%:

Type definition cache hits rate (%) =

(*type definition cache hits count* ÷ *number of type definition information acquisition requests*) × 100

Actions to be taken

Increase the value of the pd_type_def_cache_size operand.

(2) Tuning method (2)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Total type definitions cache size (TYPE-DEF CACHE TOTAL SIZE)
- Allocated type definition cache size (TYPE-DEF CACHE ALLOC SIZE)

Tuning procedure

Obtain the type definition cache utilization factor with the following formula, and tune it so that the utilization factor becomes 90% or higher:

Type definition cache utilization factor (%) =

(*maximum value of total type definitions cache size* ÷ *allocated type definition cache size*) × 100

Actions to be taken

Reduce the value of the pd_type_def_cache_size operand.

21.5.6 Tuning the buffer length for routine definition information

This section explains tuning of the buffer length for routine definition information (pd_routine_def_cache_size operand value).

(1) Tuning method (1)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Routine definition cache hits count (# OF RTN-DEF CACHE HIT)
- Number of routine definition information acquisition requests (# OF RTN-DEF GET REQ)

Tuning procedure

Obtain the routine definition cache hits rate with the following formula, and tune it so that the hits rate becomes 100%:

Routine definition cache hits rate (%) =

(*routine definition cache hits count* ÷ *number of routine definition information acquisition requests*) × 100

Actions to be taken

Increase the value of the `pd_routine_def_cache_size` operand.

(2) Tuning method (2)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Total routine definitions cache size (`RTN-DEF CACHE TOTAL SIZE`)
- Allocated routine definition cache size (`RTN-DEF CACHE ALLOC SIZE`)

Tuning procedure

Obtain the routine definition cache utilization factor with the following formula, and tune it so that the utilization factor becomes 90% or higher:

`Routine definition cache utilization factor (%) =`

(*maximum value of total routine definitions cache size* ÷ *allocated routine definition cache size*) × `100`

Actions to be taken

Reduce the value of the `pd_routine_def_cache_size` operand.

(3) Tuning method (3)

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Plug-in routine definition cache hits count (`# OF PLG-RTN CACHE HIT`)
- Number of plug-in routine definition acquisition requests (`# OF PLG-RTN GET REQ`)

Tuning procedure

Obtain the plug-in routine definition cache hits rate with the following formula, and tune it so that the hits rate becomes 100%:

`Plug-in routine definition cache hits rate (%) =`

(*plug-in routine definition cache hits count* ÷ *number of plug-in routine definition acquisition requests*) × `100`

Actions to be taken

Increase the value of the `pd_routine_def_cache_size` operand.

21.5.7 Tuning the buffer length for registry information

This section explains tuning of the buffer length for registry information (pd_registry_cache_size operand value).

Information to be referenced

See the following system activity statistical information provided by the statistics analysis utility:

- Registry cache hits count (# OF REGISTRY CACHE HIT)
- Number of registry information acquisition requests (# OF REGISTRY GET REQ)

Tuning procedure

Determine the registry cache hits rate using the following formula, and adjust the buffer length to achieve a hits rate of 100%.

Registry cache hits rate (%) =

(*registry cache hits count* ÷ *registry information collection requests count*) × 100

Actions to be taken

Increase the value of the pd_registry_cache_size operand.

21.6 Tuning the number of processes

This section explains the procedure for tuning the number of server activation processes, which is part of the system activity statistical information provided by the statistics analysis utility. The purpose is to enable the system to activate server processes in an optimal manner by analyzing the number of active processes that were actually executing services and determining whether or not there were any service requests beyond the maximum number of active processes.

The phrase *during service execution* in this section refers to the following status in a server:

- In the front-end server or single-end server, it is the status in which a HiRDB client is allocated to a process by issuing the `CONNECT` statement. This status remains until the `DISCONNECT` statement is issued.
- In a back-end server or dictionary server, it is during execution of a transaction. This status remains until the transaction is terminated.

Information to be collected

System activity statistical information is collected by the statistics analysis utility.

21.6.1 Tuning the maximum number of active processes

This section explains the procedure for tuning the specifications in the following operands that determine the maximum number of active processes:

- `pd_max_users`
- `pd_max_bes_process`
- `pd_max_dic_process`

Information to be referenced

Reference the following statistical information related to system operations:

- Number of service requests exceeding the maximum number of active processes (`#OF REQ PROCESS OVER MAX`)
- Number of service processes under service execution (`#OF PROCESS ON SERVICE`)

Purpose

- To make efficient use of memory resources by adjusting the number of active processes actually executing a service so that it is close to the maximum number of active processes specified in these operands.
- To prevent service from being denied by increasing the maximum number of

active processes (when it turns out that the estimated value is too small).

Evaluating the analysis results

When the operand value is greater than the number of server processes engaged in service execution

It is likely that the operand value can be reduced.

Reduce each server's maximum number of active processes to a value close to the maximum value for the number of server processes during service execution (`#OF PROCESS ON SERVICE`). The following should be noted in such a case:

- Depending on other servers' status, the value should not be reduced to the maximum value for the number of server processes during service execution (`#OF PROCESS ON SERVICE`). For example, this is true when service requests from multiple front-end servers may concentrate in a specific back-end server or when a particular front-end server in the unit is handling too much processing.
- In tuning `pd_max_bes_process` or `pd_max_dic_process`, if the specified value is smaller than the value of `pd_max_users`, the value of `pd_max_users` is assumed.

When there are service requests exceeding the maximum number of active processes

The operand value should be increased.

In this case, the operand value should be increased such that no service request is denied. The following should be noted for each server with respect to the tuning procedure:

- For the front-end server or single server, if a service request received by the server exceeds the maximum number of active processes, the client retries up to the following number of times:

  `HiRDB client with HiRDB Version 4.0 03-03 or earlier: 100 times`

  `HiRDB client with HiRDB Version 4.0 04-00 or later: 10 times`

  The number of service requests exceeding the maximum number of active processes (`#OF REQ PROCESS OVER MAX`) includes the retries count; this must be taken into account when this parameter is tuned.
- For a back-end server or dictionary server, a service request is not retried; therefore, the number of service requests exceeding the maximum number of active processes (`#OF REQ PROCESS OVER MAX`)

can be used for tuning.

If the value of pd_max_users is increased and the resulting value exceeds the value of pd_max_bes_process or pd_max_dic_process, the value of pd_max_bes_process or pd_max_dic_process is also assumed to increase up to the value of pd_max_users.

Actions to be taken

- For the front-end server or single server, modify the value of pd_max_users.
- For a back-end server, modify the value of pd_max_bes_process.
- For a dictionary server, modify the value of pd_max_dic_process.

21.6.2 Tuning the number of resident processes

This section explains the procedure for tuning the number of resident processes specified in the pd_process_count operand.

Information to be referenced

Reference the following statistical information related to system operations:

- Number of service processes engaged in service execution (#OF PROCESS ON SERVICE)

Purpose

Each server's process activation time is reduced along with the memory resources by specifying as the number of resident processes a value close to the average number of active processes that were actually executing the service.

Evaluating the analysis results

The process activation time can be reduced by making each server's processes resident beforehand; however, if more processes are made resident than necessary, memory resources will not be used efficiently. A reasonable number of resident processes for a server appears to be the average value of the number of service processes under service execution (#OF PROCESS ON SERVICE).

When improving the server's service execution response

If the value of the pd_process_count operand is less than the average value of the number of service processes during service execution (#OF PROCESS ON SERVICE), increase it up to about the average value. If it is increased further, some resident processes may not receive services, resulting in a waste of memory.

When reducing the memory space used by resident server processes

If the value of the pd_process_count operand is greater than the average

value of the number of service processes during service execution (#OF PROCESS ON SERVICE), reduce it down to about the average value. If it is decreased further, more requests may be received than the number of resident processes and service execution response may be delayed as much as the process activation time.

Actions to be taken

Modify the value of the pd_process_count operand.

21.6.3 Tuning the number of processes in asynchronous READ processing

This section explains how to tune the number of processes in asynchronous READ processing specified in the pd_max_ard_process operand. When the asynchronous READ facility is used, if the number of processes in asynchronous READ processing is low and the number of SQL statements to which the asynchronous READ facility is applicable is high, processing time can increase due to completion waiting during I/O processing of asynchronous READ processes.

Information to be referenced

Reference the following information in the UAP statistical report (for details about UAP statistical reports, see the manual *HiRDB Version 8 UAP Development Guide*):

- Accumulated synchronization wait time during asynchronous READ (ARWT, ARWTM)
- Accumulated database I/O time (IOTIM, IOTIMM)
- Average synchronization wait time during asynchronous READ (ARWTA, ARWTMA)
- Average synchronization I/O time during asynchronous READ (ARSTA, ARSTMA)

Purpose

The purpose is to determine the number of appropriate processes for asynchronous READ, when the asynchronous READ facility is not effective.

Evaluating the analysis results

Compare with the following times; if step 2 is longer, consider that the asynchronous wait time may be too long:

1. Accumulated database I/O time (IOTIM, IOTIMM) when the asynchronous READ facility is not used
2. Total of the average synchronization wait time during asynchronous READ (ARWTA, ARWTMA) and accumulated database I/O time (IOTIM, IOTIMM)

when the asynchronous READ facility is used

Compare the following times; if step 2 is longer, consider that the asynchronous wait time may be too long:

1. Average synchronization I/O time during asynchronous READ (ARSTA, ARSTMA) when the asynchronous READ facility is used
2. Average synchronization wait time during asynchronous READ (ARWTA, ARWTMA) when the asynchronous READ facility is used

The number of asynchronous READs (AIO-R) can be understood from the statistical information for HiRDB files related to database operations obtained by the statistics analysis utility.

Actions to be taken

If the accumulated synchronization wait time during asynchronous READ (ARWT, ARWTM) or the average synchronization wait time (ARWTA, ARWTMA) during asynchronous READ is long, increase the value of the pd_max_ard_process operand.

However, if the result of checking the number of asynchronous READs (AIO-R) is that the asynchronous READ requests are concentrated in the same RDAREA, increasing the value of this operand will have no effect. In such a case, have the RDAREA be composed of multiple HiRDB files, and place each HiRDB file on a different disk.

For the maximum effective value of the pd_max_ard_process operand, of the RDAREAs for which the number of asynchronous READ (AIO-R) is specified, specify the number that simultaneously executes the prefetch facility. If the number of processes increases, it is necessary to consider the system resources (shared memory and message queues).

If the accumulated synchronization wait time during asynchronous READ (ARWT, ARWTM) or the average synchronization wait time during asynchronous READ (ARWTA, ARWTMA) is short, the asynchronous READ facility cannot provide greater effectiveness.

21.7 Tuning indexes

When a table is updated, its indexes are also updated to maintain database conformity. When an index has been defined, the following items must be evaluated:

Items to be evaluated

- The number of times index update processing is executed increases proportionally to the number of indexes defined for the table, resulting in a commensurate increase in the SQL processing time.
- When an index is updated, update information is collected in the system log file. Therefore, if the number of indexes increases, the amount of data to be output to the system log file also increases commensurately.
- If a shortage occurs in an index page for storing updated key values, an index page split occurs. When index page split occurs, system log information on the split is collected, thereby increasing the amount of data to be stored in the system log file.

Information to be collected

Index statistical information is collected by the statistics analysis utility.

(1) Splits count (SP_NM)

Purpose

This information is collected in order to evaluate the validity of indexes. Indexes can affect performance adversely in the following case:

- If an appropriate free space ratio (`PCTFREE`) is not specified in the index definition, index page splitting may occur.

In such a case, the index definition should be modified on the basis of the index splits count.

Evaluating the analysis results

1. If index page splitting occurs frequently, the index may have been expanded by an application that involves mainly data insertion (`INSERT`) into the table, resulting in frequent index page splits.
2. If a table is partitioned and index page splitting occurs frequently in a specific RDAREA, update processing may be concentrated on a specific key range due to poor table partitioning.

Actions to be taken

Take one of the following actions:

1. Increase the free space ratio (`PCTFREE`) specified in the index definition. If a large amount of data is to be added, use the database load utility.
2. Delete any unneeded indexes.
3. If a table is partitioned, check the key range in which the index page splits are concentrated and determine whether or not to partition further.

21.8 Tuning the database

The condition analysis information for each RDAREA is important for tuning the size and orderliness of a database. The condition of a database is analyzed on the basis of the status of the RDAREAs defined in the HiRDB system and the storage status of the tables and indexes.

Information to be collected

Needed information is collected by the database analysis utility.

(1) Physical analysis information for each RDAREA

Purpose

Physical analysis information is collected for each RDAREA in order to evaluate the validity of the utilization efficiency and size of the RDAREA with respect to the defined RDAREA size and segment size.

The validity of the utilization efficiency and size of an RDAREA is evaluated by analyzing the storage status of all segments and pages in the RDAREA without taking into account the tables or indexes.

Evaluating the analysis results

When the ratio of used segments is low

Consider that there are many unused segments, and study the following:

- If one RDAREA consists of multiple HiRDB files, check the sizes of the files to determine whether or not they are being used.
- Check the amount of data to be added in the future, and the table and index definitions.

When the ratio of used segments is high and the ratio of used pages is low

Consider the following factors, and check that the ratio of free pages in the segments is not too large:

- Consider whether the RDAREA is being used efficiently.
- A common reason for a large ratio of used segments occurs when the segment size specified during RDAREA definition is larger than the volume of the RDAREA, which reduces the number of segments such that the defined tables and indexes use all of the segments.

When the ratio of used segments is high and the ratio of used pages is high

Possible causes are:

- There is not enough free space in the RDAREA.

- If the ratio of full pages is high, the size may be insufficient.

Actions to be taken

Check the validity of the segment size specified when the RDAREA was created. Determine the validity by checking the numbers of tables or index and data items stored in the RDAREA. If the ratio of full pages is too high or too low, take the following actions:

- If the ratio of full pages is too high, expand the RDAREA with the database structure modification utility. Or, move some of the tables or indexes from this RDAREA to another RDAREA.
- If the ratio of full pages is too low, the storage efficiency inside the RDAREA may have declined, resulting in fragmented data placement. In this case, use the database reorganization utility to reorganize the table by RDAREA.
- In the case of the data dictionary RDAREA, reorganize it with the database reorganization utility with `dir` specified in the `-C` option. When the data dictionary RDAREA is reorganized, reorganization of specified data dictionary tables or reorganization of all data dictionary tables can be selected. When specific data dictionary tables are not specified, the entire data dictionary RDAREA is reorganized.

(2) Logical analysis information for an RDAREA

Purpose

Logical analysis information is collected for each RDAREA in order to evaluate the validity of the following:

- Whether the size of each RDAREA matches the estimated size
- Whether the database should be reorganized

The storage status of all segments and all pages of the tables or indexes in an RDAREA should be analyzed.

Evaluating the analysis results

Table condition analysis information

1. When the ratio of used segments is high

   If there are no unused segments in the relevant RDAREAs, it can be considered that no more data can be inserted. In this case, consider the availability of free space (`PCTFREE`) for subsequent data additions and updates.

2. When the ratio of used segments is high, and the ratio of used pages is low

   Consider that the ratio of segment free pages specified in the `PCTFREE`

operand is not appropriate. In this case, consider the availability of free space (PCTFREE) for subsequent data additions and updates.

3. When there are more used pages than the estimated number of pages

   If there are VARCHAR, NVARCHAR, or MVARCHAR columns, consider whether the data length of any of these columns has exceeded 255 bytes, and determine whether or not this was considered in the calculation.

4. When the ratio of full segments and the ratio of full pages are both high

   The RDAREA size is inadequate, or disorder has occurred in the data arrangement because the following operations were performed many times:

   - Updating of null-value data to numeric or character data
   - Updating of a VARCHAR, NVARCHAR, or MVARCHAR column resulting in a longer column
   - Updating of column value to the null value, or updating of a VARCHAR, NVARCHAR, or MVARCHAR column resulting in a shorter column

Index condition analysis information

If the number of used segments and used pages is greater than what was calculated in advance, and if the addition of a large number of key values has caused index splitting to occur, consider that the page usage rate may have increased.

Actions to be taken

- Either expand the RDAREA or store some of the tables in different RDAREAs (if multiple tables are stored in the RDAREA).
- Reorganize the table with the database reorganization utility.
- Reevaluate the ratio of free pages in the segment that was specified in the PCTFREE operand during table definition.
- Reevaluate the ratio of unused area in a page that was specified in the PCTFREE operand during table definition.

(3) Logical analysis information for a table or index

Purpose

If a table was row-partitioned, check that the table and any index were partitioned properly. This can be done by comparing the amount of output data (numbers of segments and pages) and the estimated amount of data. Also, whether or not the database should be reorganized can be determined by checking the table or index storage condition.

Evaluating the analysis results

Table condition analysis information

1. When the ratio of full segments or full pages is 80% or higher

   If the ratio of used pages is much higher than expected from the estimated number of segments, consider that the table storage status may have become fragmented.

2. When a table is row-partitioned

   If the number of used segments in a specific RDAREA is large, or if the number of rows stored in a specific RDAREA is large, consider that the tables may not be optimally partitioned. Reconsider the table allocation from the estimated data volume. If hash partitioning is already being used, change the hash function. At this time, check the number of rows stored in each RDAREA, and ensure that the rows are stored uniformly.

3. When the ratio of total unused pages for a table { (*total-number-of-pages* - *total-number-of-used-pages*) ÷ *total-number-of-pages*} is less than the ratio of free pages of segments specified in the table definition

   Probably, there are too few unused pages due to repeated addition of data.

4. When there are not enough free pages to add data

   Consider that the ratio of free pages for segments is too large, or that data deletion has caused an increase in the number of used free pages.

Index condition analysis information

1. When the number of pages that store indexes calculated from the number of rows stored in the tables is less than the total number of used pages in the analysis result

   Consider that a large number of rows in a specific range were deleted, which caused the related index key values to also be deleted, resulting in used free pages being created in the index pages.

2. When the used pages are almost all full pages

   If this occurs even after the database reorganization utility has been used to reorganize the indexes, consider that the ratio of segment free pages is not optimal or that data additions have used up the free area in the pages.

Actions to be taken

Table condition analysis information

1. When the ratio of full segments or full pages is high

   If new data is to be added in the future, reorganize the corresponding table.

2. When there are many free pages (unused pages)

   Re-examine the ratio of free pages in the segment.

3. When data deletions have increased the number of used free pages

   Reorganize the tables, reorganize the indexes, or release the used free pages.

4. When there are few unused segments in an RDAREA

   Expand the RDAREA.

Index condition analysis information

1. When there are many used free pages

   Reorganize the tables, reorganize the indexes, or release the used free pages.

2. When more data is expected to be added in the future

   If the column data may occur at random for the defined index, redefine the index and specify an appropriate value in the `PCTFREE` operand.

3. When the number of used pages is much greater in a specific RDAREA than in other RDAREAs

   Reevaluate the table partitioning method. Partition the table so that the amount of partitioned data is the same in each RDAREA.

4. When there are many used free pages in specific RDAREAs only

   Reorganize the tables and indexes in those RDAREAs, or release the used free pages.

(4) Storage condition analysis for cluster key and clustering data pages

Purpose

If a cluster key is defined, check the storage disorder rate, which indicates the disorder of tables and indexes, and the number of keys stored with a duplicate structure.

Cluster key storage condition analysis

The cluster key is searched in the order of key values to display the storage location changes count where the storage location spans multiple pages or segments and the storage disorders count (rate) where of the storage location changes count, the storage sequence is opposite to the ascending order of

pages (segments) in units of pages and segments.

Page splitting increases the storage disorder rate.

Clustering data page storage condition analysis

A data page for a table with a cluster key defined is called a clustering data page.

The row data storage location changes count and the storage disorders count (rate) are displayed in units of pages and segments based on the storage location information for the row data in the cluster key when the row data is searched in the order of cluster key values.

If the storage condition becomes poor due to row addition and update processing, the storage disorder rate or the storage location changes count or both increase.

Evaluating the analysis results

Cluster key storage condition analysis information

1. When there are too many rows for the number of storage keys in an index

   The performance of retrieval processing using the index is affected adversely because the key duplication rate is high.

2. When some keys are stored with duplicate key structure

   The performance of retrieval processing using the index is affected adversely because there are (or there were) keys with a high duplication level.

3. When the storage disorder rate is high

   Retrieval performance is affected adversely because there is some disorder in the data page storage sequence.

Clustering data page storage condition analysis information

1. When the storage disorder rate is high

   The performance of accesses in the order of cluster key values is affected adversely because there is some disorder in the data page storage sequence.

2. When the storage location changes count is greater than the number of segments or pages used in the condition analysis result for each table minus 1

   The performance of accesses in the order of cluster key values is affected adversely because there is some disorder in the data page

storage sequence.

Actions to be taken

Cluster key storage condition analysis information

1. When there are too many rows for the number of storage keys in an index

   Reevaluate the column structure in the index definition.

2. When some keys are stored with duplicate key structure

   If there is any key with a high duplication level, redefine the table with the cluster key without using those index structure columns that have a high data duplication level. If in the past there was a key with a high duplication level, the duplicate key structure will be eliminated when the table is reorganized or the index is re-created by the database reorganization utility.

3. When the storage disorder rate is high

   Reorganize the table or re-create the index with the database reorganization utility.

Clustering data page storage condition analysis information

1. When the storage disorder rate is high

   Reorganize the table with the database reorganization utility.

2. The number of storage unit changes is greater than one less than the number of used segments or pages, based on condition analysis of table units.

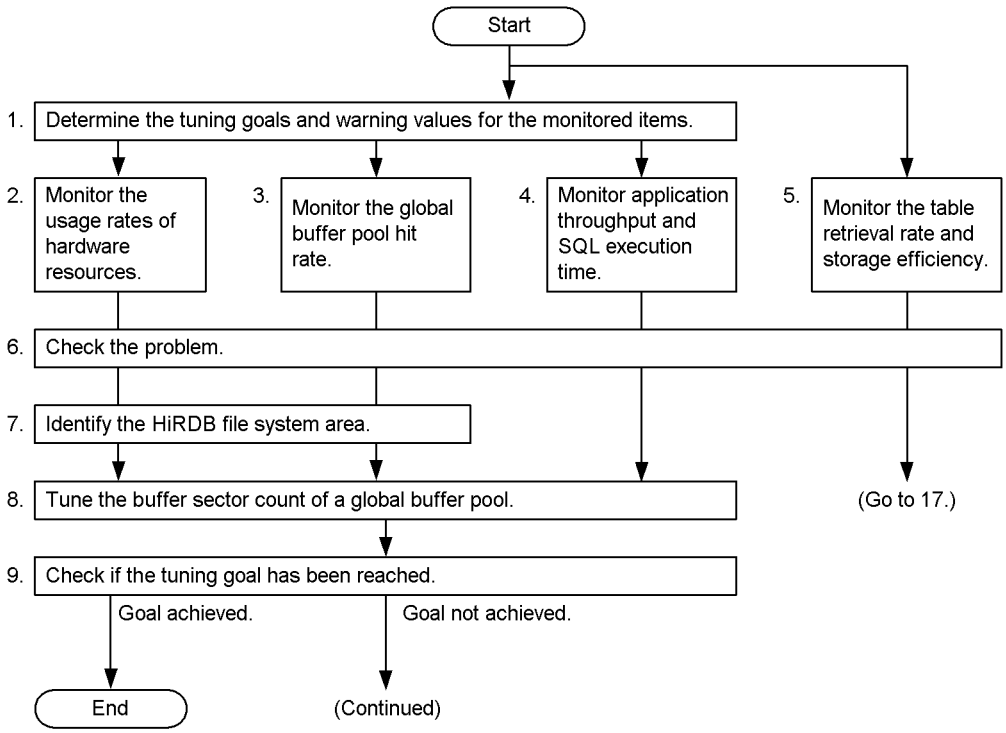
   Reorganize the table with the database reorganization utility.

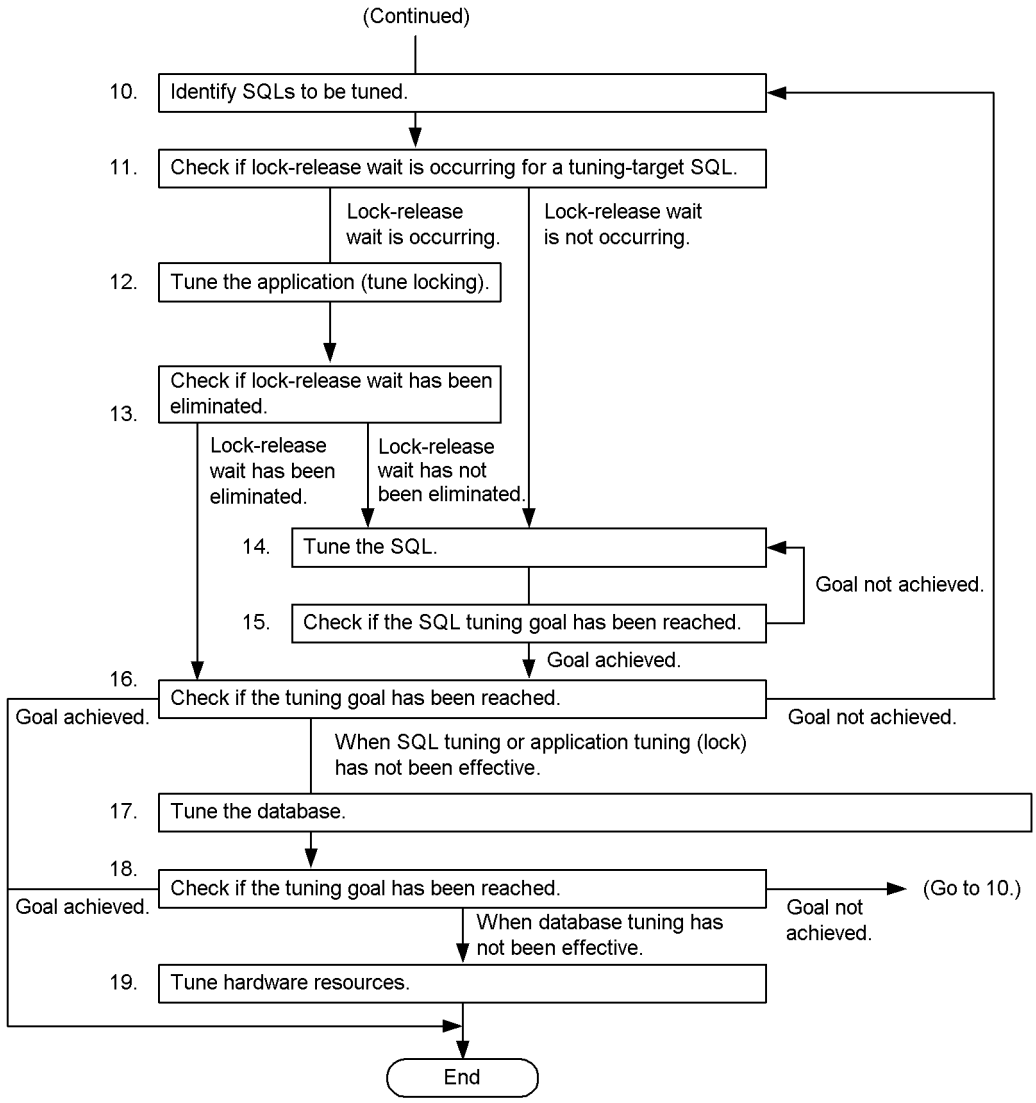
21.9 Tuning SQLs

This section explains the method of tuning SQLs that require a long time to execute or that have a large number of input/output processes.

Figure 21-2 shows the flow of SQL tuning.

Figure 21-2: SQL tuning flow





Note

The numbers to the left of the process boxes correspond to the paragraph numbers of the explanations on the following pages. For example, step 5 is explained in paragraph (5) below.

(1) Determining the tuning goals and warning values for the monitored items

Determine the tuning goals and warning values for the monitored items. Determine the

overall system performance (for example, application processing time and transaction commit count), and then set tuning goals accordingly. Also set warning values for the following monitored items:

- Hardware resource usage rate

  The items to be monitored are the usage rates of hardware resources, such as a server machine's processor and disk volumes.

- Global buffer pool status

  The item to be monitored is the global buffer pool hit rate.

- Application throughput and SQL execution time

  The items to be monitored are the transaction commit count and SQL execution time.

(2) Monitoring the usage rates of hardware resources

When the usage rates of hardware resources, such as a server machine's processor and disks, become high, tuning may be necessary. High hardware resource usage rates may cause system throughput to decline. This is the reason why you should monitor the usage rates of hardware resources.

Using the OS's `sar` command (`vmstat` command for Linux), for example, you can identify hardware resources that have high usage rates, such as the server machine's processor and disks.

Reference note:

- Use of JP1/Performance Management - Agent Option for Platform provides you with a facility for checking hardware resource usage rates. For details about JP1/Performance Management - Agent Option for Platform, see the manual *For UNIX Systems: Job Management Partner 1/ Performance Management - Agent for Platform*.
- If you are using a Hitachi disk array device (Lightning/Thunder series), you can use HiCommand Tuning Manager - Agent for RAID to collect performance data for the disk array device. For details about HiCommand Tuning Manager - Agent for RAID, see the manual *HiCommand Tuning Manager - Agent for RAID*.

(3) Monitoring the global buffer pool hit rate

When the global buffer pool hit rate declines, tuning may be necessary. You can execute the `pdbufls` command to check the global buffer pool hit rate.

Reference note:

You can also use JP1/Performance Management - Agent Option for HiRDB to check the global buffer pool hit rate.

(4) Monitoring application throughput and SQL execution time

When application throughput decline, tuning may be necessary. You can collect statistics on system operations by executing the statistics analysis utility (`pdstedit` command) and then use this information to check application throughput.

You can also use the SQL runtime warning output facility to monitor SQL execution time. For details about the SQL runtime warning output facility, see *8.9 Output of warning information about the time required for SQL execution (SQL runtime warning output facility)*.

Reference note:

You can also use JP1/Performance Management - Agent Option for HiRDB to check application throughput.

(5) Monitoring the table retrieval rate and storage efficiency

When the arrangement of table data becomes fragmented, tuning may be necessary. The `KFPH00212-I` or `KFPH22017-I` message is output when the retrieval rate or storage efficiency declines for a table, so you should watch for these messages.

Reference note:

- HiRDB provides a facility for predicting reorganization time, which predicts the date when a table will need to be reorganized. For details about the facility for predicting reorganization time, see *13.4 Predicting table reorganization time (facility for predicting reorganization time)*.
- You can also use JP1/Performance Management - Agent Option for HiRDB to collect messages related to tables whose data retrieval rates or storage efficiency have declined.

(6) Checking the problem

You should perform tuning when any of the items monitored in steps (2) through (4) exceeds its warning value. If you use OS or HiRDB commands to collect data on the items being monitored, compare the currently collected data with previously collected data to determine whether warning values have been exceeded.

If a hardware resource whose warning value has been exceeded is a processor, use an OS command to determine whether HiRDB processes are using many processors.

If a warning is issued by the SQL runtime warning output facility, go to step (8) and

perform tuning.

If the data arrangement becomes fragmented in a table being monitored in step (5) and a message is issued, go to step (17) and perform tuning.

Reference note:

If you use JP1/Performance Management - Agent for Platform, JP1/Performance Management - Agent Option for HiRDB, or HiCommand Tuning Manager - Agent Option for RAID, you can monitor items by specifying warning values for the data in the monitored items. When any of these warning values is exceeded, a warning is issued.

(7) Identifying the HiRDB file system area

When a hardware resource or global buffer pool hit rate exceeds its warning value in step (6), you must identify the HiRDB file system area to which the hardware resource is related. To collect information about the relationship between the hardware resource or global buffer pool that exceeded its warning value and a HiRDB file system area, you need to summarize the information between individual layers.

You can check `SQL_PHYSICAL_FILES` in the dictionary table to identify the relationship between a disk volume and a HiRDB file system area, the relationship between a HiRDB file system area and HiRDB files, and the relationships between HiRDB files and RDAREAs. If you are using the inner replica facility, check the `SQL_IOS_GENERATIONS` table.

For a HiRDB file system area for work table files, you can determine the name of the HiRDB file system area by checking the value specified in the `pdwork` operand. You can also check the `SQL_TABLES` table or `SQL_INDEXES` table in the dictionary table to identify the relationships between RDAREAs and tables (or indexes).

Reference note:

- If you use HiRDB CM, you can collect information about the relationships between disk volumes and tables (or indexes).
- If you are using a Hitachi disk array device (Lightning/Thunder Series), you can use HiCommand Tuning Manager - Agent for RAID Map to collect information about the relationships between ports and logical devices in the disk array device. For details about HiCommand Tuning Manager - Agent for RAID Map, see the manual *HiCommand Tuning Manager - Agent for RAID Map*.

(8) Tuning the buffer sector count of a global buffer pool

If the buffer sector count of a global buffer pool is small, system throughput may have

declined or hardware resource usage rates may have increased.

Tune the buffer sector count of the global buffer pool that is related to the HiRDB file system area identified in step (7). If a reduced global buffer pool hit rate necessitated tuning, tune the buffer sector count of the applicable global buffer pool. If application throughput or SQL execution time necessitated tuning, tune the buffer sector counts of all global buffer pools.

For details about the buffer sector count of a global buffer pool, see the manual *HiRDB Version 8 Installation and Design Guide*. For details about how to tune global buffer pools, see *21.1 Tuning global buffer pools*.

(9) Checking if the tuning goal has been reached

If the tuning goal has been reached by the tuning performed in step (8), terminate tuning. If the tuning goal has not been reached, proceed to the next step and continue tuning.

(10) Identifying SQLs to be tuned

Identify SQLs to be tuned. Also establish tuning goals for the SQLs.

Use the `pdobils` command to collect statistics on an SQL object buffer, and make the SQLs listed below the tuning targets. When you collect statistics on an SQL object buffer, you can check each SQL's execution time, number of executions, and number of inputs/outputs.

Set each SQL's target execution time, target number of executions, and target number of inputs/outputs as the tuning goals.

- SQL with the longest execution time
- SQL with the largest number of executions
- SQL with the largest number of inputs/outputs
  - SQL with the largest number of real READ counts for data pages, index pages, and directory pages; SQL with the largest number of real READ counts for LOB column data pages; or SQL with the largest number of real READ counts for list pages
  - SQL with a real WRITE count for data pages, index pages, and directory pages; SQL with a real WRITE count for LOB column data pages; or SQL with a real WRITE count for list pages
  - SQL with the largest number of updates for data pages, index pages, and directory pages; SQL with the largest number of updates for LOB column data pages; or SQL with the largest number of updates for list pages
  - SQL with the largest number of READ or WRITE counts for work table files

(11) Checking if lock-release wait is occurring for a tuning-target SQL

Execute the statistics analysis utility (`pdstedit` command) to collect statistics on system operations and check whether lock-release wait has occurred. If it has, collect the server's lock status regularly, and check whether lock-release wait is occurring. If an applicable SQL is using a resource for which there is a lock-release wait, it is safe to assume that lock-release wait is occurring for that tuning-target SQL.

If there is a lock-release wait, release the lock by performing application tuning (lock tuning) in step (12). If no lock-release wait has occurred, tune the SQL in step (14).

(12) Tuning the application (tuning locking)

If lock-release wait is occurring for a tuning-target SQL, tune the application (tune the locking). For details about application tuning (tuning locking), see *Lock* in the manual *HiRDB Version 8 UAP Development Guide*.

(13) Checking if lock-release wait has been eliminated

Check if application tuning has eliminated the lock-release wait that has occurred for the tuning-target SQL. For the checking procedure, see step (11).

If lock-release wait has been eliminated, go to step (16) and check if the tuning goal has been achieved. If lock-release wait has not been eliminated, go to step (14) and tune the SQL.

(14) Tuning the SQL

Table 21-1 shows the methods for tuning an SQL.

Table 21-1: SQL tuning methods

SQL characteristics	Assumed cause	Details	Corrective action
The real READ count for data pages, index pages, and directory pages or the total number of referencing operations is large (the number of inputs/outputs is greater than originally designed, or execution time is longer than originally designed)	Unsuitable access path to the table	Table scan is being executed when the table search range is wide.	See the concept of tuning with the access path display utility in the manual *HiRDB Version 8 Command Reference*.
		Table scan is being executed in a search for which a narrowing condition is specified (an unsuitable index is being used).	
		Index search range is wide (search is over an unnecessarily wide range).	
		Index search range has not been narrowed because there is no search condition, or the search condition is invalid.	
		A search with multiple predicates specified using AND is being performed.	
	Unsuitable joining method	Index of a join key of an inner table is unsuitable for nest-loop-join.	
		Index being used for outer join is unsuitable.	
	`BROADCAST` or `KEY RANGE PARTIAL BROAD CAST` transmission is specified as the transmission method for nest-loop-join (for a HiRDB/Parallel Server).		
READ or WRITE count exists for work table files (execution time is longer than originally designed)	Unsuitable joining method	Packet partitioning has been executed multiple times because of a large number of hash-joined inner tables or subquery searches.	See the preparations for executing hash join or subquery hash in the *HiRDB Version 8 UAP Development Guide*.
		A large volume of data is sorted in sort-merge join.	See the concept of tuning with the access path display utility in the manual *HiRDB Version 8 Command Reference*.
		Cross-join is executed.	
	Sorting was executed.	Sorting is executed for `ORDER BY` processing.	

(15) Checking if the SQL tuning goal has been reached

Check if the tuning goal has been reached for the tuning-target SQL. Collect statistics on the SQL object buffer, and check whether the SQL execution time, execution count, and input/output count are within the target ranges. If the goals have not been reached, return to step (14) and tune the SQL again.

(16) Checking if the tuning goal has been reached

Check if the tuning goal has been reached. If it has, terminate tuning. If the tuning goal has not been reached, return to step (10), identify the tuning-target SQL again, and tune it.

If SQL tuning or application tuning has not produced the desired effects, go to step (17) and tune the database.

(17) Tuning the database

If SQL tuning or application tuning does not produce the desired effects, tune the database. Tune the following items:

- Buffer sector count of a global buffer pool

  For details about the sector count of a global buffer pool, see the manual *HiRDB Version 8 Installation and Design Guide*. For details about tuning global buffer pools, see *21.1 Tuning global buffer pools*.

- Work table buffer size

  For details about work table buffer size, see the preparations for executing hash join or subquery hash in the manual *HiRDB Version 8 UAP Development Guide*.

- Table row partitioning

  For details about table row partitioning, see the *HiRDB Version 8 Installation and Design Guide*.

- Data rearrangement

  For details about data rearrangement, see *21.8 Tuning the database*.

- Deferred write processing

  For details about deferred write processing, see *21.2 Tuning deferred write processing*.

(18) Checking if the tuning goal has been reached

Check if tuning the database has achieved the tuning goal. If so, terminate tuning. If the tuning goal has not been reached, go to step (10), identify the tuning-target SQL again, and tune it.

If database tuning has not produced the desired effects, go to the next step and tune hardware resources.

(19) Tuning hardware resources

Add or augment hardware resources. Using the OS's `sar` command (`vmstat` command for Linux), for example, determine processor usage and disk usage. Add or augment the hardware resource that has the highest usage.

If the hardware resource with the highest usage is a processor, switching to a higher-performance processor may reduce processing time. If the hardware resource with the highest usage is a disk, switching to a higher-performance disk or adding disks (increasing the spindle count) in the case of a disk array device may reduce input/output time.

Reference note:

- You can also use JP1/Performance Management - Agent Option for Platform to check processor and disk volume usage.
- If you are using a Hitachi disk array device (Lightning/Thunder Series), you can use HiCommand Tuning Manager - Agent for RAID to collect performance data for the disk array device.

21.10 Tuning the system's internal processing

In some cases, you can specify an operand to facilitate the system's internal processing.

These operands are described in this section. For details about each operand, see the section on each operand in the manual *HiRDB Version 8 System Definition*.

(1) Operands for increasing performance, even if the machine load increases

The operands described here are used to select among methods that use one of the following two general approaches:

1. An approach that places a high load on the machine, but provides high performance (for a machine capable of high performance). If the machine's performance capacity is low, CPU usage may approach 100%, posing the risk of degraded OS performance.
2. An approach that places a low workload on the machine, but can ensure standard performance. Stable performance can be achieved with any type of machine.

To minimize risk, the default values for many of these operands are set to use a method that adheres to approach 2.

If your machine's performance capacity is high, you may be able to improve performance (while also increasing the load on the machine) by changing the values specified for these operands. However, if unsuitable values are specified, the CPU usage may approach 100%, posing the risk of degraded OS performance. Thus, you should be careful about changing the specification values for these operands.

Note that these operands are not described in detail here because their specification values are related to the system's complex internal processing. You should refer to the specification guidelines for each operand that are provided in the manual *HiRDB Version 8 System Definition*.

Table 21-2 lists the operands for tuning the system's internal processing.

Table 21-2: Operands for tuning the system's internal processing

Specification item	Specification condition	Operand
Lock-release detection method	For general UAP execution	`pd_lck_release_detect`
Lock-release detection interval		`pd_lck_release_detect_interval`
Processing request queuing method		`pd_server_entry_queue`
Inter-thread lock sleep method		`pd_thdlock_sleep_func`
Inter-thread lock sleep time		`pd_thdlock_retry_time`
Inter-thread spin lock spin count		`pd_thdspnlk_spn_count`
Global buffer lock-release detection interval	Global buffer lock-release-wait has occurred.	`pd_dbbuff_wait_interval`
Spin count during global buffer lock release detection		`pd_dbbuff_wait_spn_count`
Global buffer lock-release detection method	Global buffer is being used.	`pd_dbbuff_lock_release_detect`
Spin count during global buffer lock-acquisition-wait		`pd_dbbuff_lock_spn_count`
Interval during global buffer lock-acquisition-wait		`pd_dbbuff_lock_interval`

(2) Operands for specifying performance during concurrent transaction execution

Table 21-3 shows the operands for specifying performance during concurrent transaction execution.

Table 21-3: Operands for specifying performance during concurrent transaction execution

Specification item	Specification condition	Operand
Inter-thread lock release notification method	When the execution time must be the same for all transactions that are being executed concurrently.	`pd_thdlock_wakeup_lock`
Inter-thread lock release detection interval	When the execution times may differ to some extent among the transactions that are being executed concurrently.	`pd_thdlock_pipe_retry_interval`

Chapter

22. Using the Security Audit Facility

This chapter explains the environmental setup and operating procedures for the security audit facility.

22.1 Overview of the security audit facility

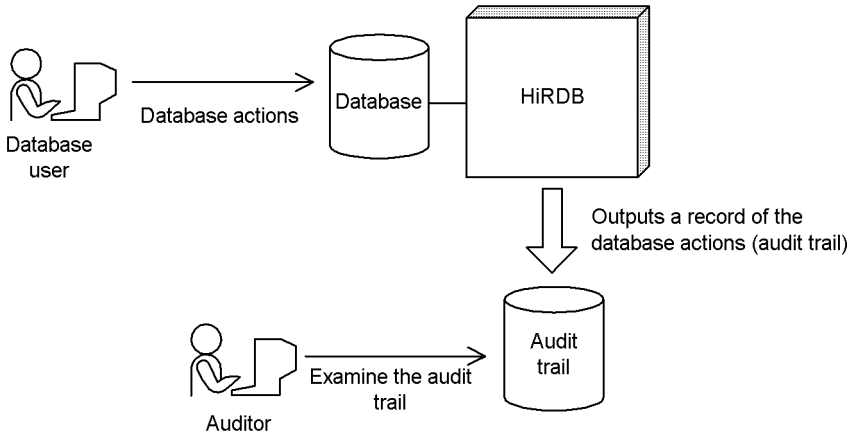
This section provides an overview of the security audit facility. The following topics are covered:

- About the security audit facility
- Triggers for collecting audit trails
- Examples of audit trail collection
- Information collected in an audit trail
- Accessing an audit trail
- System configuration requirements
- Audited events

22.1.1 About the security audit facility

HiRDB security is protected by privileges. The information that can be referenced, the information that can be updated, and the objects that can be manipulated (tables, indexes, etc.) are restricted by privileges. HiRDB can keep track of various operations on the database in order to make it possible to determine whether or not the privileges are operating appropriately. This facility is called the security audit facility, and the record of actions that is output is called an audit trail. The audit trail that is output can be examined to check for inappropriate accesses. This check can be performed by the user who holds the audit privilege (called the *auditor*). Figure 22-1 outlines the security audit facility.

Figure 22-1: Outline of the security audit facility



The audit trail collects information on who performed what action on what using what privilege. The auditor can specify the actions that are to be collected in the audit trail using the `CREATE AUDIT` statement. An audit trail is collected whenever an action is performed for which collection of an audit trail has been specified.

Reference note:

The security audit facility is not intended to strengthen security. Its purpose is to output a record of database accesses for checking that privilege operations are functioning appropriately.

22.1.2 Triggers for collecting audit trails

HiRDB collects audit trails at the following triggers:

- At the time of privilege checking for executing a command or SQL statement
- When an event terminates

HiRDB does not collect an audit trail when an SQL syntax error occurs or when an error occurs as a result of a mistake in entering a command.

(1) Audit trail collection during privilege checking

A single audit trail is collected for each privilege check. Sometimes, another SQL is executed as an extension to an executed SQL. In this case, HiRDB also collects an audit trail during privilege checking for the SQL that is executed as an extension to the original SQL. For details, see *22.12 Audit trail record items (during privilege checking)*.

(2) Audit trail collection at event termination

When an event terminates, HiRDB collects an audit trail for each object that was the target of the event. However, for a consecutive certification failure account lock state that is released because the account lock period has passed, the audit trail is collected not when the account lock period has passed, but when the next event occurs after the passage of the account lock period.

- When `CONNECT` is executed
- When `DROP CONNECTION SECURITY` is executed
- When the `pdacunlck` command is executed

For details about the audit trail output patterns at event termination, see *22.3 Audit trail output patterns*.

For details about the audit trails that are output at event termination, see *22.13 Audit trail record items (at event termination)*.

22.1.3 Examples of audit trail collection

Examples of audit trail collection are shown below.

Example 1: Collect an audit trail of table accesses

When a table is accessed, the table's access privilege (SELECT privilege) is used, and an audit trail is collected.

Table search contents (SQL specification)		Audit trail contents				
		Executor	Privilege used	Manipulated object	Manipulated object's name	Action
A user (USR1) issues the following SELECT statement: SELECT C1 FROM USR1.T1	Privilege	USR1	Table access (SELECT privilege)	Table	USR1.T1	Accesses a table (SELECT)
	Termination	USR1	—	Table	USR1.T1	Accesses a table (SELECT)
A user (USR2) issues the following SELECT statement: SELECT T1.C1,T2.C1 FROM USR1.T1 T1,USR2.T2 T2 WHERE T1.C1=T2.C1	Privilege	USR2	Table access (SELECT privilege)	Table	USR1.T1	Accesses a table (SELECT)
		USR2	Table access (SELECT privilege)	Table	USR2.T2	Accesses a table (SELECT)
	Termination	USR2	—	Table	USR1.T1	Accesses a table (SELECT)
		USR2	—	Table	USR2.T2	Accesses a table (SELECT)

Legend:

Privilege: Audit trail collected during privilege checking

Termination: Audit trail collected at event termination

—: Not applicable

Example 2: Collect an audit trail of table definitions and deletions

When a table is defined or deleted, schema owner privilege, table owner privilege, and RDAREA usage privilege are used, and an audit trail is collected.

Table search contents (SQL specification)		Audit trail contents				
		Executor	Privilege used	Manipulated object	Manipulated object's name	Action
A user (USR1) issues the following CREATE TABLE statement: CREATE TABLE T1(C1 INT) IN RDAREA1	Privilege	USR1	RDAREA usage privilege	RDAREA	RDAREA1	Creates a definition
		USR1	Owner	Schema	USR1	Creates a definition
		USR1	Owner	Table	USR1.T1	Creates a definition
	Termination	USR1	—	Table	USR1.T1	Creates a definition
A user (USR2) issues the following DROP TABLE statement: DROP TABLE T1	Privilege	USR2	Owner	Table	USR2.T1	Deletes a definition
	Termination	USR2	—	Table	USR2.T1	Deletes a definition

Legend:

Privilege: Audit trail collected during privilege checking

Termination: Audit trail collected at event termination

—: Not applicable

22.1.4 Information collected in an audit trail

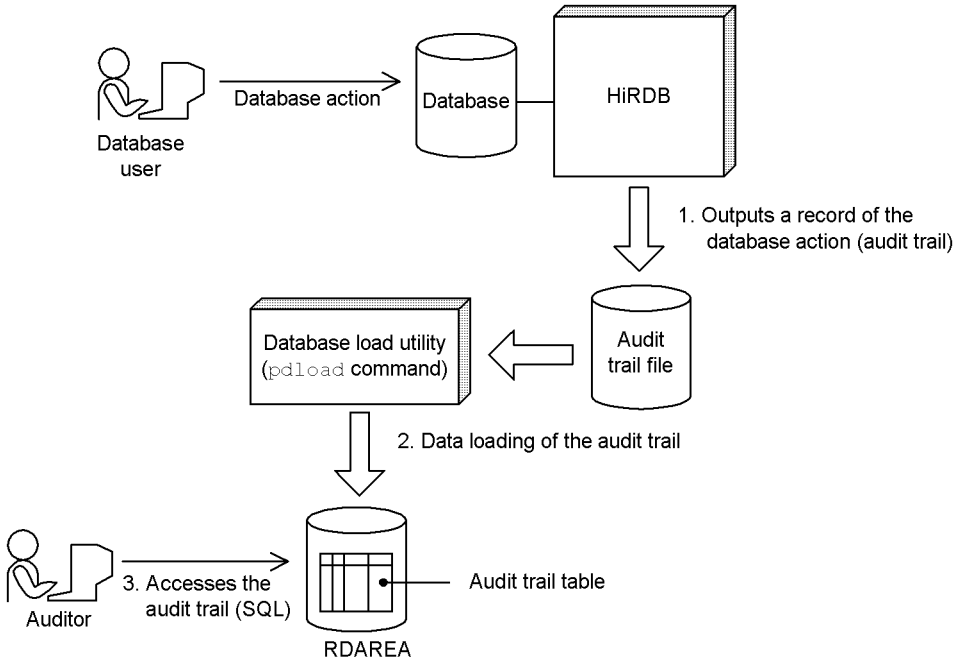
For the information collected in audit trails, see the following sections:

- *22.2 Information output to an audit trail file*
- *22.8 Audit trail table columns*
- *22.12 Audit trail record items (during privilege checking)*
- *22.13 Audit trail record items (at event termination)*

22.1.5 Accessing an audit trail

The audit trail is output to an audit trail file. The data in an audit trail file can be accessed using SQL after the data has been loaded into the audit trail table by the database load utility (pdload command). The auditor can reference an audit trail table (but cannot modify it). A user other than the auditor can access (but cannot modify) an audit trail table if the auditor has granted access privilege to that user. Figure 22-2 shows how to access the audit trail.

Figure 22-2: Accessing the audit trail



Explanation

1. When audited events are executed, an audit trail is output to an audit trail file. An audit trail file is created in a HiRDB file system area for audit trail files. For details about audited events, see *22.1.7 Audited events*.
2. The audit trail output to an audit trail file becomes the input information to the database load utility (`pdload` command) for data loading to record the data in the audit trail table. For details, see *22.7 Recording data in the audit trail table*.
3. The auditor uses the audit trail table to inspect the audit. For details about the audit trail table, see *22.8 Audit trail table columns*.

Table 22-1 shows the differences between an audit trail table and other tables.

Table 22-1: Differences between an audit trail table and other tables

Action on table	Audit trail table	Other tables
Defining a table	HiRDB administrator uses the `pdmod` command to define the table.	Each user uses `CREATE TABLE` to define the table.

Action on table		Audit trail table	Other tables
Deleting a table		Only the auditor can delete the table. Users with DBA privilege cannot delete the table.	Table owner deletes the table. Users with DBA privilege can also delete the table.
Modifying the table definition		Cannot be done.	Table owner can modify the table.
Granting access privileges to other users		Only the SELECT privilege can be granted.	SELECT, INSERT, UPDATE, and DELETE privileges can all be granted.
Loading data into a table		Can be executed only by the auditor.	Can be executed by the table owner. Can also be executed by other users who are granted access privilege.
Reorganizing a table		Can be executed only by the auditor.	Can be executed by users with DBA privilege. Can also be executed by other users who are granted access privilege.
Usage privilege to RDAREAs for storing the table		Only the auditor has the usage privilege.	Table owner has the usage privilege. Can also be used by other users who are granted the privilege.
Row-partitioning of a table		N	Y
Accessing a table	SELECT	Y	Y
	INSERT	N	Y
	UPDATE	N	Y
	DELETE	Can be performed only by the auditor.	Y
	PURGE	Can be performed only by the auditor.	Y

Legend:

Y: Can be executed.

N: Cannot be executed.

22.1.6 System configuration requirements

The following applies to use of the security audit facility:

- HiRDB/Single Server

  The security audit facility cannot be used with utility special units.

- HiRDB/Parallel Server

  The security audit facility cannot be used with a unit that does not have a server (including a unit the contains only the system manager). If the security audit

facility is to be used in such a situation, provide a front-end server, dictionary server, or back-end server in each unit.

22.1.7 Audited events

Actions that are collected in the audit trail are called audit events. Table 22-2 lists the audit events.

When the security audit facility is enabled, audit trails are output automatically by the system for some events. For other events, the auditor can select whether or not audit trails are to be collected.

Table 22-2: Audit events

Event type	Explanation and audited events	Selectability
System administrator security events	1. Security events performed by the HiRDB administrator or users with DBA privilege are audited. 2. Modifications of the setting values of the connection security facility are audited. 3. Security events performed automatically by the system are audited. An audit trail is output when the following events occur: • HiRDB startup (`pdstart` command)[1] • HiRDB termination (`pdstop` command)[1,2] • Auditor registration (`pdmod` command) • Audit trail table creation (`pdmod` command) • Audit trail file deletion (`pdaudrm` command)[3] • Audit trail collection startup[5] • Audit trail collection termination[6] • Start of audit trail file overwriting • Transition to consecutive certification failure account lock state • Release of consecutive certification failure account lock state Applicable in the following cases: ● During `CONNECT` after the account lock period has passed ● During execution of `DROP CONNECTION SECURITY` ● During execution of the `pdacunlck` command • Transition to password-invalid account lock state • Release of password-invalid account lock state • Modification of a setting value of the connection security facility: ● Permitted number of consecutive certification failures ● Account lock period ● Items to be set up for character string restrictions for passwords (including an advance check) • Execution of the `pdacnlck` command	N (an audit trail is always output).

Event type	Explanation and audited events	Selectability
Auditor security events	These are audits of events performed by the auditor. An audit trail is output when the following events occur: • Data loading into an audit trail table (pdload command) • Swapping of audit trail files (pdaudswap command) • Defining events to be audited (CREATE AUDIT)[4] • Deleting events to be audited (DROP AUDIT)[4] • Changing the auditor password (GRANT AUDIT)[4]	N (an audit trail is always output).
Session security events	These are audits of user authentication by authorization identifier and password. An audit trail is output when the following events occur: • Connection to HiRDB (CONNECT statement) • User change (SET SESSION AUTHORIZATION statement)	Y
Privilege management events	These are audits of addition or deletion of user privileges. An audit trail is output when the following events occur: • User privilege addition (GRANT statement) • User privilege deletion (REVOKE statement)	Y[7]

Event type	Explanation and audited events	Selectability
Object definition events	These are audits of object definitions, deletions, or modifications. An audit trail is output when the following events occur: • Object definition; this applies to the following SQL statements: CREATE ALIAS CREATE FOREIGN INDEX CREATE FOREIGN TABLE CREATE FUNCTION CREATE INDEX CREATE PROCEDURE CREATE PUBLIC VIEW CREATE SCHEMA CREATE SERVER CREATE TABLE CREATE TRIGGER CREATE TYPE CREATE USER MAPPING CREATE VIEW • Object deletion; this applies to the following SQL statements: DROP ALIAS DROP DATA TYPE DROP FOREIGN INDEX DROP FOREIGN TABLE DROP FUNCTION DROP INDEX DROP PROCEDURE DROP PUBLIC VIEW DROP SCHEMA DROP SERVER DROP TABLE DROP TRIGGER DROP USER MAPPING DROP VIEW • Object modification; this applies to the following SQL statements: ALTER PROCEDURE ALTER ROUTINE ALTER TABLE ALTER TRIGGER COMMENT	Y[7]

Event type	Explanation and audited events	Selectability
Object manipulation events	These are audits of object manipulations. An audit trail is output when the following events occur: • Table reference (SELECT statement) • Table row insertion (INSERT statement) • Table row update (UPDATE statement) • Table row deletion (DELETE statement) • Table deletion of all rows (PURGE TABLE statement) • Stored procedure execution (CALL statement) • Table lock control (LOCK TABLE statement) • List creation (ASSIGN LIST statement)	Y[7]
Utility operation event	Security events related to object operations performed by a utility or command are audited. An audit trail is output when any of the following is executed: • Database load utility (pdload command) Target object: TABLE • pddefrev command Target objects: ALIAS, PROCEDURE, TABLE, TRIGGER, and VIEW • Database reorganization utility (pdrorg command) Target object: TABLE • Dictionary import/export utility (pdexp command) Target objects: ALIAS, PROCEDURE, TABLE, TRIGGER, and VIEW • Integrity check utility (pdconstck command): TABLE	Y[7, 8]

Legend:

Y: Can be selected

N: Cannot be selected

[1] HiRDB/Parallel Server server unit startup and termination are not regarded as audit events.

[2] Normal termination and planned termination are regarded audit events. Forced termination and abnormal termination are not regarded as audit events. To audit forced termination or abnormal termination, use the messages output by HiRDB or the OS.

The following termination commands are not audited:

- pdstop -f
- pdstop -f -q
- pdstop -f -x *host-name*
- pdstop -f -u *unit-identifier*
- pdstop -f -s *server-name*

- `pdstop -f -u` *unit-identifier* `-s` *server-name*
- `pdstop -z`
- `pdstop -z -q`
- `pdstop -z -c`
- `pdstop -z -s` *server-name*

[3] Creation of an audit trail file is not regarded as an audit event. To audit creation of audit trail files, use the OS's audit facility.

[4] An output trail is also output when the database definition utility (`pddef` command) or the interactive SQL execution utility (`pdsql` command) is executed.

[5] An audit trail is output when an audit trail is to be collected because the `pdaudbegin` command is executed, or when HiRDB is started.

[6] An audit trail is output when HiRDB is terminated normally or by a planned termination when the `pdaudend` command is executed or an audit trail is collected.

[7] If the event-target object in a privilege control event, object definition event, object operation event, or utility operation event is an audit trail table, a view table that uses an audit trail as the base table, or a list that uses an audit trail as the base table, an audit trail is output unconditionally when the event terminates. You can select whether to output an audit trail during privilege checking. However, because the database load utility (`pdload` command) executed for an audit trail table is included as an auditor security event, the audit trail at event termination and during privilege checking is output unconditionally.

[8] When the database reorganization utility (`pdrorg` command) is used to reload a dictionary table, an audit trail is output unconditionally.

22.2 Information output to an audit trail file

Table 22-3 lists the information output to an audit trail file.

Table 22-3: Information output to an audit trail file

Collected information	Explanation
User identifier	Authorization identifier of user who executed the audited event
Event execution date	Date the event was executed
Event execution time	Time the event was executed
Event execution duration	Amount of time required to execute the event (in microseconds)
Event type	Type of event (see Table *22-16 Event types and subtypes*)
Event subtype	Event's subtype (see Table *22-16 Event types and subtypes*)
Event success or failure	Result of event execution (whether or not privilege checking was successful)
Privilege used	Privilege used when the event was executed
UAP name	UAP name specified in client environment definition's `PDCLTAPNAME` operand
Service name	Name of the service requested by the UAP that issued the event. If it is an OpenTP1 SUP (service use program), this is the name of the service requested of the SPP (service provider program); if it is TP1/Message Control, this is the name of the service requested of the MHP (message processing program).
IP address	IP address of the client that executed the UAP that issued the event*
Process number	Process ID of the UAP that issued the event*
Thread number	Thread ID of the UAP that issued the event*
Host name	Target host name of the UAP connection that issued the event
Unit identifier	Target unit identifier of the UAP connection that issued the event
Server name	Target front-end server name or single-server name of the UAP connection that issued the event
Connect number	Connect number of the user that issued the event
SQL number	Event SQL number

Collected information	Explanation
Object owner name	Owner name of the object that is the target of the event privilege check
Object name	Object name of the object that is the target of the event privilege check
Object type	Object type of the object that is the target of the event privilege check
Added, deleted, or changed privilege	Privilege added, deleted, or changed due to the event
One of the following is acquired: • User identifier of user who granted, deleted, or changed a privilege • User identifier associated with the event target	• Identifier of the user whose privilege was granted, deleted, or changed by the event • Authorization identifier that became the event target
Security audit facility operand values	Values of operands related to security audit facility (values applicable at time of HiRDB startup)
Audit trail type	Indicator of privilege checking or event termination
SQL code or termination code	Termination code of the SQL, utility, or command
Audit trail file name at swapping source	Audit trail file name at the swapping source when swapping occurs
Audit trail file name at swapping destination	Audit trail file name at the swapping destination when swapping occurs
Type of setting change for connection security facility	Type of setting change for the connection security facility (the change type is set also when a password is changed)
Operand value related to connection security facility (before change)	Operand value related to the connection security facility before the change
Operand value related to connection security facility (after change)	Operand value related to the connection security facility after the change
Audit trail table option	Flag when the event option target is an audit trail table, a view table that uses an audit trail as the base table, or a list that uses an audit trail
Number of accesses	Number of rows searched, inserted, updated, and deleted in an object (base table, view table, foreign table, table alias, or list) by the event

Note

The information that is collected depends on the event. For a list of the information that is collected for each event, see *22.12 Audit trail record items*

(during privilege checking) and *22.13 Audit trail record items (at event termination)*.

* When introducing applications under Open/TP1 or when introducing web servers and other products, information on the application connected to HiRDB is collected, rather than information on the application executed by the end user.

22.3 Audit trail output patterns

This section explains the audit trail output patterns.

22.3.1 Output patterns during privilege checking

A single audit trail is collected during a single privilege check. However, in the exceptional cases shown in Table 22-4, multiple audit trails are collected during a single privilege check.

Table 22-4: Event execution units and audit trail record output units

Events	Execution unit	Target output units	Number of output records
GRANT, REVOKE	User	Users	Number of target users
	Group	Groups	Number of target groups
	Role	Roles	Number of target roles
CREATE CONNECTION SECURITY, DROP CONNECTION SECURITY	Setting value of the connection security facility	Setting values of the connection security facility	Number of setting values of the connection security facility

When multiple privilege checks occur in a single event, audit trails are output as follows:

- When any one of multiple privileges is required

  An audit trail is output during one of the privilege checks.

  Example

  To unload someone else's table, you must have either DBA privilege or SELECT privilege.

  Whether the DBA privilege check or the SELECT privilege check was successful is output in the audit trail.

- When all of multiple privileges are required

  Audit trails are output during all the privilege checks.

  Example

  To reorganize someone else's table, you must have the INSERT, DELETE, and SELECT privileges.

  If the INSERT, DELETE, and SELECT privilege checks are all successful,

audit trails are output for all three checks. If an error occurs in midstream, audit trails are output for the successful privilege checks. Failure trails are output for the unsuccessful privilege checks.

22.3.2 Output patterns at event termination

Multiple audit trails are collected for each event (one audit trail for each event-target object). As a rule, the same contents are output for all items, except for the operation-target object information column of the audit trail table. However, if the result of an event is different for each operation target, such as a utility, different contents are output for each termination code. Examples of audit trail output patterns are described below.

(1) Multiple target objects or target users

(a) Multiple target objects

For example, when `SELECT * FROM "T1","T2"` is executed, two audit trail lines are output, one for object `T1` and one for `T2`, as shown below. In this case, the same termination code is set for both.

...	Event subtype	...	Object name	...	Termination code	...
...	SEL	...	T1	...	XXX	...
...	SEL	...	T2	...	XXX	...

An event, such as routine re-creation or the utility for reorganizing a database in units of schemas, may succeed or fail depending on the object. In this case, a different termination code is output for each object. Table 22-5 lists events that sometimes have multiple target objects.

Table 22-5: Events that sometimes have multiple target objects (event execution units and audit trail record output units)

Events	Execution unit	Target output units	Number of output records	SQL code or termination code
GRANT, REVOKE	User	Users	Number of target users	Same value for all event-target objects
	Group	Groups	Number of target groups	
	Role	Roles	Number of target roles	
CREATE CONNECTION SECURITY, DROP CONNECTION SECURITY	Setting value of the connection security facility	Setting values of the connection security facility	Number of setting values of the connection security facility	
Data manipulation SQL	Base table	Base tables	Number of target base tables	
	View table	View tables	Number of target view tables	
	Table alias	Table aliases	Number of target table aliases	
	Foreign table	Foreign tables	Number of target foreign tables	
ALTER ROUTINE, ALTER PROCEDURE, ALTER TRIGGER	Multiple routines or triggers[1]	Procedures, functions, or triggers	One for each target routine	Different value for each event-target object
pdexp, pddefrev	Base table	Base tables	Number of target base tables	
	Viewed table	View tables	Number of target view tables	
	Table alias	Table aliases	Number of target table aliases	
	Stored procedure	Stored procedures	Number of target stored procedures	
	Trigger	Triggers	Number of target trigger count	

Events	Execution unit	Target output units	Number of output records	SQL code or termination code
pdrorg	Schema,[2] base table, or index	Base tables	Number of target base tables	
pdaudbegin, pdaudend	Unit	Units	Number of target units	

[1] If routine information cannot be collected, NULL is entered for the object information and only a single audit trail line is output.

[2] In the case of reorganizing in units of schemas, if the table information in the schema cannot be collected, schema information is entered for the object information, and only a single audit trail line is output.

(b) Subqueries

For example, when `UPDATE "T1" SET "C1" = (SELECT "C1" FROM "T2" WHERE CODE = '01') WHERE CODE = '01'` is executed, information on the accessed objects is output as in (a) above. In this case, audit trails of the actions performed on the objects are output for the event type and subtypes as shown below. The same value is entered for the SQL code in the audit trail at each event's termination.

...	Event subtype	...	Object name	...	Termination code	...
...	UPD	...	T1	...	XXX	...
...	SEL	...	T2	...	XXX	...

(2) SQL that runs internally

In some cases, another SQL is executed as an extension of an executed SQL. If an SQL that is executed as an extension when an event is executed is defined as the audit target, information is collected on the target object of the SQL that is executed as an extension, or on the user. In this case, the event type and subtype of the SQL that is executed as an extension are entered as the event type and subtype of the object or user. Furthermore, for all objects, the same value is entered for the SQL code of each audit trail.

For example, if a table is deleted as an extension of `DROP SCHEMA`, object information on the table is output as the accessed object, separately from schema information. If the deletion process is terminated by an error, an audit trail is output for each user and each object that has been accessed up to the time of the error.

In procedures and triggers, an SQL is executed automatically as an extension of

another SQL. In this case, an event termination audit trail is output for each SQL that is executed as an extension. In the case of a trigger, because the executor switches temporarily to the trigger owner, the trigger owner is entered as the event executor. In the case of a procedure, an event termination audit trail is also output when a CALL statement is terminated. If the procedure is nested, an event termination audit trail is output for each CALL statement. Table 22-6 shows audit trail record output units for trigger and procedure execution.

Table 22-6: Audit trail record output units for trigger and procedure execution

Type		Event executor (authorization identifier value)	Audit trail record output unit
Trigger	User trigger	Trigger owner	SQL units in the trigger
	System trigger (internal trigger of a referential constraint, etc.)	Trigger owner	SQL units in the system trigger
Procedure		Procedure executor	SQL units in the procedure + CALL statement

If an error occurs in an SQL in a trigger or procedure, the same SQL record is output as the event results of the SQL at that point in time, the event results of the SQL that became the trigger, and the results of the CALL statement. Table 22-7 shows the error locations in a trigger and the audit trail (SQL code) details. Table 22-8 shows the error locations in a procedure and the audit trail (SQL code) details.

Table 22-7: Error locations in a trigger and audit trail (SQL code) details

BEFORE trigger SQL result [1]	Trigger SQL result [2]	AFTER trigger SQL result [3]	Audit trail (SQL code) details		
			Audit trail of [1]	Audit trail of [2]	Audit trail of [3]
Error	—	—	SQL record of [1]	SQL record of [1]	N
Normal	Error	—		SQL record of [2]	N
	Normal	Error		SQL record of [3]	SQL record of [3]
		Normal	0	0	0

Legend:

—: Not applicable

N: No audit trail (SQL code) is output.

Table 22-8: Error locations in a procedure and audit trail (SQL code) details

Nesting?	Details of event for which audit trail is to be collected					Audit trail (SQL code) details				
	[1]	[2]	[3]	[4]	[5]	Audit trail of [1]	Audit trail of [2]	Audit trail of [3]	Audit trail of [4]	Audit trail of [5]
No	E	—	—	—	E	SQL record of [1]	N	N	N	SQL record of [1]
	Nrm	—	—	—	E	0	N	N	N	SQL record of [5]
					Nrm	0	N	N	N	0
Yes	E	—	—	—	E	SQL record of [1]	N	N	N	SQL record of [1]
	Nrm	E	E	—	E	0	SQL record of [2]	SQL record of [2]	N	SQL record of [2]
		Nrm	E	—	E	0	0	SQL record of [3]	N	SQL record of [3]
			Nrm	E	E	0	0	0	SQL record of [4]	SQL record of [4]
				Nrm	E	0	0	0	0	SQL record of [5]
					Nrm	0	0	0	0	0

Legend:

—: Not applicable

E: Error

Nrm: Normal

N: No audit trail (SQL code) is output.

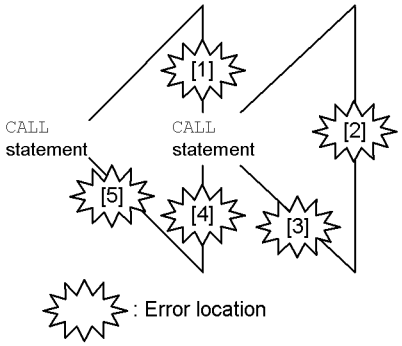
[1]: Result of SQL in a procedure before nesting

[2]: Result of SQL in a procedure after nesting

[3]: Result of a nested CALL statement

[4]: Result of SQL in a procedure before nesting

[5]: Result of a CALL statement



(3) Dynamic SQL

In the case of a dynamic SQL, the entire process from PREPARE to execution is treated as a single SQL, and an event termination audit trail is output for that unit. An event termination audit trail during execution is output when OPEN, CLOSE or EXECUTE is completed. Figure 22-3 shows the flow of a dynamic SQL depending on the type of data manipulation SQL.

Figure 22-3: Flow of a dynamic SQL depending on the type of data manipulation SQL

Flow of dynamic SQL	Data manipulation SQL		
	For SELECT (with a cursor)	For UPDATE, DELETE, INSERT, CALL, or PURGE	
↓	PREPARE ↓ OPEN ↓ FETCH ↓ CLOSE	PREPARE ↓ EXECUTE	EXECUTE IMMEDIATE

Explanation

The shaded SQLs are triggers for output of an audit trail. Note that the audit trail collection method for PREPARE, CLOSE and OPEN depends on where an error occurs. Table 22-9 shows whether an audit trail is output depending on the success or failure of an event during dynamic SQL execution.

Table 22-9: Audit trail output depending on the success or failure of an event during dynamic SQL execution

SQL result			Collection specification		
			Upon success	Upon failure	Both
PREPARE successful	OPEN (EXECUTE) successful	CLOSE successful	Output on CLOSE (EXECUTE)	Not output	Output on CLOSE (EXECUTE)
		CLOSE failed	Not output	Output on CLOSE (EXECUTE)	Output on CLOSE (EXECUTE)
	OPEN (EXECUTE) failed		Not output	Output on OPEN (EXECUTE)	
PREPARE failed	—		Not output	Output on PREPARE	
EXECUTE IMMEDIATE			Output on EXECUTE IMMEDIATE		

Legend:

—: Not applicable

Note

An event termination audit trail is output when the cursor is closed by executing the CLOSE statement internally without having to execute it explicitly. For the timing, see the section on the CLOSE statement in the manual *HiRDB Version 8 SQL Reference*.

22.3.3 Relationships among audit trails

How audit trails that are output during privilege checking are related to audit trails output at event termination and how multiple records are related to events can be checked from the following information:

- For an SQL, check the combination of the server name, CONNECT serial number, and SQL serial number that are output in an audit trail.
- For a command or utility, check the combination of the process ID and host name that are output in an audit trail.

For details about audit trail output contents, see *22.8 Audit trail table columns*.

22.4 Environment settings

The following is the procedure for making the environment settings for the security audit facility.

Procedure

1. Specify HiRDB system definition operands related to the security audit facility.
2. Create a HiRDB file system area for the audit trail files.
3. Register the auditor, create the RDAREA for storing the audit trail table, and create the audit trail table.
4. Define the audit events.

Steps 1-3 are performed by the HiRDB administrator, and step 4 is performed by the auditor. The procedure step numbers correspond to the section numbers of the explanation that follows. For example, step 3 above is explained in section 22.4.3 below.

22.4.1 Security audit facility operand specifications

Executor: HiRDB administrator

Table 22-10 lists the operands that can be specified for using the security audit facility.

Table 22-10: Operands specified for using the security audit facility

Operand	Explanation
`pd_audit`	Specifies whether or not collection of an audit trail is to start from the time HiRDB starts: `Y`: Collect an audit trail from the time of HiRDB startup. `N`: Do not start collecting an audit trail at the time HiRDB starts. Even if N is specified in this operand, an audit trail can be collected by executing the `pdaudbegin` command. If `Y` is specified in the `pd_audit` operand, or if the `pdaudbegin` command is executed, audit trails for the following events are collected unconditionally: • System administrator security events • Auditor security events For other events, use `CREATE AUDIT` to specify whether an audit trail is to be collected. For details, see *22.4.4 Audit event definition*.

Operand	Explanation
pd_aud_file_name	Specifies the HiRDB file system area to be used for the audit trail files. HiRDB creates the audit trail files in this HiRDB file system area. This operand must be specified when the security audit facility is used. If it is not specified, the security audit facility cannot be used. When this operand is specified, HiRDB will not start if an access error occurs in the HiRDB file system area for the audit trail files during startup of HiRDB (or unit for a HiRDB/Parallel Server).
pd_aud_max_generation_size	Specifies the maximum size of an audit trail file.
pd_aud_max_generation_num*	Specifies the maximum number of audit trail file generations to be created in the HiRDB file system area for audit trail files.
pd_aud_no_standby_file_opr	Specifies the processing when there are no available swappable audit trail files: down: When there is one or fewer swappable audit trail files available, HiRDB (or unit for a HiRDB/Parallel Server) is to be terminated forcibly. forcewrite (default): When there are no swappable audit trail files available, a data load waiting audit trail file (excluding files in shutdown status) is to be forcibly made the swap target so that audit trail output will continue. For details about the status of audit trail files, see *22.6 Operation of audit trail files*.
pd_aud_async_buff_size	Specifies the buffer length to be used when the audit trail is output asynchronously.
pd_aud_async_buff_count	Specifies the number of buffers to be used when the audit trail is output asynchronously.
pd_aud_async_buff_retry_intvl	Specifies the interval at which buffer monitoring is to be retried until an unused buffer is allocated when all the buffers used for asynchronous output of an audit trail are in use.
pd_aud_file_wrn_pnt	Specifies that a warning message is to be output when the number of unswappable audit trail files reaches a warning level. Specify for the warning value a value that is less than the maximum number of audit trail file generations specified in the pd_aud_max_generation_num operand.

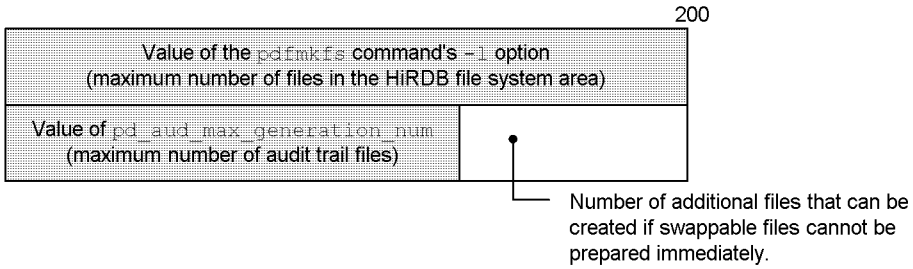
* Specify the value of the pd_aud_max_generation_num operand so that it satisfies the following condition:

- Value of pd_aud_max_generation_num < value of pdfmkfs command's -l *option*

The -l option specifies the maximum number of files to be created in the HiRDB

file system area used for the audit trail files, which is discussed later. If the value of the operand is specified so that this condition is satisfied, then if swappable files cannot be prepared immediately, you will be able to increase the value of the pd_aud_max_generation_num operand. Figure 22-4 shows the recommended specification.

Figure 22-4: Recommended relationship between the value of pd_aud_max_generation_num and the -l option



22.4.2 Creation of the HiRDB file system area for the audit trail files

Executor: HiRDB administrator

Use the pdfmkfs command to create the HiRDB file system area to be used by HiRDB for creation of the audit trail files. The following should be considered when you create the HiRDB file system area for audit trail files:

- Specify the HiRDB file system area name specified in the pd_aud_file_name operand.
- Create the HiRDB file system area for the audit trail files as a character special file. Do not create it as a normal file.
- For a HiRDB/Parallel Server, create a HiRDB file system area for audit trail files in each unit.
- For the -k option (usage), specify SYS or SVR. The recommended value is SYS.
- The target for specifying the -n option (HiRDB file system area length) is as follows:

  Recommended value: Value of pd_aud_max_generation_size × 200 + 20 (megabytes)

  Minimum value: Value of pd_aud_max_generation_size × value of pd_aud_max_generation_num + 20 (megabytes)
- The target for specifying the -l option (maximum number of file generations in the HiRDB file system area) is as follows:

  Recommended value: 200 (maximum value of

pd_aud_max_generation_num)

Minimum value: Value of pd_aud_max_generation_num

22.4.3 Auditor registration, creation of the RDAREA to store the audit trail table, and creation of the audit trail table

Executor: HiRDB administrator

Execute the database structure modification utility (pdmod command), and perform the following. All of the following tasks can be performed simultaneously with a single issuance of the pdmod command.

- Register the auditor
- Create the RDAREA that will store the audit trail table
- Create the audit trail table

(1) Register the auditor

Register the auditor with the pdmod command's create auditor statement. The auditor can perform the following operations:

- Data load the audit trail table
- Swap audit trail files
- Access, update, and delete the audit trail table

Notes

- The registered auditor cannot be deleted and updated at the same time.
- Create a user without DBA privilege as the auditor. The HiRDB administrator cannot be the auditor.
- Only one person can be registered as the auditor (there cannot be multiple auditors).
- The auditor cannot hold the DBA privilege. Table 22-11 lists the user privileges that can be held by the auditor.

Table 22-11: User privileges that can be held by the auditor

Type of user privilege	Yes/ No	Remarks
Audit privilege	Yes	These privileges are granted automatically when the auditor is registered.
CONNECT privilege	Yes	
Schema definition privilege	Yes	

Type of user privilege	Yes/ No	Remarks
DBA privilege	No	The auditor cannot hold the DBA privilege.
RDAREA usage privilege	Yes	The privilege to use the RDAREA that stores the audit trail table must be granted by a user who has the DBA privilege. This also applies to use of other RDAREAs.
Table access privilege	Yes	The auditor can hold audit trail table access privilege. The privilege to access other tables must be granted by each table's owner.

Legend:

Yes: This privilege can be held.

No: This privilege cannot be held.

Notes when the Directory Server linkage facility is used

- Register the auditor's user information with the Directory Server.
- The auditor changes the password with the `GRANT AUDIT` statement. It must be changed to the password registered with the Directory Server.

(2) Create the RDAREA to store the audit trail table

Create the RDAREA where the audit trail table will be stored using the `pdmod` command's `create rdarea` statement. The following should be considered when creating the RDAREA for storing the audit trail table:

- Make the RDAREA type a user RDAREA.
- Grant RDAREA usage privilege only to the auditor. Do not make it a shared RDAREA and do not grant usage to it to other users. Specify the RDAREA usage privilege with the `create rdarea` statement's `for user used by` operand.
- Allocate global buffers to the added RDAREA.
- The audit trail table can be stored in an existing RDAREA, but the audit trail table cannot be stored in a shared RDAREA or in an RDAREA to which users other than the auditor have usage privileges. In such a case, change the RDAREA usage privilege so that only the auditor has it.
- Before re-creating or modifying the RDAREA that stores the audit trail table, the auditor must delete the audit trail table.

(3) Create the audit trail table

Create the audit trail table with the `pdmod` command's `create audit table` statement. The following should be considered when creating the audit trail table:

- Only one audit trail table can be created.

- The audit trail table can be deleted and redefined. To delete the audit trail table, the auditor executes the DROP TABLE statement. Users other than the auditor cannot delete the audit trail table. To re-create the audit trail table after it has been deleted, the HiRDB administrator executes the pdmod command's create audit table statement.
- The audit trail table cannot be row partitioned.
- The table definition of the audit trail table cannot be modified.

22.4.4 Audit event definition

Executor: Auditor

System administrator security events and auditor security events are included in the events to be audited. CREATE AUDIT is used to define whether or not the following are audited:

- Session security events
- Privilege management events
- Object definition events
- Object manipulation events
- Utility manipulation events

Use DROP AUDIT to remove these audit events from being audited. For details about CREATE AUDIT and DROP AUDIT, see the manual *HiRDB Version 8 SQL Reference*.

22.5 Operating procedure

This section explains the actions performed by the HiRDB administrator and the tasks performed by the auditor when the security audit facility is used.

22.5.1 Actions performed by the HiRDB administrator

(1) Collect an audit trail

The HiRDB administrator uses one of the following methods to collect an audit trail:

- Specify `Y` for the `pd_audit` operand

  In this case, an audit trail will be collected beginning at the time of HiRDB startup.

- Execute the `pdaudbegin` command

  In this case, the audit trail will be collected beginning at the time the command is executed.

To stop collection of the audit trail, execute the `pdaudend` command. Only the HiRDB administrator can execute this command. The auditor cannot use this command.

Reference note:

- If HiRDB is restarted, the previous operating status is inherited. If an audit trail was being collected, it will be collected after restart; if an audit trail was not being collected, it will not be collected after restart.
- If HiRDB undergoes a normal startup, then rather than the previous operating status, the specification of the `pd_audit` operand takes precedence. Even if an audit trail was being collected, if `pd_audit=N` is specified, no audit trail will be collected after a normal startup. If an audit trail was not being collected but `pd_audit=Y` is specified, an audit trail will be collected after the normal startup.

(2) Delete audit trail files

Audit trail files can be deleted with the `pdaudrm` command. Only the HiRDB administrator can use this command. The auditor cannot use this command.

(3) Manipulate the RDAREA that stores the audit trail table

The HiRDB administrator manipulates the RDAREA that stores the audit trail table. For example, the HiRDB administrator can perform the following actions:

- Back up the RDAREA that stores the audit trail table

- Recover the RDAREA that stores the audit trail table
- Modify the structure of the RDAREA that stores the audit trail table (extend the RDAREA, etc.)
- Add or remove usage privileges for the RDAREA that stores the audit trail table*

* This is performed when the RDAREA that stores the audit trail table is modified. For example, the HiRDB administrator can remove usage privileges for the RDAREA before it is modified, and grant an auditor usage privileges for the RDAREA after it is modified.

(4) Create an HiRDB file system area for the audit trail files

The HiRDB administrator manipulates the HiRDB file system area for the audit trail files. For example, the HiRDB administrator performs the following actions:

- Delete the audit trail table by deleting the HiRDB file system area (delete with an OS command)*
- Back up the HiRDB file system area with the `pdfbkup` command
- Recover the HiRDB file system area with the `pdfrstr` command
- Initialize the HiRDB file system area with the `pdfmkfs` command
- Delete the HiRDB files that store the audit trail table with the `pdfrm` command

* HiRDB does not output this event as part of the audit trail. Use the OS's audit facility to audit this event.

22.5.2 Actions performed by the auditor

Actions 1 and 2 below are performed periodically, and the remaining actions are performed as needed.

(1) Check the status of the audit trail files

Check the status of the audit trail files with the `pdls -d aud` command. Check the following:

- Whether or not there are data load waiting audit trail files
- Whether or not there are swappable audit trail files

For details about the statuses of audit trail files, see *22.6 Operation of audit trail files*.

(2) Record data in the audit trail table (data load the audit trail table)

Record data (the output audit trail) in the audit trail table. Use the database load utility (`pdload` command) to load the data into the audit trail table from a data load waiting audit trail file. For details about the data load procedure, see *22.7 Recording data in the audit trail table*.

(3) Use the audit trail table

Use the audit trail table to perform audits. For details about the columns of the audit table, see *22.8 Audit trail table columns*.

(4) Swap audit trail files

Use the `pdaudswap` command to swap audit trail files. For example, to load the data from the current audit trail file to the audit trail table, swap audit trail files with the `pdaudswap` command and then perform the data load. The current audit trail file cannot be data loaded.

Note:

If either of the following conditions is satisfied, the `pdaudswap` command cannot be executed:

- There are no swappable audit trail files
- No audit trail files have been generated

(5) Manipulate the audit trail table

The auditor manipulates the audit trail table. For example, the auditor performs the following actions:

- Grant access privileges to the audit trail table

  Only the SELECT privilege for the audit trail table and a view table that uses the audit trail table as the base table can be granted to other users. SELECT privilege can also be removed. The INSERT, UPDATE, and DELETE privileges cannot be granted to other users. The auditor adds and removes access privileges to the audit trail table. Access privilege to the audit trail table cannot be granted to anyone other than the auditor.

- Define indexes for the audit trail table

  Indexes can be defined for the audit trail table. For details about the audit table columns, see *22.8 Audit trail table columns*.

  The audit trail table cannot be row partitioned, nor can the table definition be modified.

- Reorganize the audit trail table

  The auditor reorganizes the audit trail table. Users other than the auditor cannot reorganize the audit trail table.

- Delete the audit trail table

  The audit trail table can be deleted with `DROP TABLE`. The auditor deletes the audit trail table. Users other than the auditor cannot delete the audit trail table.

To re-create the audit trail table, the HiRDB administrator uses the `pdmod` command's `create audit` table statement.

(6) Add and delete audit events

Audit events can be added with `CREATE AUDIT`. Unnecessary audit events can be deleted with `DROP AUDIT`.

(7) Change the password

The auditor's password can be changed with `GRANT AUDIT`. The auditor changes the password.

(8) Delete the auditor's schema

If there is no audit trail table, the auditor's schema can be deleted. The auditor and DBA privilege holders can delete the auditor's schema.

22.6 Operation of audit trail files

This section explains in detail the creation of audit trail files. The following topics are covered:

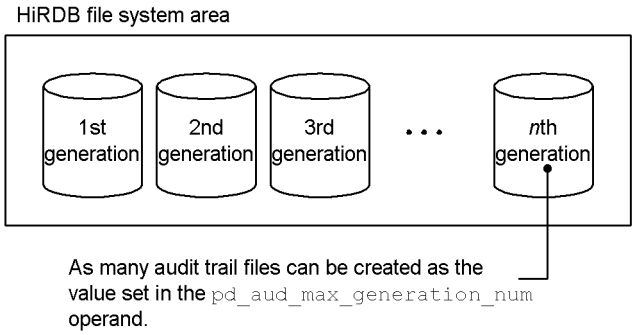
- Creation of audit trail files
- Status of audit trail files
- Swapping of audit trail files

22.6.1 Creation of audit trail files

(1) Audit trail file creation convention

HiRDB creates audit trail files automatically in the HiRDB file system area for audit trail files. The first time an audit trail file is created, it becomes the first generation, followed by the second generation, the third generation, and so on. Figure 22-5 shows the procedure for creating audit trail files.

Figure 22-5: Audit trail file creation



The maximum number of audit trail files that can be created in the HiRDB file system area is specified in the `pd_aud_max_generation_num` operand. However, if the value of the `pdfmkfs` command's `-l` option is smaller than the value of the `pd_aud_max_generation_num` operand, the value of the `-l` option becomes the maximum. Also, if there becomes insufficient space in the HiRDB file system area, it will not be possible to create as many audit trail files as provided for by the specification of the `pd_aud_max_generation_num` operand.

(2) Naming of audit trail files

The audit trail files are named according to the following convention:

`pdaud`*unit-identifier*`001-200.aud`

Example: If the unit identifier is `UNT1` and `pd_aud_max_generation_num=100`

1st generation file name: pdaudUNT1001.aud

2nd generation file name: pdaudUNT1002.aud

:

99th generation file name: pdaudUNT1099.aud

100th generation file name: pdaudUNT1100.aud

If files other than audit trail files are created in the HiRDB file system area for audit trail files, they must not be files whose names will be inconsistent with these naming conventions.

(3) Output to audit trail file

Audit trail output format

For the audit trail output format, the pd_aud_async_buff_size operand can be used to select synchronous output or asynchronous output.

Buffer flush opportunities for asynchronous output

If asynchronous output is selected, when an opportunity to output to the audit trail file occurs, the audit trail is stored temporarily in a buffer used for asynchronous output. The audit trail stored in a buffer used for asynchronous output is output to the audit trail file at the following opportunities; these are called asynchronous output buffer flush opportunities:

- When the buffer used for asynchronous output becomes full
- When the pdaudswap command is executed
- When the pdaudend command is executed
- When the pdstop command is used for normal termination or planned termination of HiRDB (or unit)

Audit trail output processing

Audit trail output processing to the audit trail file is as follows, depending on the output format:

- For synchronous output

  If there are no audit trail files, an audit trail file is created at the first audit trail collection opportunity, and the audit trail is output. If there is at least one audit trail file, the audit trail is output to the current file at each output trail collection opportunity.

- For asynchronous output

  If there are no audit trail files, an audit trail file is created at the first asynchronous output buffer flush opportunity, and the audit trail is output. If

there is at least one audit trail file, the audit trail is output to the current file at each asynchronous output buffer flush opportunity.

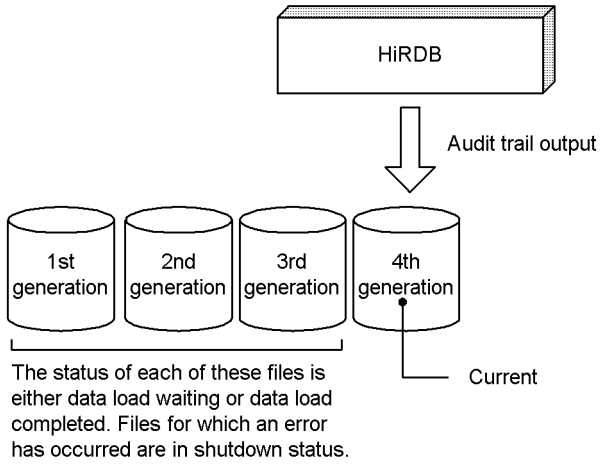
22.6.2 Status of audit trail files

HiRDB manages audit trail files in several statuses, as shown and explained in Table 22-12 and Figure 22-6. The status of audit trail files can be checked with the `pdls -d aud` command.

Table 22-12: Audit trail file statuses

File status	Explanation
Current	This is the file to which the audit trail can be output currently. There is always one file in this status. Files other than the current file are considered to be on standby.
Data load completed	This is a file from which data loading to the audit trail table has been completed. Such a file can be used as the swap target when audit trail file swapping occurs.
Data load waiting	This is a file from which data loading to the audit trail table has not been completed. Such a file cannot be used as the swap target when for audit trail file swapping. However, depending on the value of the `pd_aud_no_standby_file_opr` operand, it may be possible to use a file in this status as the swap target.
Shutdown	This is an audit trail file that cannot be used due to an error during audit trail I/O or with the audit trail file header. Files in this status cannot be used as an audit trail file swap target. Files that are not shut down are called normal files.

Figure 22-6: Audit trail file statuses



22.6.3 Swapping of audit trail files

Changing the audit trail output destination (changing the file that is the current file) is

called audit trail file swapping.

(1) When audit trail files are swapped

Table 22-13 explains the conditions under which audit trail files are swapped.

Table 22-13: Conditions under which audit trail files are swapped

Condition for swapping	Explanation
File is full	When the output audit trail becomes full, the file is swapped. The size of an audit trail file is specified in the `pd_aud_max_generation_size` operand.
Error in the current file	If an error occurs during I/O processing to the current file, the file is swapped.
`pdaudswap` command is executed	When the `pdaudswap` command is executed, the file is swapped.
HiRDB is restarted	When HiRDB (or unit for a HiRDB/Parallel Server) is restarted, the file is swapped. In the case of normal startup or a restart after planned termination, the file is not swapped and the file that was the current file at the time of the previous HiRDB termination continues to be used.

In the following cases, the file that should be swapped in is replaced automatically by another file:

- Generation of the audit trail file that was to be the swap target failed
- Update to the management information failed

(2) Audit trail file swap target

The processing depends on whether or not the number of audit trail files used has reached the maximum (value of `pd_aud_max_generation_num`).

- If the number of audit trail files has not reached the maximum (value of `pd_aud_max_generation_num`), another audit trail file is created, and that file becomes the swap target. The smallest unused generation number is allocated as the file generation number.
- If the number of audit trail files has reached the maximum number (value of `pd_aud_max_generation_num`), the data load completed file (excluding files in shut down status) with the oldest update date becomes the swap target.

(3) HiRDB processing when there are no files available for the swap target

If there are no files available for the swap target, HiRDB processing depends on the value of the `pd_aud_no_standby_file_opr` operand.

If `forcewrite` (the default) is specified

If there are no swappable files available, a data load waiting audit trail file (excluding files on shutdown status) is forcibly made the swap target. The data

load waiting file with the oldest update date becomes the swap target.

If `down` is specified

If there is one or fewer swappable audit trail files available, HiRDB (or the unit for a HiRDB/Parallel Server) is terminated forcibly. For the procedure at this time, see *22.10(3) When HiRDB is terminated forcibly because there are no swappable target audit trail files*.

If the `pd_aud_file_wrn_pnt` operand is specified, a warning message can be output when the number of unswappable audit trail files reaches a warning level.

22.7 Recording data in the audit trail table

Executor: Auditor

Use the HiRDB supplied audit trail data recording UOC to record data in the audit trail table. This UOC operates by extending the database load utility (pdload command). It provides a facility for load data output to an audit trail file into the audit trail table. Only the auditor can perform data loading to the audit trail table.

Hint:

When the pdload command is executed, either the PDUSER environment variable must be used to specify the auditor's authorization identifier and password, or the auditor's authorization identifier must be specified in the pdload command's -u option and the password must be entered when the pdload command executes.

22.7.1 Example 1: Data loading from specified audit trail files

This example loads two data load waiting files (file names pdaudUNT1001.aud and pdaudUNT1002.aud) into the audit trail table (table name SQL_AUDIT_TRAIL). The following is the specification of the pdload command for this example:

Example pdload command specification

```
pdload -b -W SQL_AUDIT_TRAIL /pdload/control_file
```

Explanation

- The -b and -W options must be specified.
- Specify the audit trail table for the table name.
- The following explains the contents of the control information file:

Example of specifying the control information file

```
source bes1:(uoc)
srcuoc param='dir=/secarea,file=(pdaudUNT1001.aud,pdaudUNT1002.aud)'
```

Explanation

source bes1:(uoc): Specifies the server name (bes1) for the HiRDB/Parallel Server. You can specify a server name in the unit where the audit trail files are located. You may specify any server name in the unit, but if you specify a

back-end server, you can eliminate a communication bottleneck.

`dir`: Specifies the HiRDB file system area that stores the audit trail files to be data loaded.

`file`: Specifies the names of the audit trail files to be data loaded.

If indexes have been defined for the audit trail table, use the `idxwork` statement and the `sort` statement.

22.7.2 Example 2: Data loading from all audit trail files in the HiRDB file system area

This example loads all data load waiting audit trail files in the HiRDB file system area (/secarea) into the audit trail table (table name SQL_AUDIT_TRAIL). The following is the specification of the pdload command for this example:

Example pdload command specification

```
pdload -b -W SQL_AUDIT_TRAIL /pdload/control_file
```

Explanation

- The -b and -W options must be specified.
- Specify the audit trail table for the table name.
- The following explains the contents of the control information file:

Example of specifying the control information file

```
source bes1:(uoc)
srcuoc param='dir=/secarea,file=all,mode=normal'
```

Explanation

source bes1:(uoc) : Specifies the server name (bes1) for the HiRDB/Parallel Server. You can specify a server name in the unit where the audit trail files are located. You may specify any server name in the unit, but if you specify a back-end server, you can eliminate a communication bottleneck.

dir: Specifies the HiRDB file system area that stores the audit trail files to be data loaded.

file=all: Specifies that all audit trail files in the HiRDB file system area are to be data loaded. However, the current file and files in shutdown status are excluded.

mode=normal: Specifies that data load completed files are not to be data loaded.

Remarks

If mode=force is specified, data load completed files are also data loaded. You would specify force in the situation explained below. Note that when force is specified, the same data may be recorded twice, so care must be taken not to record data that has already been recorded.

- If the audit trail table's data was deleted inadvertently and there is no backup,

it would be appropriate to data load all audit trail files in order to recover the audit trail table

Table 22-14 shows whether data loading can be executed, depending on the audit trail file status and user specified values.

Table 22-14: Whether or not data is loaded depending on the audit trail file status and user specified values

Audit trail file status		User specified value		
Shutdown status	**Data load status**	**mode**	**file=*file-name***	**file=all**
Not shutdown	Data load waiting	normal	Executes data load processing from the audit file.	
	Data load completed		Skips data load processing from the audit file, and continues processing.	Does not perform data load processing from the audit file.
Shutdown	Data load waiting			
	Data load completed			
Not shutdown	Data load waiting	force	Executes data load processing from the audit file.	This is a user specification error.
	Data load completed			
Shutdown	Data load waiting		Skips data load processing from the audit file, and continues processing.	
	Data load completed			
—	Current	—	Skips data load processing from the audit file, and continues processing.	Does not perform data load processing from the audit file.

Legend:

—: Not applicable.

22.7.3 Procedure when an error occurs during data loading

This section explains the procedure when an error occurs during data loading.

(1) When executing in the log acquisition mode or the pre-update log acquisition mode

Procedure

1. Check the progress of data load processing. Because a commit occurs for each audit trail file during data loading, data loading is completed when the `KFPL00800-I` message is output for an audit trail file.

   Use the `pdls -d aud` command to check the audit trail file status. Files for which data loading has been completed enter data load completed status.

2. Check whether or not the KFPL23202-E message has been output. Data loading has been completed for an audit trail file for which the KFPL23202-E message is output, but the file's status failed to change. Therefore, such files remain in data load waiting status, which means that if the pdload command is re-executed the data may be recorded again. For this reason, you should use the pdaudrm command to delete such files before re-executing the pdload command.
3. Re-execute the pdload command. There is no need to change the option specifications or the control information file specification.

(2) When executing in the logless mode

Procedure

1. Use the pdrstr command and a backup made before data loading was executed to recover the RDAREA that stores the audit trail table.
2. Audit trail files for which data loading has been completed enter data load completed status. Therefore, even if the pdload command is re-executed, these files in data load completed status will not be data loaded. In this case, specify mode=force and re-execute the pdload command.
3. Re-execute the pdload command. If file=all is specified, specify the control information file to be data loaded in the file operand.

22.8 Audit trail table columns

Table 22-15 shows the columns of the audit trail table.

Table 22-15: Audit trail table columns

Column name	Column content	Column data type	NULL value specification
USER_NAME	Authorization identifier of the person who executed the audited event. For a command or utility executed by the HiRDB administrator or a general OS user, the OS user name is output.	MVARCHAR(30)	No
EXEC_DATE	Date the event was executed. Event execution date means the date according to the standard time at which the server that issued the audit trail output request requests output of an audit trail to the audit trail buffer.	DATE	No
EXEC_TIME	Time the event was executed. Event execution time means the time according to the standard time at which the server that issued the audit trail output request requests output of an audit trail to the audit trail buffer.	TIME	No
EXEC_TIME_MICRO	Time the event was executed (microseconds). Event execution time means the time according to the standard time at which the server that issued the audit trail output request requests output of an audit trail to the audit trail buffer.	INTEGER	No
EVENT_TYPE	Event type. For details about the event types and subtypes, see *(1) Event types and subtypes*.	CHAR(3)	No
EVENT_SUBTYPE	Event subtype. For details about the event types and subtypes, see *(1) Event types and subtypes*.	CHAR(3)	No
EVENT_RESULT	Result of event execution; this item flags whether or not privilege checking was successful: S: Privilege checking or event succeeded. F: Privilege checking or event failed. U: Part of the event failed.	CHAR(1)	No

Column name	Column content	Column data type	NULL value specification
USED_PRIVILEGE	Privilege used when the event executed. When the event is terminated, three consecutive blank symbols (Δ) are output. AUD: Audit privilege CNT: CONNECT privilege DBA: DBA privilege DEL: DELETE privilege INS: INSERT privilege OWN: Owner RDA: RDAREA usage privilege SCH: Schema definition privilege SEL: SELECT privilege SYS: HiRDB administrator UPD: UPDATE privilege	CHAR(3)	No
UAP_NAME	Name of the UAP that issued the event.[1] If the UAP name is fewer than 30 characters, it is padded with trailing blank symbols (Δ). In all other cases, the NULL value is output.	VARCHAR(30)	Yes
SERVICE_NAME	Name of the service requested by the UAP that issued the event. If an OpenTP1 UAP, this is the name of the service requested by SPP or MHP. If not an OpenTP1 UAP, 31 consecutive asterisks (*) are output. In all other cases, the NULL value is output.	VARCHAR(31)	Yes
IP_ADDRESS	IP address that issued the event. The NULL value is output when the IP address cannot be identified.	VARCHAR(63)	Yes
PROCESS_ID	Process ID that issued the event. The NULL value is output when the process ID cannot be identified. For a command or utility, the process ID of the executed command is output. At the start of audit trail overwriting by the system, the process ID of the audit trail daemon is output. In all other cases, the process ID of the single server or front-end server that connected is output.	INTEGER	Yes

Column name	Column content		Column data type	NULL value specification
THREAD_ID	Thread ID that issued the event. The NULL value is output when the thread ID cannot be identified.		INTEGER	Yes
HOST_NAME	Name of the host that received the event. The host name specified in the -x option of the pdunit operand is output. If a system switchover facility is used, the host name of the primary system is output. The NULL value is output when the name of the host that received the event cannot be identified.		VARCHAR(32)	Yes
UNIT_NAME	Unit identifier that received the event. The unit identifier specified in the -u option of the pdunit operand is output. The NULL value is output when the unit identifier that received the event cannot be identified.		CHAR(4)	Yes
SERVER_NAME	Name of the server that received the event. For a HiRDB/Single Server, this is the name of the single server; for a HiRDB/Parallel Server, this is the name of the front-end server. This is output when the server name can be identified. The server name specified in the -s option of the pdstart operand is output. The NULL value is output when the name of the server that received the event cannot be identified.		VARCHAR(8)	Yes
CONNECT_NUMBER	Connection number of the event issuer. The NULL value is output when the connection number cannot be identified.		INTEGER	Yes
SQL_NUMBER	Event SQL number. The NULL value is output when the event SQL number cannot be identified.		INTEGER	Yes
OBJECT_SCHEMA	Object information detail[2]	Owner of the object for which the event privilege is being checked. The NULL value is output when the object owner cannot be identified.	MVARCHAR(30)	Yes

Column name	Column content		Column data type	NULL value specification
OBJECT_NAME		Name of the object for which the event privilege is being checked. The NULL value is output when the object name cannot be identified.	MVARCHAR(30)	Yes
OBJECT_TYPE		Type of object for which the event privilege is being checked. The NULL value is output when the object type cannot be identified. ALS: Alias AUF: Audit trail file FID: Foreign index FNC: Function FSV: Foreign server FTB: Foreign table IDX: Index LST: List PRC: Procedure RDA: RDAREA SCH: Schema TBL: Table TRG: Trigger TYP: Data type USM: User Mapping VIW: View table	CHAR(3)	Yes
PRIVILEGE_TYPE	Privilege added, deleted, or modified by the event. The NULL value is output when the privilege cannot be identified. AUD: Audit privilege CNT: CONNECT privilege DBA: DBA privilege DEL: DELETE privilege INS: INSERT privilege RDA: RDAREA usage privilege SCH: Schema definition privilege SEL: SELECT privilege UPD: UPDATE privilege		CHAR(3)	Yes

Column name	Column content	Column data type	NULL value specification
PRIVILEGE_SCHEMA	User authorization identifier added, deleted, or modified by the event. Or, the authorization identifier of the user who was the target of the event. The NULL value is output when the authorization identifier cannot be identified. • GRANT or REVOKE: Authorization identifier, group ID, or role name to which a privilege was added, from which a privilege was deleted, or whose privilege was modified. • Transition to or release from consecutive certification failure account lock state: Authorization identifier that was locked or released. • Transition to or release from password-invalid account lock state: Authorization identifier that was locked or released.	MVARCHAR(30)	Yes
SECURITY_OPERAND	Values of the security audit facility operands. These are the operand values at HiRDB startup. For all cases other than startup, the NULL value is output. For details about the operand values for the security audit facility, see *(2) Operand values for the security audit facility*.	VARCHAR(256)	Yes
AUDIT_TRAIL_TYPE	Audit trail type. A value is output to differentiate between an audit trail showing an event result and an audit trail showing the result of privilege checking during event execution. Privilege checking: NULL Event termination: E	CHAR(1)	Yes
SQL_CODE	SQL code or termination code. When an audited event is terminated, an SQL code or termination code is output for the SQL event or utility (or command). For privilege checking, the NULL value is output. This code indicates the event success or failure value that is set when the event terminates. For details about each code in the audit record entry and whether or not the event is successful, see *(3) SQL code or termination code indicating event success or failure*.	INTEGER	Yes

Column name	Column content	Column data type	NULL value specification
FROM_AUDFILE_NAME	Audit trail file name at the swapping source. The NULL value is output when the audit trail cannot be identified.	MVARCHAR(30)	Yes
TO_AUDFILE_NAME	Audit trail file name at the swapping destination. The NULL value is output when the audit trail cannot be identified.	MVARCHAR(30)	Yes
SECURITY_PARM_TYPE	Modification type related to the connection security facility.[3]	CHAR(4)	Yes
BEFORE_SECURITY_PARM	Setting value of the connection security facility before change.[3] If the setting value is fewer than 10 characters, it is padded for output with trailing blanks (setting value + blanks = 10 characters).	CHAR(10)	Yes
AFTER_SECURITY_PARM	Setting value of the connection security facility after change.[3] If the setting value is fewer than 10 characters, it is padded for output with trailing blanks (setting value + blanks = 10 characters).	CHAR(10)	Yes
AUDIT_TABLE_OPTION	Audit trail table option. This item is not output for privilege checking. It is output in the following cases: • The event-target object is an audit trail table. • The event-target object is a view table created using an audit trail table as the base table. • The event-target object is a list created using an audit trail table as the base table.	CHAR(1)	Yes

Column name	Column content	Column data type	NULL value specification
	Y: Manipulation-target object is an audit trail table. V: Manipulation-target object is a view table created using an audit trail table as the base table. L: Manipulation-target object is a list created using an audit trail table as the base table. NULL value: For cases other than those listed above or for privilege checking For details about whether or not audit trail table options are output, see *(5) Whether or not audit trail table options are output*.		
ACCESS_COUNT	Number of rows searched, inserted, updated, and deleted by the user for an object (base table, view table, foreign table, table alias, or list). The access count that can be acquired is from 0 to 2147483647. If acquisition of the access count fails, the NULL value is output. For details about the access count, see *(6) Details about the access count*.	INTEGER	Yes

Notes

- For a column for which the NULL value is specified, NULL is set for the NOT NULL constraint. For a column for which the NULL value is not specified, NOT NULL is set for the NOT NULL constraint.
- When version upgrading changes the column structure of an audit trail table and data prior to version upgrading is registered in the audit trail table after version upgrading, the NULL value is set in the newly added columns of the existing rows.

[1] To use a UAP name as a search key, specify CAST to convert the attribute to CHAR(30) and specify the result in the conditional expression.

Example:

```
SELECT * FROM SQL_AUDIT_TRAIL
    WHERE UAP_NAME=CAST('UAP11' AS CHAR(30));
```

[2] During privilege checking:

The object that is the target of privilege checking is output. However, when the privilege of the executor is checked, nothing is output because there is no object that is checked.

Examples:

- Target object when the access privilege to table `T1` is checked: `Table T1`
- Target object when whether or not the executor has the DBA privilege is checked: NULL value

At event termination:

The object that is the target of the event is output. If no object is specified as the event target, `table` is output as the object type.

Example:

- Target object when a table is searched: The searched table

[3] A modification type is output in the following cases:

- When a setting value of the connection security facility is changed
- When a password is registered or modified

For details about the information that is recorded, see *(4) Information that is recorded when the connection security facility is used*.

(1) Event types and subtypes

Table 22-16 provides details about the event types and subtypes.

Table 22-16: Event types and subtypes

Audit event	Event type value	Event subtype value	Event
System administrator security events	`SYS`	`STR`	HiRDB startup
		`STP`	HiRDB termination
		`MOD`	Auditor registration or audit trail table creation (`pdmod` command)
		`ARM`	Audit trail file deletion (`pdaudrm` command)
		`ABG`	Audit trail collection startup (`pdaudbegin` command, HiRDB startup)
		`AEN`	Audit trail collection termination (`pdaudend` command, HiRDB termination)
		`OVW`	Start of audit trail file overwriting
		`CLK`	Transition to consecutive certification failure account lock state
		`CUL`	Release of consecutive certification failure account lock state

Audit event	Event type value	Event subtype value	Event
		PLK	Transition to password-invalid account lock state
		PUL	Release of password-invalid account lock state
		SPR	Modification of an operand related to the connection security facility
		ULK	Execution of the pdacnlck command
Auditor security events	AUD	ALD	Data load the audit trail table (pdload command)
		ASW	Audit trail file swap (pdaudswap command)
		CRT	Audit event definition (CREATE AUDIT)
		DRP	Audit event deletion (DROP AUDIT)
		GRT	Auditor password change (GRANT AUDIT)
Session security events	SES	CNT	Connection to HiRDB (CONNECT statement)
		ATH	User modification (SET SESSION AUTHORIZATION statement)
Privilege management events	PRV	GRT	User privilege addition (GRANT statement)
			Password change (adds information indicating a password change to the set modification type for the connection security facility)
		RVK	User privilege deletion (REVOKE statement)
Object definition events	DEF	CRT	Object definition (CREATE type SQL)
		DRP	Object deletion (DROP type SQL)
		ALT	Object modification (ALTER type SQL)

Audit event	Event type value	Event subtype value	Event
Object manipulation events	ACS	SEL	Table retrieval (SELECT statement)
		INS	Table row insertion (INSERT statement)
		UPD	Table row modification (UPDATE statement)
		DEL	Deletion of a row from a table (DELETE statement)
		PRG	Deletion of all rows from a table (PURGE TABLE statement)
		CAL	Execution of a stored procedure (CALL statement)
		LCK	Table lock control (LOCK TABLE statement)
		ASN	List creation (ASSIGN LIST statement)
Utility manipulation event	UTL	LOD	Database load utility (pdload command)
		ORG	Database reorganization utility (pdrorg command)
		EXP	Dictionary import/export utility (pdexp command)
			pddefrev command
		CST	Integrity check utility (pdconstck command)

(2) Operand values for the security audit facility

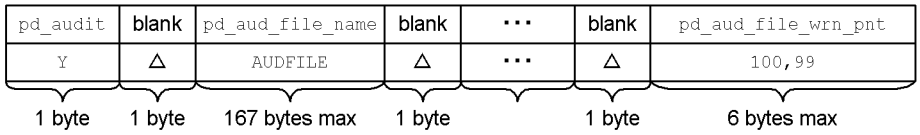
Table 22-17 lists the operand values related to the security audit facility. Figure 22-7 shows the data format used when operand values are recorded.

Table 22-17: Values of security audit facility operands

Operand	Data type	Stored value
pd_audit	CHAR(1)	Y or N
pd_aud_file_name	VARCHAR(167)	Name of HiRDB file system area used for audit trail files
pd_aud_max_generation_size	VARCHAR(4)	1-5240
pd_aud_max_generation_num	VARCHAR(3)	2-200
pd_aud_no_standby_file_opr	VARCHAR(10)	down or forcewrite
pd_aud_async_buff_size	VARCHAR(7)	0, 4096-6553600
pd_aud_async_buff_count	VARCHAR(4)	1-6500

Operand	Data type	Stored value
pd_aud_file_wrn_pnt	VARCHAR(6)	0-100 and 0-99

Figure 22-7: Data format for output of security audit facility operand values



(3) SQL code or termination code indicating event success or failure

Table 22-18 provides details about the SQL codes or termination codes that indicate event success or failure.

Table 22-18: SQL code or termination code indicating event success or failure

Event	SQL code or termination code value	Meaning	Event success or failure	Remarks
Various SQLs	0 or greater	Success	S	—
	Negative	Failure	F	
pdstart	0	Success	S	—
	4	Success	S	
	8	Failure	F	An audit trail cannot be collected in the following cases: • Failure occurred before the audit trail daemon started. • Failure of the pdstart command caused the unit to terminate abnormally.
pdstop	0	Success	S	If a failure occurred after the audit trail daemon had terminated, a successful audit trail is output.
	4	Success	S	—
	8	Failure	F	
pdmod (auditor registration, audit trail creation)	0	Success	S	—
	8	Failure	F	

Event	SQL code or termination code value	Meaning	Event success or failure	Remarks
pdaudrm	0	Success	S	—
	8	Failure	F	
pdaudbegin	0	Success	S	—
	4	Partial failure	U	
	8	Failure	F	An audit trail cannot be collected in the following cases: • All generations of audit trail files have been used. • The pdaudbegin command failed because there is no generation of audit trail files that is in unload wait status.
pdaudend	0	Success	S	—
	8	Failure	F	
pdacunlck	0	Success	S	—
	8	Failure	F	
pdaudswap	0	Success	S	During overwriting of the audit trail file, the names of the audit trail files at the swapping source and swapping destination are output.
	8	Failure	F	During overwriting of the audit trail file, the name of the audit trail file at the swapping source is output. The NULL value is output for the audit trail file name at the swapping destination.
pdload	0	Success	S	—
	4	Success	S	
	8	Failure	F	
pddefrev	0	Success	S	—
	4	Success	S	
	8	Failure	F	
	12	Failure	F	

Event	SQL code or termination code value	Meaning	Event success or failure	Remarks
pdrorg	0	Success	S	—
	4	Success	S	
	8	Failure	F	
pdexp	0	Success	S	—
	4	Success	S	
	8	Failure	F	
	12	Failure	F	
pdconstck	0	Success	S	—
	4	Success	S	End of warning (there is an integrity violation)
	8	Failure	F	—

Legend:

—: Not applicable

(4) Information that is recorded when the connection security facility is used

Table 22-19 provides the information that is recorded when the connection security facility is used.

Table 22-19: Information that is output when the connection security facility is used

Facility	Event type	Event subtype	Modification type	Value that is output after modification
Permitted number of consecutive certification failures	SYS	SPR	ECNT	1-10 or NULL value
Account lock period			LCKM	10-44640, UNLIMITED, or NULL value
Minimum number of bytes for a password			MINL	1-15 or NULL value
Authorization identifier specification prohibition			USID	RESTRICT, UNRESTRICT, or NULL value
Single character-type specification prohibition			SMLN	
Password change*	PRV or AUD	GRT	CPWD	NULL value

Note

The NULL value is output in the following cases:

- For a value that is to be output before or after a change, no value is found in the corresponding column in the SQL_SYSPARAMS dictionary table (for example, when a new setting is specified).
- For a NULL value
- When a value cannot be identified.

During an advance check of password character string restrictions, an audit trail is collected for each modification type. However, the NULL value is output for the value that is to be output before or after the change.

*

HiRDB cannot determine from the event type, event subtype, and SQL code only whether the password was actually changed during execution of a GRANT statement. Only when a registered user password is changed during execution of a GRANT statement is CPWD set as the modification type. Table 22-20 shows the modification types that are output when a password is changed.

Table 22-20: Modification types that are output when a password is changed

Does a password exist for the registered user?	GRANT CONNECT, DBA, or AUDIT execution result	Modification type
Password exists.	Password change	CPWD
	Password deletion	CPWD
	No password change	NULL value
Password does not exist.	Changes to "Password exists."	CPWD
	Retains "Password does not exist."	NULL value

Note

Users and passwords referred to here are those managed by HiRDB.

(5) Whether or not audit trail table options are output

Table 22-21 shows whether or not audit trail table options are output.

Table 22-21: Audit trail table option output

Execution result	Failure cause	Is audit trail table option output?
Success	—	Output
Failure*	The target object does not exist.	Not output (NULL value)
	An error occurred because the target object is an audit trail table.	Output
	Error other than one listed above	Output (not output when information cannot be collected)

Legend:

—: Not applicable

* To determine whether a table is an audit trail table when no audit trail table option is output, check the combination of the object owner and object name in the audit record entry, for example.

(6) Details about the access count

Table 22-22 provides the details of the access count.

Table 22-22: Details about the access count

Access method		Access count
SQL	SELECT	Number of rows accessed (fetched until CLOSE was successful or failed) by the user on the result set using the SELECT statement. For block transfer, the number of rows in a block is treated as the access count. If multiple tables are specified for a query, the same access count is used for all the applicable tables. The access count is set to the NULL value in either of the following cases: • Table is specified in a subquery (for details, see Table 22-23) • Table is specified in the right-hand operand in the EXCEPT [ALL] set operation (for an example, see Figure 22-13)
	INSERT	Number of rows inserted
	UPDATE	Number of rows updated
	DELETE	Number of rows deleted
Utility	pdload	Data load count per table
	pdrorg	Unload count per table and reload count per table
	pdconstck	Total number of key values resulting in limitation violation

Table 22-23: Access count by subquery

Specified location in table		Access count	Remarks
Scalar subquery	Where a value expression is permitted	NULL value	For an example, see Figure 22-14
Row subquery	Where a row value constructor is permitted	NULL value	For an example, see Figure 22-15
	SET clause in UPDATE statement	NULL value	For an example, see Figure 22-16
Table subquery	Right-hand term of IN predicate	NULL value	For an example, see Figure 22-17
	Right-hand term of qualified predicate	NULL value	For an example, see Figure 22-18
	EXISTS predicate	NULL value	For an example, see Figure 22-19
	Derived table in FROM clause	Non-NULL value*	For an example, see Figure 22-20

*

If the result of searching the derived table in the FROM clause is not returned directly to the user, the access count is the NULL value (for an example, see Figure 22-21).

Figures 22-8 through 22-21 show output examples of the access count when the access method is SELECT.

Figure 22-8: Output example of access count (part 1)

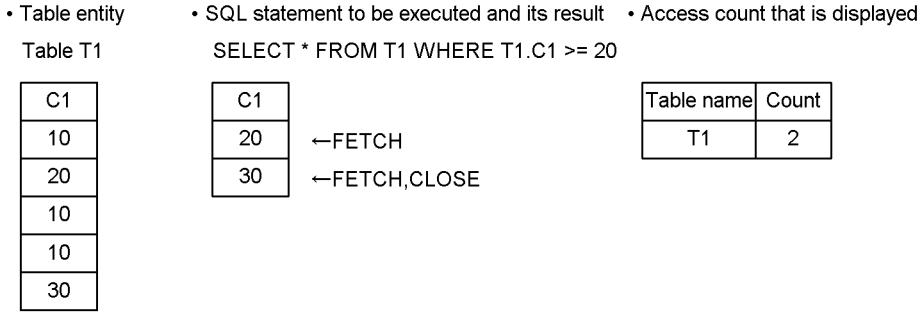


Figure 22-9: Output example of access count (part 2)

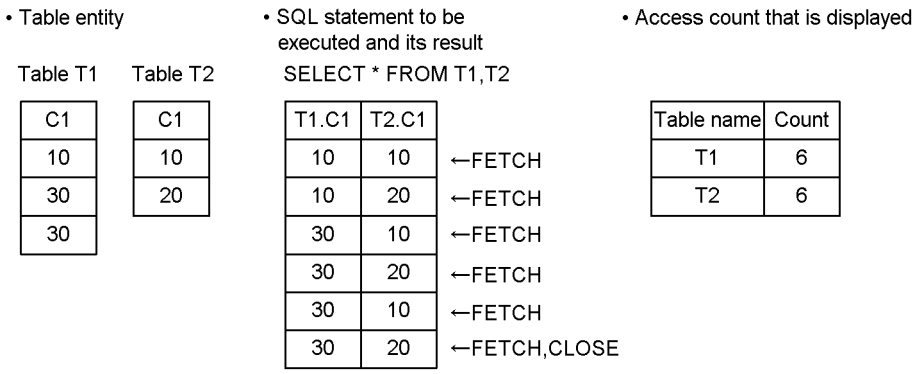
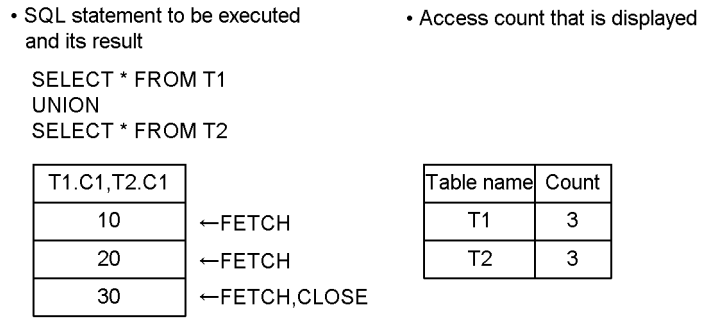
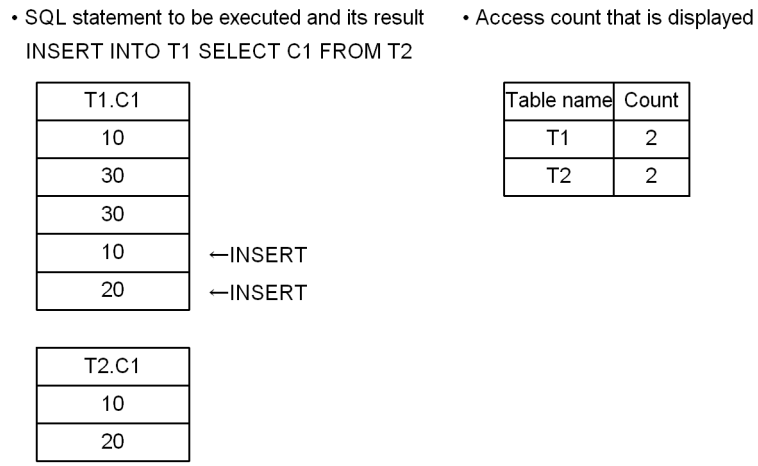


Figure 22-10: Output example of access count (part 3)



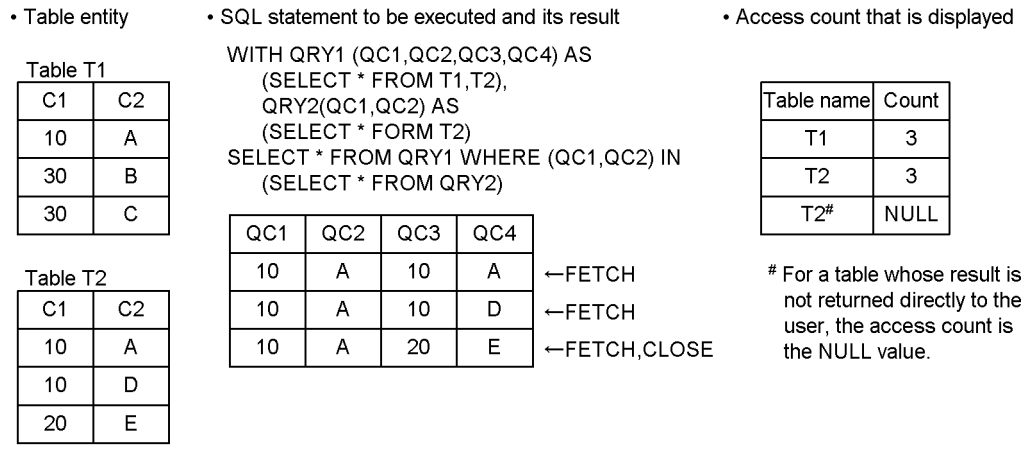
Note: The actual table is the same as in Figure 22-9.

Figure 22-11: Output example of access count (part 4)



Note: The actual table is the same as in Figure 22-9.

Figure 22-12: Output example of access count (part 5)



• Table entity

Table T1

C1	C2
10	A
30	B
30	C

Table T2

C1	C2
10	A
10	D
20	E

• SQL statement to be executed and its result

```
WITH QRY1 (QC1,QC2,QC3,QC4) AS
    (SELECT * FROM T1,T2),
    QRY2(QC1,QC2) AS
    (SELECT * FORM T2)
SELECT * FROM QRY1 WHERE (QC1,QC2) IN
    (SELECT * FROM QRY2)
```

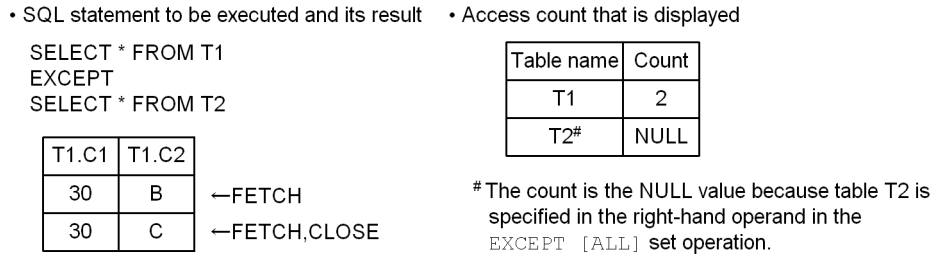
QC1	QC2	QC3	QC4	
10	A	10	A	←FETCH
10	A	10	D	←FETCH
10	A	20	E	←FETCH,CLOSE

• Access count that is displayed

Table name	Count
T1	3
T2	3
T2#	NULL

For a table whose result is not returned directly to the user, the access count is the NULL value.

Figure 22-13: Output example of access count (part 6)



• SQL statement to be executed and its result

```
SELECT * FROM T1
EXCEPT
SELECT * FROM T2
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

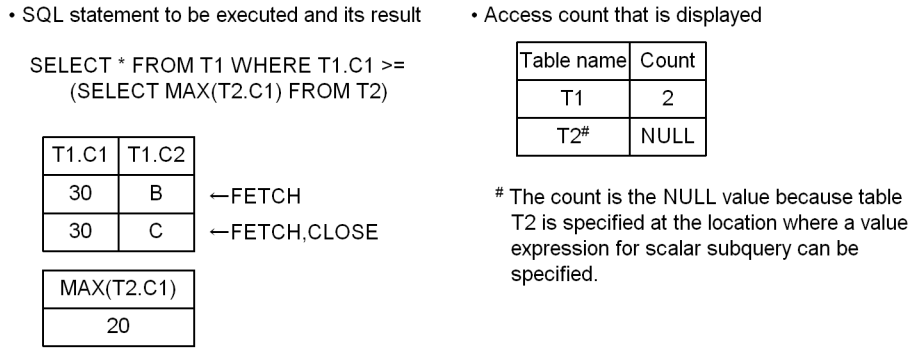
• Access count that is displayed

Table name	Count
T1	2
T2#	NULL

The count is the NULL value because table T2 is specified in the right-hand operand in the `EXCEPT [ALL]` set operation.

Note: The actual table is the same as in Figure 22-12.

Figure 22-14: Output example of access count (part 7)



• SQL statement to be executed and its result

```
SELECT * FROM T1 WHERE T1.C1 >=
    (SELECT MAX(T2.C1) FROM T2)
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

MAX(T2.C1)
20

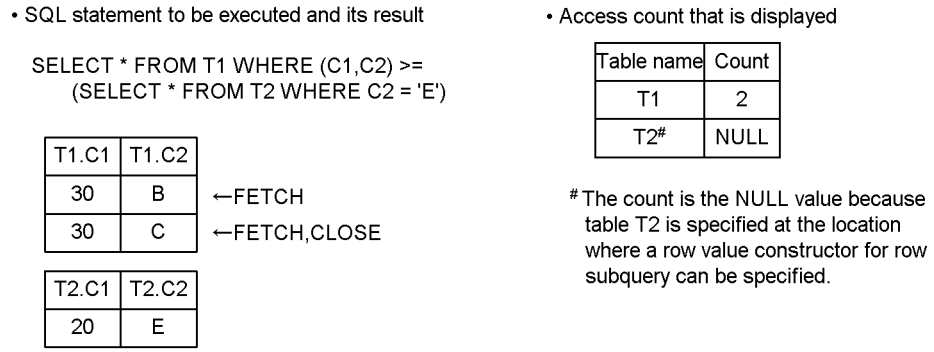
• Access count that is displayed

Table name	Count
T1	2
T2#	NULL

The count is the NULL value because table T2 is specified at the location where a value expression for scalar subquery can be specified.

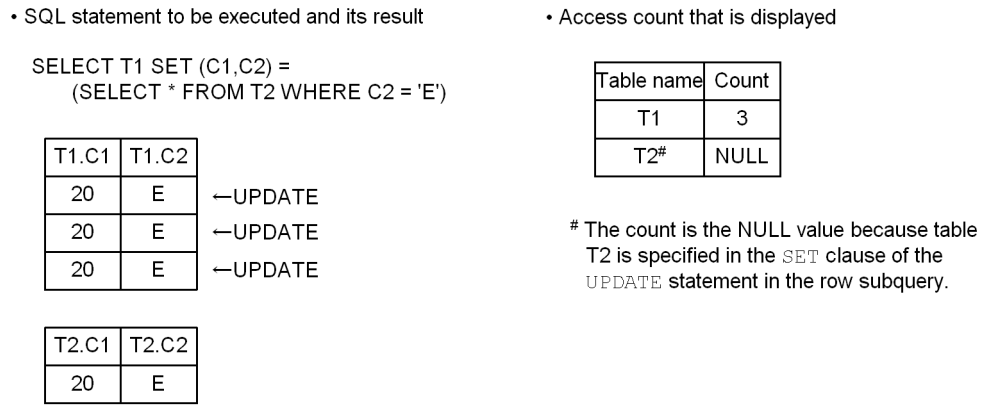
Note: The actual table is the same as in Figure 22-12.

Figure 22-15: Output example of access count (part 8)



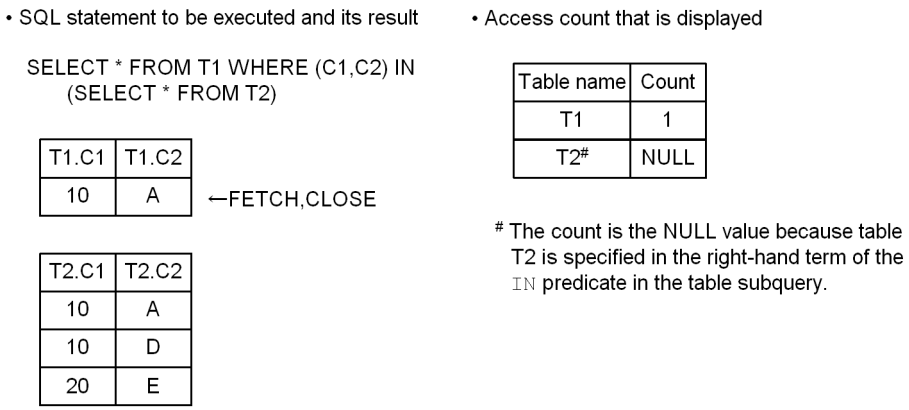
Note: The actual table is the same as in Figure 22-12.

Figure 22-16: Output example of access count (part 9)



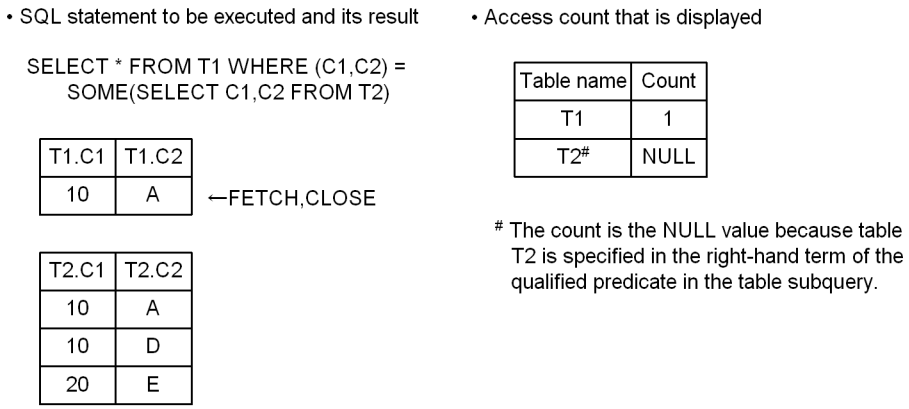
Note: The actual table is the same as in Figure 22-12.

Figure 22-17: Output example of access count (part 10)



Note: The actual table is the same as in Figure 22-12.

Figure 22-18: Output example of access count (part 11)



Note: The actual table is the same as in Figure 22-12.

Figure 22-19: Output example of access count (part 12)

• SQL statement to be executed and its result

```
SELECT * FROM T1 WHERE NOT EXISTS
    (SELECT * FROM T2 WHERE T1,C1 = T2.C1)
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

T2.C1	T2.C2
10	A

• Access count that is displayed

Table name	Count
T1	2
T2#	NULL

The count is the NULL value because table T2 is specified in the EXISTS predicate in the table subquery.

Note: The actual table is the same as in Figure 22-12.

Figure 22-20: Output example of access count (part 13)

• SQL statement to be executed and its result

```
SELECT * FROM (SELECT * FROM T1) AS D1
```

T1.C1	T1.C2	
10	A	←FETCH
30	B	←FETCH
30	C	←FETCH,CLOSE

• Access count that is displayed

Table name	Count
T1#	3

The count is not the NULL value because table T2 is specified as the derived table in the FROM clause of the table subquery, but the result of searching the derived table in the FROM clause is not returned directly to the user.

Note: The actual table is the same as in Figure 22-12.

Figure 22-21: Output example of access count (part 14)

• SQL statement to be executed and its result

```
SELECT * FROM T1 WHERE (C1,C2) IN
 (SELECT * FROM (SELECT * FROM T2) AS D1)
```

T1.C1	T1.C2	
10	A	←FETCH,CLOSE

T2.C1	T2.C2
10	A
10	D
20	E

• Access count that is displayed

Table name	Count
T1	1
T2#	NULL

The count is the NULL value because table T2 is specified as the derived table in the FROM clause of the table subquery.

Note: The actual table is the same as in Figure 22-12.

22.9 Narrowing the audit trails

By narrowing the audit trails, you can acquire only specific audit trails.

You can narrow audit trails by defining a condition in the `CREATE AUDIT` definition SQL statement, and then use `DROP AUDIT` as needed to drop audit trails.

(1) Selection items

Table 22-24 lists and describes the selection items that can be specified as audit trail narrowing conditions.

Table 22-24: Selection items that can be specified as audit trail narrowing conditions

Selection item	Specification	Description	Record items in the corresponding audit trail
Operation type	Required	Narrows the audit trails to be acquired to audit trails of a specified operation type.	• Event type • Event subtype
Trail type	Required	Narrows the audit trails to be acquired to audit trails of a specified trail type.	• Audit trail type
Event success/ failure	Required	Narrows the audit trails to be acquired on the basis of event success or failure.	• Event success or failure
Object*	Optional	Acquires audit trails by narrowing the objects that became the target of a specific event.	• Object owner • Object name • Object type

*

The following rules apply to object specification:

- Definition of an audit event can be specified even for a nonexistent object name (so that the trails can be narrowed down when an object is created by `CREATE`).
- The definition of an audit event cannot be changed even if the object specified in the audit event definition is deleted by another SQL statement, or if one of its attributes (such as its name) is changed. The following are examples:

  Example 1: An audit event definition is specified for table `T1`, but the table is renamed from `T1` to `T2`. The specification of `T1` remains unchanged in the audit event definition.

  Example 2: An audit event definition is specified for table `T1`, but table `T1`

is dropped by DROP TABLE. The audit event definition remains as is.

- In the case of privilege checking trails, trails are acquired only when the target object for privilege checking matches the target object for the event. If you acquire audit trails by narrowing down by object, Hitachi recommends that you acquire audit trails at the time of event termination. Table 22-25 shows whether or not there is output from privilege checking when trails are narrowed down by object.

Table 22-25: Whether or not there is output from privilege checking when trails are narrowed by object

Privilege used	Whether or not there are objects	Objects that can be narrowed
DBA	No	None
SCH	No	None
CNT	No	None
RDA	Yes (RDA)	Objects listed at left
SEL	Yes (FTB, LST, TBL, VIW)	Objects listed at left
INS	Yes (FTB, TBL, VIW)	Objects listed at left
DEL	Yes (FTB, TBL, VIW)	Objects listed at left
UPD	Yes (FTB, TBL, VIW)	Objects listed at left
AUD	No	None*
SYS	Yes (AUF, TBL)	--
OWN	Yes (FID, FNC, FSV, FTB, IDX, PRC, SCH, TBL, TRG, TYP, VIW)	Objects listed at left

Legend:

--: If the audit facility is executing, audit trails are always output regardless of the audit event definition.

Note:

For details about the privileges that are used and the symbols for indicating whether or not there is an object, see Table *22-15 Audit trail table column*.

*

In the case of an auditor's security event, if the security audit facility is executing, audit trails are always output regardless of the audit event definition.

If you have specified a data dictionary table for the target object, specify the object type, authorization identifier, and table identifier as described in Table 22-26.

Table 22-26: Object type, authorization identifier, and table identifier when a data dictionary table is specified

Operation type	Object type	Authorization identifier	Table identifier
Object operation event	VIEW	MASTER	Table identifiers of the data dictionary tables, excluding the data dictionary tables used by the system
Utility operation event	TABLE	Omitted*	Table identifiers of all data dictionary tables

*

Even when the authorization identifier is omitted, '(Data dictionary)' is stored in the object owner column in the data dictionary table SQL_AUDITS.

(2) Available selections

The following selections are available:

- For each CREATE AUDIT, you can combine desired selection items (operation type, trail type, event success/failure, and object) to create a single audit event definition. The AND condition applies to these selection items. Note that specification of an object is optional.
- For each CREATE AUDIT, you can specify only one of operation type, trail type, event success/failure, or object. To specify multiple values for each of these items, execute as many CREATE AUDIT statements as there are values. If multiple audit event definitions are created, the OR condition applies to the audit event definitions, in which case an audit trail that satisfies any of the audit event definitions is acquired.

To acquire audit trails when the target of an object operation event is the table "USER1"."T1" and the target of the audit is the termination trail of an object definition event, define as follows:

```
CREATE AUDIT AUDITTYPE EVENT FOR ACCESS ON TABLE "USER1"."T1"
CREATE AUDIT AUDITTYPE EVENT FOR DEFINITION
```

The following describes the trails that are acquired and the trails that are not when the above audit event is defined:

Trails that are acquired:

- Event termination trail when the table "USER1"."T1" is searched

- Event termination trail when the CREATE statement is executed

Trails that are not acquired:

- Trail when a table other than "USER1"."T1" is searched
- Trail of a connection event
- Trail of a utility operation event

(3) Combinations of selection items

Some combinations of selection items serve no purpose. For such a combination of selection items, the executed CREATE AUDIT results in an error.

An example is when the object table "USER1"."T1" is specified in CONNECT for a session security event.

(4) Security audit information buffer

To use the security audit facility, a security audit information buffer is required. Therefore, you must estimate the shared memory requirement for the security audit information buffer.

You can do this by estimating manually the specification value and then specifying the value in the pd_audit_def_buffer_size operand, or by having the system determine the value automatically (omit the pd_audit_def_buffer_size operand). When the system determines the value, a margin is added to the memory size to ensure sufficient space. The memory size required is determined by the number of entries of the object that is the target of the narrowed audit. Thus, to determine the size of memory to allocate, the system both adds the value 100 to the number of entries for the object that has been defined, and also multiplies this number of entries by 1.2. It then applies the higher of the two results as the size of memory to allocate. The following shows the margin value:

Number of entries for an object that has already been defined as the target of a narrowed audit	Condition	Margin value
0	None	Value for 100 entries for the object that is the target of the narrowed audit
1 or greater	$N + 100 > \times 1.2$	Value for 100 entries for the object that is the target of the narrowed audit
	$N + 100 \leq N \times 1.2$	Value for $N \times 0.2$ entries

Legend:

N: Number of entries for the object that has already been defined as the target of the narrowed audit

If the required amount of memory cannot be allocated when the security audit information buffer is created, the actions described in Tables 22-27 and 22-28 must be taken.

Note:

If the security audit information buffer cannot store definition information for all audit events, processing may slow down because accesses must be made to the data dictionary table.

Table 22-27: HiRDB operation and actions to be taken when security audit information buffer is created (during HiRDB startup)

pd_audit_def_buffer_size operand specification	Allocation of shared memory	HiRDB operation	Action
Specified	Failure	Does not start. In this case, HiRDB displays the KFPD00031-E message.	Take one of the following actions: • Increase the OS's shared memory size. • Provide free space in the OS's shared memory. • Decrease the pd_audit_def_buffer_size operand value.
	Success	Starts. If the definition information for all the audit events is not stored in the security audit information buffer, HiRDB displays the KFPD00032-W message.	Because performance may decline, re-estimate the pd_audit_def_buffer_size operand value.
Omitted	Failure	Starts, but does not create the security audit information buffer. In this case, HiRDB displays the KFPD00032-W message.	Because performance may decline, take one of the following actions: • Increase the OS's shared memory size. • Provide free space in the OS's shared memory. If neither of the above actions can be taken, specify a value in the pd_audit_def_buffer_size operand that is less than the value determined automatically by the system.
	Success	Starts.	No action is required.

Table 22-28: HiRDB operation and actions to be taken when security audit information buffer is created (during HiRDB operation)

pd_audit_def_buffer_size operand specification	Overflow of definition information for audit event in security audit information buffer	HiRDB operation	Action
Specified	Yes	Stores as much definition information for audit events as possible in the security audit information buffer and then resumes processing. In this case HiRDB displays the KFPD00032-W message.	Re-estimate the value of the pd_audit_def_buffer_size operand according to the displayed KFPD00032-W message. If no action is taken, performance may decline.
	No	Stores the definition information for all the audit events in the security audit information buffer and then resumes processing.	No action is required.
Omitted	Yes	Stores as much definition information for audit events as possible in the security audit information buffer and then resumes processing. In this case HiRDB displays the KFPD00032-W message.	Restart HiRDB. The system re-calculates the size and creates a security audit information buffer. If the KFPD00032-W message is displayed when HiRDB restarts, take one of the following actions: • Increase the OS's shared memory size. • Provide free space in the OS's shared memory. If neither of the above actions can be taken, specify a value in the pd_audit_def_buffer_size operand that is less than the value determined automatically by the system.
	No	Stores the definition information for all the audit events in the security audit information buffer and then resumes processing.	No action is required.

When the pd_audit_def_buffer_size operand is omitted, the specification value is determined automatically by the system. If definitions of audit events increase during HiRDB operation, the size of the security audit information buffer increases the next time HiRDB starts. This means that the size of the security audit information buffer may increase each time HiRDB is started.

(5) HiRDB operation in the event of an error in the security audit information buffer

This subsection describes HiRDB's operation in the event of an error in the security audit information buffer.

(a) At HiRDB startup

The security audit information buffer is created when HiRDB starts. If an error occurs, the HiRDB operation depends on whether the size of the security audit information buffer is being determined automatically by the system or manually by the user.

- When the system determines the size automatically

  HiRDB displays a message and starts operation with the size of the security audit information buffer set to 0 (security audit information buffer is not created). However, in the case of a dictionary access error (rollback is required), the HiRDB operation is as follows:

  For a HiRDB/Single Server:

  In the case of an error, such as a disk failure, HiRDB displays a message and immediately terminates abnormally (unit shutdown), because it would most likely be impossible to continue operation when the definition information for audit events cannot be acquired after starting operation.

  For a HiRDB/Parallel Server:

  HiRDB retries creation of the security audit information buffer a specified number of times, because the dictionary server may have not started. If the error remains after the specified number of retries, HiRDB places the front-end server in SUSPEND status.

- When the user determines the size manually

  HiRDB displays a message and then resumes processing. However, if the specified shared memory size cannot be allocated for the security audit information buffer, HiRDB displays a message and immediately terminates abnormally (unit shutdown).

Table 22-29 describes the causes of errors during HiRDB startup and the HiRDB operations.

Table 22-29: Causes of errors during HiRDB startup and HiRDB operations

Cause of error		HiRDB operation	
		pd_audit_def_buffer_size operand is omitted	pd_audit_def_buffer_size operand is specified
Area allocation error	Shared memory for buffer	Starts with size 0	Cannot start
	Process private memory for dictionary search	Starts with size 0	Allocates shared memory and then resumes processing
Communication error		Starts with size 0	Allocates shared memory and then resumes processing
Dictionary access error	Rollback is not required	Starts with size 0	Allocates shared memory and then resumes processing
	Rollback is required	Cannot start*	Allocates shared memory and then resumes processing

*

For a HiRDB/Parallel Server, if the error cannot be corrected after the specified number of restart attempts, HiRDB places the front-end server in SUSPEND status.

For a HiRDB/Single Server, HiRDB immediately terminates abnormally.

(b) During HiRDB operation

If an error occurs while HiRDB is checking the definition information for audit events, HiRDB outputs the corresponding audit trail regardless of the definition of audit events.

If an error occurs during SQL execution, HiRDB also outputs audit trails. In this case, an error may also occur when the definition information for an audit event is acquired during output of an audit trail. Table 22-30 describes the combinations of errors, the SQL codes to be set, and whether or not rollback is required. If an error occurs during output of audit trails, HiRDB ignores that error and resumes processing.

Table 22-30: Combinations of errors, SQL codes to be set, and whether or not rollback is required

Status before acquisition of audit event definition	Status during acquisition of audit event definition	SQL code to be set	Whether or not rollback is required
Normal	Normal	0	No
	Error requiring rollback	SQL code during acquisition of audit event definition	Yes
	Error not requiring rollback	SQL code before acquisition of audit event definition	No
Error requiring rollback	Normal	SQL code before acquisition of audit event definition	Yes
	Error requiring rollback	SQL code before acquisition of audit event definition	Yes
	Error not requiring rollback	SQL code before acquisition of audit event definition	Yes
Error not requiring rollback	Normal	SQL code before acquisition of audit event definition	No
	Error requiring rollback	SQL code before acquisition of audit event definition	Yes
	Error not requiring rollback	SQL code before acquisition of audit event definition	No

(6) Change of security audit information buffer status

When an event occurs, the status of the security audit information buffer changes, such as from disabled to enabled. The following table shows the changes in the security audit information buffer status when an event occurs.

Event	Status of security audit information buffer			
	Initial status (before HiRDB start)	Disabled status (no information has been set)	Enabled status (information has been set)	Disabled status (old information remains)
	1	2	3	4
Completion of HiRDB startup processing	→2	--	--	--
Access to security audit information buffer	--	→3	→3	→3
Change to audit event definition (execution of `CREATE AUDIT` or `DROP AUDIT`)	--	→2	→4	→4

Legend:

→*n*: When the event occurs, the security audit information buffer is placed in the status indicated by the number *n*.

--: Not applicable

22.10 Audit trail file error handling

(1) When an error occurs in an audit trail file

When an error occurs in an audit trail file, the file is placed in shutdown status. An audit trail file in shutdown status cannot be used except to be deleted. Take action according to the following procedure:

Procedure

1. Identify the audit trail file in shutdown status from the `KFPS05704-E` message.

   You can use the `pdls -d aud` command to identify the generation in which the error occurred. If HiRDB is running, the auditor executes the `pdls -d aud` command.

2. Use the `pdaudrm` command to delete the audit trail files in shutdown status. To delete forcibly the files in data load waiting status, specify the `-f` option.

Files in shutdown status cannot be data loaded. Therefore, if they are not data loaded, the entire audit trail will be lost.

(2) When an error occurs in the HiRDB file system area used for audit trail files (when all audit trail files are in shutdown status)

This explains the procedure when an error has occurred in the HiRDB file system area used for audit trail files, causing all audit trail files to be placed in shutdown status.

The HiRDB processing depends on the value of the `pd_aud_no_standby_file_opr` operand. If `down` is specified, HiRDB (or unit for a HiRDB/Parallel Server) is terminated forcibly. If `forcewrite` (default) is specified, audit trail output stops, but HiRDB continues in operation. The HiRDB administrator uses the procedure explained below.

Procedure

1. If HiRDB is operating, use the `pdstop` command to terminate HiRDB normally. If normal termination is not possible, use the `pdstop -f` command to terminate HiRDB forcibly.

2. Use the `pdfmkfs` command to re-create the file system area for audit trail files. If the HiRDB file system area cannot be used due to a fault such as s disk error, create another HiRDB file system area for audit trail files on a different disk.

3. If you changed the file system area for the audit trail files, also change the specification of the `pd_aud_file_name` operand.

4. If there is a backup of the file system area for audit trail files, use the

pdfrstr command to recover the HiRDB file system area.

5. Use the pdstart command to start HiRDB.

(3) When HiRDB is terminated forcibly because there are no swappable target audit trail files

If down is specified in the pd_aud_no_standby_file_opr operand and only one swappable file remains, terminate HiRDB forcibly (for a HiRDB/Parallel Server, terminate the unit). The HiRDB administrator uses the procedure explained below.

(a) For a HiRDB/Single Server

Select one of the following procedures:

Procedure 1: When pd_aud_max_generation_num < 200

1. Specify 200 for the pd_aud_max_generation_num operand.
2. Use the pdstart command to start HiRDB.
3. Use the pdload command to data load the files in data load waiting status.

Procedure 2: When pd_aud_max_generation_num = 200

1. Specify N in the pd_audit operand.
2. Use the pdstart command to start HiRDB.
3. Use the pdload command to data load the files in data load waiting status.
4. Use the pdaudbegin command to start collection of the audit trail.

(b) For a HiRDB/Parallel Server

Select one of the following procedures:

Procedure 1: When pd_aud_max_generation_num < 200

1. Use the pdstop -f command to terminate HiRDB forcibly.
2. Specify 200 for the pd_aud_max_generation_num operand.
3. Use the pdstart command to start HiRDB.
4. Use the pdload command to data load the files in data load waiting status.

Procedure 2: When pd_aud_max_generation_num = 200

1. Use the pdstop -f command to forcibly terminate HiRDB.
2. Specify N in the pd_audit operand.
3. Use the pdstart command to start HiRDB.
4. Use the pdload command to data load the files in data load waiting status.
5. Use the pdaudbegin command to start collection of the audit trail.

22.11 Linkage with other facilities

This section explains security audit facility linkage with other facilities.

(1) Linkage with the system switchover facility

- Whether or not audit trail collection is inherited by the switchover target system is determined from the termination mode of the switching HiRDB. If the switching system is restarted, the status before system switchover is inherited. If the switchover target system terminates normally, it follows the specification of the `pd_audit` operand.
- If the system switchover facility is used in the monitor mode, an audit trail is collected when system switchover occurs and the standby HiRDB system (the standby unit for a HiRDB/Parallel Server) is started. An audit trail is not collected when system switchover occurs in server mode.
- If system switchover occurs due to an error, HiRDB does not correctly collect the audit trail before the switchover.

(2) Linkage with plug-ins

If the `pdplgrgst` command is used to register, delete, or upgrade plug-ins with HiRDB, HiRDB uses the `pdplgrgst` command extension to issue the following SQL statements:

- `CREATE` statements for functions, abstract data types, or indexes provided by the plug-ins
- `DELETE` statements for the plug-in definition information stored in the dictionary tables
- `DROP` statements for functions, abstract data types, or indexes provided by the plug-ins

If audit events are set for these SQL statements, an audit trail is collected for these SQL statements using the `pdplgrgst` command extension.

(3) Linkage with the inner replica facility

An audit trail is not collected for pair volumization (disk duplexing). Use the OS's auditing facility to audit pair volumization.

(4) Linkage with the HiRDB External Data Access facility

An audit trail is collected for SQL statements used by the HiRDB External Data Access facility.

22.12 Audit trail record items (during privilege checking)

The audit trail record items during privilege checking are described below.

The record items for *User identifier* through *Process ID* are described in Part 1; the record items for *Thread ID* through *SQL code/termination code* is described in Part 2, and the record items for *Name of swapping source audit trail file* and subsequent are described in Part 3.

For details about the event types and subtypes, see Table *22-16 Event types and subtypes*. For details about the privileges that were used, see *USED_PRIVILEGE* in Table *22-15 Audit trail table columns*.

■ Audit trail record items during privilege checking (part 1)

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[3]	Object name
System administrator security events	HiRDB start	pdstart	Y	Y	Y	Y	SYS	STR	Y	SYS	—	—	—	Y	—	Y	Y	—	—	—	—	—
	HiRDB stop	pdstop	Y	Y	Y	Y	SYS	STP	Y	SYS	—	—	—	Y	—	Y	Y	—	—	—	—	—
	Auditor registration	pdmod	Y	Y	Y	Y	SYS	MOD	Y	SYS	"pdmod"	—	—	Y	—	Y	—	—	—	—	—	—
	Audit trail table creation	pdmod	Y	Y	Y	Y	SYS	MOD	Y	SYS	"pdmod"	—	—	Y	—	Y	—	—	—	—	Y	Y
	Audit trail file deletion	pdaudrm	Y	Y	Y	Y	SYS	ARM	Y	SYS	—	—	—	Y	—	Y	Y	—	—	—	—	Y
	Audit trail collection start	pdaudbegin	Y	Y	Y	Y	SYS	ABG	Y	SYS	"pdaudbegin"	—	—	Y	—	Y	Y	—	—	—	—	—
		HiRDB start	Y	Y	Y	Y	SYS	ABG	Y	SYS	"pdstart"	—	—	Y	—	Y	Y	—	—	—	—	—
	Audit trail collection termination	pdaudend	Y	Y	Y	Y	SYS	AEN	Y	SYS	"pdaudend"	—	—	Y	—	Y	Y	—	—	—	—	—
		HiRDB termination	Y	Y	Y	Y	SYS	AEN	Y	SYS	"pdstop"	—	—	Y	—	Y	Y	—	—	—	—	—
	Release of consecutive certification failure account lock state	pdacunlck (when all DBA privilege holders are in account lock state)	Y	Y	Y	Y	SYS	ULK	Y	SYS	"pdacunlck"	—	—	Y	—	Y	—	—	—	—	—	—
	Setting value modification for connection security facility	CREATE CONNECTION SECURITY, DROP CONNECTION SECURITY	Y	Y	Y	Y	SYS	SPR	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
Auditor security events	Data loading in audit trail table	pdload	Y	Y	Y	Y	AUD	ALD	Y	AUD	"pdload"	—	—	Y	—	Y	—	—	—	—	—	—
	Swapping of audit trail files	pdaudswap	Y	Y	Y	Y	AUD	ASW	Y	AUD	—	—	—	Y	—	Y	Y	—	—	—	—	—
	Audited event definition	CREATE AUDIT	Y	Y	Y	Y	AUD	CRT	Y	AUD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Audited event deletion	DROP AUDIT	Y	Y	Y	Y	AUD	DRP	Y	AUD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Auditor password change	GRANT AUDIT	Y	Y	Y	Y	AUD	GRT	Y	AUD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[3]	Object name
Session security events	Connection to HiRDB	`CONNECT`	Y	Y	Y	Y	SES	CNT	Y	CNT	Y	S	Y	Y	—	Y	Y	Y	Y	—	—	—
	User change	`SET SESSION AUTHORIZATION`	Y	Y	Y	Y	SES	ATH	Y	CNT	Y	S	Y	Y	—	Y	Y	Y	Y	—	—	—
Privilege control events	Privilege granting	`GRANT DBA`	Y	Y	Y	Y	PRV	GRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`GRANT SCHEMA`	Y	Y	Y	Y	PRV	GRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`GRANT CONNECT`	Y	Y	Y	Y	PRV	GRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`GRANT RDAREA`	Y	Y	Y	Y	PRV	GRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`GRANT` *access-privilege*	Y	Y	Y	Y	PRV	GRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	GRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	GRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
	Privilege revocation	`REVOKE DBA`	Y	Y	Y	Y	PRV	RVK	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`REVOKE SCHEMA`	Y	Y	Y	Y	PRV	RVK	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`REVOKE CONNECT`	Y	Y	Y	Y	PRV	RVK	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`REVOKE RDAREA`	Y	Y	Y	Y	PRV	RVK	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		`REVOKE` *access-privilege*	Y	Y	Y	Y	PRV	RVK	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	RVK	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	RVK	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
Object definition events	`ALTER`	`ALTER PROCEDURE`	Y	Y	Y	Y	DEF	ALT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		`ALTER ROUTINE`	Y	Y	Y	Y	DEF	ALT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		`ALTER TABLE` (other than `ADD RDAREA` and `CHANGE RDAREA`)	Y	Y	Y	Y	DEF	ALT	Y	RDA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		`ALTER TABLE ADD RDAREA` `ALTER TABLE CHANGE RDAREA`	Y	Y	Y	Y	DEF	ALT	Y	RDA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		`ALTER TRIGGER`	Y	Y	Y	Y	DEF	ALT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		`COMMENT`	Y	Y	Y	Y	DEF	ALT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[3]	Object name
Object definition events (cont'd)	CREATE	CREATE ALIAS	Y	Y	Y	Y	DEF	CRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	CRT	Y	SCH	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		CREATE FOREIGN INDEX	Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE FOREIGN TABLE	Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE FUNCTION	Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE INDEX (Format 1)	Y	Y	Y	Y	DEF	CRT	Y	RDA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE INDEX (Format 2)	Y	Y	Y	Y	DEF	CRT	Y	RDA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE PROCEDURE	Each SQL statement in a procedure is output as event type DEF and event subtype CRT. For details about the event type and subtype with which the other SQL statements are output, see the applicable SQL statement.																			
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE SCHEMA	Y	Y	Y	Y	DEF	CRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	CRT	Y	SCH	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		CREATE SERVER	Y	Y	Y	Y	DEF	CRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		CREATE TABLE	Y	Y	Y	Y	DEF	CRT	Y	RDA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE TRIGGER	Each SQL statement in a trigger SQL statement is output as event type DEF and event subtype CRT from the CREATE PROCEDURE statement that is executed as an extension. For details about the event type and subtype with which the other SQL statements are output, see the applicable SQL statement.																			
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE TYPE	Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE USER MAPPING	Y	Y	Y	Y	DEF	CRT	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		CREATE VIEW	Y	Y	Y	Y	DEF	CRT	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE PUBLIC VIEW	Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	CRT	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL	Audit record items (part 1)																				
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success / failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[3]	Object name
Object definition events (cont'd)	DROP	DROP ALIAS	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP DATA TYPE	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP FOREIGN INDEX	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP FOREIGN TABLE	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP FUNCTION	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP INDEX	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP PROCEDURE	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP SCHEMA	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP SERVER	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP TABLE	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP TRIGGER	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP USER MAPPING	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		DROP VIEW DROP PUBLIC VIEW	Y	Y	Y	Y	DEF	DRP	Y	DBA	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
			Y	Y	Y	Y	DEF	DRP	Y	OWN	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
Object operation events	Operations	DELETE (static SQL)	Y	Y	Y	Y	ACS	DEL	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	DEL	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	DEL	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		EXECUTE[1]	Depends on the pre-processed SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.																			
		EXECUTE IMMEDIATE	Depends on the specified SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.																			
		INSERT (static SQL)	Y	Y	Y	Y	ACS	INS	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	INS	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	INS	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		LOCK TABLE (shared) (static SQL)	Y	Y	Y	Y	ACS	LCK	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		LOCK TABLE (locked) (static SQL)	Y	Y	Y	Y	ACS	LCK	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	LCK	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		OPEN[1] SELECT	Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL		Audit record items (part 1)																			
				User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success / failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[3]	Object name
Object operation events (cont'd)	Operations (cont'd)	PREPARE[1]	ASSIGN LIST (Format 1)	Y	Y	Y	Y	ACS	ASN	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			DELETE	Y	Y	Y	Y	ACS	DEL	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	DEL	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	DEL	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			INSERT	Y	Y	Y	Y	ACS	INS	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	INS	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	INS	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			LOCK TABLE (shared)	Y	Y	Y	Y	ACS	LCK	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			LOCK TABLE (locked)	Y	Y	Y	Y	ACS	LCK	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	INS	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			UPDATE	Y	Y	Y	Y	ACS	UPD	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			PURGE TABLE	Y	Y	Y	Y	ACS	PRG	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			SELECT	Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		PURGE (static SQL)		Y	Y	Y	Y	ACS	PRG	Y	DEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		UPDATE (static SQL)		Y	Y	Y	Y	ACS	UPD	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	UPD	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		SELECT, 1-line SELECT statement (static SQL)		Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	SEL	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
	Embedded language	ALLOCATE CONNECTION HANDLE		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		FREE CONNECTION HANDLE		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		DECLARE CONNECTION HANDLE SET		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		DECLARE CONNECTION HANDLE		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		GET CONNECTION HANDLE		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		COPY		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		GET DIAGNOSTICS		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		COMMAND EXECUTE		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Event		Command/ SQL	Audit record items (part 1)																				
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success / failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[3]	Object name
Utility operation events	Creation	pdload[2]	Y	Y	Y	Y	UTL	LOD	Y	INS	"pdload"	—	—	Y	—	Y	—	—	—	—	Y	Y
			Y	Y	Y	Y	UTL	LOD	Y	DEL	"pdload"	—	—	Y	—	Y	—	—	—	—	Y	Y
		pdexp pddefrev	Y	Y	Y	Y	UTL	EXP	Y	DBA	"pdexp"	—	—	Y	—	Y	—	—	—	—	—	—
	Application	pdrorg	Y	Y	Y	Y	UTL	ORG	Y	DBA	"pdrorg"	—	—	Y	—	Y	—	—	—	—	—	—
			Y	Y	Y	Y	UTL	ORG	Y	SEL	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
			Y	Y	Y	Y	UTL	ORG	Y	INS	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
			Y	Y	Y	Y	UTL	ORG	Y	DEL	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
			Y	Y	Y	Y	UTL	ORG	Y	AUD	"pdrorg"	—	—	Y	—	Y	—	—	—	—	—	—
			Y	Y	Y	Y	UTL	ORG	Y	OWN	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
		pdconstck	Y	Y	Y	Y	UTL	CST	Y	DBA	"pdconstck"	—	—	Y	—	Y	—	—	—	—	—	—
			Y	Y	Y	Y	UTL	CST	Y	OWN	"pdconstck"	—	—	Y	—	Y	—	—	—	—	Y	Y

■ Audit trail record items during privilege checking (part 2)

Event		Command/ SQL	Audit record items (part2)													Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
System administrator security events	HiRDB start	pdstart	—	—	—	Y	—	—	—	—	—	—	—	—	—	1	—	Unit in which pdstart command was executed
	HiRDB stop	pdstop	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	MGR
	Auditor registration	pdmod	—	AUD	Y	—	—	—	—	—	—	—	—	—	—	AUD + ATB	—	DS
	Audit trail table creation	pdmod	TBL	—	—	—	—	—	—	—	—	—	—	—	—			
	Audit trail file deletion	pdaudrm	AUF	—	—	—	—	—	—	—	—	—	—	—	—	1	—	MGR
	Audit trail collection start	pdaudbegin	—	—	—	—	—	—	—	—	—	—	—	—	—			
		HiRDB start	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	Active units
	Audit trail collection termination	pdaudend	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	MGR
		HiRDB termination	—	—	—	—	—	—	—	—	—	—	—	—	—		—	Terminated units
	Release of consecutive certification failure account lock state	pdacunlck (when all DBA privilege holders are in account lock state)	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
	Setting value modification for connection security facility	CREATE CONNECTION SECURITY, DROP CONNECTION SECURITY	—	—	—	—	—	—	—	—	—	—	—	—	—			
Auditor security events	Data loading in audit trail table	pdload	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	See section *22.14*
	Swapping of audit trail files	pdaudswap	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	MGR
	Audited event definition	CREATE AUDIT	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
	Audited event deletion	DROP AUDIT	—	—	—	—	—	—	—	—	—	—	—	—	—			
	Auditor password change	GRANT AUDIT	—	AUD	Y	—	—	—	—	—	—	—	—	—	—			

Event		Command/ SQL	Audit record items (part 2)														SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count	Number of records		
Session security events	Connection to HiRDB	CONNECT	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	FES
	User change	SET SESSION AUTHORIZATION	—	—	—	—	—	—	—	—	—	—	—	—	—			
Privilege control events	Privilege granting	GRANT DBA	—	DBA	Y	—	—	—	—	—	—	—	—	—	—	USR	—	DS
		GRANT SCHEMA	—	SCH	Y	—	—	—	—	—	—	—	—	—	—			
		GRANT CONNECT	—	CNT	Y	—	—	—	—	—	—	—	—	—	—			
		GRANT RDAREA	—	RDA	Y	—	—	—	—	—	—	—	—	—	—			
		GRANT *access-privilege*	FTB	Y	Y	—	—	—	—	—	—	—	—	—	—	USR × ACS		
			TBL	Y	Y	—	—	—	—	—	—	—	—	—	—			
			VIW	Y	Y	—	—	—	—	—	—	—	—	—	—			
	Privilege revocation	REVOKE DBA	—	DBA	Y	—	—	—	—	—	—	—	—	—	—	USR	—	DS
		REVOKE SCHEMA	—	SCH	Y	—	—	—	—	—	—	—	—	—	—			
		REVOKE CONNECT	—	CNT	Y	—	—	—	—	—	—	—	—	—	—			
		REVOKE RDAREA	—	RDA	Y	—	—	—	—	—	—	—	—	—	—			
		REVOKE *access-privilege*	FTB	Y	Y	—	—	—	—	—	—	—	—	—	—	USR × ACS	DROP VIEW	
			TBL	Y	Y	—	—	—	—	—	—	—	—	—	—			
			VIW	Y	Y	—	—	—	—	—	—	—	—	—	—			
Object definition events	ALTER	ALTER PROCEDURE	—	—	—	—	—	—	—	—	—	—	—	—	—	1	CREATE PROCEDURE (note that objects PRC and SCH in CREATE PROCEDURE are not output)	DS
			PRC	—	—	—	—	—	—	—	—	—	—	—	—	PRC		
		ALTER ROUTINE	—	—	—	—	—	—	—	—	—	—	—	—	—	1		
			FNC	—	—	—	—	—	—	—	—	—	—	—	—	FNC		
			PRC	—	—	—	—	—	—	—	—	—	—	—	—	PRC		
			TRG	—	—	—	—	—	—	—	—	—	—	—	—	TRG		
		ALTER TABLE (other than ADD RDAREA and CHANGE RDAREA)	RDA	—	—	—	—	—	—	—	—	—	—	—	—	RDA + RLB + RID	DROP VIEW, DROP INDEX, DROP TRIGGER, DROP PUBLIC VIEW	FES or DS
			IDX	—	—	—	—	—	—	—	—	—	—	—	—	IDX		
			PRC	—	—	—	—	—	—	—	—	—	—	—	—	PRC		
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	TBL		
			TRG	—	—	—	—	—	—	—	—	—	—	—	—	TRG		
			VIW	—	—	—	—	—	—	—	—	—	—	—	—	VIW		
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	RDA	—	—	—	—	—	—	—	—	—	—	—	—	RDA + RLB + RID + CLS + PRM	—	FES or DS
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	TBL		
		ALTER TRIGGER	—	—	—	—	—	—	—	—	—	—	—	—	—	1	CREATE PROCEDURE (note that objects PRC and SCH in CREATE PROCEDURE are not output)	DS
			TRG	—	—	—	—	—	—	—	—	—	—	—	—			
		COMMENT	TBL	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS

Event		Command/ SQL	Audit record items (part 2)														Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
Object definition events (cont'd)	CREATE	CREATE ALIAS	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	FES or DS
			—	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE FOREIGN INDEX	FID	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
			FTB	—	—	—	—	—	—	—	—	—	—	—	—			
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE FOREIGN TABLE	FSV	—	—	—	—	—	—	—	—	—	—	—	—			
			FTB	—	—	—	—	—	—	—	—	—	—	—	—			
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE FUNCTION	FNC	—	—	—	—	—	—	—	—	—	—	—	—	1	—	FES or DS
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE INDEX (Format 1)	RDA	—	—	—	—	—	—	—	—	—	—	—	—	RDA	—	DS
			IDX	—	—	—	—	—	—	—	—	—	—	—	—	1		
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE INDEX (Format 2)	RDA	—	—	—	—	—	—	—	—	—	—	—	—	RDA		
			IDX	—	—	—	—	—	—	—	—	—	—	—	—	1		
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE PROCEDURE	Each SQL statement in a procedure is output as event type DEF and event subtype CRT. For details about the event type and subtype with which the other SQL statements are output, see the applicable SQL statement.															—	FES or DS
			PRC	—	—	—	—	—	—	—	—	—	—	—	—	1		
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE SCHEMA	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
			—	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE SERVER	—	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE TABLE	RDA	—	—	—	—	—	—	—	—	—	—	—	—	RDA + RID + RLB + CLS + PRM	CREATE TRIGGER (when generating a trigger for performing a constraint action)	DS
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	1		
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE TRIGGER	Each SQL statement in a trigger SQL statement is output as event type DEF and event subtype CRT from the CREATE PROCEDURE statement that is executed as an extension. For details about the event type and subtype with which the other SQL statements are output, see the applicable SQL statement.															CREATE PROCEDURE	DS
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	1		
			TRG	—	—	—	—	—	—	—	—	—	—	—	—			
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE TYPE	TYP	—	—	—	—	—	—	—	—	—	—	—	—	1	· CREATE PROCEDURE (when there is a member procedure) · CREATE FUNCTION (when there is a member function)	DS
			SCH	—	—	—	—	—	—	—	—	—	—	—				
		CREATE USER MAPPING	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
		CREATE VIEW	FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES or DS
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
			VIW	—	—	—	—	—	—	—	—	—	—	—	—			
			VIW	—	—	—	—	—	—	—	—	—	—	—	—	1		
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		CREATE PUBLIC VIEW	FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL[4]	—	FES
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
			VIW	—	—	—	—	—	—	—	—	—	—	—	—			

Event		Command/ SQL	Audit record items (part 2)														SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count	Number of records		
Object definition events (cont'd)	DROP	DROP ALIAS	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	FES or DS
			ALS	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP DATA TYPE	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP FUNCTION, DROP PROCEDURE	DS
			FNC	—	—	—	—	—	—	—	—	—	—	—	—	FNC		
			PRC	—	—	—	—	—	—	—	—	—	—	—	—	PRC		
			TYP	—	—	—	—	—	—	—	—	—	—	—	—	1		
		DROP FOREIGN INDEX	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
			FID	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP FOREIGN TABLE	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP VIEW, DROP FOREIGN INDEX, DROP PUBLIC VIEW	DS
			FTB	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP FUNCTION	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP DATA TYPE	DS
			FNC	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP INDEX	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
			IDX	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP PROCEDURE	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP DATA TYPE	DS
			PRC	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP SCHEMA	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP TABLE, DROP VIEW, DROP FOREIGN TABLE, DROP INDEX, DROP FOREIGN INDEX, DROP PROCEDURE, DROP FUNCTION, DROP DATA TYPE, DROP TRIGGER, DROP PUBLIC VIEW	DS
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP SERVER	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
			FSV	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP TABLE	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP VIEW, DROP INDEX, DROP TRIGGER, DROP PROCEDURE, DROP PUBLIC VIEW	DS
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP TRIGGER	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP PROCEDURE	DS
			TRG	—	—	—	—	—	—	—	—	—	—	—	—			
		DROP USER MAPPING	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	DS
		DROP VIEW DROP PUBLIC VIEW	—	—	—	—	—	—	—	—	—	—	—	—	—	1	DROP VIEW, DROP PUBLIC VIEW	DS
			VIW	—	—	—	—	—	—	—	—	—	—	—	—			
Object operation events	Operations	DELETE (static SQL)	FTB	—	—	—	—	—	—	—	—	—	—	—	—	—	1	Trigger SQL statement (including trigger for performing a constraint action)	FES
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	—			
			VIW	—	—	—	—	—	—	—	—	—	—	—	—	—			
		EXECUTE[1]	Depends on the pre-processed SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.														Pre-processed SQL	FES
		EXECUTE IMMEDIATE	Depends on the specified SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.														Specified SQL	FES
		INSERT (static SQL)	FTB	—	—	—	—	—	—	—	—	—	—	—	—	—	1	· SELECT · Trigger SQL statement (including trigger for performing a constraint action)	FES
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	—			
			VIW	—	—	—	—	—	—	—	—	—	—	—	—	—			

Event		Command/ SQL		Audit record items (part 2)														Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
				Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
Object operation events (cont'd)	Operations (cont'd)	LOCK TABLE (shared) (static SQL)		FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
		LOCK TABLE (locked) (static SQL)		FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
				FTB	—	—	—	—	—	—	—	—	—	—	—	—			
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
				FTB	—	—	—	—	—	—	—	—	—	—	—	—			
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
		OPEN[1]	SELECT	FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
		PREPARE[1]	ASSIGN LIST (Format 1)	TBL	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
			DELETE	FTB	—	—	—	—	—	—	—	—	—	—	—	—	1	PREPARE(SELECT)	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
			INSERT	FTB	—	—	—	—	—	—	—	—	—	—	—	—	1	PREPARE(SELECT)	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
			LOCK TABLE (shared)	FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
			LOCK TABLE (locked)	FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
				FTB	—	—	—	—	—	—	—	—	—	—	—	—			
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
				FTB	—	—	—	—	—	—	—	—	—	—	—	—			
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
			UPDATE	FTB	—	—	—	—	—	—	—	—	—	—	—	—	1	PREPARE(SELECT)	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
			PURGE TABLE	TBL	—	—	—	—	—	—	—	—	—	—	—	—	1	—	FES
			SELECT	FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	PREPARE(SELECT)	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
		PURGE (static SQL)		TBL	—	—	—	—	—	—	—	—	—	—	—	—	1	—	FES
		UPDATE (static SQL)		FTB	—	—	—	—	—	—	—	—	—	—	—	—	1	· SELECT · Trigger SQL statement (including trigger for performing a constraint action)	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			
		SELECT, 1-line SELECT statement (static SQL)		FTB	—	—	—	—	—	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	—	—	—	—	—	—	—	—	—			
				VIW	—	—	—	—	—	—	—	—	—	—	—	—			

Event		Command/ SQL	Audit record items (part 2)														SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count	Number of records		
Object operation events (cont'd)	Embedded language	ALLOCATE CONNECTION HANDLE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		FREE CONNECTION HANDLE	—	—	—	—	—	—	—	—	—	—	—	—	—			
		DECLARE CONNECTION HANDLE SET	—	—	—	—	—	—	—	—	—	—	—	—	—			
		DECLARE CONNECTION HANDLE	—	—	—	—	—	—	—	—	—	—	—	—	—			
		GET CONNECTION HANDLE	—	—	—	—	—	—	—	—	—	—	—	—	—			
		COPY	—	—	—	—	—	—	—	—	—	—	—	—	—			
		GET DIAGNOSTICS	—	—	—	—	—	—	—	—	—	—	—	—	—			
		COMMAND EXECUTE	—	—	—	—	—	—	—	—	—	—	—	—	—			
Utility operation events	Creation	pdload[2]	TBL	—	—	—	—	—	—	—	—	—	—	—	—	1	—	See section *22.14*
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
		pdexp pddefrev	—	—	—	—	—	—	—	—	—	—	—	—	—			
	Application	pdrorg	—	—	—	—	—	—	—	—	—	—	—	—	—			
			TBL	—	—	—	—	—	—	—	—	—	—	—	—	TBL		
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			
			—	—	—	—	—	—	—	—	—	—	—	—	—	1		
			SCH	—	—	—	—	—	—	—	—	—	—	—	—			
		pdconstck	—	—	—	—	—	—	—	—	—	—	—	—	—	1		
			TBL	—	—	—	—	—	—	—	—	—	—	—	—			

Legend:

Y: Information is collected.

S: Information is collected sometimes via the XA interface.

—: Information is not collected; or, not applicable.

ACS: Number of privileges specified (4 if all is specified).

ATB: Number of audit trail tables created with the `pdmod` command (1 if audit trail tables were created; 0 if audit trail tables were not created).

AUD: Number of auditors created with the `pdmod` command (1 if auditors were created; 0 if auditors were not created).

CLS: 1 if FOR CLUSTER KEY clause is specified; 0 if it is not specified.

FNC: Number of functions to be re-created.

IDX: Number of related indexes.

PRC: Number of procedures to be re-created.

PRM: 1 if FOR PRIMARY KEY clause is specified; 0 if it is not specified.

RDA: Number of RDAREAs specified.

RLB: Number of RDAREAs for LOB data.

RID: Number of index RDAREAs.

TBL: Number of tables specified by a SELECT statement, SELECT clause, or LOCK statement, or the number of tables to be processed by a utility.

TRG: Number of triggers to be re-created.

USR: Number of specified users or groups.

VIW: Number of target view tables.

[1] When the SQL type is SELECT, INSERT, UPDATE, or DELETE, usage privilege is also checked during pre-processing by a PREPARE statement and an audit trail is collected. As a result, the number of output records is doubled. The timing for usage privilege checking is described below. The underlined SQL statements check usage privilege and collect an audit trail.

When the SQL type is SELECT

- When preprocessing by a PREPARE statement is not performed
```
EXEC SQL DECLARE C1 CURSOR FOR SELECT * FROM T1;
EXEC SQL OPEN C1;
```

- When preprocessing by a PREPARE statement is performed
```
EXEC SQL PREPARE S1 FROM 'SELECT * FROM T1';
EXEC SQL DECLARE C1 CURSOR FOR S1;
EXEC SQL OPEN C1;
```

When the SQL type is INSERT (same for UPDATE and DELETE)

- When preprocessing by a PREPARE statement is not performed
```
EXEC SQL INSERT INTO T1(C1) VALUES(1);
```

- When preprocessing by a PREPARE statement is performed
```
EXEC SQL PREPARE S1 FROM 'INSERT
                          INTO T1(C1) VALUES(?)';
EXEC SQL EXECUTE S1 FOR:data;
```

[2] Applies to cases in which a table is not or cannot be recognized as an audit trail table.

[3] When the target object is a public view table, PUBLIC is output as the object owner.

[4] When a base table is a view table or public view table, the record count is increased for each real table or foreign table that is used as the base table.

22.13 Audit trail record items (at event termination)

The audit trail record items at event termination are described below.

The record items for *User identifier* through *Process ID* are described in Part 1; the record items for *Thread ID* through *SQL code/termination code* is described in Part 2, and the record items for *Name of swapping source audit trail file* and subsequent are described in Part 3.

For details about the event types and subtypes, see *Table 22-16 Event types and subtypes*. For details about the privileges that were used, see *USED_PRIVILEGE* in *Table 22-15 Audit trail table columns*.

■ Audit trail record items at event termination (part 1)

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/ failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[1]	Object name
System administrator security events	HiRDB start	pdstart	Y	Y	Y	Y	SYS	STR	Y	'△△△'	—	—	—	Y	—	Y	Y	—	—	—	—	—
	HiRDB stop	pdstop	Y	Y	Y	Y	SYS	STP	Y	'△△△'	—	—	—	Y	—	Y	Y	—	—	—	—	—
	Auditor registration	pdmod	Y	Y	Y	Y	SYS	MOD	Y	'△△△'	"pdmod"	—	—	Y	—	Y	—	—	—	—	—	—
	Audit trail table creation	pdmod	Y	Y	Y	Y	SYS	MOD	Y	'△△△'	"pdmod"	—	—	Y	—	Y	—	—	—	—	Y	Y
	Audit trail file deletion	pdaudrm	Y	Y	Y	Y	SYS	ARM	Y	'△△△'	"pdaudrm"	—	—	Y	—	Y	Y	—	—	—	—	Y
	Audit trail collection start	pdaudbegin	Y	Y	Y	Y	SYS	ABG	Y	'△△△'	"pdaudbegin"	—	—	Y	—	Y	Y	—	—	—	—	—
		HiRDB start	Y	Y	Y	Y	SYS	ABG	Y	'△△△'	"pdstart"	—	—	Y	—	Y	Y	—	—	—	—	—
	Audit trail collection termination	pdaudend	Y	Y	Y	Y	SYS	AEN	Y	'△△△'	"pdaudend"	—	—	Y	—	Y	Y	—	—	—	—	—
		HiRDB termination	Y	Y	Y	Y	SYS	AEN	Y	'△△△'	"pdstop"	—	—	Y	—	Y	Y	—	—	—	—	—
	Audit trail file overwriting	Automatic overwriting by the system	Y	Y	Y	Y	SYS	OVW	Y	'△△△'	—	—	—	Y	—	Y	Y	—	—	—	—	—
		Manual overwriting by pdaudswap	Y	Y	Y	Y	SYS	OVW	Y	'△△△'	"pdaudswap"	—	—	Y	—	Y	Y	—	—	—	—	—
	Transition to consecutive certification failure account lock state	Connection failure	Y	Y	Y	Y	SYS	CLK	Y	'△△△'	—	—	—	Y	—	Y	Y	—	—	—	—	—
	Release of consecutive certification failure account lock state	Connection after account lock period has passed	Y	Y	Y	Y	SYS	CUL	Y	'△△△'	—	—	—	Y	—	Y	Y	Y	Y	Y	—	—
		DROP CONNECTION SECURITY FOR CONNECT	Y	Y	Y	Y	SYS	CUL	Y	'△△△'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		pdacunlck	Y	Y	Y	Y	SYS	CUL	Y	'△△△'	"pdacunlck"	—	—	Y	—	Y	—	—	—	—	—	—
	Transition to password-invalid account lock state		Y	Y	Y	Y	SYS	PLK	Y	'△△△'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Release from password-invalid account lock state		Y	Y	Y	Y	SYS	PUL	Y	'△△△'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Setting value modification for connection security facility	CREATE CONNECTION SECURITY, DROP CONNECTION SECURITY	Y	Y	Y	Y	SYS	SPR	Y	'△△△'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Execution of pdacunlck		Y	Y	Y	Y	SYS	ULK	Y	'△△△'	"pdacunlck"	—	—	Y	—	Y	—	—	—	—	—	—

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[1]	Object name
Auditor security events	Data loading in audit trail table	pdload	Y	Y	Y	Y	AUD	ALD	Y	'ΔΔΔ'	"pdload"	—	—	Y	—	Y	—	—	—	—	Y	Y
	Swapping of audit trail files	pdaudswap	Y	Y	Y	Y	AUD	ASW	Y	'ΔΔΔ'	"pdaudswap"	—	—	Y	—	Y	Y	—	—	—	—	—
	Audited event definition	CREATE AUDIT	Y	Y	Y	Y	AUD	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Audited event deletion	DROP AUDIT	Y	Y	Y	Y	AUD	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
	Auditor password change	GRANT AUDIT	Y	Y	Y	Y	AUD	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
Session security events	Connection to HiRDB	CONNECT	Y	Y	Y	Y	SES	CNT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	—	—	—
	User change	SET SESSION AUTHORIZATION	Y	Y	Y	Y	SES	ATH	Y	'ΔΔΔ'	S	S	S	Y	—	Y	Y	Y	Y	Y	—	—
Privilege control events	Privilege granting	GRANT DBA	Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		GRANT SCHEMA	Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		GRANT CONNECT	Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		GRANT RDAREA	Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
		GRANT *access-privilege*	Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	GRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
	Privilege revocation	REVOKE DBA	Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		REVOKE SCHEMA	Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		REVOKE CONNECT	Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	—
		REVOKE RDAREA	Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
		REVOKE *access-privilege*	Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	PRV	RVK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success / failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[1]	Object name
Object definition events	ALTER	ALTER PROCEDURE	Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		ALTER ROUTINE	Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		ALTER TABLE **(other than** ADD RDAREA **and** CHANGE RDAREA**)**	Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		ALTER TABLE ADD RDAREA, ALTER TABLE CHANGE RDAREA	Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		ALTER TRIGGER	Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		COMMENT	Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	DEF	ALT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
	CREATE	CREATE ALIAS	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE FOREIGN INDEX	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE FOREIGN TABLE	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE FUNCTION	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE INDEX **(Format 1)**	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE INDEX **(Format 2)**	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE PROCEDURE	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE SCHEMA	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE SERVER	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE TABLE	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE TRIGGER	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE TYPE	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CREATE USER MAPPING[2]	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
		CREATE VIEW, CREATE PUBLIC VIEW	Y	Y	Y	Y	DEF	CRT	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
	DROP	DROP ALIAS	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP DATA TYPE	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP FOREIGN INDEX	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP FOREIGN TABLE	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP FUNCTION	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP INDEX	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP PROCEDURE	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP SCHEMA	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP SERVER	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP TABLE	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP TRIGGER	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DROP USER MAPPING[2]	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	—	Y
		DROP VIEW, DROP PUBLIC VIEW	Y	Y	Y	Y	DEF	DRP	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL	Audit record items (part 1)																			
			User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[1]	Object name
Object operation events	Operations	CALL	Y	Y	Y	Y	ACS	CAL	Y	' '	S	S	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		DELETE (static SQL)	Y	Y	Y	Y	ACS	DEL	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	DEL	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	DEL	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	DEL	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
		EXECUTE	Depends on the pre-processed SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.																			
		EXECUTE IMMEDIATE	Depends on the specified SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.																			
		INSERT (static SQL)	Y	Y	Y	Y	ACS	INS	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	INS	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	INS	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y
			Y	Y	Y	Y	ACS	INS	Y	' '	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL		Audit record items (part 1)																				
				User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success/failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[1]	Object name
Object operation events (cont'd)	Operations (cont'd)	LOCK TABLE (shared) (static SQL)		Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		LOCK TABLE (locked) (static SQL)		Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		OPEN	SELECT[3]	Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		CLOSE	SELECT	Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		PREPARE	ASSIGN LIST (Format 1, Format 2)	Y	Y	Y	Y	ACS	ASN	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			DELETE	Y	Y	Y	Y	ACS	DEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	DEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	DEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	DEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			INSERT	Y	Y	Y	Y	ACS	INS	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	INS	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	INS	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	INS	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			LOCK TABLE (shared)	Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			LOCK TABLE (locked)	Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	LCK	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			UPDATE	Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			PURGE TABLE	Y	Y	Y	Y	ACS	PRG	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	PRG	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
			SELECT[3]	Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y

Event		Command/ SQL		Audit record items (part 1)																			
				User identifier	Event execution date	Event execution time	Event execution time (microseconds)	Event type	Event subtype	Event success / failure	Privilege used	UAP name	Service name	IP address	Process ID	Thread ID	Host name	Unit identifier	Server name	Connect number	SQL number	Object owner[1]	Object name
Object operation events (cont'd)	Operations (cont'd)	PURGE (static SQL)		Y	Y	Y	Y	ACS	PRG	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	PRG	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		UPDATE (static SQL)		Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	UPD	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
		SELECT, 1-line SELECT statement (static SQL)[3]		Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
				Y	Y	Y	Y	ACS	SEL	Y	'ΔΔΔ'	Y	S	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
Utility operation events	Creation	pdload[4]	By table	Y	Y	Y	Y	UTL	LOD	Y	'ΔΔΔ'	"pdload"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By RDAREA	Y	Y	Y	Y	UTL	LOD	Y	'ΔΔΔ'	"pdload"	—	—	Y	—	Y	—	—	—	—	Y	Y
		pdexp pddefrev	By table	Y	Y	Y	Y	UTL	EXP	Y	'ΔΔΔ'	"pdexp"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By view table	Y	Y	Y	Y	UTL	EXP	Y	'ΔΔΔ'	"pdexp"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By alias table	Y	Y	Y	Y	UTL	EXP	Y	'ΔΔΔ'	"pdexp"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By procedure	Y	Y	Y	Y	UTL	EXP	Y	'ΔΔΔ'	"pdexp"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By trigger	Y	Y	Y	Y	UTL	EXP	Y	'ΔΔΔ'	"pdexp"	—	—	Y	—	Y	—	—	—	—	Y	Y
	Application	pdrorg	By schema and table	Y	Y	Y	Y	UTL	ORG	Y	'ΔΔΔ'	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By index	Y	Y	Y	Y	UTL	ORG	Y	'ΔΔΔ'	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
			By RDAREA	Y	Y	Y	Y	UTL	ORG	Y	'ΔΔΔ'	"pdrorg"	—	—	Y	—	Y	—	—	—	—	Y	Y
		pdconstck		Y	Y	Y	Y	UTL	CST	Y	'ΔΔΔ'	"pdconstck"	—	—	Y	—	Y	—	—	—	—	Y	Y

■ Audit trail record items at event termination (part 2)

Event		Command/ SQL	Audit record items (part 2)													Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification, or identifier of user targeted by event	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
System administrator security events	HiRDB start	pdstart	—	—	—	Y	E	Y	—	—	—	—	—	—	—	1	—	Unit in which pdstart command was executed
	HiRDB stop	pdstop	—	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	Unit in which pdstop command was executed
	Auditor registration	pdmod	—	AUD	Y	—	E	Y	—	—	—	—	—	—	—	AUD + ATB	—	DS
	Audit trail table creation	pdmod	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
	Audit trail file deletion	pdaudrm	AUF	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	MGR
	Audit trail collection start	pdaudbegin	—	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	Each unit
		HiRDB start	—	—	—	—	E	Y	—	—	—	—	—	—	—		—	Each active unit
	Audit trail collection termination	pdaudend	—	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	Each unit
		HiRDB termination	—	—	—	—	E	Y	—	—	—	—	—	—	—			Each terminated unit
	Audit trail file overwriting	Automatic overwriting by the system	AUF	—	—	—	E	Y	Y	Y	—	—	—	—	—	1	—	Each unit
		Manual overwriting by pdaudswap	AUF	—	—	—	E	Y	Y	Y	—	—	—	—	—			

Event		Command/ SQL	Audit record items (part 2)													Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification, or identifier of user targeted by event	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
System administrator security events (cont'd)	Transition to consecutive certification failure account lock state	Connection failure	—	—	Y	—	E	Y	—	—	—	—	—	—	—	1	—	FES
	Release of consecutive certification failure account lock state	Connection after account lock period has passed	—	—	Y	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		DROP CONNECTION SECURITY FOR CONNECT	—	—	Y	—	E	Y	—	—	—	—	—	—	—	USR		DS
		pdacunlck	—	—	Y	—	E	Y	—	—	—	—	—	—	—			
	Transition to password-invalid account lock state		—	—	Y	—	E	Y	—	—	—	—	—	—	—	USR	—	DS
	Release from password-invalid account lock state		—	—	Y	—	E	Y	—	—	—	—	—	—	—			
	Setting value modification for connection security facility	CREATE CONNECTION SECURITY, DROP CONNECTION SECURITY	—	—	—	—	E	Y	—	—	Y	Y	Y	—	—	SPR	—	FES
	Execution of pdacunlck		—	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	DS
Auditor security events	Data loading in audit trail table	pdload	TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y	1	—	See section *22.14*
	Swapping of audit trail files	pdaudswap	AUF	—	—	—	E	Y	Y	Y	—	—	—	—	—	1		MGR
	Audited event definition	CREATE AUDIT	—	—	—	—	E	Y	—	—	—	—	—	—	—	1		FES
	Audited event deletion	DROP AUDIT	—	—	—	—	E	Y	—	—	—	—	—	—	—	1		
	Auditor password change	GRANT AUDIT	—	AUD	Y	—	E	Y	—	—	Y	—	—	—	—	1		
Session security events	Connection to HiRDB	CONNECT	—	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
	User change	SET SESSION AUTHORIZATION	—	—	—	—	E	Y	—	—	—	—	—	—	—			
Privilege control events	Privilege granting	GRANT DBA	—	DBA	Y	—	E	Y	—	—	Y	—	—	—	—	USR	—	FES
		GRANT SCHEMA	—	SCH	Y	—	E	Y	—	—	—	—	—	—	—			
		GRANT CONNECT	—	CNT	Y	—	E	Y	—	—	Y	—	—	—	—			
		GRANT RDAREA	RDA	RDA	Y	—	E	Y	—	—	—	—	—	—	—	USR × RDA		
		GRANT *access-privilege*	FTB	Y	Y	—	E	Y	—	—	—	—	—	—	—	USR × ACS	—	FES
			TBL	Y	Y	—	E	Y	—	—	—	—	—	Y	—			
			VIW	Y	Y	—	E	Y	—	—	—	—	—	Y	—			
	Privilege revocation	REVOKE DBA	—	DBA	Y	—	E	Y	—	—	—	—	—	—	—	USR	—	FES
		REVOKE SCHEMA	—	SCH	Y	—	E	Y	—	—	—	—	—	—	—			
		REVOKE CONNECT	—	CNT	Y	—	E	Y	—	—	—	—	—	—	—			
		REVOKE RDAREA	RDA	RDA	Y	—	E	Y	—	—	—	—	—	—	—	USR × RDA		
		REVOKE *access-privilege*	FTB	Y	Y	—	E	Y	—	—	—	—	—	—	—	USR × ACS	DROP VIEW	FES
			TBL	Y	Y	—	E	Y	—	—	—	—	—	Y	—			
			VIW	Y	Y	—	E	Y	—	—	—	—	—	Y	—			

Event		Command/ SQL	Audit record items (part 2)													Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification, or identifier of user targeted by event	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
Object definition events	ALTER	ALTER PROCEDURE	PRC	—	—	—	E	Y	—	—	—	—	—	—	—	PRC	—	FES
		ALTER ROUTINE	PRC	—	—	—	E	Y	—	—	—	—	—	—	—	1		
		ALTER TABLE (other than ADD RDAREA and CHANGE RDAREA)	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—	TBL	DROP VIEW, DROP INDEX, DROP TRIGGER, DROP PUBLIC VIEW	FES
		ALTER TABLE ADD RDAREA, ALTER TABLE CHANGE RDAREA	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—	TBL	—	FES
		ALTER TRIGGER	TRG	—	—	—	E	Y	—	—	—	—	—	—	—	1		
		COMMENT	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
			VW	—	—	—	E	Y	—	—	—	—	—	Y	—			
			FTB	—	—	—	E	Y	—	—	—	—	—	—	—			
	CREATE	CREATE ALIAS	ALS	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		CREATE FOREIGN INDEX	FID	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE FOREIGN TABLE	FTB	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE FUNCTION	FNC	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE INDEX (Format 1)	IDX	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE INDEX (Format 2)	IDX	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE PROCEDURE	PRC	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE SCHEMA	SCH	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE SERVER	FSV	—	—	—	E	Y	—	—	—	—	—	—	—			
		CREATE TABLE	TBL	—	—	—	E	Y	—	—	—	—	—	—	—	1	CREATE TRIGGER (when generating a trigger for performing a constraint action)	FES
		CREATE TRIGGER	TRG	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		CREATE TYPE	TYP	—	—	—	E	Y	—	—	—	—	—	—	—	1	· CREATE PROCEDURE (when there is a member procedure) · CREATE FUNCTION (when there is a member function)	FES
		CREATE USER MAPPING[2]	USM	—	Y	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		CREATE VIEW, CREATE PUBLIC VIEW	VW	—	—	—	E	Y	—	—	—	—	—	Y	—			

Event		Command/ SQL	Audit record items (part 2)													Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
			Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification, or identifier of user targeted by event	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
Object definition events (cont'd)	DROP	DROP ALIAS	ALS	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		DROP DATA TYPE	TYP	—	—	—	E	Y	—	—	—	—	—	—	—	1	DROP FUNCTION, DROP PROCEDURE	FES
		DROP FOREIGN INDEX	FID	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		DROP FOREIGN TABLE	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	1	DROP VIEW, DROP FOREIGN INDEX, DROP PUBLIC VIEW	FES
		DROP FUNCTION	FNC	—	—	—	E	Y	—	—	—	—	—	—	—	1	DROP DATA TYPE	FES
		DROP INDEX	IDX	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	
		DROP PROCEDURE	PRC	—	—	—	E	Y	—	—	—	—	—	—	—	1	DROP DATA TYPE	
		DROP SCHEMA	SCH	—	—	—	E	Y	—	—	—	—	—	—	—	1	DROP TABLE, DROP VIEW, DROP FOREIGN TABLE, DROP INDEX, DROP FOREIGN INDEX, DROP PROCEDURE, DROP FUNCTION, DROP DATA TYPE, DROP TRIGGER, DROP PUBLIC VIEW	FES
		DROP SERVER	FSV	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		DROP TABLE	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—	1	DROP VIEW, DROP INDEX, DROP TRIGGER, DROP PROCEDURE, DROP PUBLIC VIEW	FES
		DROP TRIGGER	TRG	—	—	—	E	Y	—	—	—	—	—	—	—	1	DROP PROCEDURE	FES
		DROP USER MAPPING[2]	USM	—	PUBLIC	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		DROP VIEW, DROP PUBLIC VIEW	VIW	—	—	—	E	Y	—	—	—	—	—	Y	—	1	DROP VIEW, DROP PUBLIC VIEW	FES
Object operation events	Operations	CALL	PRC	—	—	—	E	Y	—	—	—	—	—	—	—	1	—	FES
		DELETE (static SQL)	FTB	—	—	—	E	Y	—	—	—	—	—	—	Y	1	· SELECT · Trigger SQL statement	FES
			TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
			VIW	—	—	—	E	Y	—	—	—	—	—	Y	Y			
			ALS	—	—	—	E	Y	—	—	—	—	—	—	Y			
		EXECUTE	Depends on the pre-processed SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.														Pre-processed SQL	FES
		EXECUTE IMMEDIATE	Depends on the specified SQL statement. Data manipulation SQL and control SQL statements output the same audit trail as is output when PREPARE is executed for the SQL statement. Definition SQL statements output the same audit trail as is output by the definition SQL statement of an object definition event.														Specified SQL	FES
		INSERT (static SQL)	FTB	—	—	—	E	Y	—	—	—	—	—	—	Y	1	· SELECT · Trigger SQL statement	FES
			TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
			VIW	—	—	—	E	Y	—	—	—	—	—	Y	Y			
			ALS	—	—	—	E	Y	—	—	—	—	—	—	Y			

Event		Command/ SQL		Audit record items (part 2)															SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
				Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification, or identifier of user targeted by event	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count	Number of records		
Object operation events (cont'd)	Operations (cont'd)	LOCK TABLE (shared) (static SQL)		FTB	—	—	—	E	Y	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
		LOCK TABLE (locked) (static SQL)		FTB	—	—	—	E	Y	—	—	—	—	—	—	—	TBL		
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
		OPEN	SELECT[3]	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	TBL		
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
				LST	—	—	—	E	Y	—	—	—	—	—	—	—	1		
		CLOSE	SELECT	FTB	—	—	—	E	Y	—	—	—	—	—	—	Y	TBL		
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	Y			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	Y			
				LST	—	—	—	E	Y	—	—	—	—	—	—	Y	1		
		PREPARE	ASSIGN LIST (Format 1, Format 2)	LST	—	—	—	E	Y	—	—	—	—	—	Y	—	1		
			DELETE	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	1	PREPARE(SELECT)	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
			INSERT	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	1	PREPARE(SELECT)	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
			LOCK TABLE (shared)	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	TBL	—	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
			LOCK TABLE (locked)	FTB	—	—	—	E	Y	—	—	—	—	—	—	—			
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
			UPDATE	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	1	PREPARE(SELECT)	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
			PURGE TABLE	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—	1	—	FES
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
			SELECT[3]	FTB	—	—	—	E	Y	—	—	—	—	—	—	—	TBL	PREPARE(SELECT)	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	—			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
				LST	—	—	—	E	Y	—	—	—	—	—	Y	—	1		

Event		Command/ SQL		Audit record items (part 2)														Number of records	SQL statement that may be executed as extension	Output unit (HiRDB/ Parallel Server)
				Object type	Privilege granted, revoked, or modified	User identifier of privilege granting, revocation, or modification, or identifier of user targeted by event	Audit-related operand definition value	Audit trail type	SQL code/ termination code	Name of swapping source audit trail file	Name of swapping destination audit trail file	Security facility modification type	Setting value before security facility modification	Setting value after security facility modification	Audit trail table option	Access count			
Object operation events (cont'd)	Operations (cont'd)	PURGE (static SQL)		TBL	—	—	—	E	Y	—	—	—	—	—	Y	—	1	—	FES
				ALS	—	—	—	E	Y	—	—	—	—	—	—	—			
		UPDATE (static SQL)		FTB	—	—	—	E	Y	—	—	—	—	—	—	Y	1	· SELECT · Trigger SQL statement	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	Y			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	Y			
		SELECT, 1-line SELECT statement (static SQL)[3]		FTB	—	—	—	E	Y	—	—	—	—	—	—	Y	TBL	—	FES
				TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
				VIW	—	—	—	E	Y	—	—	—	—	—	Y	Y			
				ALS	—	—	—	E	Y	—	—	—	—	—	—	Y			
Utility operation events	Creation	pdload[4]	By table	TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y	1	—	See section *22.14*
			By RDAREA	TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
		pdexp pddefrev	By table	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—	TBL		
			By view table	VIW	—	—	—	E	Y	—	—	—	—	—	Y	—	VIW		
			By alias table	ALS	—	—	—	E	Y	—	—	—	—	—	—	—	ALS		
			By procedure	PRC	—	—	—	E	Y	—	—	—	—	—	—	—	PRC		
			By trigger	TRG	—	—	—	E	Y	—	—	—	—	—	—	—	TRG		
	Application	pdrorg	By schema and table	TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y	TBL		
			By index	TBL	—	—	—	E	Y	—	—	—	—	—	Y	—			
			By RDAREA	TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y			
		pdconstck		TBL	—	—	—	E	Y	—	—	—	—	—	Y	Y	1		

Legend:

Y: Information is collected.

S: Information is collected sometimes via the XA interface.

—: Information is not collected; or, not applicable.

ACS: Number of privileges specified (4 if all is specified).

ALS: Number of target table aliases.

ATB: Number of audit trail tables created with the `pdmod` command (1 if audit trail tables were created; 0 if audit trail tables were not created).

AUD: Number of auditors created with the `pdmod` command (1 if auditors were created; 0 if auditors were not created).

CLS: `1` if FOR CLUSTER KEY clause is specified; `0` if it is not specified.

FNC: Number of functions to be re-created.

IDX: Number of related indexes.

PRC: Number of procedures to be re-created.

PRM: `1` if FOR PRIMARY KEY clause is specified; `0` if it is not specified.

RDA: Number of RDAREAs specified.

RLB: Number of RDAREAs for LOB data.

RID: Number of index RDAREAs.

SPR: Number of changed connection security facility settings.

TBL: Number of tables specified by a SELECT statement, SELECT clause, or LOCK statement, or the number of tables to be processed by a utility.

TRG: Number of triggers to be re-created.

USR: Number of specified users or groups.

VIW: Number of target view tables.

[1] When the target object is a public view table, `PUBLIC` is output as the object owner.

[2] A foreign server name is output as the user mapping object name. `PUBLIC` is output as the event target user identifier.

[3] If a `WITH` clause is used, the object type is output as follows:

```
WITH Q1(C1) AS (SELECT C1 FROM AAA) SELECT C1 FROM BBB
```

Type	AAA	BBB
Base table	`TBL`	`TBL`
Foreign table	`FTB`	`FTB`
View table	`VIW`	`VIW`
Query name	Not output	Not output

Note

If an error occurs before the type can be determined, `TBL` is output for all items.

[4] Applies to cases in which a table is not or cannot be recognized as an audit trail table.

22.14 Audit trail output destination unit during utility execution (HiRDB/Parallel Server only)

This section explains the output destination unit of an audit trail that is output during utility execution. Tables 22-31 and 22-32 show the output destination units of audit trails that are output during utility execution.

Table 22-31: Audit trail output destination unit during utility execution (Part 1)

Condition following utility execution		Audit trail output destination unit
Normal termination		See *Table 22-32* for the audit trail output destination unit.
Abnormal termination	After utility server startup	
	Before utility server startup	Audit trail is output to the unit where the command was entered.

Table 22-32: Audit trail output destination unit during utility execution (Part 2)

Utility	Execution-target object		Output destination unit	
			At event termination	During privilege checking
pdload	In table units	Base table only	Unit where the input file is located	Unit where the command was entered
		Base table + BLOB		
		BLOB only		
	In RDAREA units	Base table only		
		Base table + BLOB		
		BLOB only		

<table>
<tr><th colspan="5" rowspan="2">Utility</th><th colspan="3" rowspan="2">Execution-target object</th><th colspan="2">Output destination unit</th></tr>
<tr><th>At event termination</th><th>During privilege checking</th></tr>
<tr><td rowspan="6">pdrorg</td><td rowspan="5">-k rorg
-k unld
-k reld</td><td rowspan="3">-g option specified</td><td colspan="3">—</td><td>Unit where the unload file was created</td><td rowspan="8">Unit where the command was entered</td></tr>
<tr><td rowspan="2">-g option not specified</td><td colspan="3">Dictionary table reorganization</td><td>Unit where the dictionary server is located</td></tr>
<tr><td rowspan="3">User table reorganization</td><td colspan="2">Table reorganization</td><td rowspan="2">Unit where the unload file was created</td></tr>
<tr><td rowspan="2">Row-partitioned table</td><td>Reorganization in RDAREA units</td></tr>
<tr><td colspan="2">-g option not specified</td><td>Reorganization in table units</td><td>Unit where the server specified in the first unload or lobunld statement specified is located</td></tr>
<tr><td colspan="2">-k ixrc</td><td colspan="3">—</td><td rowspan="3">Unit where the command was entered</td></tr>
<tr><td colspan="2">-k ixor</td><td colspan="3">—</td></tr>
<tr><td colspan="2">-k ixmk</td><td colspan="3">—</td></tr>
<tr><td colspan="3">pdexp, pddefrev</td><td colspan="3">—</td><td>Unit where the export file is located</td><td>Unit where the export file is located</td></tr>
<tr><td colspan="3">pdconstck</td><td colspan="3">—</td><td>Unit where the command was entered</td><td>Unit where the command was entered</td></tr>
</table>

Legend:

—: Not applicable

22.15 Notes on version upgrading

This section provides notes about version upgrading when the security audit facility is used. For details about version upgrading, see the manual *HiRDB Version 8 Installation and Design Guide*.

(1) When the audit trail format for audit trail files is changed

If version upgrading changes the audit trail format for audit trail files, take the following action before version upgrading:

- Use the `pdaudswap` command to swap audit trail files and use the `pdload` command to register the content of the file before swapping in the audit trail table.

Reference note:

Because there are no triggers for output of an audit trail during HiRDB termination and for swapping audit trail files before HiRDB starts after a version upgrade, an audit trail in the old format collected during HiRDB termination and an audit trail in the new format collected during HiRDB startup will both exist in the same file. For this reason, HiRDB is designed to permit you to load data into the audit trail table even when audit trails in old and new formats coexist in the same file.

(2) When the column structure of the audit trail table is changed

Version upgrading may add columns to the audit trail table. The column structure of the audit trail table can be changed by HiRDB by executing the `pdvrup` command. If data already exists in the audit trail table, NULL values are set in the added columns.

(3) When the number of audit trail events increases

If version upgrading increases the number of audit trail events, the volume of audit trail output may increase. Therefore, before version upgrading, re-estimate the size required on the disk that stores audit trail files.

This is applicable when audit trails are to be output for all events (`CREATE AUDIT FOR ANY` is specified). In such a case, if version upgrading increases the number of new events (such as utility manipulation events), audit trails are also collected for the utility manipulation events.

If you do not wish to collect an audit trail for a newly added audited event, do not use `CREATE AUDIT FOR ANY` when defining audit trail collection. Instead, define individual audit trails (`CREATE AUDIT FOR SESSION`, `CREATE AUDIT FOR PRIVILEGE`, ...).

(4) Notes on version-upgrading failure

Execute the pdload command only after version upgrading succeeds. If this command is executed before version upgrading succeeds, the following problems may arise:

- Upgrading to version 07-02 or later from a version earlier than 07-02 (but later than version 07-00)

  When HiRDB is started after version upgrading, an audit trail in the new format is output to the audit trail file. If the older version is restored from this status, and then the pdload command is used to register an audit trail in the audit trail table, correct operation cannot be guaranteed. In such a case, after the older version has been restored, use the pdfmkfs command to initialize the HiRDB file system area for audit trail files.

- Upgrading from version 07-02 or later

  In 07-02 or later versions, if the format of the audit trail that is output does not match the format supported by the applicable version, the KFPS05753-W message is output and the audit trail is deleted.

(5) Backing up the RDAREAs that store an audit trail table

Before version upgrading, make a backup of the RDAREAs that store an audit trail table. If version upgrading fails and you need to revert to the older version, restore the RDAREAs that store the audit trail table from the backup.

If no backup was made, use DROP TABLE to delete the audit trail table, then use the pdmod command to re-create the audit trail table. In this case, the data stored in the audit trail table is deleted.

Chapter

23. Using the Connection Security Facility

This chapter explains how to use the connection security facility, which is designed to enhance the security of HiRDB systems.

23.1 Overview of the connection security facility

This section provides an overview of the connection security facility. The following items are explained.

- About the connection security facility
- Password character string restrictions
- Limit on the number of consecutive certification failures

23.1.1 About the connection security facility

One of the means of enhancing system security is to use passwords. HiRDB can set up a password for each user. However, if a simple password that can be easily guessed is used (for example, using the person's authorization identifier or birth date as the password), there is a heightened risk that an unauthorized user may be able to guess the password and infiltrate the system. To minimize the possibility of unauthorized use of passwords, Hitachi recommends that you use the connection security facility. Table 23-1 provides an overview of the connection security facility.

Table 23-1: Overview of the connection security facility

Functions	Explanation
Password character string restrictions	You can set up restrictions on the character strings used as passwords. For example, you can prohibit passwords such as `AAAAA` or `zzzzz`. Prohibiting simple passwords enhances password security.
Limit on the number of consecutive certification failures	When a user enters an invalid password, certification of that user fails and the user is not connected to HiRDB. If user certification fails more than a certain number of times in succession, HiRDB can be set to deny to that user the right to connect to HiRDB. For this purpose, a limit is set on the number of consecutive failures permitted to achieve user certification, and anytime a user exceeds the limit, the user is barred from connecting to HiRDB. For example, the limit could be set so that denial of connection rights to HiRDB will be implemented whenever a user enters an invalid password more than three times in a row.

Use of both of these functions makes unauthorized access based on guessing at a password more difficult, resulting in enhanced security.

Note:

You cannot use the Directory Server linkage facility and the connection security facility at the same time. If you use the Directory Server linkage facility, you must disable the settings for the connection security facility.

23.1.2 Password character string restrictions

(1) Restrictions that can be set for passwords

Table 23-2 explains the restrictions that can be set for passwords.

Table 23-2: Restrictions that can be set for passwords

Item	Explanation
Specifiable minimum in bytes	You can specify a minimum number of bytes that must be used for a password.
Prohibition on use of the authorization identifier	You can prohibit use of a person's authorization identifier in his or her password character string.
Prohibition on use of only one type of characters*	You can prohibit the use of only one type of characters in a password, such as only upper-case letters or only lower-case letters.

* The characters that can be specified for passwords can be classified into the following three types:

- Uppercase letters (A-Z, #, @, \)
- Lowercase letters (a-z)
- Numbers (0-9)

Hint:

You cannot specify separate password character string restrictions for different users. The specified password character string restrictions will apply uniformly to all HiRDB users (including users with the DBA privilege and the auditor).

Reference note:

Whether or not this facility is used, the following specification rule applies to passwords:

- A password can consist of alphanumeric characters, but must begin with an alphabetic character.

(2) Effect on existing users

When password character string restrictions are first established, any user whose existing password does not conform to the restrictions is placed in what is called *password-invalid account lock state*. A user placed in this status can no longer connect to HiRDB.

To release a user from password-invalid account lock state, the user's password must be changed. For details about changing a password, see *23.4.1 Releasing individual*

users from password-invalid account lock state.

Before establishing password character string restrictions, you can determine how many users will be placed in password-invalid account lock state because of the restrictions. For details about determining the number of such users, see *23.5 Checking for users who will be placed in password-invalid account lock state*.

(3) Effect on new users

`GRANT DBA`, `GRANT AUDIT`, or `GRANT CONNECT` is used to set a password for a new user. If that password violates a restriction, execution of the `GRANT` statement will not be successful.

(4) Setting method

You use `CREATE CONNECTION SECURITY` to set password character string restrictions. For details about the setting procedure, see *23.2 Setting password character string restrictions*.

For details about password character string restrictions, see sections *23.2* through *23.8* and *23.12*.

23.1.3 Limit on the number of consecutive certification failures

(1) Limits that can be set

When a user enters an invalid password, certification of that user fails and the user is not connected to HiRDB. If user certification fails more than a certain number of times in succession (*permitted number of consecutive certification failures*), HiRDB can be set to deny connection rights to HiRDB to that user.

For example, if the permitted number of consecutive certification failures is set to `3`, a user who fails user certification four times in a row as a result of entering an invalid password is placed in *consecutive certification failure account lock state*. A user who is placed in this status no longer has the right to connect to HiRDB.

Reference note:

You cannot specify separate limits on the permitted number of consecutive certification failures for different users. The specified limit will apply to all HiRDB users (including users with the DBA privilege and the auditor).

You can also specify the period during which a user is to be kept in consecutive certification failure account lock state; this is called the *account lock period*. For example, if the account lock period is set to `1` (hour), a consecutive certification failure account lock state remains in effect for a user for one hour. When the hour has passed, the consecutive certification failure account lock state is cancelled and the user is again permitted to attempt to connect to HiRDB.

Reference note:

- You can also set the account lock period to be permanent (to not expire automatically).
- You can cancel a consecutive certification failure account lock state before the account lock period has expired. For details about the cancellation procedure, see *23.11 Releasing consecutive certification failure account lock state*.

(2) How to count the number of failures

Only entry of up to 30 bytes of an invalid password is counted as a failure. The following situations do not constitute a failure:

- Entry of an invalid authorization identifier (specifying a non-existent authorization identifier).
- Entry of more than 30 bytes for a password.

Reference note:

- The count of consecutive certification failures is not reset to `0` even if the user waits for a while. For example, if a user fails once, then fails again after waiting for an hour, the count of consecutive certification failures increments to `2`.
- The count of consecutive certification failures is not reset to `0` even when HiRDB terminates.
- The count of consecutive certification failures remains valid even if the user attempts to connect from a different client machine. For example, if a user fails once from machine A and then fails from machine B, the count of consecutive certification failures increments to `2`.
- The count of consecutive certification failures remains valid even if the user tries to connect from a different front-end server. For example, if a user fails once from front-end server A and then fails from front-end server B, the count of consecutive certification failures increments to `2`.
- Command and utility executions are also counted.
- When the consecutive certification failure account lock state is cancelled, the count of consecutive certification failures is reset to `0`.

(3) Setting method

You use `CREATE CONNECTION SECURITY` to set the limit on the number of

consecutive certification failures. For details of the setting procedure, see *23.9 Setting and cancelling the limit on number of consecutive certification failures*.

For details about applying the limit on number of consecutive certification failures, see sections *23.9* through *23.12*.

(4) Required RDAREA

When this facility is used, the system-defined `ADD_INTERVAL` scalar function is used to check for the consecutive certification failure account lock state. For this reason, a data dictionary LOB RDAREA is required; if no data dictionary LOB RDAREA is available, you must create one.

23.2 Setting password character string restrictions

Executor: DBA privilege holder

This section explains the procedure for initial set-up of password character string restrictions. You must perform the steps in the order they are shown below, beginning with step (1).

(1) Evaluate the restrictions that can be set for passwords

You should evaluate the restrictions that can be set for passwords. The items you should consider are shown in Table 23-3.

Table 23-3: Restrictions that can be set for passwords

Restriction	Explanation
Specifiable minimum in bytes	Specify in bytes the minimum number of characters that can be used for a password. The specifiable range of the minimum number of characters for a password is between `6` and `15`.
Prohibition on use of the authorization identifier	Specify whether or not inclusion of the person's authorization identifier in the password character string is to be prohibited. If prohibition is specified, the following passwords would be prohibited: Examples of prohibited passwords when the authorization identifier is `K001`: `K001`, `abK001`, `K00165`, `GTK001KL`
Prohibition on use of only one type of characters	Specify whether or not use of only one type of characters for a password is to be prohibited. If prohibition is specified, the following passwords would be prohibited: Examples of prohibited passwords: `HUDGTX`, `jkfgytud`, `D@MK#B\`

(2) Check for users who will be in violation of the specified restrictions

You should check in advance for existing users whose existing password will not conform to the proposed restrictions. Because the nonconforming users will be placed in password-invalid account lock state, they will no longer be permitted to connect to HiRDB. Before establishing restrictions, you should identify the users whose existing password will be in violation of the restrictions. For details about the identification procedure, see *23.5 Checking for users who will be placed in password-invalid account lock state*.

Reference note:

Setting up password character string restrictions may cause some users to be placed in password-invalid account lock state. You should notify all users in advance about the imminent establishment of password restrictions, and inform them that they may have to change their password before the implementation date. Once the implementation date arrives, check for users who are in violation of the restrictions and contact them.

(3) Change passwords

Use a `GRANT` statement to change the password of a user whose existing password does not conform to the password restrictions. Examples follow:

Example 1

Change the password of `USER01` to `f51HD7tc`:

```
GRANT CONNECT TO USER01 IDENTIFIED BY "f51HD7tc"
```

Reference note:

Each user can change his or her own password. Passwords can also be changed by a DBA privilege holder.

Example 2

Change the password of DBA privilege holder `ADMIN01` to `gd4A@sPL`:

```
GRANT DBA TO ADMIN01 IDENTIFIED BY "gd4A@sPL"
```

Example 3

Change the auditor's password to `a0h7Fc3K`:

```
GRANT AUDIT IDENTIFIED BY "a0h7Fc3K"
```

(4) Set the password character string restrictions

Use `CREATE CONNECTION SECURITY` to set the desired password character string restrictions.

Note:

If the password of a DBA privilege holder or of the auditor does not conform to the restrictions (even if that is the only person whose password does not conform), execution of `CREATE CONNECTION SECURITY` will not be successful.

An example of specifying CREATE CONNECTION SECURITY follows:

Example

The following password character string restrictions are to be set:

- The minimum number of bytes for a password is to be set to 8.
- Inclusion of the authorization identifier in the password is to be prohibited.
- Use of only one type of characters in a password is to be prohibited.

```
CREATE CONNECTION SECURITY
    FOR PASSWORD
      MIN LENGTH 8                  ...1
      USER IDENTIFIER RESTRICT      ...2
      SIMILAR RESTRICT              ...3
```

Explanation

1. Sets that each password must be at least 8 bytes in length.
2. Prohibits inclusion of the user's authorization identifier in his or her password. To prohibit, specify RESTRICT; to not prohibit, specify UNRESTRICT.
3. Prohibits use of only one type of characters in a password. To prohibit, specify RESTRICT; to not prohibit, specify UNRESTRICT.

(5) Check for users in password-invalid account lock state

Check for users in password-invalid account lock state. For the checking procedure, see *23.4.1(1) Check for users in password-invalid account lock state*.

23.3 Changing a password character string restriction

This section explains the procedure for changing a password character string restriction.

23.3.1 Special notes on changing password character string restrictions

(1) Relaxing a password character string restriction

When there are users in password-invalid account lock state, relaxing a password character string restriction may release some users from the password-invalid account lock state. Before changing password character string restriction, make sure it is acceptable to release these users from the password-invalid account lock state.

The following is the procedure for identifying the users who will be released from the password-invalid account lock state:

Procedure

1. Check for users in the password-invalid account lock state. For the checking procedure, see *23.4.1(1) Check for users in password-invalid account lock state*.
2. Perform an advance check of the restriction that is being considered. For the procedure for checking in advance, see *23.5 Checking for users who will be placed in password-invalid account lock state*.
3. Identify the applicable users based on the differences in the results from steps 1 and 2.

(2) Strengthening one restriction while relaxing another

Care must be exercised in strengthening one restriction while relaxing another restriction. For example, a user may attempt to change a password before the implementation date in order to avoid violation of a new restriction. However, the user may not be able to make the change because it violates the current restriction. In such a case, the password must be changed so that it satisfies both the current and the proposed restrictions.

23.3.2 Procedure for changing a password character string restriction

Executor: DBA privilege holder

This section explains the procedure for changing a password character string restriction. You must perform the steps in order they are shown below, beginning with step (1).

(1) Check the restrictions that are currently in force

Information on the password character string restrictions that have been specified is stored in the SQL_SYSPARMS dictionary table. To determine the restrictions that are currently in effect, search SQL_SYSPARMS and check the specification information on password character string restrictions. An example of such a search follows:

Example

Check the password restrictions that are currently in effect:

```
SELECT FUNCTION_KEY,PARAM_KEY,INT_VALUE,CHAR_VALUE
    FROM MASTER.SQL_SYSPARAMS
      WHERE FUNCTION_KEY='PASSWORD'
```

Execution results

```
FUNCTION_KEY   PARAM_KEY        INT_VALUE  CHAR_VALUE
-------------------------------------------
PASSWORD       MIN_LENGTH       8          8
PASSWORD       USER_IDENTIFIER  NULL       RESTRICT
PASSWORD       SIMILAR          NULL       RESTRICT
```

Explanation

The following settings are in effect:

- Minimum length (number of bytes) for a password: 8
- Inclusion of the authorization identifier in the password: Prohibited (RESTRICT)
- Use of only one type of characters in a password: Prohibited (RESTRICT)

If not prohibited, UNRESTRICT is displayed.

(2) Evaluate the restrictions to be set for passwords

Evaluate the password restrictions you intend to set or change. For details, see Table *23-3 Restrictions that can be set for passwords*.

(3) Check for users who will be in violation of the specified restrictions

Check in advance for existing users whose existing password will not conform to the proposed new set of restrictions. Because the nonconforming users will be placed in password-invalid account lock state, they will no longer be permitted to connect to HiRDB. Before establishing a restriction, you should identify the users whose existing password will be in violation of the restriction. For details about the identification procedure, see *23.5 Checking for users who will be placed in password-invalid account lock state*.

Reference note:

Setting up or changing a password character string restriction may cause some users to be placed in password-invalid account lock state. You should notify all users in advance about the imminent change in password restrictions and inform them that they may have to change their password before the implementation date. Once the implementation date arrives, check for users who are in violation of the new set of restrictions and contact them.

(4) Change passwords

Use a GRANT statement to change the password of a user whose existing password does not conform to the new set of password restrictions. Examples follow:

Example 1

Change the password of USER01 to f51HD7tc:

```
GRANT CONNECT TO USER01 IDENTIFIED BY "f51HD7tc"
```

Reference note:

Each user can change his or her own password. Passwords can also be changed by a DBA privilege holder.

Example 2

Change the password of DBA privilege holder ADMIN01 to gd4A@sPL:

```
GRANT DBA TO ADMIN01 IDENTIFIED BY "gd4A@sPL"
```

Example 3

Change the auditor's password to a0h7Fc3K:

```
GRANT AUDIT IDENTIFIED BY "a0h7Fc3K"
```

(5) Change the password character string restrictions

The procedure for changing the password character string restrictions follows:

Procedure

1. Use DROP CONNECTION SECURITY to cancel the password character string restrictions. For the cancellation procedure, see *23.7 Cancelling the password character string restrictions*.
2. Use CREATE CONNECTION SECURITY to specify the new password

character string restrictions.

Reference note:

Even if you are not changing all the password character string restrictions (for example, you wish to change only the minimum number of bytes for a password), you must first use `DROP CONNECTION SECURITY` to cancel all the password character string restrictions, then use `CREATE CONNECTION SECURITY` to specify the new set of restrictions.

(6) Check for users in password-invalid account lock state

Check for users in password-invalid account lock state. For the checking procedure, see *23.4.1(1) Check for users in password-invalid account lock state*.

23.4 Releasing the password-invalid account lock state

This section explains how to release the password-invalid account lock state.

23.4.1 Releasing individual users from password-invalid account lock state

Executor: DBA privilege holder

This section explains the procedure for releasing individual users from password-invalid account lock state. You must perform the steps in order they are shown below, beginning with step (1).

(1) Check for users in password-invalid account lock state

Check for users in password-invalid account lock state. An example follows:

Example

Display the authorization identifiers of users in password-invalid account lock state:

```
SELECT USER_ID
    FROM MASTER.SQL_USERS
      WHERE PWD_LOCK_TIME IS NOT NULL
```

Execution results

```
USER_ID
----------
USER1
USER2
```

Explanation

`USER1` and `USER2` are listed as being in password-invalid account lock state.

Reference note:

If a user is in password-invalid account lock state, the date and time the user was placed in this status are set in the `PWD_LOCK_TIME` column of the `SQL_USERS` dictionary table. If a user is not in password-invalid account lock state, the NULL value is set in the `PWD_LOCK_TIME` column.

(2) Change passwords

Use a `GRANT` statement to change the password of a user who is in password-invalid account lock state. Examples follow:

Example

Change the password of user USER01 to f51HD7tc:

```
GRANT CONNECT TO USER01 IDENTIFIED BY "f51HD7tc"
```

23.4.2 Releasing all users from password-invalid account lock state

Executor: DBA privilege holder

You use DROP CONNECTION SECURITY to cancel the password character string restrictions. This action releases all users from password-invalid account lock state.

Example

Cancel the password character string restrictions and release all users from password-invalid account lock state:

```
DROP CONNECTION SECURITY FOR PASSWORD
```

23.5 Checking for users who will be placed in password-invalid account lock state

Executor: DBA privilege holder

This section explains the procedure for checking in advance for users who will be placed in password-invalid account lock state. You must perform the steps in the order they are shown below, beginning with step (1).

(1) Execute CREATE CONNECTION SECURITY with the TEST option specified

Execute `CREATE CONNECTION SECURITY` with the `TEST` option specified. When the `TEST` option is specified, a *violation type code* is set in the `PASSWORD_TEST` column of the `SQL_USERS` dictionary table on the line for any user who would be in violation of a proposed restriction specified in the same `CREATE CONNECTION SECURITY` statement.

Reference note:

When the `TEST` option is specified, only checking of the proposed password character string restrictions is performed. Violators are not placed in password-invalid account lock state.

The following example checks in advance for users who would be placed in password-invalid account lock state by proposed character string restrictions:

Example

Check for violators if the following character string restrictions were set for passwords:

- Minimum length (number of bytes) for a password: 8
- Inclusion of the authorization identifier in the password: Prohibited
- Use of only one type of characters in a password: Prohibited

```
CREATE CONNECTION SECURITY
    FOR PASSWORD TEST              ...1
      MIN LENGTH 8                 ...2
      USER IDENTIFIER RESTRICT     ...3
      SIMILAR RESTRICT             ...4
```

Explanation

1. Specifies the `TEST` option for checking in advance.
2. The minimum number of bytes permitted for a password would be 8.

3. Inclusion of the user's authorization identifier in that user's password would be prohibited.
4. Use of only one type of characters in a password would be prohibited.

When this SQL statement is executed, the passwords of all users who are registered in SQL_USERS are checked, and a violation type code is set in the PASSWORD_TEST column of SQL_USERS for any user whose password would be in violation of any of the specified proposed restrictions.

Hint:

- If the specification contents of the CREATE CONNECTION SECURITY SQL for an advance check are not the same as the specification contents when the password character string restrictions are actually set, even users who corrected their passwords may be placed in password-invalid account lock state. It is important that the CREATE CONNECTION SECURITY SQL specified for the advance check be the same as the SQL executed for setting the password character string restrictions (other than for the TEST option).
- After the advance check is executed, instruct only the users who will be in violation of the new character string restrictions to change their passwords. Also make sure that the password of any new user who is registered complies with the new password restrictions.

(2) Search the PASSWORD_TEST column

Violation type codes are set in the PASSWORD_TEST column of SQL_USERS. Search the PASSWORD_TEST column to identify users whose existing password will cause them to be placed in password-invalid account lock state. Table 23-4 shows the violation type codes that are set. If there is no violation, the NULL value is set.

Table 23-4: Violation type codes set in the PASSWORD_TEST column

Order number	Item	Violation type code set in the PASSWORD_TEST column
1	Violation of the minimum number of bytes for a password	L
2	Violation of the prohibition on inclusion of the authorization identifier in the password	U
3	Violation of the prohibition on use of only one type of characters in a password	S

Note

If a password violates multiple items, only one violation type code is set, depending on the item order numbers. For example, if a password violates the items with order numbers 1 and 2, `L` (the violation type code for item 1) is set.

Examples of checks for users in password-invalid account lock state are shown below.

Example 1

Obtain a list of users who are in violation of a password character string restriction:

```
SELECT USER_ID
   FROM MASTER.SQL_USERS
     WHERE PASSWORD_TEST IS NOT NULL
```

Execution results

```
USER_ID
-----------
USER1
USER2
USER3
```

Explanation

`USER1`, `USER2`, and `USER3` are in violation of a password character string restriction.

Example 2

Obtain a list of DBA privilege holders and auditors who are in violation of a password character string restriction:

```
SELECT USER_ID
    FROM MASTER.SQL_USERS
      WHERE PASSWORD_TEST IS NOT NULL
        AND (DBA_PRIVILEGE = 'Y' OR AUDIT_PRIVILEGE = 'Y')
```

Execution results

```
USER_ID
-----------
AUDITOR1
DBA1
DBA2
```

Explanation

DBA privilege holders `DBA1` and `DBA2`, as well as the auditor (`AUDITOR1`), are in violation of a password character string restriction.

23.6 Privilege granting or revocation for users in password-invalid account lock state

(1) Granting privileges to users in password-invalid account lock state

The DBA privilege cannot be granted to a user (who has the CONNECT privilege) who is in password-invalid account lock status. You must first release the user from password-invalid account lock state, then you can grant the DBA privilege.

(2) Deleting privileges from users in password-invalid account lock state

You use a REVOKE statement to revoke privileges from users in password-invalid account lock state.

23.7 Cancelling the password character string restrictions

Executor: DBA privilege holder

You use `DROP CONNECTION SECURITY` to cancel the password character string restrictions. This operation returns the system to its status before the password character string restrictions were set.

Example

Cancel the password character string restrictions currently in effect:

```
DROP CONNECTION SECURITY FOR PASSWORD
```

Reference note:

You cannot delete the restrictions individually (for example, you cannot cancel only the setting for the minimum number of bytes for a password).

Note:

Cancelling the password character string restrictions releases the password-invalid account lock state. As a result, any users who had been in this status and who should not be allowed to connect to HiRDB become able to connect. You should first check for the users in password-invalid account lock state.

23.8 Relationships between password character string restrictions and other facilities

This section explains the relationships between password character string restrictions and other facilities.

23.8.1 Notes on using a Directory Server linkage facility

You cannot set up password character string restrictions while you are using a Directory Server linkage facility. If password character string restrictions have already been set up when you begin using a Directory Server linkage facility, the restrictions will no longer be effective.

Before using a Directory Server linkage facility, it is strongly advised that you cancel the password character string restrictions.

Reference note:

If you use a Directory Server linkage facility without cancelling the password character string restrictions, and then stop using the Directory Server linkage facility, the following problems may occur:

- Users who were in violation of the password character string restrictions will not have been placed in password-invalid account lock state.
- Users who are not in violation of the password character string restrictions may be placed in password-invalid account lock state.
- DBA privilege holders and the auditor may be placed in password-invalid account lock state.

23.8.2 Notes on using the security audit facility

You should take note of the following when you use the security audit facility:

- The password character string restrictions are not checked when the `pdmod` command is used to register the auditor. After registering the auditor, it is important that you immediately use a `GRANT AUDIT` statement to change the auditor's password.
- You cannot set a user who is in password-invalid account lock state as the auditor.

23.9 Setting and cancelling the limit on number of consecutive certification failures

This section explains how to set and cancel the limit on the number of consecutive certification failures. The following items are explained:

- Setting a new limit on the number of consecutive certification failures
- Cancelling the limit on the number of consecutive certification failures
- Changing the limit on the number of consecutive certification failures
- Checking the permitted number of consecutive certification failures and the account lock period

23.9.1 Setting a new limit on the number of consecutive certification failures

Use `CREATE CONNECTION SECURITY` to set limits on the number of consecutive certification failures. Examples follow:

Example 1

Set limits on the number for consecutive certification failures as follows:

- Permitted number of consecutive certification failures: 3
- Account lock period: 30 minutes

```
CREATE CONNECTION SECURITY
    FOR CONNECT PERMISSION COUNT 3      ...1
                LOCK 30 MINUTE          ...2
```

Explanation

1. Specifies the permitted number of consecutive certification failures.
2. Specifies an account lock period.

Example 2

Set limits on the number of consecutive certification failures as follows:

- Permitted number of consecutive certification failures: 5
- Account lock period: Permanent

```
CREATE CONNECTION SECURITY
    FOR CONNECT PERMISSION COUNT 5      ...1
                LOCK UNLIMITED          ...2
```

Explanation

1. Specifies the permitted number of consecutive certification failures.
2. Specifies an account lock period.

23.9.2 Cancelling the limit on the number of consecutive certification failures

Use DROP CONNECTION SECURITY to cancel the limit on the number of consecutive certification failures. An example follows:

Example

Cancel the limit on the number of consecutive certification failures.

```
DROP CONNECTION SECURITY
    FOR CONNECT
```

Reference note:

Cancelling the limit on the number of consecutive certification failures releases the consecutive certification failure account lock state.

23.9.3 Changing the limit on the number of consecutive certification failures

To change the limit on the number of consecutive certification failures, first use DROP CONNECTION SECURITY to cancel the existing limit, and then use CREATE CONNECTION SECURITY to set a new limit. An example follows:

Example

Change the limit on the number of consecutive certification failures as follows:

- Permitted number of consecutive certification failures: 5
- Account lock period: 10 hours

```
DROP CONNECTION SECURITY
    FOR CONNECT                          ...1
CREATE CONNECTION SECURITY
    FOR CONNECT PERMISSION COUNT 5       ...2
                LOCK 10 HOUR             ...3
```

Explanation

1. Cancels the existing limit on the number of consecutive certification

failures.

2. Specifies a new permitted number of consecutive certification failures.
3. Specifies an account lock period.

23.9.4 Checking the permitted number of consecutive certification failures and the account lock period

Executor: DBA privilege holder or auditor

To check the permitted number of consecutive certification failures and the account lock period that have been set, you search the SQL_SYSPARAMS dictionary table. An example follows:

Example

Check the permitted number of consecutive certification failures and the account lock period that have been set:

```
SELECT * FROM MASTER.SQL_SYSPARAMS
    WHERE PARAM_KEY='PERMISSION_COUNT'
    OR PARAM_KEY='LOCK_MINUTE'
```

Execution results 1

```
PARAM_KIND           FUNCTION_KEY   PARAM_KEY        INT_VALUE  CHAR_VALUE
-------------------------------------------------------------------------
CONNECTION_SECURITY  CONNECT        PERMISSION_COUNT 2          2
CONNECTION_SECURITY  CONNECT        LOCK_MINUTE      1440       1440
```

Explanation

The permitted number of consecutive certification failures is 2, and the account lock period is 1440 minutes.

The account lock period is stored in SQL_SYSPARAMS as a number of minutes.

Execution result 2

```
PARAM_KIND           FUNCTION_KEY   PARAM_KEY        INT_VALUE  CHAR_VALUE
-------------------------------------------------------------------------
CONNECTION_SECURITY  CONNECT        PERMISSION_COUNT 2          2
CONNECTION_SECURITY  CONNECT        LOCK_MINUTE      NULL       UNLIMITED
```

Explanation

The permitted number of consecutive certification failures is 2, and the account lock period is permanent (UNLIMITED).

23.10 Checking for users in consecutive certification failure account lock state

Executor: DBA privilege holder or auditor

You search a dictionary table to check for users in consecutive certification failure account lock state. Search examples are provided below.

Because the system-defined ADD_INTERVAL scalar function is used, a data dictionary LOB RDAREA is required. If no data dictionary LOB RDAREA is available, you must create one.

Example 1

Display a list of users in consecutive certification failure account lock state (when the account lock period is not set to permanent):

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CAST(CON_LOCK_TIME AS CHAR(19)) >=
  (SELECT MASTER.ADD_INTERVAL(CAST(CURRENT_TIMESTAMP(0) AS
CHAR(19)),-INT_VALUE)
    FROM MASTER.SQL_SYSPARAMS
      WHERE PARAM_KEY = 'LOCK_MINUTE_CODE')
      AND CON_LOCK_TIME IS NOT NULL
```

Execution results

```
USER_ID CON_LOCK_TIME
--------------------
USER1   2005-01-19 11:37:58
USER2   2005-01-19 12:06:11
```

Explanation

USER1 and USER2 are in consecutive certification failure account lock state.

Example 2

Display a list of users in consecutive certification failure account lock state (when the account lock period is set to permanent):

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CON_LOCK_TIME IS NOT NULL
```

Execution results

```
USER_ID CON_LOCK_TIME
--------------------
USER1   2005-01-19 13:17:23
USER2   2005-01-19 13:17:35
```

Explanation

USER1 and USER2 are in consecutive certification failure account lock state.

Example 3

Display a list of users who are not in consecutive certification failure account lock state (when the account lock period is not set to permanent):

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CAST(CON_LOCK_TIME AS CHAR(19)) <
  (SELECT MASTER.ADD_INTERVAL(CAST(CURRENT_TIMESTAMP(0) AS
CHAR(19)),-INT_VALUE)
    FROM MASTER.SQL_SYSPARAMS
      WHERE PARAM_KEY = 'LOCK_MINUTE_CODE')
      OR CON_LOCK_TIME IS NULL
```

Execution results

```
USER_ID  CON_LOCK_TIME
--------------------
USER1    NULL
AUDITOR1 NULL
```

Explanation

USER1 and AUDITOR1 are not in consecutive certification failure account lock state.

Example 4

Display a list of users who are not in consecutive certification failure account lock state (when the account lock period is set to permanent):

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CON_LOCK_TIME IS NULL
```

Execution results

```
USER_ID  CON_LOCK_TIME
--------------------
USER1    NULL
AUDITOR1 NULL
```

Explanation

USER1 and AUDITOR1 are not in consecutive certification failure account lock state.

Note:

If the front-end server and the dictionary server are located in different server machines, you must ensure that the times at the server machines are synchronized. If the times are not synchronized, valid search results may not be obtained.

Reference note:

The date/time a user was placed in consecutive certification failure account lock state is stored in the SQL_USERS dictionary table. This date/time information is retained even after the consecutive certification failure account lock state has been released. The date/time information is cleared the first time user certification (CONNECT) is successful.

23.11 Releasing consecutive certification failure account lock state

Executor: HiRDB administrator

You use the `pdacunlck` command to release the consecutive certification failure account lock state. After checking for users in consecutive certification failure account lock state, you can release those users from consecutive certification failure account lock state. Examples follow:

Example 1

Release user `USER01` from consecutive certification failure account lock state:

```
pdacunlck USER01
```

Example 2

Release users `USER01` and `USER02` from consecutive certification failure account lock state:

```
pdacunlck USER01,USER02
```

Example 3

Release all users from consecutive certification failure account lock state:

```
pdacunlck ALL
```

Reference note:

If you use `DROP CONNECTION SECURITY` to cancel the limit on the number of consecutive certification failures, all users who are currently in consecutive certification failure account lock status are released from that status.

23.12 Notes on using the connection security facility

This section provides notes on using the connection security facility. The following items are explained:

- Releasing a double lock
- Notes on restoring a dictionary RDAREA

23.12.1 Releasing a double lock

It is possible for a user to be placed in both password-invalid account lock state and consecutive certification failure account lock state. This state is called a *double lock*.

To release a double lock, you must release both the password-invalid account lock state and the consecutive certification failure account lock state. The order in which these states are released does not matter.

23.12.2 Notes on restoring a dictionary RDAREA

After a backup of dictionary RDAREAs is made, the password-invalid account lock state or the consecutive certification failure account lock state may change. Therefore, when you restore dictionary RDAREAs, you should use both the backup of the dictionary RDAREAs and the unload log file to restore the dictionary RDAREAs to their most recent status.

Chapter

24. Using the Directory Server Linkage Facility

This chapter explains the environment setup and operating procedures for the Directory Server linkage facility.

This chapter contains the following sections:

24.1 Overview of the Directory Server linkage facility

This section provides an overview of the Directory Server linkage facility. This section covers the following topics:

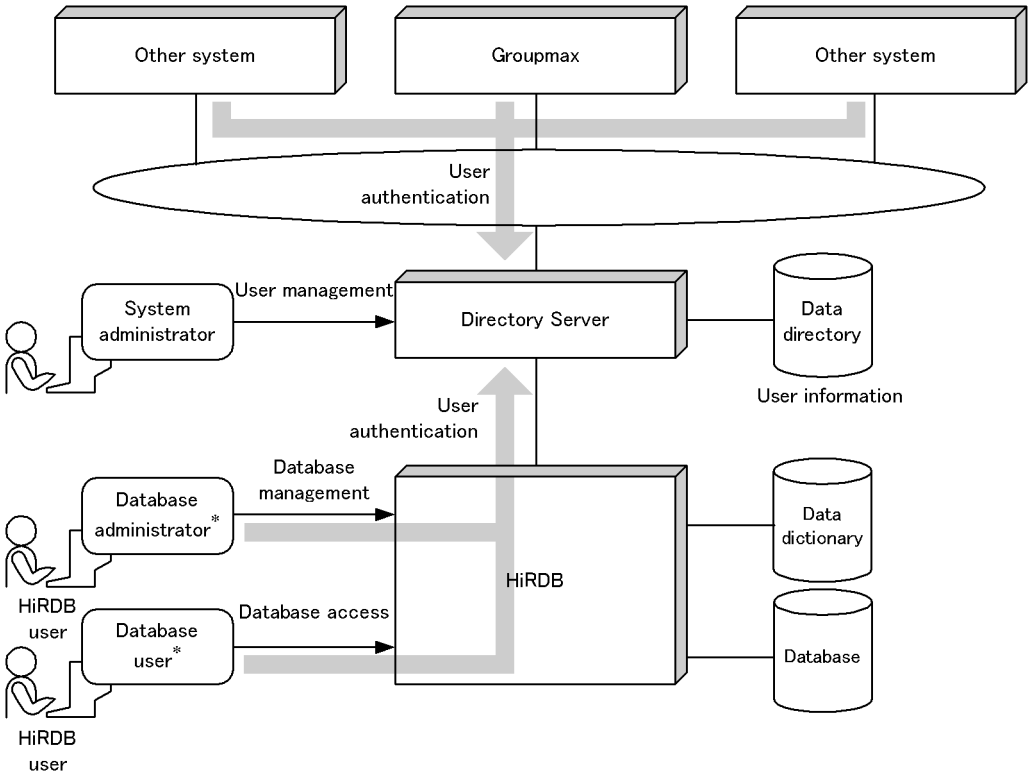
- About the Directory Server linkage facility
- Directory Servers that can be linked
- Capabilities of the Directory Server linkage facility

24.1.1 About the Directory Server linkage facility

The *Directory Server linkage facility* makes it possible to use a Directory Server to manage and authenticate HiRDB users. A *Directory Server* is a program that uses a release type protocol called LDAP to provide service through the Internet or through a distributed system environment such as an intranet. This service, called the *Directory Service*, uses centralized management to reduce the workload of the system administrator.

Use of a Directory Server makes it possible to centrally manage organizational and user information (such as user IDs, passwords, departments, job titles) that traditionally has been managed separately on a variety of systems such as HiRDB or Groupmax. A Directory Server also makes it possible to reference and update the user information from multiple work sites. Figure 24-1 provides an overview of the Directory Server linkage facility.

Figure 24-1: Overview of the Directory Server linkage facility



* Directory Server performs user authentication for users who wish to access HiRDB. User information (user IDs and passwords) must be registered into Directory Server in advance.

24.1.2 Directory servers that can be linked

HiRDB can link with Sun Java System Directory Server. This is referred to as the *Sun Java System Directory Server linkage facility* or, simply, as the directory server linkage facility.

Required knowledge

The descriptions in this manual assume that the reader is familiar with the following:

- Directory Servers
- HiRDB security facility

For details about Directory Servers, see *Sun Java System Directory Server*. For

details about the HiRDB security facility, see *2. Security Definition*.

Conditions

- The following are the requirements (applicable OSs) for using the directory server linkage facility:
  - HP-UX
  - Solaris 8
  - AIX 5L

  The HiRDB must be a POSIX library version that operates in the 32-bit mode. When you set up the HiRDB environment, you use the `pdsetup` command to select the POSIX library version.
- The HiRDB LDAP Option must be installed to use the Sun Java System Directory Server linkage facility.

24.1.3 Capabilities of the Directory Server linkage facility

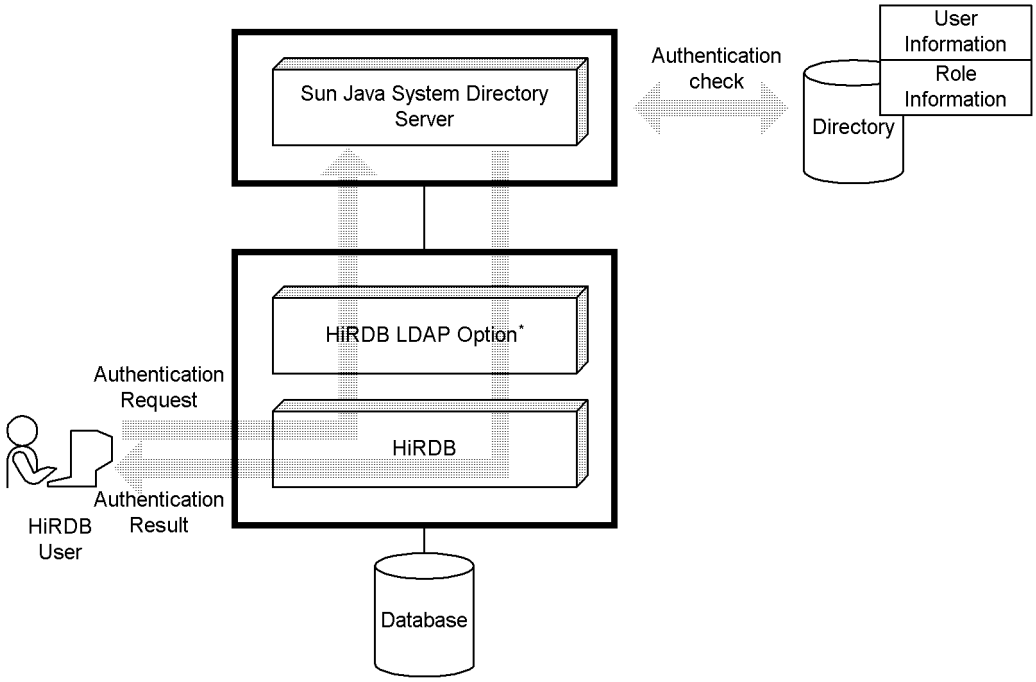
(1) Centralized management by the Directory Server of users connected to HiRDB

Directory Server provides centralized management of the CONNECT privilege information that was previously managed by HiRDB and performs user authentication whenever a user attempts to connect to HiRDB.

Traditionally, the CONNECT privilege has been granted by the DBA privilege holder to those users who need it. When the Directory Server linkage facility is used, there is no need to grant the CONNECT privilege to users. When user information (user IDs and passwords) is registered into Directory Server, the CONNECT privilege is granted automatically to all the registered users.

HiRDB manages DBA privileges, audit privileges, schema definition privileges, RDAREA usage privileges, and table access privileges. Figures 24-2 provide overviews of user authentication.

Figure 24-2: Overview of user authentication (for the Sun Java System Directory Server linkage facility)



Explanation

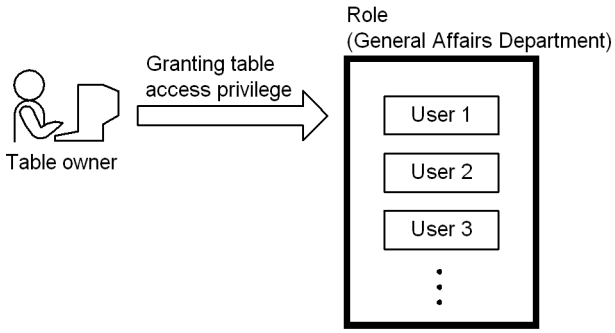
When a user attempts to connect to HiRDB, Sun Java System Directory Server performs user authentication. If the user's ID and password have been registered in Sun Java System Directory Server, the user is permitted to connect to HiRDB.

(2) Granting of table access to roles

The Sun Java System Directory Server employs the concept of roles. Groups of people based on job titles or departments are registered in a Directory Server as separate roles. Then, by granting table access privileges to a role, the administrator can grant table access privileges to all users who belong to that role. The administrator can manage separately the table access privileges for the various roles. Figure 24-3 shows the granting of table access privileges to a role.

To provide table access to a role, the administrator must use a role name, which the Sun Java System Directory Server applies as a filter.

Figure 24-3: Granting table access to a role



Explanation

If the owner of a table grants access to a role (for example, to the General Affairs Department), all users in the General Affairs Department will be able to access that table.

24.2 System configuration

24.2.1 Software configuration

The conditions of HiRDB LDAP Option determine the operating system and product that are required. The following products are required to run the Sun Java System Directory Server linkage facility:

- Sun Java System Directory Server
- iPlanet Console (Sun ONE Console) *
- HiRDB LDAP Option

* Needed when a graphical user interface is used to register directory information such as user names.

Notes

Hitachi recommends caution when both of the following conditions apply:

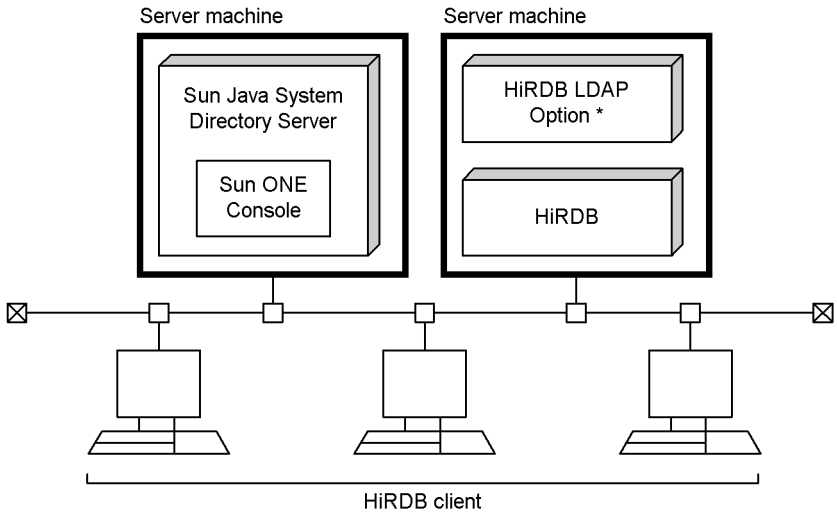
- Privileges are granted to a role
- DABroker (a version earlier than 02-06) is used to acquire table access privilege information

In such a case, users cannot access the table access privilege information granted to their role. If 02-06 or a later version of DABroker is used, users can access the table access privilege information.

24.2.2 Example system configurations

Figure 24-4 shows an example of a system configuration (for a HiRDB/Single Server) when the Sun Java System Directory Server linkage facility is used. You must install HiRDB LDAP Option on the server machine where the HiRDB/Single Server is located.

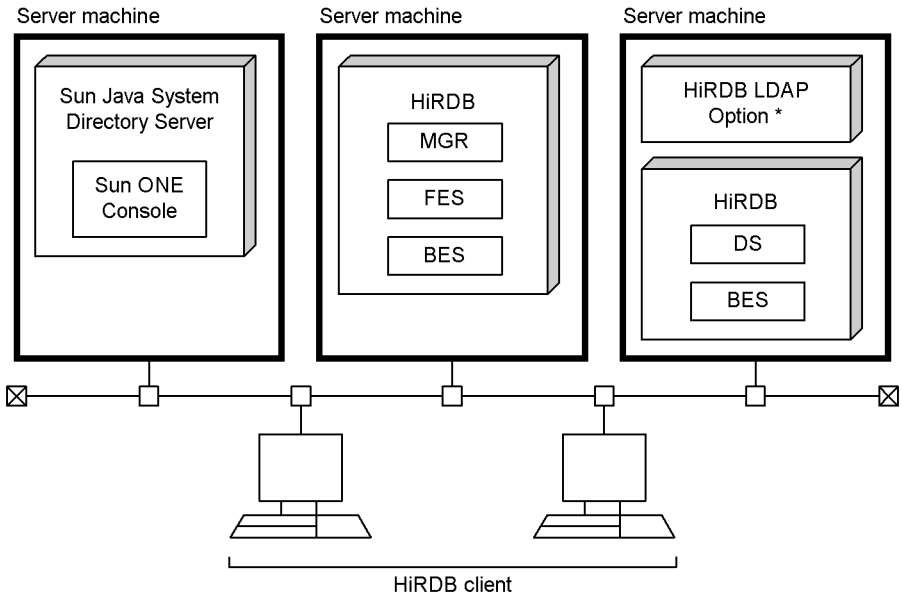
Figure 24-4: Example system configuration using the Sun Java System Directory Server linkage facility (for a HiRDB/Single Server)



* Sun ONE Directory Runtime is included with HiRDB LDAP Option.

Figure 24-5 shows an example of a system configuration (for a HiRDB/Parallel Server) when the Sun Java System Directory Server linkage facility is used. You must install HiRDB LDAP Option on the server machine where the dictionary server is located.

Figure 24-5: Example system configuration using Sun Java System Directory Server linkage facility (for a HiRDB/Parallel Server)



* Sun ONE Directory Runtime is included with HiRDB LDAP Option.

24.3 Environment setup

This section explains the environment setup for the Directory Server linkage facility.

24.3.1 Notes on HiRDB environment setup

To use the Directory Server Linkage Facility, the POSIX version must be used. To use the POSIX library version, specify the `-l` option in the `pdsetup` command that is executed during environment setup for HiRDB. When HiRDB is first installed, the `-l` option should be specified in the `pdsetup` command.

If HiRDB is already running but the POSIX library version is not being used, use the following procedure to change HiRDB to the POSIX library version:

Procedure

1. Use `pdstop` command to terminate HiRDB normally.
2. Use the `pdsetup -d` command to delete HiRDB from the OS; choose `y` in response to the message. In the case of a HiRDB/Parallel Server, execute the `pdsetup -d` command at all server machines.
3. Execute `pdsetup` command with `POSIX` specified in the `-l` option. In the case of a HiRDB/Parallel Server, execute the `pdsetup -l` command at all server machines.
4. Use the `pdstart` command to start HiRDB normally.

HiRDB Version 5.0 and older versions do not support the POSIX library version. Therefore, when HiRDB is upgraded from Version 5.0 or older, use this procedure to change HiRDB to the POSIX library version.

If the `-l` option was not specified in the `pdsetup` command when a new HiRDB was installed, use this procedure to change HiRDB to the POSIX library version.

24.3.2 Procedure for setting up environment for Directory Server linkage facility

This section explains the procedure for setting up an environment for the Directory Server linkage facility. The following assumes that the environment for HiRDB is already set up and HiRDB is ready to be started.

Procedure

1. Install the Directory Server.
2. Users and roles must be registered into the Sun Java System Directory Server.
3. Terminate HiRDB.

4. Set up the environment for HiRDB LDAP Option.
5. Specify the `pd_directory_server` operand.
6. Grant privileges to users.[1]
7. Delete unneeded CONNECT privileges.[2]

The procedure step numbers correspond to the paragraph numbers in the explanation that follows. For example, step 3 above is explained in paragraph (3) below.

[1] Perform this step when you install HiRDB at a server for the first time.

[2] Perform this step when HiRDB is already running.

(1) Install the Directory Server

Install the Directory Server. For details about installing the Sun Java System Directory Server, see *Sun Java System Directory Server*.

(2) Register users and roles in the Directory Server

Register in the Sun Java System Directory Server the users who wish to connect to HiRDB. The HiRDB administrator must also be registered. Defined roles can also be registered. For details about registering users or roles into the Sun Java System Directory Server, see *Sun Java System Directory Server*.

(a) Notes on registering users

1. Be sure to specify the users' passwords when you register users in the Directory Server. You can register users without specifying passwords, but such users will not be able to connect to HiRDB.
2. User IDs must conform to the naming conventions for HiRDB authorization identifiers (i.e., a user ID must consist of up to eight bytes of upper- and/or lower-case alphabetic characters and/or numeric digits).
3. A user ID that is the same as a HiRDB reserved word cannot be used for connecting to HiRDB (for a list of the reserved words, see the manual *HiRDB Version 8 SQL Reference*).
4. All user IDs must be unique.
5. Note that user and role information is case sensitive. For details, see *24.3.3 Handling upper-case and lower-case letters specified in user IDs, passwords, and roles*.

(b) Notes on registering roles

1. Register roles that use a filter.
2. Role names must conform to the naming conventions for HiRDB. A role name

must consist of up to 30 bytes of upper- and/or lower-case alphabetic characters and/or numeric digits. Double-byte characters are not allowed.

3. If a role name is the same as a HiRDB reserved word, table access privileges cannot be granted to that group. For a list of reserved words, see the manual *HiRDB Version 8 SQL Reference*.
4. Check that no registered role name is the same as an existing role name or user ID in the Sun Java System Directory Server.
5. All role names must be unique.
6. Note that user and role information is case sensitive. For details, see *24.3.3 Handling upper-case and lower-case letters specified in user IDs, passwords, and roles*.

(c) When HiRDB is already running

The users registered into HiRDB can be identified by referencing the `SQL_USERS` dictionary table; an example follows:

Example

Display the authorization identifiers of all users registered in HiRDB:

```
SELECT USER_ID FROM MASTER.SQL_USERS
```

(3) Terminate HiRDB

Use the `pdstop` command to terminate HiRDB normally.

(4) Set up the environment for HiRDB LDAP Option

Install HiRDB LDAP Option and execute the `pdopsetup` command. For details about installation and executing the `pdopsetup` command, see the manual *HiRDB Version 8 Installation and Design Guide*.

After installation is complete, create a HiRDB LDAP Option environment definition file. For details about HiRDB LDAP Option environment definition files, see *24.7 Creating the HiRDB LDAP Option environment definition file*.

(5) Specify the pd_directory_server operand

Specify the `pd_directory_server` operand to use the Directory Server linkage facility. Then, use the `pdstart` command to start HiRDB normally.

(6) Grant privileges to the users

This task should be performed when HiRDB is installed for the first time. Grant privileges to the users who are registered in Hitachi Directory Server. For the procedure for granting privileges, see *24.4 User privileges setup*.

(7) Delete unneeded CONNECT privileges

This task should be performed if HiRDB had been installed and operated previously. Because CONNECT privileges are now managed by Hitachi Directory Server, HiRDB's CONNECT privilege management information is no longer needed. Use the REVOKE statement to delete the CONNECT privilege of all users except for the following:

- DBA privilege holders (including the HiRDB administrator)[1]
- Audit privilege holders[1]
- Schema definition privilege holders[2]

The following SQL example retrieves all users who do not have DBA or schema definition privilege (users who have CONNECT privilege only or CONNECT privilege and table access privileges only):

Example

```
SELECT USER_ID FROM MASTER.SQL_USERS
    WHERE DBA_PRIVILEGE = 'N' AND SCHEMA_PRIVILEGE = 'N'
      AND AUDIT_PRIVILEGE <> 'Y'
```

[1] CONNECT privileges of DBA privilege holders and audit privilege holders cannot be deleted.

[2] The CONNECT privilege of a user whose schema exists cannot be revoked. Revoking the CONNECT privilege when there is no schema also revokes the schema definition privilege. If a schema definition privilege is revoked inadvertently, grant the schema definition privilege again. Delete a schema only when it is no longer needed.

Remarks

Although HiRDB can be run without deleting the CONNECT privileges, the CONNECT privilege information will remain in the HiRDB dictionary as unnecessary information that is not used. If unneeded privilege information is left, a different user with the same ID who registers subsequently may manipulate definitions and tables using that privilege. If it is decided not to delete the CONNECT privileges, safeguard against such an occurrence by ensuring that the user information registered in Directory Server matches exactly the privilege information registered in HiRDB.

24.3.3 Handling upper-case and lower-case letters specified in user IDs, passwords, and roles

This section describes the handling of upper-case and lower-case letters specified in user IDs, passwords, and roles.

(1) Differences in handling by HiRDB and the Directory Server

HiRDB does not differentiate between upper-case and lower-case letters in user IDs, passwords, and role names. HiRDB handles lower-case letters as upper-case letters. To differentiate between upper-case and lower-case letters, the string must be enclosed in quotation marks (`"`). For example, `USERA`, `usera`, `userA`, and `USerA` are all regarded as the character string `USERA`. In contrast, `"USERA"`, `"usera"`, `"userA"`, and `"USerA"` are handled as different character strings.

For details with respect to the Sun Java System Directory Server, see *Sun Java System Directory Server*. Table 24-1 describes typical handling of upper-case and lower-case letters by the Directory Servers.

Table 24-1: Handling of upper-case and lower-case letters by the Directory Servers

Item	Handling by Directory Servers
User IDs	Does not differentiate between upper- and lower-case letters. Example: `USERA`, `usera`, `userA`, and `USerA` are all regarded as the same character string.
Passwords	Differentiates between upper- and lower-case letters. Example: `HiRDB`, `HIRDB`, and `hirdb` are regarded as different passwords.
Role name	Does not differentiate between upper- and lower-case letters. Example: `GROUPA`, `groupa`, `GrouPA`, and `gRoUpA` are all regarded as the same character string.

(2) Setting up case sensitivity in a Directory Server

Table 24-2 provides guidelines for setting up case sensitivity in a Directory Server.

Table 24-2: Guidelines for setting up case sensitivity in a Directory Server

Condition		Differentiation between upper- and lower-case letters
When Directory Server is used in HiRDB only	When HiRDB differentiates between upper- and lower-case letters	Set Directory Server to differentiate between upper- and lower-case letters
	When HiRDB does not differentiate between upper- and lower-case letters	Set Directory Server to not differentiate between upper- and lower-case letters
When Directory Server is used in multiple products including HiRDB		Decide whether or not Directory Server is to differentiate between upper- and lower-case letters by considering other products as well. Before changing the handling of upper- and lower-case letters for the sake of HiRDB, ensure that other products will not be affected adversely by the change.

Notes on differentiation between upper- and lower-case letters

1. Suppose that a user called `USERA` is registered in Directory Server. In this event, either USERA or usera can connect to HiRDB. However, if `GRANT SCHEMA TO "usera"` is specified when the schema definition privilege is granted, this SQL will be accepted. However, HiRDB will recognize that the schema definition privilege was granted to usera. Consequently, only usera can use the schema definition privilege, and USERA cannot. A similar situation occurs when DBA privilege or an access privilege is granted.
2. If `GRANT SELECT ON T1 TO GROUP "groupa"` is specified when access privileges are granted to roles, all users belonging to `groupa` as well as users belonging to `GROUPA` will be granted table access privileges.

(3) About HiRDB operation

When specifying user IDs, passwords, or role names in HiRDB, Hitachi recommends specifying in a consistent manner (i.e., using only upper-case letters or enclosing letters in quotation marks).

24.4 User privileges setup

Before setting up user privileges, check that all users have been registered into the Directory Server. Also check that the HiRDB administrator, DBA privilege holders and auditors are also registered. Any users not registered in the Directory Server will not have CONNECT privileges and will not be able to connect to HiRDB.

24.4.1 DBA privilege setup

A DBA privilege holder uses the GRANT statement to grant DBA privileges to other users. DBA privileges cannot be granted to roles. Specify in the GRANT statement's IDENTIFIED BY operand the passwords that are registered in the Directory Server. Check that a password is specified for every user.

When DBA privilege is granted, that information is registered in SQL_USERS.

Example

Grant DBA privilege to the user whose authorization identifier is USR01 and password is HIR01:

```
GRANT DBA TO USR01 IDENTIFIED BY HIR01
```

24.4.2 Auditor privilege setup

Use the pdmod command to register auditors into the Directory Server. Auditors can then use the GRANT AUDIT statement to change their passwords to the passwords registered in the Directory Server.

24.4.3 CONNECT privilege setup

There is no need to grant HiRDB's CONNECT privilege to users. User are not registered in HiRDB's SQL_USERS dictionary table.

Be sure to specify the users' passwords when you registering users in the Directory Server. You can register users without specifying passwords, but such users will not be able to connect to HiRDB.

24.4.4 Schema definition privilege setup

A DBA privilege holder uses the GRANT statement to grant the schema definition privilege to other users. The schema definition privilege cannot be granted to a role. When schema definition privilege is granted, that information is registered in SQL_USERS.

Example

Grant schema definition privilege and RDAREA usage privilege (RDAREA name is RDAREA01) to a user who creates tables (authorization identifier is USR02):

```
GRANT SCHEMA TO USR02
GRANT RDAREA RDAREA01 TO USR02
```

24.4.5 RDAREA usage privilege setup

A DBA privilege holder uses the GRANT statement to grant the RDAREA usage privilege to other users. The RDAREA usage privilege cannot be granted to roles.

Example

Grant schema definition privilege and RDAREA usage privilege (RDAREA name is RDAREA01) to a user who creates tables (authorization identifier is USR02):

```
GRANT SCHEMA TO USR02
GRANT RDAREA RDAREA01 TO USR02
```

24.4.6 Table access privilege setup

The owner of a table uses the GRANT statement to grant access privileges to other users or roles.

(1) Granting the table access privilege to a user

Example

Grant SELECT privilege for a table (*authorization-identifier*.*table-identifier* is USR02.T01) to a user (authorization identifieris USR03):

```
GRANT SELECT ON USR02.T01 TO USR03
```

Note:

If a role name is the same as a user ID and access privileges were granted to both names, it may not be possible to distinguish between the names when acquiring table access privileges information. For details, see *24.5.2 Acquiring table access privileges information*.

(2) Granting table access privileges to a role

Table access is granted to roles that use filters.

Example

Grant SELECT privilege for a table (*authorization-identifier*.*table-identifier* is USER02.T01) to a role (role name is GRP01):

```
GRANT SELECT ON USR02.T01 TO GROUP GRP01
```

Remarks

When the following SQL is executed, the SELECT privilege, INSERT privilege, or UPDATE privilege for a table (T01) is granted to a user (USR01) who belongs to a role (GRP01).

```
GRANT SELECT ON T01 TO PUBLIC
GRANT INSERT ON T01 TO GROUP GRP01
GRANT UPDATE ON T01 TO USR01
```

Note:

- When access privilege for a table is granted to a role, users belonging to that role cannot create a view table for that table.
- Access privilege for a table can be granted to a user or role that is not registered in the Directory Server. However, the individual user or a user belonging to the role cannot access the table.

(3) A user moves to a different role (when granting access privilege for a table to a different role)

When a user moves to a different role, it is not necessary to use the `GRANT` statement or `REVOKE` statement to change access privilege for a table. For example, when a user (`USR01`) moves from the Finance Department to General Administration, simply change the department name in the user's user information registered in Directory Server and execute the `pdgrprfl` command. `USR01` will then use the table access privileges granted to the General Administration group and will no longer be able to use the table access privileges granted to the Finance Department group.

If table access privileges have been granted directly to the user, it is necessary to reevaluate those table access privileges (i.e., it may be necessary to revoke some table access privileges).

24.5 Operating procedures

24.5.1 Adding, modifying, or deleting a user or role

(1) Deleting a user or role

When you delete a user who is already registered in the Directory Server, you must also delete the privileges granted to that user. When you delete a role, you must also delete the access privileges for tables that were granted to that role.

(2) Execute the pdgrprfl command

To add, modify, or delete a user or role already registered in the Directory Server, execute the `pdgrprfl` command. Then, execute the `pdgrprfl` command to refresh the user and role information stored in HiRDB LDAP Option and HiRDB.

If the `pdgrprfl` command is not executed after adding, modifying, or deleting a user or role, this user information will not match the user information stored in the Directory Server. If the `pdgrprfl` command cannot always be executed immediately after updating Directory Server information, you should set the command to execute at a regular interval (such as once every few hours or once a day).

Update the user or role information that is stored in HiRDB LDAP Option, or HiRDB when you start HiRDB (this is the same as executing the `pdgrprfl` command).

(3) A user is deleted accidentally

If a user is deleted accidentally, the privilege information granted to that user remains in HiRDB. Register that user in the Directory Server again and the user will have access to all privileges previously granted.

(4) Check to see if user information is registered in Directory Server

Execute the `pdusrchk` command to check for user information in the Directory Server.

24.5.2 Acquiring table access privileges information

Search the `SQL_TABLE_PRIVILEGES` dictionary table to acquire access privileges information for a table. See the following example.

Example

```
SELECT TABLE_SCHEMA, TABLE_NAME, GRANTOR, GRANTEE,
SELECT_PRIVILEGE,
    INSERT_PRIVILEGE,DELETE_PRIVILEGE, UPDATE_PRIVILEGE,
GRANTEE_TYPE
  FROM MASTER.SQL_TABLE_PRIVILEGES
       WHERE TABLE_SCHEMA <> 'HiRDB'
         AND ((GRANTEE_TYPE IS NULL AND GRANTEE IN
```

```
(USER,'PUBLIC'))
         OR (GRANTEE_TYPE='G' AND
           IS_USER_CONTAINED_IN_HDS_GROUP(GRANTEE) IS TRUE))
```

Note: `IS_USER_CONTAINED_IN_HDS_GROUP` is a scalar function.

Reference note:

- If a role name is the same as a user ID and table access privileges were granted to both names, it may not be possible to distinguish between the names when acquiring table access privileges information. To distinguish between the two names, either execute the SQL such as in the example above or use DABroker (version 02-06 or later) to acquire the table access privileges information.
- When you use DABroker (a version earlier than 02-06), table access privileges information granted to user roles cannot be acquired.

24.5.3 Suspending the Directory Server linkage facility

The procedure for suspending the Directory Server linkage facility is explained below.

Procedure

1. If table access privileges have been granted to roles, revoke the privileges for the roles while using the Directory Server linkage facility.
2. Use the `pdstop` command to normally terminate HiRDB.[1]
3. Delete the `pd_directory_server` operand and suspend usage of the Directory Server linkage facility. Once this operand has been deleted, privileges for roles cannot be revoked.[2]
4. Use the `pdstart` command to start HiRDB. Use HiRDB to check user privileges.
5. Register in HiRDB all the necessary user privileges.[3]
6. Delete HiRDB user information from the Directory Server as necessary.

[1] The system reconfiguration command (`pcdhgconf` command) makes it possible to modify HiRDB system definitions while HiRDB is operating. In such a case, it is not necessary to terminate HiRDB. Note that HiRDB Advanced High Availability must be installed in order to use this command. For details about modifying HiRDB system definitions while HiRDB is operating, see *9.2 Modifying HiRDB system definitions while HiRDB is running (system reconfiguration command)*.

[2] If the pd_directory_server operand is deleted without deleting the privileges for roles, the following problems will occur:

- If a role uses the same name as a user ID and access privileges were granted to both names, it may not be possible to distinguish between the names when acquiring table access privileges information. For details, see *24.5.2 Acquiring table access privileges information*.

[3] User privileges must be registered for the following users:

- Users with the DBA privilege (including the HiRDB administrator)

  Users have already been registered into HiRDB. Registering users is not necessary unless they were deleted while HiRDB was operating. However, if a user's password was not specified when DBA privileges were granted, that user will not be able to use the granted DBA privilege. If this situation occurs, use the GRANT DBA statement or GRANT CONNECT statement to register the user's password.

- Users with the schema definition privilege

  Users have already been registered into HiRDB. Registering users is not necessary unless they were deleted while HiRDB was operating. However, if a user who does not have DBA privilege was granted schema definition privilege, no password is registered for that user. If this situation occurs, use the GRANT CONNECT statement to register the user's password.

- Users with the audit privilege

  Users have already been registered into HiRDB. Registering users is not necessary unless they were deleted while HiRDB was operating.

- All other users

  If CONNECT privileges was not registered into HiRDB for some users, register this privilege into HiRDB for these users.

- Accessing tables

  If table access privileges were granted only to the role you belong to, you will not have table access privileges and will not be able to access any tables. If this situation occurs, request the administrator to grant you table access privileges. Also, you will only be able to use privileges information that was granted to your role. Request the administrator to provide the necessary privileges information to you.

- Incorrect passwords

  If the password registered into HiRDB is incorrect, use the GRANT CONNECT statement to modify the registered password.

24.6 Operations in the event of an error

The operating procedures in the event of an error are explained in this section.

(1) In the event of a HiRDB error

Even if a HiRDB error occurs, it will not affect the Directory Server. Resolve the HiRDB error.

(2) In the event of a Directory Server error

If a Directory Server error occurs, the Directory Server will no longer perform user authentication and users will not be able to connect to HiRDB. For details about resolving errors, see *Sun Java System Directory Server*.

(a) Running HiRDB alone until the error is recovered

The procedure for running only HiRDB until an error in Directory Server is recovered is explained below.

Procedure

To run HiRDB alone:

1. Use `pdstop` command to terminate HiRDB normally.*
2. Delete `pd_directory_server` operand.
3. Use `pdstart` command to start HiRDB normally.
4. Use `GRANT` statement to grant CONNECT privilege to users who will be accessing HiRDB. Specify in the `GRANT` statement user IDs and passwords registered in Directory Server.
5. If table access privileges were granted to a role, use the `GRANT` statement to grant access privileges to the users belonging to that role.

* The system reconfiguration command (`pcdhgconf` command) makes it possible to modify the HiRDB system definition while HiRDB is operating. In such a case, it is not necessary to terminate HiRDB. Note that HiRDB Advanced High Availability must be installed in order to use this command. For details about modifying HiRDB system definitions while HiRDB is operating, see *9.2 Modifying HiRDB system definitions while HiRDB is running (system reconfiguration command)*.

(b) Resuming operation using the Directory Server linkage facility

Explained below is the procedure for resuming operation of the Directory Server linkage facility after recovering from a Directory Server error explained above in (a).

Procedure

To resume operations:

1. Use `pdstop` command to terminate HiRDB normally.*
2. Specify the `pd_directory_server` operand.
3. Use `pdstart` command to start HiRDB normally.
4. Use `REVOKE` statement to revoke CONNECT privileges and access privileges granted in procedure (a).

* The system reconfiguration command (`pcdhgconf` command) makes it possible to modify the HiRDB system definition while HiRDB is operating. In such a case, it is not necessary to terminate HiRDB. Note that HiRDB Advanced High Availability must be installed in order to use this command. For details about modifying HiRDB system definitions while HiRDB is operating, see *9.2 Modifying HiRDB system definitions while HiRDB is running (system reconfiguration command)*.

24.7 Creating the HiRDB LDAP Option environment definition file

You create the HiRDB LDAP Option environment definition file in the path `$PDDIR/hirdb_ldap_sods/conf/pdsodsenv.txt`. The operands specified in the HiRDB LDAP Option environment definition file are explained below.

(1) Operands

1) `LDAPHOST` Host name of the Sun Java System Directory Server

~<identifier> <<localhost>>

Specifies the host name of Sun Java System Directory Server. You must specify this operand.

2) `LDAPPORT` Port number of the Sun Java System Directory Server

~<unsigned-integer> ((1-65535)) <<389>>

Specifies the TCP/IP port number that the Sun Java System Directory Server uses during LDAP communication. You must specify this operand.

3) `BINDDN` Bind DN for searching roles

~<identifier> <<anonymous>>

Specifies the DN in the management registry used for communicating with a remote server.

4) `BINDPASSWORD` Password for searching roles

~<identifier> <<anonymous>>

Specifies the password used when searching roles.

5) `ROLEBASEDN` Base DN for searching roles

~<identifier>

Specifies the base DN to be used for searching roles. You must specify this operand.

6) `ROLESCOPE` {`base`|`one`|`sub`}

Specifies the scope for searching roles. You must specify this operand.

`base`: Search only the layer of the search starting point.

`one`: Search the layer that is directly under the search starting point.

`sub`: Search the layer of the search starting point and all layers under it.

7) `UIDKEY` Attribute to use as the key for searching users

~<user-ID> <<uid>>

Specifies the attribute to be used as the key for searching users. You must specify this operand.

8) USERBASEDN Base DN for searching users

~<identifier>

Specifies base DN to be used for searching users. You must specify this operand.

9) USERSCOPE {base|one|sub}

Specifies the scope for searching users. Be sure to specify this operand.

base: Search only the layer of the search starting point.

one: Search the layer that is directly under the search starting point.

sub: Search the layer of the search starting point and all layers under it.

10) NETWORKTIMELIMIT Monitoring time for communication timeout

~<unsigned-integer>((0, 1-65535)) <<120>> (seconds)

Specifies in seconds the communication timeout value for the Sun Java System Directory Server. Specifying 0 in this operand disables timeout monitoring. Specify the value for this operand as follows:

NETWORKTIMELIMIT-value < *pd_watch_time-value* < *PDCWAITTIME-value*

- NETWORKTIMELIMIT: Time limit for Sun Java System Directory Server communication.
- pd_watch_time: Maximum execution time for SQL (specified in the HiRDB system definition).
- PDCWAITTIME: Maximum wait time for the HiRDB client (specified in the client environment definition).

11) FILTERPREFIX Search filter prefix

~<identifier> <<(&(objectclass=inetOrgPerson))>>

Specifies a prefix for the search filter. The specified character string is prefixed to the search filter used when the Sun Java System Directory Server searches the DNs of user entries.

12) SERCHSUFFIX Search filter suffix

~<identifier> <<) >>

Specifies a suffix for the search filter. The specified character string is suffixed to the search filter used when the Sun Java System Directory Server searches the DNs of user entries.

13) SERCHTIMELIMIT Time limit for searching DNs of user entries

~<unsigned-integer> ((1-999)) <<60>> (seconds)

Specifies in seconds a maximum amount of time for the Sun Java System Directory Server to search the DNs of user entries.

14) RUNTIMEPATH Storage directory name for Sun ONE Directory Runtime

~<path-name> <<$PDDIR/hirdb_ldap_sods/sodruntime>>

Specifies as an absolute path name the storage directory for Sun ONE Directory Runtime. Use the Sun ONE Directory Runtime that is in the path specified by this operand. If no path name is specified, then the path name of Sun ONE Directory Runtime included with HiRDB LDAP Option is assumed.

If you specify a space for the path, an error may result (because such a file does not exist).

(2) Example definition

The following is an example definition of a HiRDB LDAP Option environment definition file:

```
LDAPHOST            host1
LDAPPORT            389
BINDDN              cn=USERA,ou=soft,o=hitachi
BINDPASSWORD        password
ROLEBASEDN          ou=soft,o=hitachi
ROLESCOPE           sub
UIDKEY              uid
USERBASEDN          ou=soft,o=hitachi
USERSCOPE           sub
NETWORKTIMELIMIT    120
FILTERPREFIX        (&(objectclass=inetOrgPerson)
SERCHSUFFIX         )
SERCHTIMELIMIT      60
RUNTIMEPATH
```

(3) Notes

- Press the **Enter** key at the end of each line (including the last line).
- Enter a space or tab between an operand name and its value.
- If the same operand is specified more than once, the most recent specification is effective.

Chapter

25. Using the System Switchover Facility

This chapter describes the environment setup and operation procedures for the system switchover facility. The explanations in this chapter assume a familiarity with cluster software (HA monitor, MC/ServiceGuard, VERITAS Cluster Server, Sun Cluster, HACMP, or ClusterPerfect).

25.1 Overview of the system switchover facility

The objective of a system switchover facility is to improve the reliability and responsiveness of the system being used. This section explains the HiRDB system switchover facility and covers the following topics:

- System switchover facility
- Standby-less system switchover facility
  - Standby-less system switchover (1:1) facility
  - Standby-less system switchover (effects distributed) facility
- Application criteria for system switchover facilities
- Cluster software supported by HiRDB
- Monitor mode and server mode

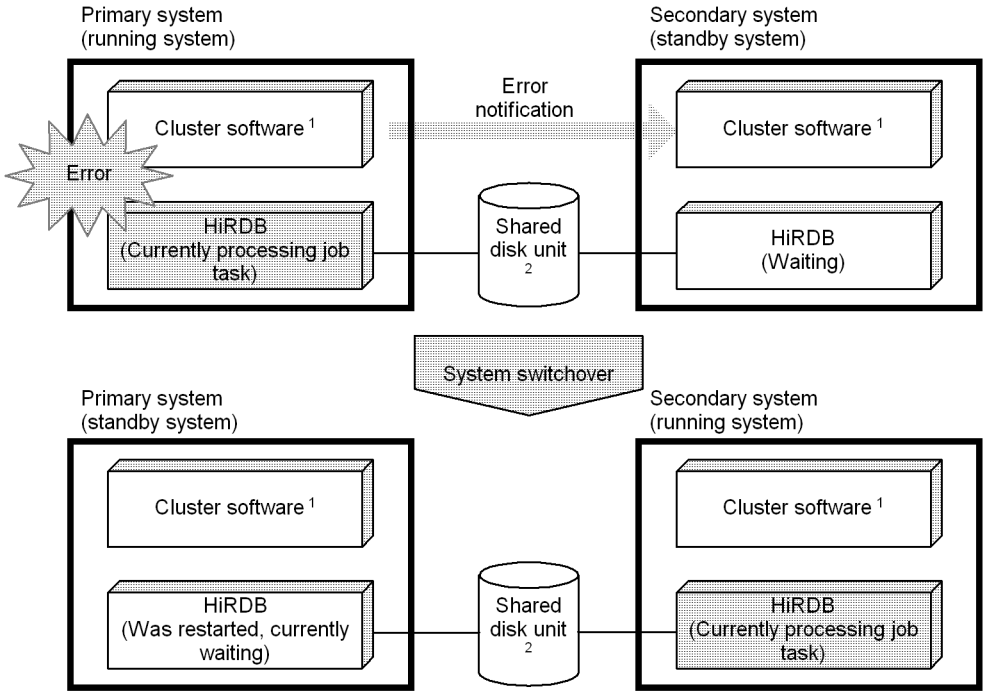
25.1.1 System switchover facility (standby system switchover facility)

If a standby HiRDB is available in addition to the HiRDB that is currently performing work processing and an error occurs on the server machine or HiRDB performing the work processing, the work processing can be switched automatically to the standby HiRDB. The function that switches the system in this manner is called the *system switchover facility (standby system switchover facility)*. Work processing is suspended when an error occurs but before the processing is switched to the standby HiRDB. The purpose of the system switchover facility is to minimize the system down time after an error occurs.

The system switchover facility is used in a cluster system configuration that uses more than one server machine. A HiRDB/Single Server performs system switchover at the system level (note that a utility special unit cannot perform system switchover). A HiRDB/Parallel Server performs system switchover at the unit level.

The system performing the work processing is called the *running system*, and the system that is standing by is called the *standby system*. Each time system switchover is performed, the running system becomes the standby system and the standby system becomes the running system. Also, to differentiate between the two systems during system construction and environment setup, the first system to start as the running system is called the *primary system* and the system that starts initially as the standby system is called the *secondary system*. Although the running system and the standby system change designations when system switchover occurs, the primary system and secondary system do not ever change those designations. Figure 25-1 provides an overview of the system switchover facility (standby system switchover facility).

Figure 25-1: Overview of the system switchover facility (standby system switchover facility)



[1] In this manual, a product that executes system switchover is called *cluster software*. For details about the cluster software supported by HiRDB, see *25.1.4 Cluster software supported by HiRDB*.

[2] For details about shared disk units, see *25.5.2 Preparing a shared disk unit*.

Explanation

If an error occurs on the running system, the standby system is notified of the error, the systems are switched, and the standby system takes over the work processing as the running system.

Notes about using ClusterPerfect

This manual refers to the running system of ClusterPerfect as the primary system and the secondary system of ClusterPerfect as the standby system.

25.1.2 Standby-less system switchover facilities

There are two types of system switchover. One is standby system switchover; the standby system switchover facility was discussed above. The other is standby-less system switchover, which consists of two facilities:

- Standby-less system switchover (1:1) facility
- Standby-less system switchover (effects distributed) facility

A standby-less system switchover facility can be applied only to a HiRDB/Parallel Server's back-end servers; it cannot be applied to a unit that contains servers other than back-end servers.

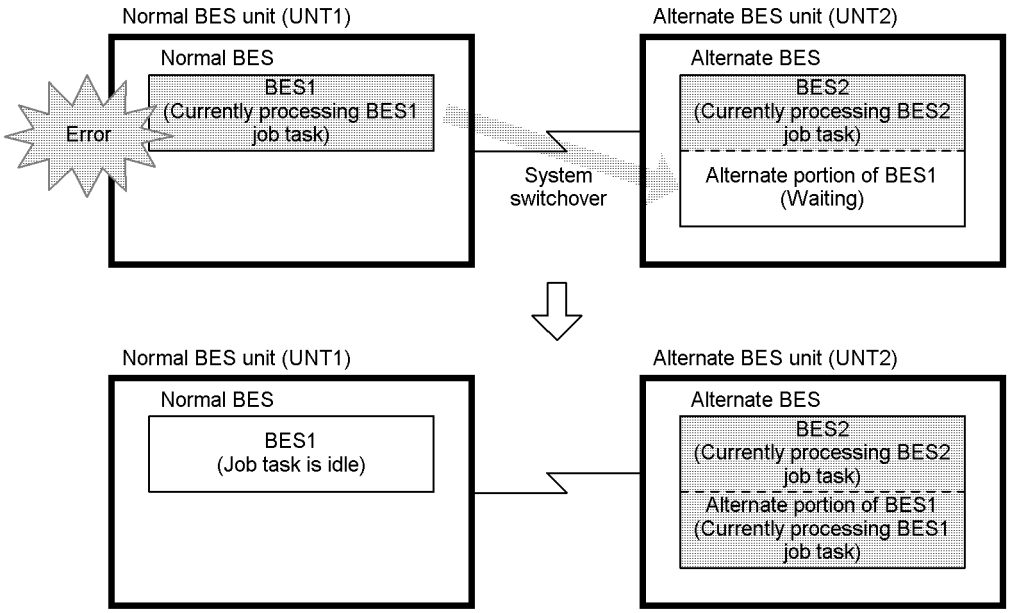
In contrast to the standby system switchover facility, a standby-less system switchover facility does not require that standby system units be prepared. When an error occurs, instead of switching over to a standby system unit, the system is switched over to another unit on the running system so that the work processing is taken over by an active back-end server. This is the function of the standby-less system switchover facilities.

(1) Standby-less system switchover (1:1) facility

The standby-less system switchover (1:1) facility switches from a unit in which an error has occurred to a pre-designated back-end server unit that assumes the processing; i.e., there is a one-to-one relationship between the original unit and the unit to which processing is switched in the event of an error.

A back-end server that releases a process when an error occurs is called a *normal BES*, and a back-end server that takes over the process is called an *alternate BES*. Also, the unit of the normal BESs is called the *normal BES unit*, and the unit of the alternate BESs is called the *alternate BES unit*. Figure 25-2 provides an overview of the standby-less system switchover (1:1) facility.

Figure 25-2: Overview of the standby-less system switchover (1:1) facility



Explanation

- BES1 and BES2 are both usually performing work processing.
- If an error occurs on the normal BES unit (UNT1), the system is switched over and its processing is taken over by the alternate BES. The portion of the processing assumed by the alternate BES is called the *alternate portion*, and the act of taking over that processing by the alternate portion is called *alternating*.
- After the error is resolved, the normal BES unit is started and the processing that the alternate BES assumed is switched back to the normal BES. In this way, this processing returns to normal status; this resumption of the original processing is called *switching the system back*.

Hint:

The concepts of the primary system in the standby system switchover facility and in the standby-less system switchover (1:1) facility are compared below:

- Think of the primary system as the normal BES unit, and think of the secondary system as the alternate BES unit.
- During normal operation, think of the normal BES unit as the running system and think of the alternate portion as the standby system. In the alternating unit after alternation, think of the alternate portion as the running system, and think of the normal BES unit as the standby system.

Reference note:

Because a unit that is running is used as the target for system switchover, a standby server machine is not needed. Therefore, in the case of standby-less system switchover, IP addresses are not transferred.

(a) Conditions

All the following conditions must be satisfied to use the standby-less system switchover (1:1) facility:

- HiRDB Advanced High Availability must be installed.
- Hitachi HA Toolkit Extension must be installed (it is not necessary for the HA monitor cluster software to be installed).
- The system switchover facility must be operating in the server mode.

(b) Advantages of the standby-less system switchover facility

The standby-less system switchover (1:1) facility provides the following advantages over the standby system switchover facility:

- Because it is not necessary to prepare a standby system unit, you use system resources more efficiently. However, when system switchover occurs, the load on the alternate BES increases, which may affect the processing performance of the server.
- Because server processes and the system server are started in advance, the amount of time required to perform system switchover is reduced to approximately the same amount of time used by the rapid system switchover facility. For details about the rapid system switchover facility, see *25.18 Reducing system switchover time (user server hot standby, rapid system switchover facility)*.

Table 25-1 lists the resources that are needed when a standby system unit is standing by and after system switchover is performed.

Table 25-1: Resources needed when a standby unit is standing by and after system switchover is performed

Item		HiRDB system server processes	HiRDB server processes	Shared memory for unit controller	Shared memory for lock pool	Shared memory for global buffer
Standby-less system switchover (1:1) facility		Yes[1]	—[2,3]	Yes[4]	Yes	—[5]
Standby-less system switchover (effects distributed) facility		Δ[6,7]	—[3,8]	Yes[9]	Yes	—[10]
Standby system switchover facility	User server host standby	No	Yes	No	No	No
	Rapid system switchover facility	Yes	Yes	Yes	Yes	Yes
	All others	No	No	No	No	No

Legend:

Yes: Resource is allocated while on standby and is also used after system switchover.

No: Resource is allocated at the time of system switchover, when it becomes the running system.

Δ: Some resources are allocated and used after system switchover, when the standby system becomes the running system.

—: Resource is not secured.

[1] Some processes of system server processing generate processes while they are standing by. Because other system servers share system server processes of the alternate BES unit, no resources are needed specifically for the alternate portion.

[2] The maximum number of back-end server processes is the value for `pd_max_bes_process` of the alternate BES. This value is the sum of the alternating processes and the non-alternating processes. Therefore, only a limited number of users may be able to connect after a system switchover.

[3] If the value of `pd_process_count` (the number of resident processes) and the number of back-end server processes already activated when system switchover was performed is less than the value of `pd_max_bes_process`, additional back-end server processes can be activated. Be sure to set the OS's operating system parameters so there will be enough processes, virtual memory, ports, etc., for the operating system

after system switchover is performed. Note also that activating additional back-end server processes may cause a temporary drop in performance after system switchover has been performed.

[4] Shared memory of the alternate portion is secured when the alternate BES unit starts.

[5] The global buffers used by alternate BESs are shared when alternating processes. Therefore, these buffer are not secured after system switchover occurs. For details about allocation of global buffers during alternating, see *25.5.7 Definition of global buffers (standby-less system switchover (1:1) facility only)*.

[6] Because system server processes are shared on a unit-by-unit basis with the accepting units, no resources are required exclusively for the guest areas.

[7] A system server process for a back-end server generates a process when it becomes the running system.

[8] The maximum permissible number of HiRDB server (back-end server) processes in a unit after system switchover can normally be defined as the combined total of the number of processes for each back-end server and the number of processes for the guests (`pd_ha_max_server_process`).

[9] When an accepting unit is started, shared memory is allocated for the guest areas.

[10] Shared when the global buffer normally used by the back-end server is shared with the accepting unit. Therefore, it is not allocated after system switchover. For details about sharing a global buffer, see *25.5.8 Definition of global buffers (standby-less system switchover (effects distributed) facility only)*.

For details about a back-end server's resource usage status when the standby-less system switchover (effects distributed) facility is used, see *25.1.2(2) Standby-less system switchover (effects distributed) facility*.

(c) Rules for defining normal BES units and alternate BES units

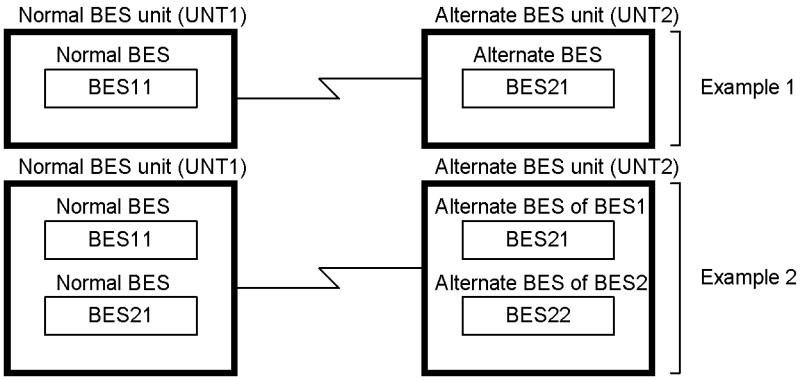
The rules for defining normal BES units and alternate BES units are explained below.

- Only a back-end server can be defined for a normal BES unit or alternate BES unit. If another type of server is defined, the standby-less system switchover facility cannot be applied to that unit.
- There must be one normal BES unit for every alternate BES unit.
- There must be one normal BES for every alternate BES.
- You can define more than one normal BES in a normal BES unit. When you do this, be sure to also define the same number alternate BESs in the alternate BES unit.

Figure 25-3 shows examples of valid configurations of a normal BES unit and alternate

BES unit. Figure 25-4 shows examples of invalid configurations.

Figure 25-3: Examples of valid configurations of a normal BES unit and an alternate BES unit



An alternate BES is defined with the `-c` option in the `pdstart` operand. Example specifications of the `pdstart` operand are shown in Examples 1 and 2 below.

Example 1

```
pdstart -t BES -s bes11 -u UNT1 -c bes21
pdstart -t BES -s bes21 -u UNT2
```

Explanation

`-s bes11`: Specifies a normal BES.

`-c bes21`: Specifies an alternate BES.

Example 2

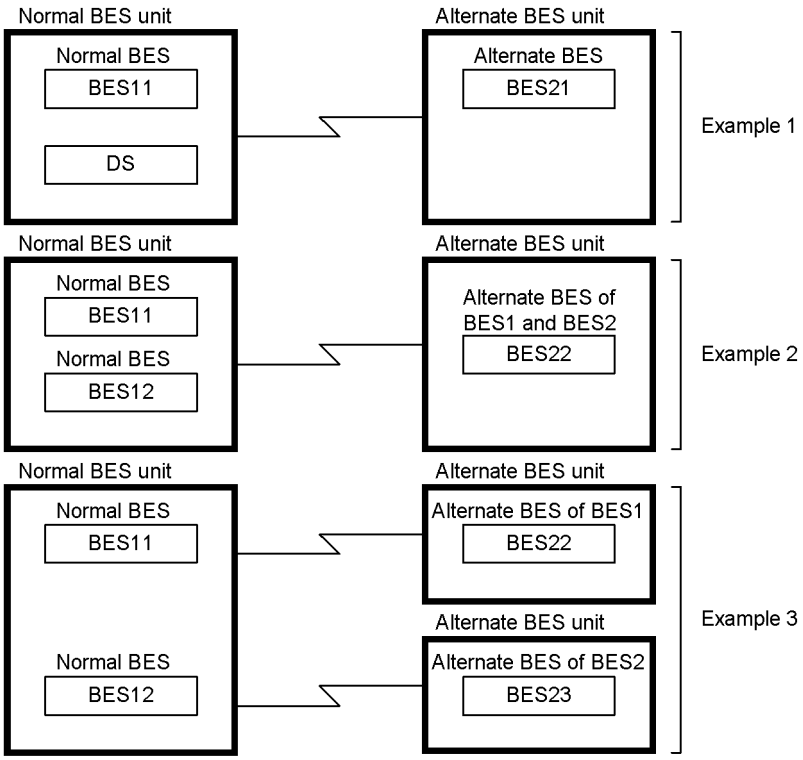
```
pdstart -t BES -s bes11 -u UNT1 -c bes21
pdstart -t BES -s bes12 -u UNT1 -c bes22
pdstart -t BES -s bes21 -u UNT2
pdstart -t BES -s bes22 -u UNT2
```

Explanation

`-s bes11, -s bes12`: Specifies a normal BES.

`-c bes21, -c bes22`: Specifies an alternate BES.

Figure 25-4: Examples of invalid configurations of a normal BES unit and an alternate BES unit



(2) Standby-less system switchover (effects distributed) facility

(a) Overview

When an error occurs, the standby-less system switchover (effects distributed) facility distributes processing requests intended for the back-end servers in the unit where the error occurred to multiple running units, where these processing requests can be executed. The standby-less system switchover (effects distributed) facility does not require standby server machines or standby units, and thus uses system resources more efficiently. After an error occurs, the processing workload increases at each unit that assumes server processing for the failed node; as a result, transaction-processing performance may be impacted negatively. However, because the processing requests intended for the servers in the failed unit are shared and executed by multiple units, the additional load per unit is kept low and degradation of system performance is minimized.

The standby-less system switchover (effects distributed) facility switches over back-end servers by distributing them, and the switchover destinations can be

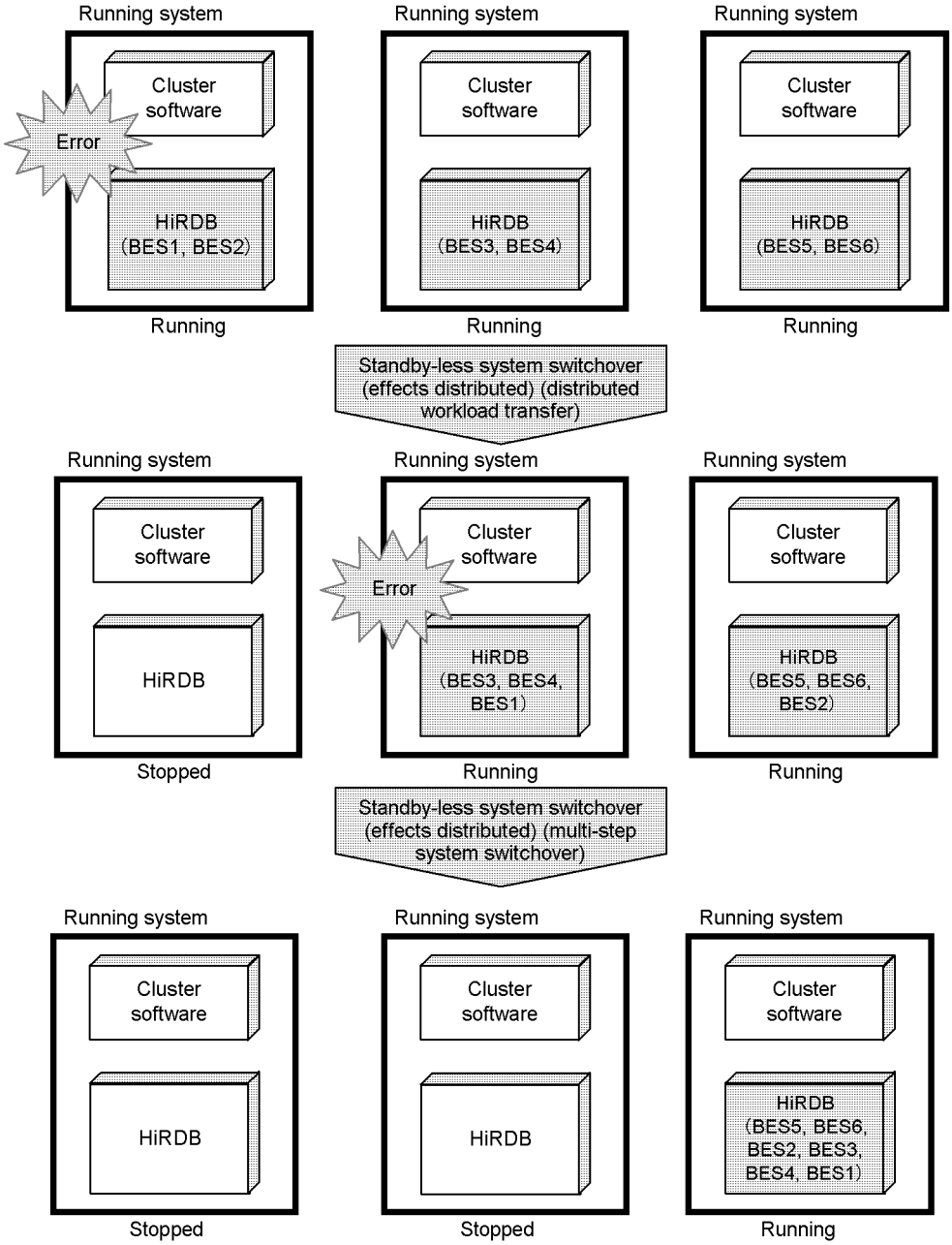
distributed among multiple units. Moreover, if an error occurs in a unit to which the original unit was switched, switching can be performed again to other running units, where processing can be continued; this is called *multi-step system switchover*. Multi-step system switchover cannot be performed in a system that uses the standby-less system switchover (1:1) facility; if an error occurs at a unit to which processing was switched in the case of the standby-less system switchover (1:1) facility, processing for the failed unit cannot be assumed and continued elsewhere.

The standby-less system switchover (effects distributed) facility is appropriate for a system whose resources must always be used efficiently and in which performance degradation because of an error must be minimized.

In the standby-less system switchover (effects distributed) facility, a back-end server defined in the original unit is called a *host BES*, and a back-end server that is accepted by another unit is called a *guest BES*. The unit where the host BESs are defined is called the *regular unit*, and the unit where a guest BES is located is called the *accepting unit*. All accepting units must be defined as an *HA group*. The back-end server resources that correspond to a guest BES constitute a *guest area*.

Figure 25-5 provides an overview of the standby-less system switchover (effects distributed) facility (distributed workload transfer and multi-step system switchover).

Figure 25-5: Overview of the standby-less system switchover (effects distributed) facility (distributed workload transfer and multi-step system switchover)



(b) Conditions

All the following conditions must be satisfied to use the standby-less system switchover (effects distributed) facility:

- HiRDB Advanced High Availability must be installed.
- Hitachi HA Toolkit Extension must be installed (not required if the cluster software is HA monitor).
- The system switchover facility must be used in the server mode.
- The standby-less system switchover (effects distributed) facility is applicable to back-end server units that consist only of back-end servers.
- A unit to which the standby-less system switchover (effects distributed) facility is to be applied must consist of one or more primary system back-end servers. This facility cannot be applied to an accepting-only unit.

(c) Resource usage status

Table 25-2 shows the usage status of back-end server resources when the standby-less system switchover (effects distributed) facility is being applied.

Table 25-2: Usage status of back-end server resources when the standby-less system switchover (effects distributed) facility is applied

Back-end server type	Back-end server status	Resource usage status
Host BES	Accepting status	An area of the size required by the back-end server's definition is created.
	Running	An area of the size required by the back-end server's definition is used.
Guest BES	Accepting status	For each resource, a guest area of the largest resource size is created in the guest server.
	Running	Within the prepared guest area, an area that matches the size required by the back-end server's definition is used.

(d) Operation of the standby-less system switchover (effects distributed) facility

When the standby-less system switchover (effects distributed) facility is used and an error occurs in a regular unit, that unit's primary BESs are moved automatically to various accepting units where they execute their processing as guest BESs. If a BES at the unit where the error occurs is itself a guest BES, it also is moved automatically to an accepting unit where it continues to execute processing as a guest BES. As is the case with the standby system switchover facility, no intervention is required from the HiRDB administrator.

Table 25-3 lists the various types of errors that can occur and whether or not system switchover occurs when standby-less system switchover (effects distributed) is used.

Table 25-3: System switchover depending on error cause when standby-less system switchover (effects distributed) facility is used

Unit's status	Starting or Stopping	Running	
Server's status	**Starting or Stopping**	**Starting or Stopping**	**Running**
Slow-down detected	Not applicable	Unit terminates abnormally. System switchover occurs.	Unit terminates abnormally. System switchover occurs.
System log full	Not applicable	Unit terminates abnormally. System switchover does not occur.	Unit terminates abnormally. System switchover does not occur.
Database path error	Not applicable	Unit terminates abnormally. System switchover occurs (only the first time).	Unit terminates abnormally. System switchover occurs (only the first time).
Back-end server terminated forcibly	Back-end server terminates abnormally. System switchover does not occur.	Back-end server terminates abnormally. System switchover does not occur.	Back-end server terminates abnormally. System switchover does not occur.
System terminated forcibly	Unit terminates abnormally. System switchover does not occur.	Unit terminates abnormally. System switchover does not occur.	Unit terminates abnormally. System switchover does not occur.
System failure	Unit terminates abnormally. System switchover does not occur.	Unit terminates abnormally. System switchover does not occur.	Unit terminates abnormally terminated System switchover occurs.

In the event of system switchover, the host BESs and any guest BESs that are running in the unit are switched over to other units. The back-end servers may be switched to different destinations.

The standby-less system switchover (effects distributed) facility switches systems automatically when various types of errors occur. If an error occurs in an accepting unit after an error had occurred in a regular unit, the back-end servers of the primary system and the guest BESs running in the failed accepting unit move to remaining running units and execute their processing as guest BESs; no intervention is required from the HiRDB administrator. The move destination of each back-end server is determined by the HA monitor definition (cluster software definition when Hitachi HA Toolkit Extension is used).

When a unit runs out of its free guest area, the standby-less system switchover (effects distributed) facility cancels the accepting status of all guest BESs that are not running.

The acceptability of a guest area is not affected by the operation of the host BES. Table 25-4 shows automatic cancellation and resetting of acceptability depending on the free space in the guest area.

When acceptability is reset automatically, all servers that are acting as running systems in other units within the HA group enter accepting status. During this process, even those back-end servers whose acceptability was stopped intentionally by entry of a command (`monsbystp` or `pdstop -q -s` *back-end-server-name*) also become accepting. If the number of BESs that can be accepted within an HA group is exceeded, resulting in reduced-mode operation, any server that is stopped is not returned to accepting status.

Table 25-4: Automatic cancellation and resetting of acceptability depending on the free space in the guest area

Unused guest area in the unit	Guest BES acceptability	
	Guest BESs active in other units	Guest BESs inactive in other units
Disappeared	Cancelled automatically	No change (being cancelled)
Generated	Reset automatically	No change

Note

This table excludes the situation in which the `monsbystp` command or the `pdstop` command (`pdstop -u` *accepting-unit-ID* `-s` *server-ID*) was used to cancel acceptability intentionally.

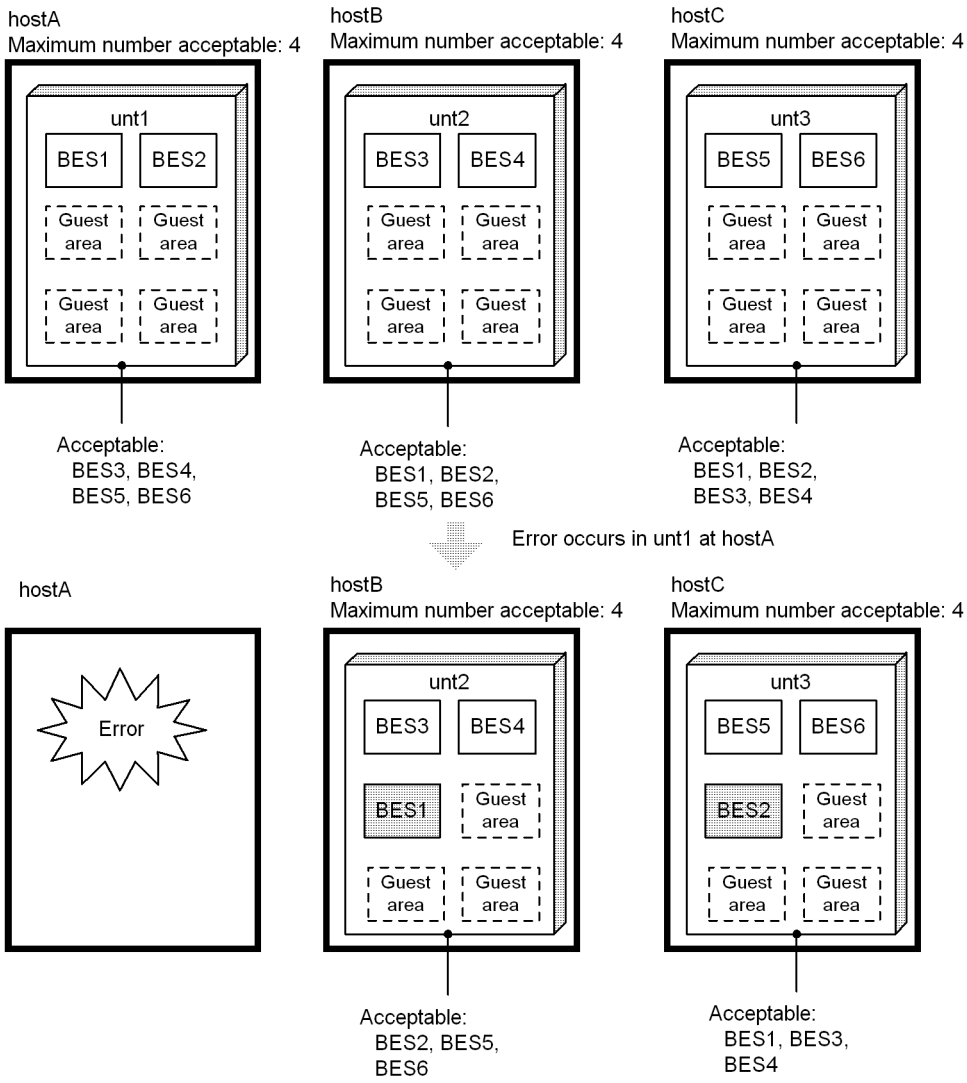
(e) Examples of system switchover using the standby-less system switchover (effects distributed) facility

■ Example of system switchover during normal operations

Figure 25-6 shows an example of system switchover during a time of normal operations.

When an error occurs at hostA, BES1 moves to unt2 and executes processing as a guest BES; BES2 moves to unt3 and executes processing as a guest BES.

Figure 25-6: Example of system switchover during normal operations



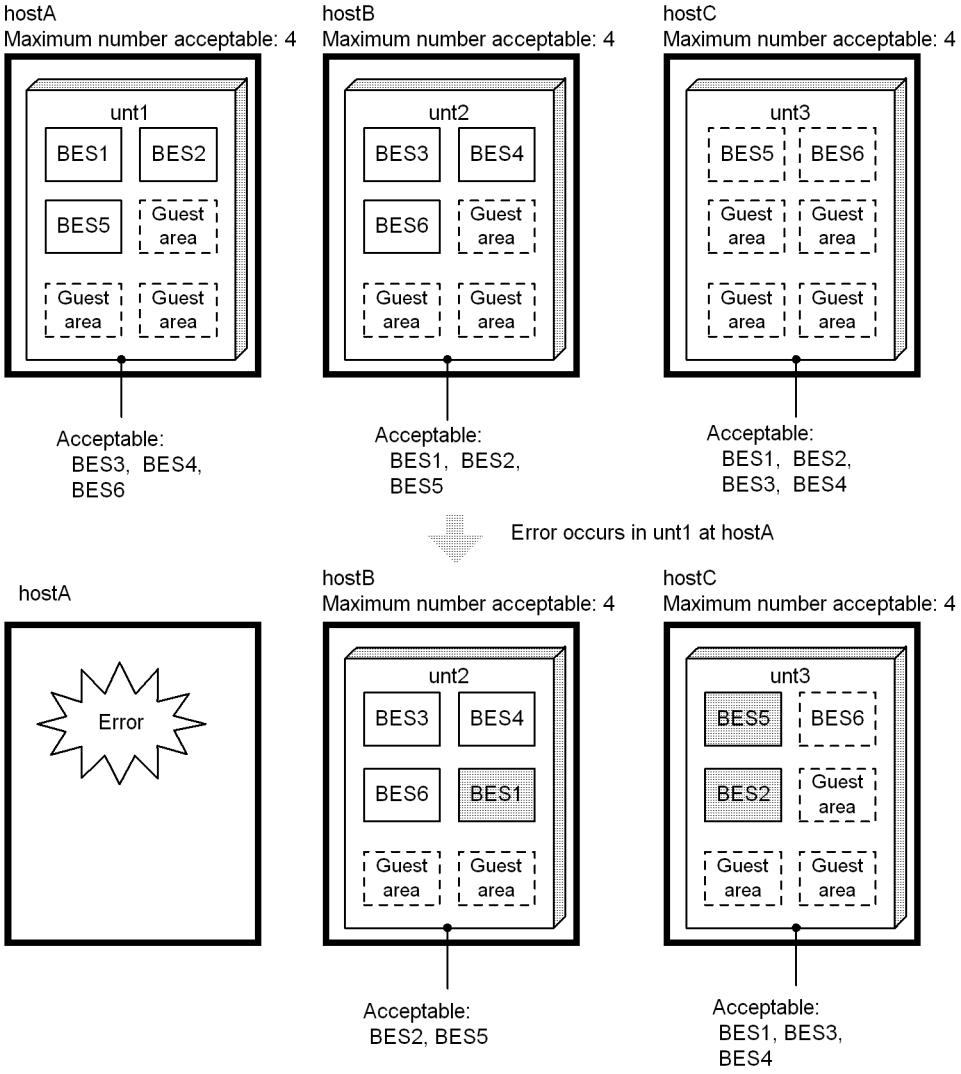
■ Example of system switchover at a host that has accepted guest BESs

Figure 25-7 shows an example of system switchover at a host that has accepted guest BESs. In this example, after a server machine has been restored but before it is reactivated, an error occurs in another server machine.

If an error occurs in hostA while BES5 is executing processing at unt1 as a guest BES, the individual back-end servers behave as follows:

- BES1 moves to unt2 and executes processing as a guest BES.
- BES2 moves to unt3 and executes processing as a guest BES.
- BES5 returns to unt3 and executes processing as a host BES.

Figure 25-7: Example of system switchover at a host that has accepted guest BESs



Unbalanced unit loading

Whether the workloads will become unbalanced among the units following

system switchover depends on the priorities assigned to the standby systems in the cluster software definition. If the priorities of the standby systems are set appropriately, workloads will be balanced even after multiple units have terminated abnormally in any combination.

However, if system switchover occurs as a result of an error in another unit after a unit has been recovered from an error, the workloads may become unbalanced. For this reason, you should check the allocation of servers after system switchover. You use the `pdls -d ha` or `pdls -d svr` command to check the allocation of servers.

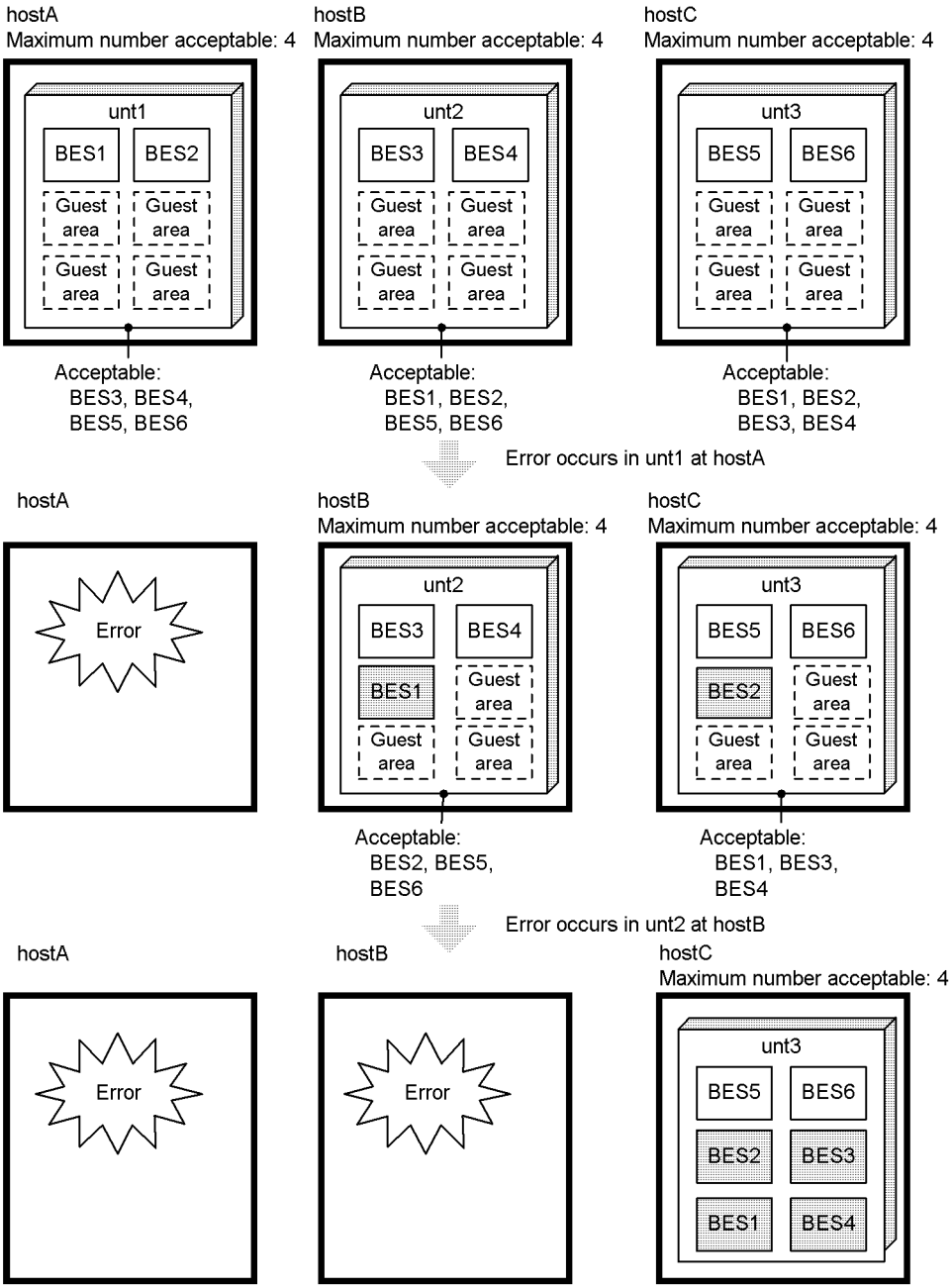
If the server allocation is unbalanced, you should use planned system switchover to modify the allocation of the servers.

Additionally, as soon as possible after a unit has been recovered from an error, you should return each BES to the unit where it is defined. This helps prevent server allocation from becoming unbalanced. In the example shown in Figure 25-7, hosts have accepted guest BESs; if BES5 and BES6 are returned to unt3 before the error occurs in host A, the error in hostA will not result in unbalanced server allocation.

■ System switchover when a series of errors occurs (when all back-end servers are in accepting status)

Figure 25-8 shows an example of system switchover when a series of errors occurs.

Figure 25-8: Example of system switchover when a series of errors occurs



Explanation

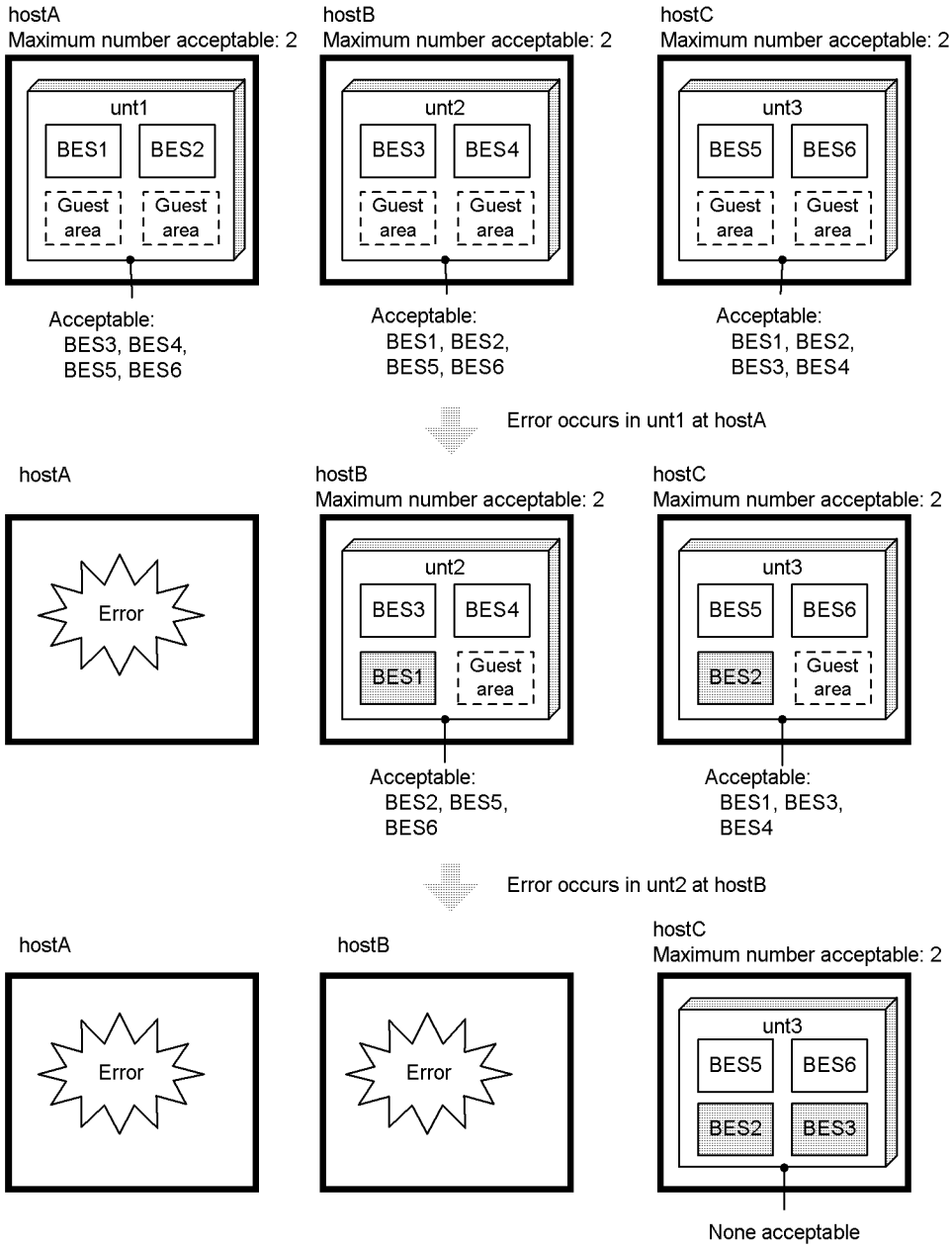
An error occurs at hostA, so BES1 is moved to unt2 and BES2 is moved to unt3, where they execute processing as guest BESs. Because the maximum number of BESs that can be accepted by unt2 and unt3 is 4 each, they can now each accept three more servers. In other words, all back-end servers running at all other units can be accepted.

If an error occurs subsequently at hostB, BES1, BES3, and BES4 running in unt2 all move to unt3 and execute processing as guest BESs. No back-end server stops.

■ System switchover when a series of errors occurs (when the number of BESs that can be accepted is insufficient)

Figure 25-9 shows an example of system switchover when a series of errors occurs but the number of BESs that can be accepted is insufficient.

Figure 25-9: Example of system switchover when a series of errors occurs but the number of BESs that can be accepted is insufficient



Explanation

An error occurs in hostA, so BES1 is moved to unt2 and BES2 is moved to unt3, where they execute processing as guest BESs. Because the maximum number of BESs that can be accepted by unt2 and unt3 is 2 each, they can now each accept only one more server. In other words, not all the back-end servers running at other units can be accepted.

If an error occurs subsequently at hostB, only BES3 running in unt2 moves to unt3 and executes processing as a guest BES; BES1 and BES4 stop.

If it is critical that processing continue at back-end servers even when a series of errors occurs, you must set an appropriately large value for the maximum number of BESs that can be accepted.

System switchover when the number of BESs that can be accepted is insufficient

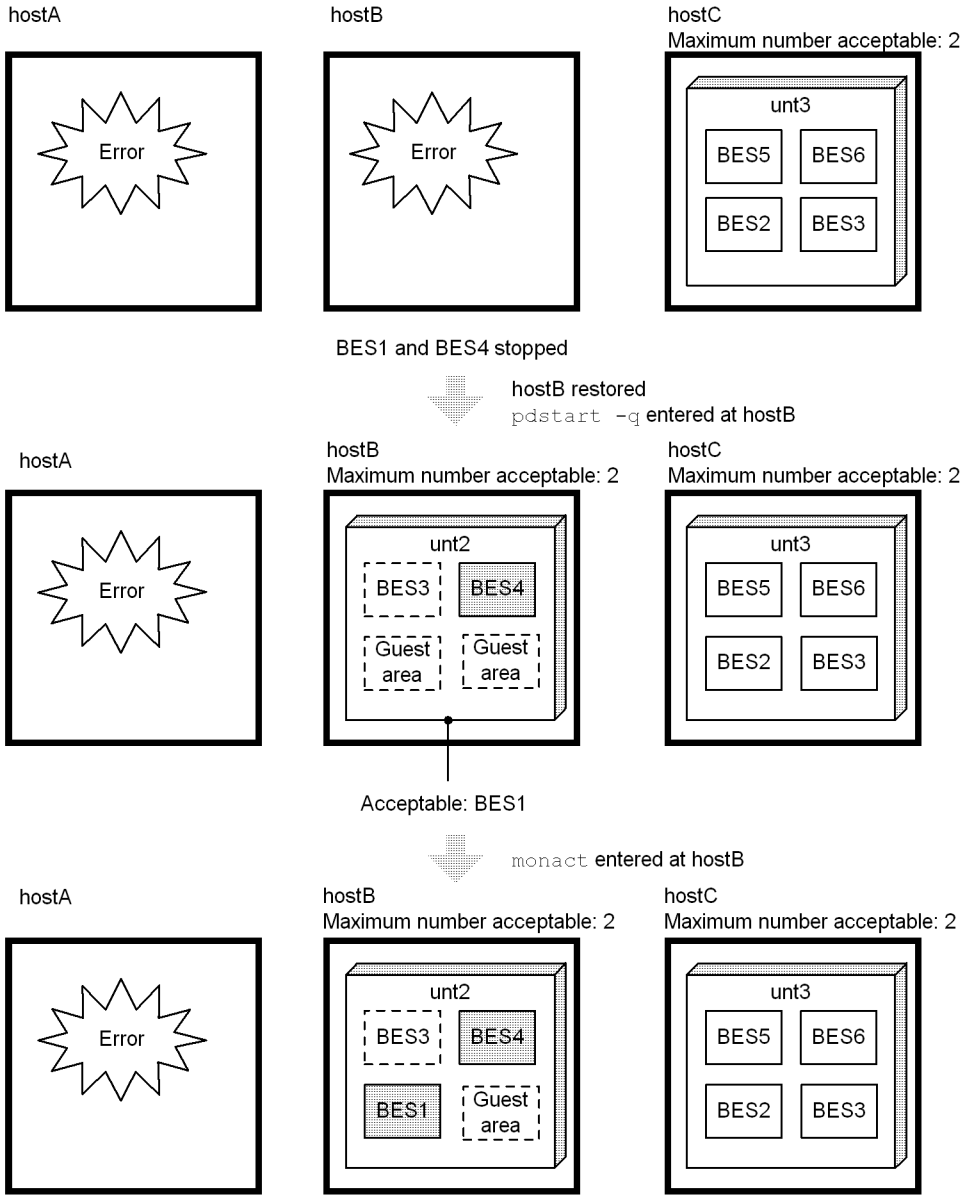
Whether a back-end server in a unit is switched to another unit or is stopped when an error occurs is determined by the priority for system switchover that is assigned to each back-end server. This order is determined by an action of the cluster software.

In this example, only BES3 was moved. However, depending on the action of the cluster software, BES1 or BES4 might have been moved.

■ Example of the action to take when an error occurs while the number of BESs that can be accepted is insufficient

Figure 25-10 shows an example of the action to take when an error occurs while the number of BESs that can be accepted is insufficient.

Figure 25-10: Example of the action to take when an error occurs while the number of BESs that can be accepted is insufficient



Explanation

HostB is recovered from an error and unt2 starts. As a result, BES4 starts as the running system in unt2 and BES1 can be accepted.

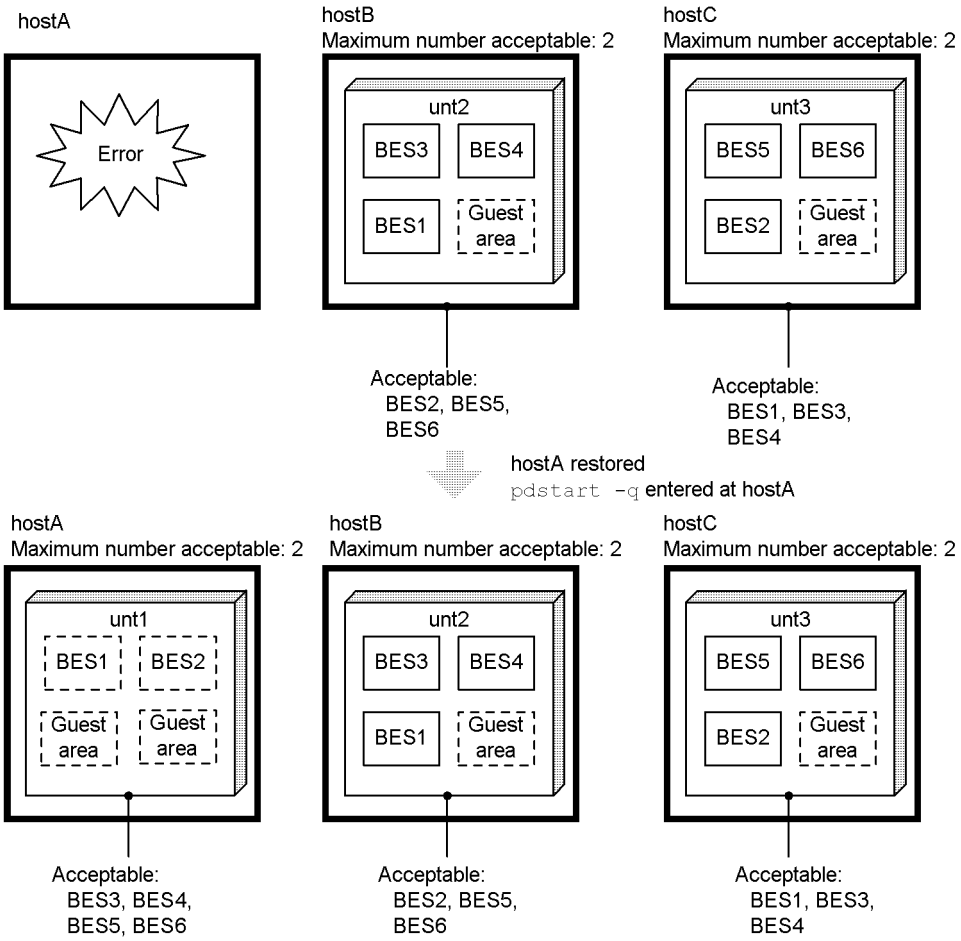
Next, when the `monact` command is entered for BES1 in unt2, BES1 begins processing as a guest BES (when Hitachi HA Toolkit Extension is used, an online command of the cluster software is used).

■ Example of how to avoid a shortage in the number of BESs that can be accepted

If you cannot set a large value for the number of BESs that can be accepted, you must correct errors as soon as possible in order to prevent server stoppage as a result of a shortage in the number of BESs that can be accepted.

Figure 25-11 shows an example of how to avoid a shortage in the number of BESs that can be accepted.

Figure 25-11: Example of how to avoid a shortage in the number of BESs that can be accepted



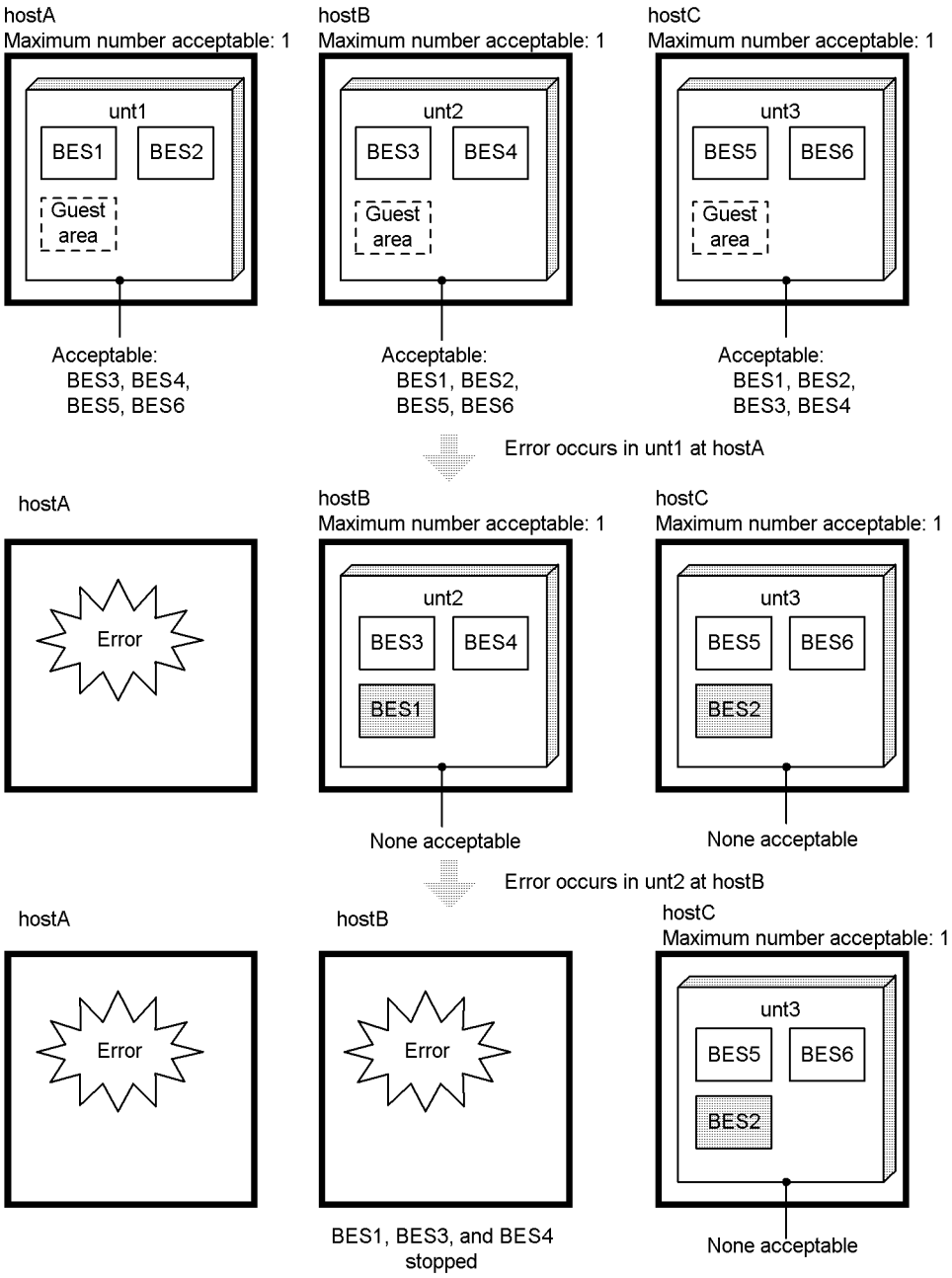
Explanation

HostA is recovered from an error and unt1 starts. As a result, BES1 and BES2 start as standby systems in unt1, so that BES3, BES4, BES5, BES6 become accepting. Therefore, even if an error occurs in hostB, BES1 and BES4 can continue processing at hostA while BES3 can continue processing at hostC.

■ System switchover when a series of errors occurs (when no BES can be accepted)

Figure 25-12 shows an example in which system switchover cannot be executed when a series of errors occurs.

Figure 25-12: Example in which system switchover cannot be executed when a series of errors occurs



Explanation

An error occurs in hostA, so BES1 is moved to unt2 and BES2 is moved to unt3, where they execute processing as guest BESs. Because the maximum number of BESs that can be accepted by unt2 and unt3 is 1 each, neither of them can now accept any more servers.

If an error occurs subsequently at hostB, BES1, BES3, and BES4 stop.

If it is critical that processing continue at back-end servers, even when a series of errors occurs, you must set an appropriately large value for the maximum number of BESs that can be accepted. The action to take when no more servers can be accepted and the method of avoiding server stoppage are the same as indicated above for the case where the number of BESs that can be accepted is insufficient.

25.1.3 Application criteria for the system switchover facilities

(1) Application criteria

Table 25-5 shows the application criteria for the system switchover facilities.

Table 25-5: System switchover facility application criteria

Importance	Standby system switchover facility	Standby-less system switchover (1:1) facility	Standby-less system switchover (effects distributed) facility
Transaction performance	Best	Poor	Good
	• Maintains performance even when an error occurs. • Guarantees the same throughput as during normal operations.	• Load from alternate processing when an error occurs may impact processing performance. • Load concentration on certain units reduces system throughput.	• Load from alternate processing when an error occurs may impact processing performance. • Load distribution is used to minimize system throughput degradation.
Resource requirement	Poor	Best	Best
	Needs standby server machines or system resources for standby units.	Does not need standby server machines or system resources for standby units.	Does not need standby server machines or system resources for standby units.
Handling of a series of errors	Best	Poor	Best
	Can handle based on linkage with cluster software (only when IP addresses are inherited, and not when the rapid system switchover facility is used).	Cannot handle because there is only one fixed alternate server.	Can handle based on linkage with cluster software.

(2) Usability of each system switchover facility when another system switchover facility is already being used

Table 25-6 shows when each system switchover facility can be applied to a unit when a different system switchover facility is already being applied to another unit within the same system.

Table 25-6: Usability of each system switchover facility when another system switchover facility is already being used for another unit within the same system

Facility already being used	Usability of the standby system switchover facility	Usability of the standby-less system switchover (1:1) facility	Usability of the standby-less system switchover (effects distributed) facility
Standby system switchover facility	Y	Y	Y
Standby-less system switchover (1:1) facility	Y	Y	N
Standby-less system switchover (effects distributed) facility	Y	N	Y*

Legend:

Y: Can be used.

N: Cannot be used.

* Only one HA group can be used at the switching destination.

25.1.4 Cluster software supported by HiRDB

The cluster software that HiRDB supports is listed in Table 25-7.

Table 25-7: Cluster software supported by HiRDB

Cluster software supported by HiRDB	Operating system			
	HP-UX	Solaris	AIX 5L	Linux
HA monitor	Yes	No	Yes	Yes
MC/ServiceGuard	Yes	No	No	No
VERITAS Cluster Server	No	Yes	No	No
Sun Cluster	No	Yes	No	No
HACMP	No	No	Yes	No

Cluster software supported by HiRDB	Operating system			
	HP-UX	Solaris	AIX 5L	Linux
ClusterPerfect	No	No	No	Yes

Legend:

Yes: Supported

No: Not supported

Note

- For details about each type of cluster software, see its manual.
- Supported functions differ depending on the type of cluster software used. For details about the functions supported by each type of cluster software, see Table *25-8 Functional differences between the monitor mode and the server mode* and Table *25-9 Cluster software that can be operated in the server mode*.

25.1.5 Monitor mode and server mode

(1) Functional differences between the monitor mode and the server mode

The two operation modes for the system switchover facility are the monitor mode and the server mode. When operating in the monitor mode, only system failures are monitored. When operating in the server mode, both system and server failures are monitored. Also, system switchover can be performed more quickly in the server mode. The functional differences between the monitor mode and the server mode are listed in Table 25-8.

Table 25-8: Functional differences between the monitor mode and the server mode

Item or function		Monitor mode	Server mode
Failures that are monitored	System failures[1]	Yes	Yes
	Server failures[2]	No	Yes
Function that reduces system switchover time	User server standby[3]	No	Yes
	Rapid system switchover facility[3]	No	Yes
Standby-less system switchover facility	Standby-less system switchover (1:1) facility	No	Yes
	Standby-less system switchover (effects distributed) facility	No	Yes

Legend:

Yes: Is monitored or can be used.

No: Is not monitored or cannot be used.

[1] Errors such as those listed below are assumed to be system failures. However, the conditions for system failures are dependent on the cluster software specifications. For details, check the cluster software documentation or other applicable documentation.

- Hardware failures
- Operating system failures
- Power outages
- Cluster software failures
- System slowdowns

[2] Errors such as those listed below are assumed to be server failures.

- HiRDB abnormal termination (HiRDB/Parallel Server)
- HiRDB slowdown (or unit slowdown in the case of a HiRDB/Parallel Server)
- Database path error

[3] This is a function that reduces the system switchover time. For details about user server hot standby and the rapid system switchover facility, see *25.18 Reducing system switchover time (user server hot standby, rapid system switchover facility)*.

(2) Cluster software that can be operated in the server mode

Some cluster software cannot operate in the server mode. Cluster software that can be operated in the server mode is listed in Table 25-9.

Table 25-9: Cluster software that can be operated in the server mode

Cluster software	Monitor mode	Server mode
HA monitor	Yes	Yes
MC/ServiceGuard	Yes	Yes
VERITAS Cluster Server	Yes	Yes
Sun Cluster	Yes	No
HACMP	Yes	No
ClusterPerfect	Yes	No

Legend:

Yes: Can be operated in this mode.

No: Cannot be operated in this mode.

Note:

In the Linux (EM64T) version, the system switchover facility in the server mode is not supported.

(3) Products required for operation in the server mode

The products required to operate the system switchover facility in the server mode are listed in Table 25-10.

Table 25-10: Products required for operation in the server mode

Function	HiRDB Advanced High Availability	Hitachi HA Toolkit Extension
Server mode	--	Required#
User server hot standby	--	Required#
Rapid system switchover facility	--	Required#
Standby-less system switchover (1:1) facility	Required	Required#
Standby-less system switchover (effects distributed) facility	Required	Required#

Legend:

Required: This product is required for operation in the server mode.

--: Not required.

When the cluster software being used is HA monitor, Hitachi HA Toolkit Extension is not required.

25.2 System configuration examples

This section explains system configuration examples for when the system switchover facility is in use.

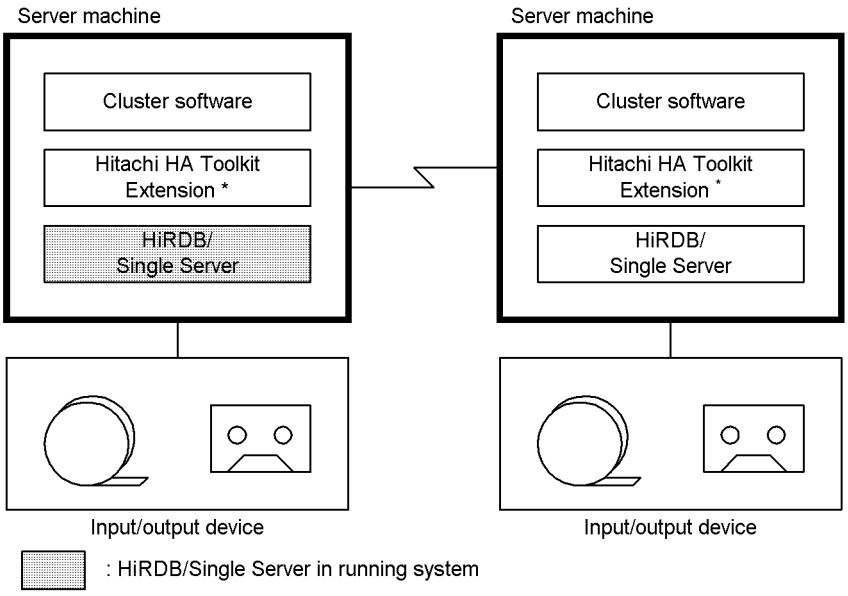
25.2.1 System configuration examples of a HiRDB/Single Server (standby system switchover)

A HiRDB/Single Server performs system switchover at the system level. System configuration examples of a HiRDB/Single Server are explained below.

(1) Example of a 1-to-1 switchover configuration

A configuration in which there is a one-to-one correspondence between the running system and the standby system is called a 1-to-1 switchover configuration. This configuration is appropriate when it is important to guarantee response time when system switchover becomes necessary. The drawback of this configuration is that you cannot utilize the resources of the server machine in the standby system (you do not have access to the resources of one of your two server machines). Figure 25-13 shows an example of a 1-to-1 switchover configuration for a HiRDB/Single Server.

Figure 25-13: System configuration example for HiRDB/Single Servers (1-to-1 switchover configuration)



* This product is required in order to operate the system switchover facility in the

server mode. However, it is not required when the cluster software in use is HA monitor.

Remarks

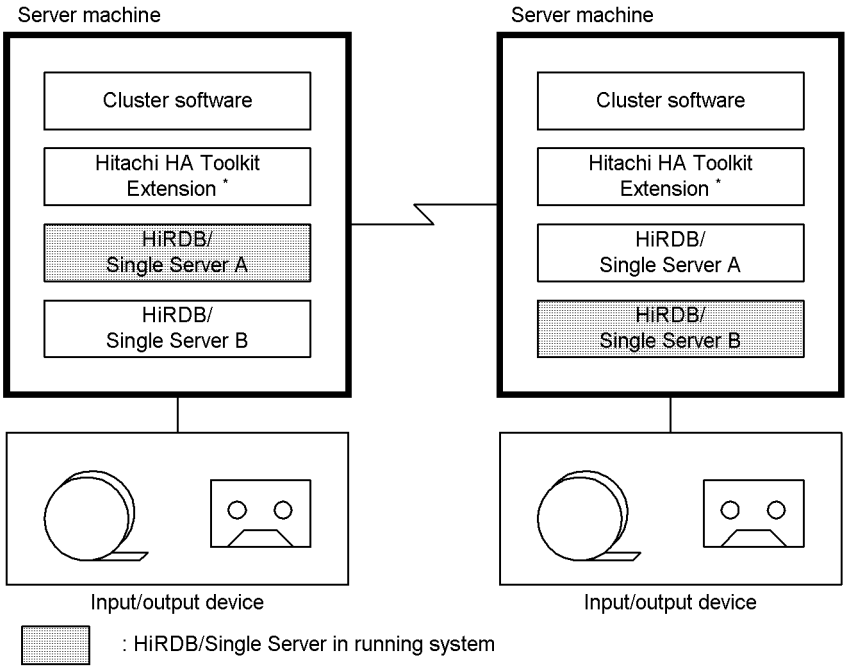
- Each server machine requires an input/output device for use by utilities.
- To execute a utility after system switchover has occurred, you must create on the local server machine the input/output files required to execute that utility.

(2) Example of a mutual system switchover configuration

A system configuration in which the running system has a system on the same server machine that acts as a mutual standby system (standby system for another HiRDB/Single Server) is called a mutual system switchover configuration. This configuration can be applied when multiple HiRDB/Single Servers are being used. This configuration is appropriate for making efficient use of server machine resources. However, response time slows when system switchover occurs.

Figure 25-14 shows an example of a system configuration that includes mutual system switchover for HiRDB/Single Servers. In this example, the mutual system switchover configuration is applied to HiRDB/Single Servers A and B.

Figure 25-14: System configuration example for HiRDB/Single Servers (mutual system switchover)



* This product is required in order to operate the system switchover facility in the server mode. However, it is not required when the cluster software in use is HA monitor.

Remarks

- Each server machine requires an input/output device for use by utilities.
- To execute a utility after system switchover has occurred, you must create on

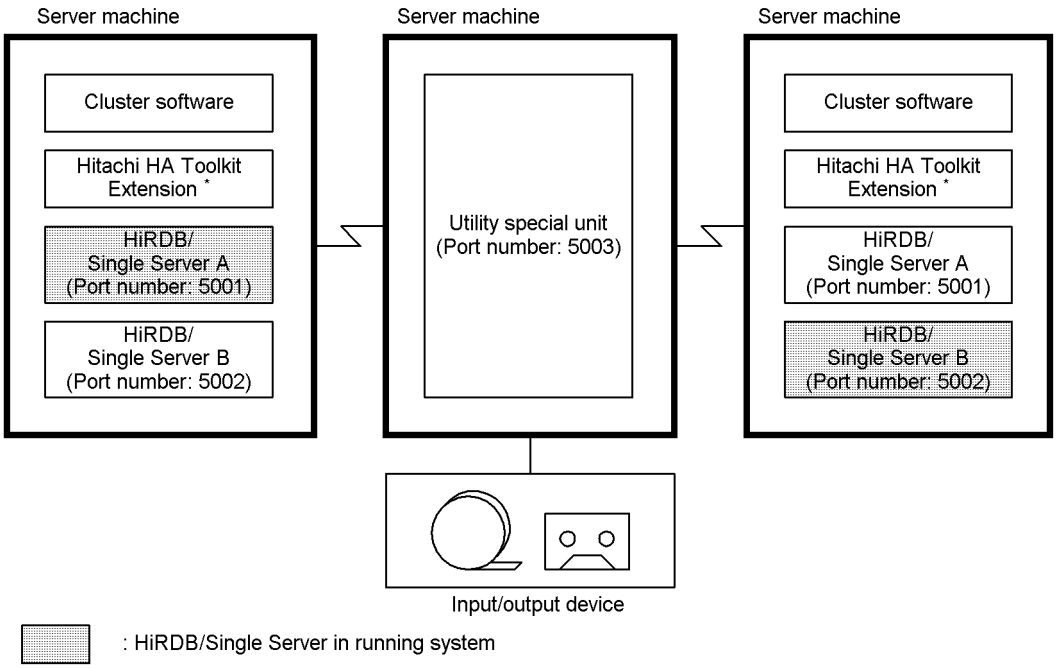
the local server machine the input/output files required to execute that utility.

- In the case of a system switchover configuration, if a host name is specified in a utility control statement, you must change the host name when the systems are switched.

(3) Sharing a utility special unit among multiple HiRDB/Single Servers

Figure 25-15 shows an example of sharing a utility special unit among multiple HiRDB/Single Servers. Note that system switchover cannot be performed on utility special units.

Figure 25-15: Sharing a utility special unit among multiple HiRDB/Single Servers

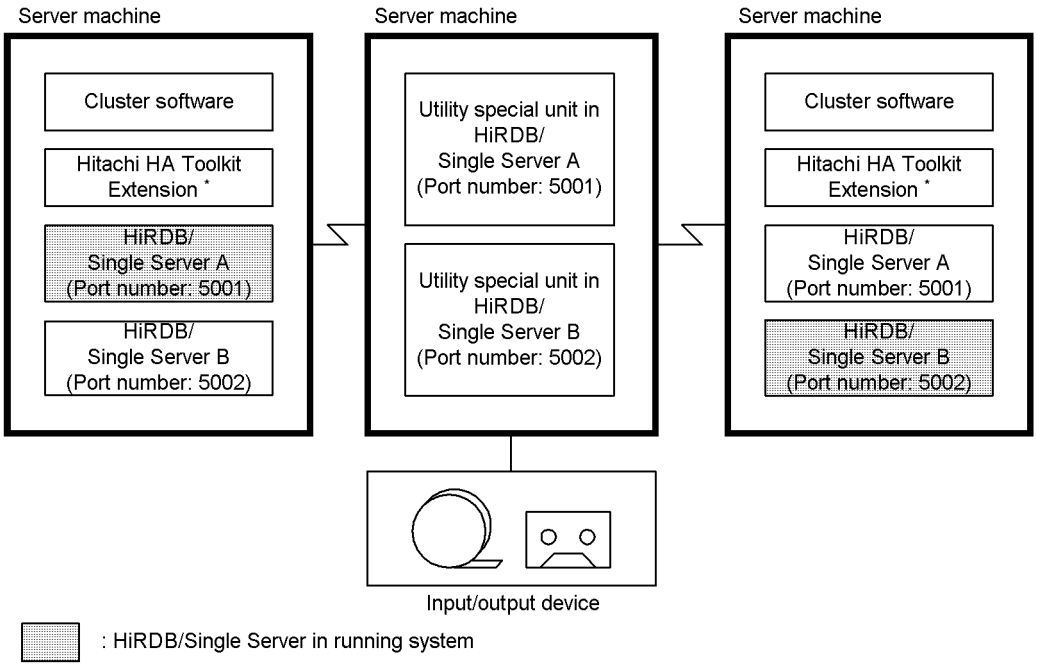


* This product is required in order to operate the system switchover facility in the server mode. However, it is not required when the cluster software in use is HA monitor.

(4) Setting up a 1:1 correspondence between HiRDB/Single Servers and utility special units

Figure 25-16 shows an example of setting up a 1:1 correspondence between HiRDB/Single Servers and utility special units.

Figure 25-16: Setting up a 1:1 correspondence between HiRDB/Single Servers and utility special units

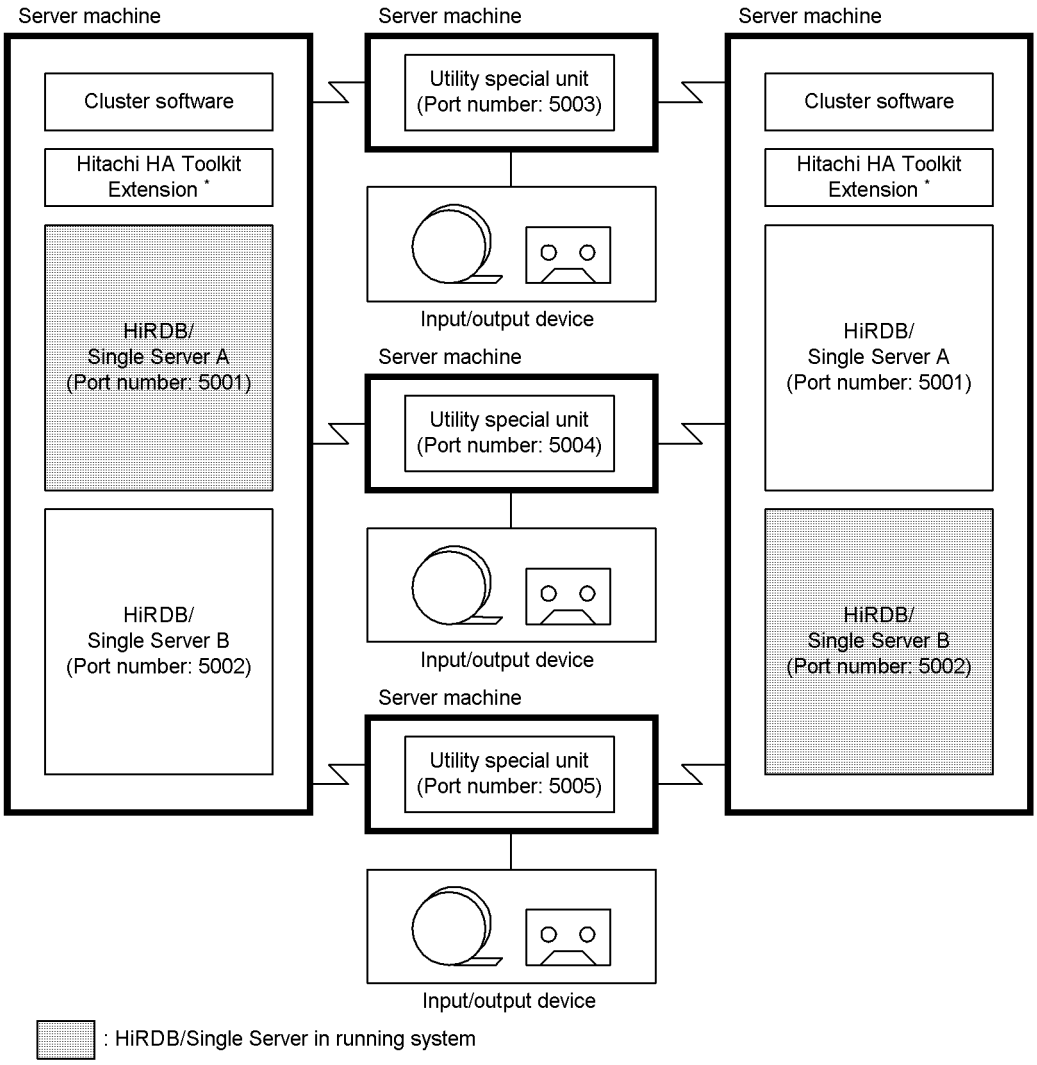


* This product is required in order to operate the system switchover facility in the server mode. However, it is not required when the cluster software in use is HA monitor.

(5) Setting up an m:n correspondence between HiRDB/Single Servers and utility special units

Figure 25-17 shows an example of setting up an *m*:*n* correspondence between HiRDB Single Servers and utility special units.

Figure 25-17: Setting up a m:n correspondence between HiRDB/Single Servers and utility special units



* This product is required in order to operate the system switchover facility in the server mode. However, it is not required when the cluster software in use is HA

monitor.

Explanation

- The host whose name was specified in executing the utility can select the utility special unit to be used.
- Setting up multiple utility special units can increase the reliability of your system.

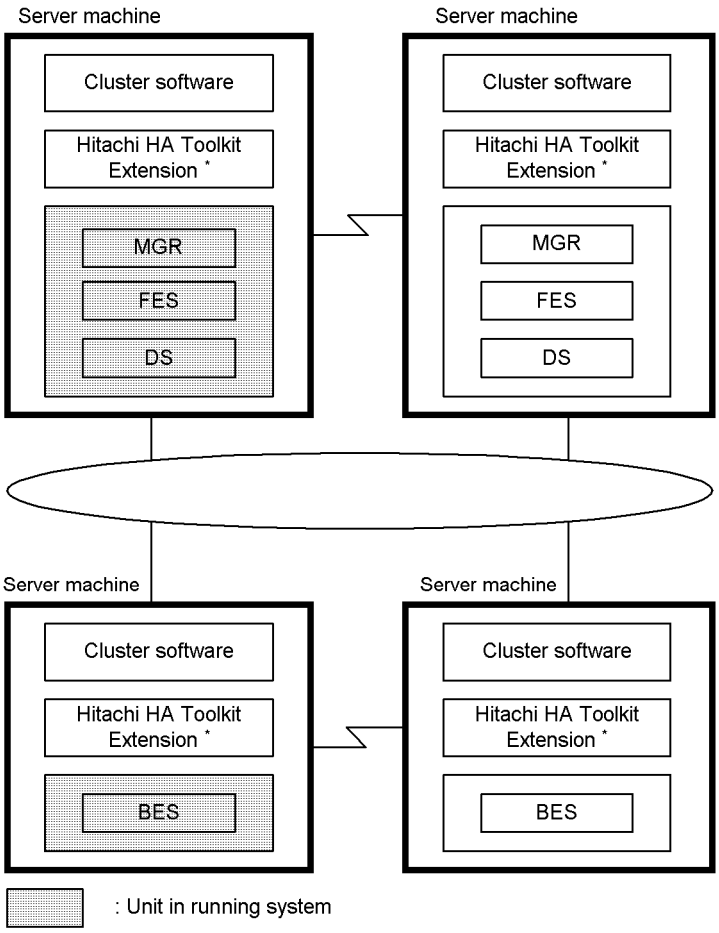
25.2.2 System configuration examples of a HiRDB/Parallel Server

A HiRDB/Parallel Server performs system switchover at the unit level. System configuration examples for a HiRDB/Parallel Server are explained below.

(1) Example of a 1-to-1 switchover configuration

A configuration in which there is a one-to-one correspondence between the running system and the standby system is called a 1-to-1 switchover configuration. This configuration is appropriate when it is important to guarantee response time when system switchover becomes necessary. The drawback of this configuration is that you cannot utilize the resources of the server machine in the standby system (you do not have access to the resources of one of your two server machines). Figure 25-18 shows an example of a 1-to-1 switchover configuration for a HiRDB/Parallel Server.

Figure 25-18: System configuration example for a HiRDB/Parallel Server (1-to-1 switchover configuration)



* This product is required in order to operate the system switchover facility in the server mode. However, it is not required when the cluster software in use is HA monitor.

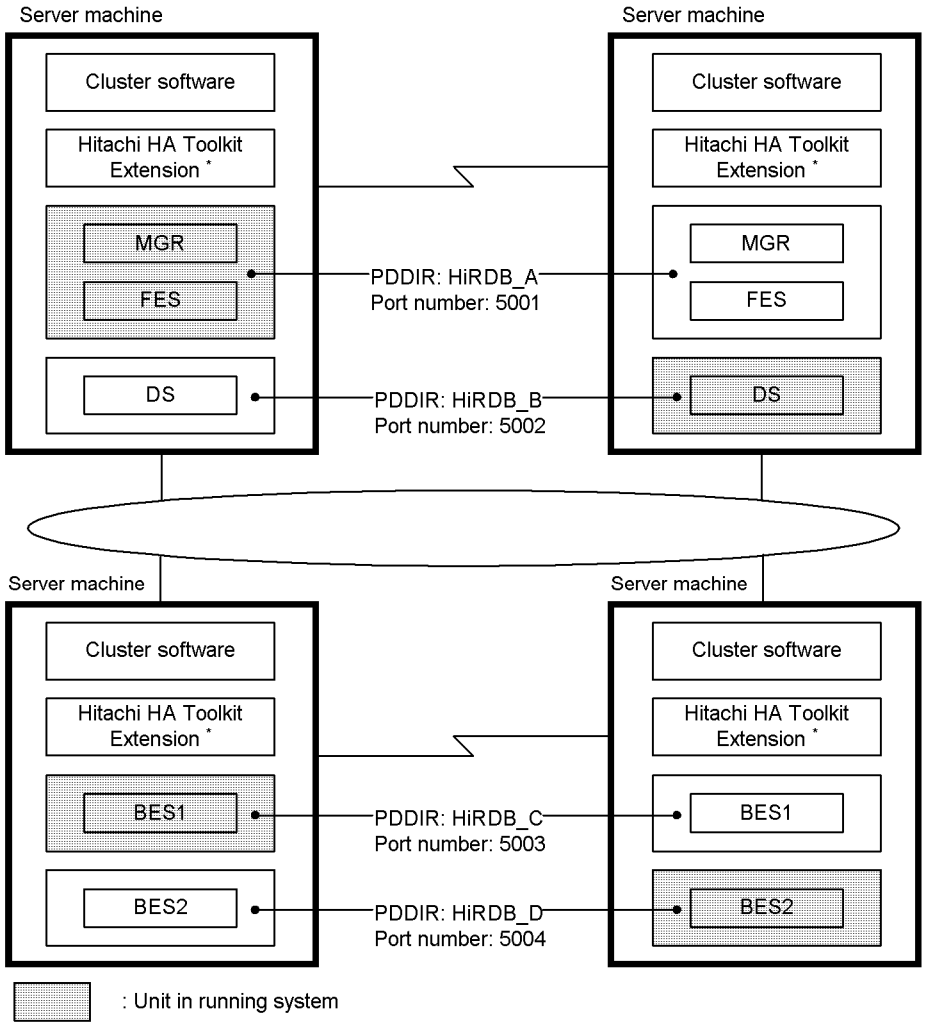
(2) Example of mutual system switchover configuration

A system configuration in which the running system has a system on the same server machine that acts as its mutual standby system (standby system for another unit) is called a mutual system switchover configuration. In a HiRDB/Parallel Server, systems can be switched over in units. Therefore, whereas a mutual system switchover configuration cannot be applied to a single HiRDB/Single Server, it can be applied to a single HiRDB/Parallel Server. When a mutual system switchover configuration is

applied to a HiRDB/Parallel Server, the running unit and the standby unit (standby unit for another unit) can be positioned within the same server machine. This configuration is appropriate for making efficient use of server machine resources. However, response time slows when system switchover occurs.

Figure 25-19 shows an example of a mutual system switchover configuration for a HiRDB/Parallel Server.

Figure 25-19: System configuration example for a HiRDB/Parallel Server (mutual system switchover)



* This product is required in order to operate the system switchover facility in the

server mode. However, it is not required when the cluster software in use is HA monitor.

For examples of the HiRDB system definitions for the mutual system switchover configuration, see the manual *HiRDB Version 8 System Definition*.

Note:

If you use the mutual system switchover configuration (or the 2-to-1 switchover configuration), two units may run on a single server machine. Therefore, you must pay attention to the following:

- The HiRDB directory names
- The port numbers

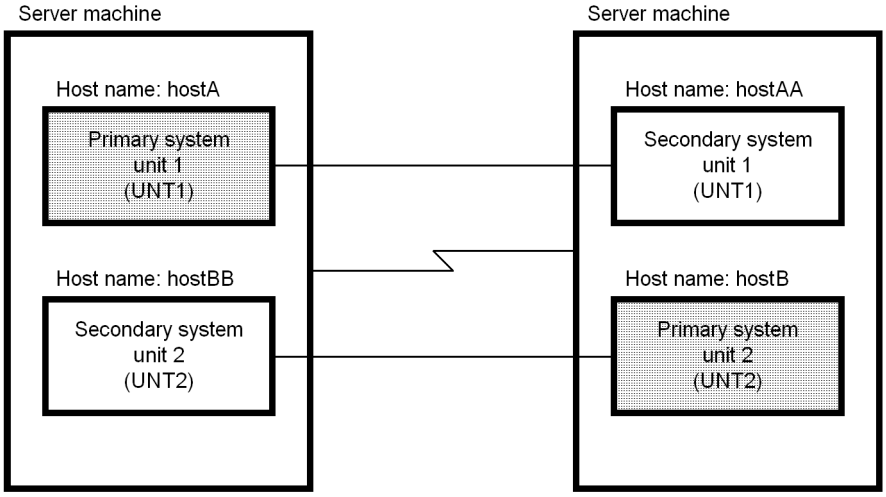
When running these two units on a single server machine, give each unit a different name and port number. Also, if IP addresses will not be inherited after system switchover, you must also pay attention to the following:

- Host names and IP addresses

Provide each unit on the same server machine with a unique host name and IP address.

Figure 25-20 shows an example of correct host name setup; Figure 25-21 shows an example of incorrect host name setup.

Figure 25-20: Example of correct host name setup

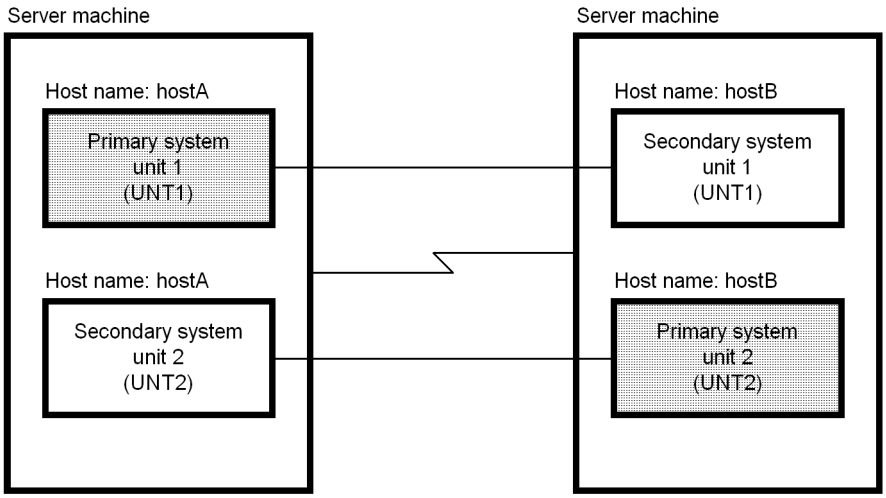


The following is an example of correct `pdunit` operand specifications:

```
pdunit -x hostA -u UNT1  ...  -c hostAA
pdunit -x hostB -u UNT2  ...  -c hostBB
```

Furthermore, host names corresponding to different IP addresses must be specified for hostA, hostAA, hostB, and hostBB.

Figure 25-21: Example of incorrect host name setup



The following is an example of incorrect `pdunit` operand specifications:

```
pdunit -x hostA -u UNT1  ...  -c hostB
pdunit -x hostB -u UNT2  ...  -c hostA
```

Explanation

If IP addresses are not to be inherited in a mutual system switchover configuration, the host names specified in the `-x` and `-c` options of the `pdunit` operand must all be different (host names cannot be duplicated).

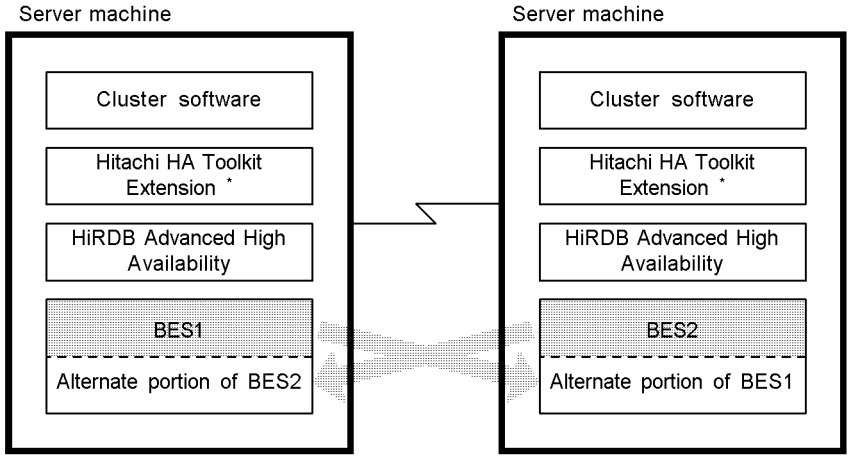
25.2.3 System configuration examples of standby-less system switchover (1:1)

This section describes typical system configuration examples of standby-less system switchover (1:1).

(1) Mutual alternating configuration

In this configuration, two back-end servers become each other's alternate BESs. Figure 25-22 shows a system configuration example of a mutual alternating configuration.

Figure 25-22: System configuration example of a mutual alternating configuration



* When the cluster software being used is HA monitor, Hitachi HA Toolkit Extension is not required.

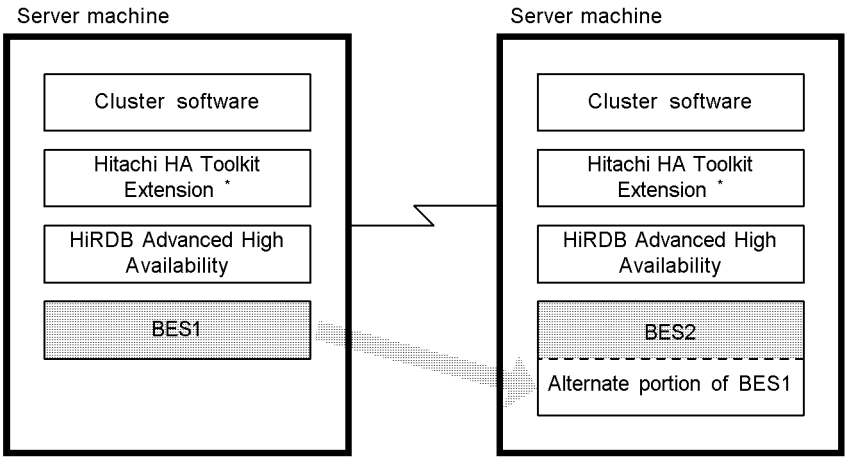
Explanation

- BES1 is the alternate BES for BES2. When an error occurs in BES2, the alternate portion of BES2 takes over the BES2 processing.
- BES2 is the alternate BES for BES1. When an error occurs in BES1, the alternate portion of BES1 takes over the BES1 processing.

(2) One-way alternating configuration

This configuration alternates only in a single direction in 1:1 standby-less system switchover. Figure 25-23 shows a system configuration example of a one-way alternating configuration (2-node configuration).

Figure 25-23: System configuration example of a one-way alternating configuration (2-node configuration)



* When the cluster software being used is HA monitor, Hitachi HA Toolkit Extension is not required.

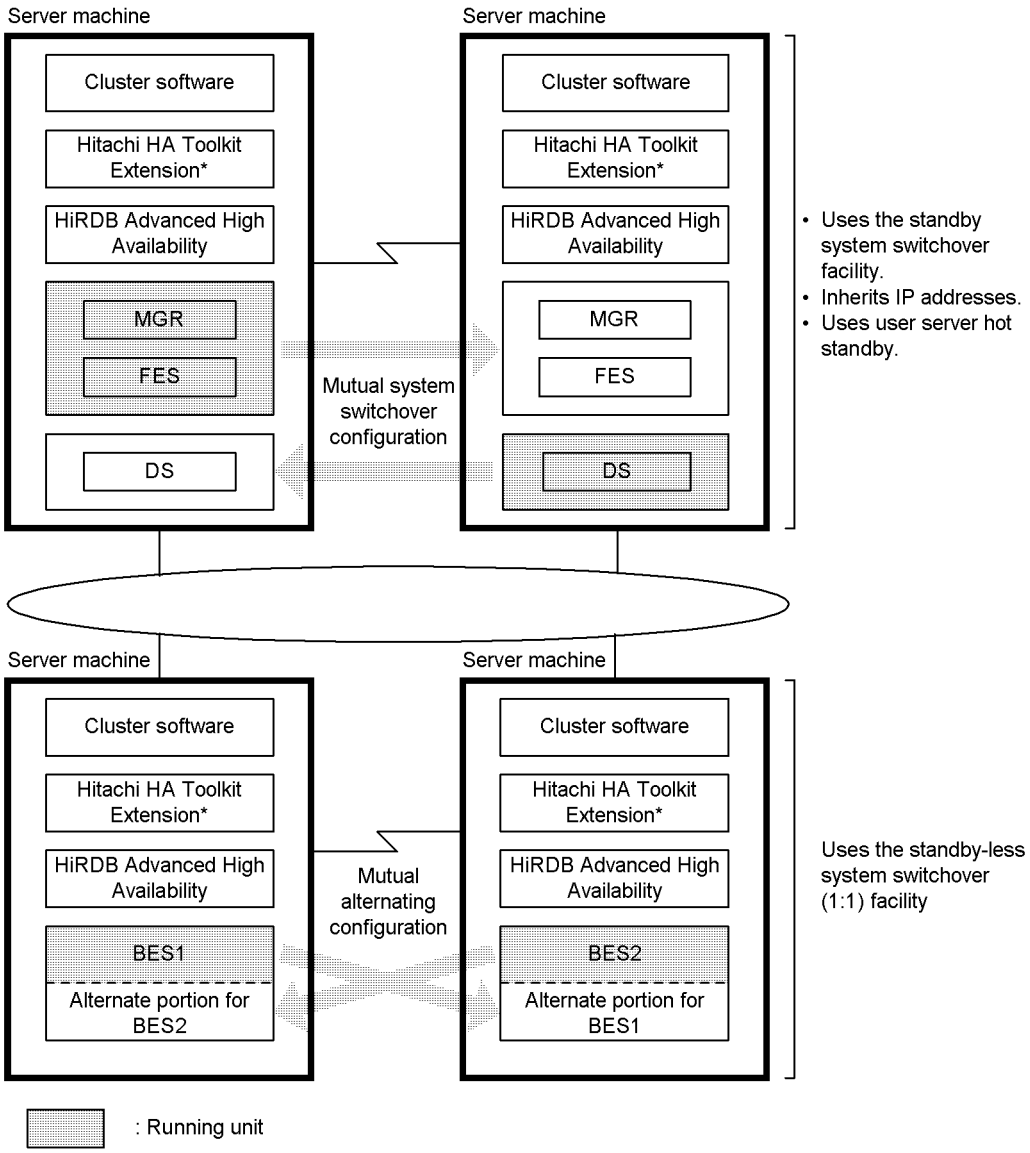
Explanation

BES2 is the alternate BES for BES1. When an error occurs in BES1, the alternate portion of BES1 takes over the BES1 processing. When an error occurs in BES2, BES1 does not take over the BES2 processing.

(3) System configuration example combining standby-less (1:1) and standby system switchover

Figure 25-24 shows a system configuration example combining standby-less (1:1) and standby system switchover. For example HiRDB system definitions for this system configuration, see the manual *HiRDB Version 8 System Definition*.

Figure 25-24: System configuration example combining standby-less (1:1) and standby system switchover



* When the cluster software being used is HA monitor, Hitachi HA Toolkit Extension is not required.

Explanation

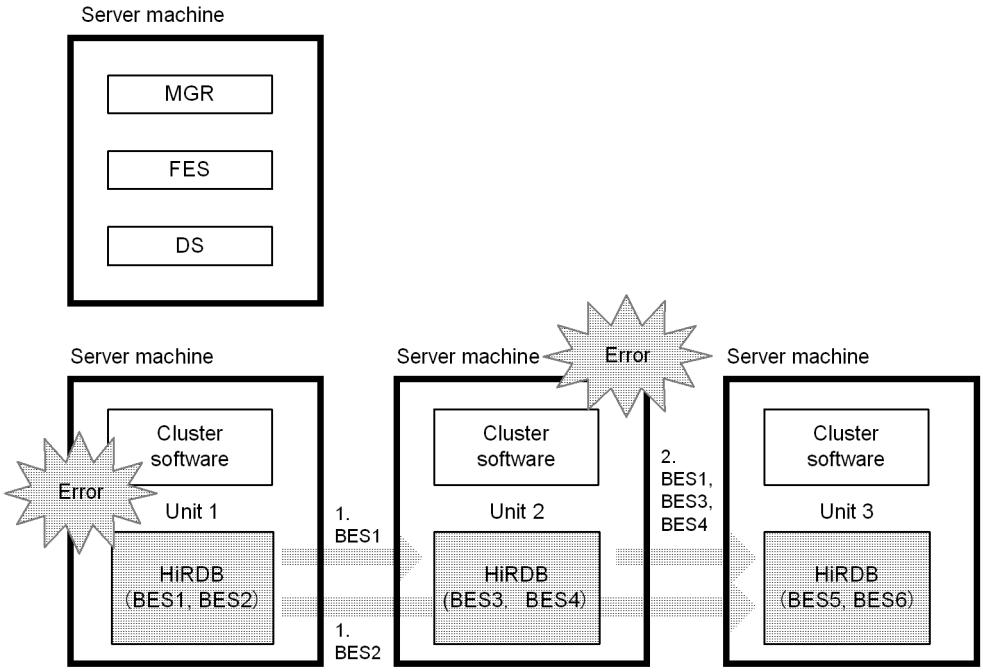
- The standby system switchover facility (mutual system switchover configuration) is applied to the MGR, FES, and DS units. To facilitate operation after system switchover, a configuration that inherits IP addresses is used. Also, user server hot standby is applied.
- The standby-less system switchover (1:1) facility (mutual alternate configuration) is applied to BES units.
- HiRDB Advanced High Availability must be installed on all server machines. Server machines that do not apply the standby-less system switchover facility and server machines that do not apply the system switchover facility must also run HiRDB Advanced High Availability.

25.2.4 System configuration examples of standby-less system switchover (effects distributed)

(1) System configuration example

Figure 25-25 shows a system configuration example of standby-less system switchover (effects distributed). When an error occurs in the regular unit, processing intended for the back-end servers in the primary system where the error occurred is divided in units of back-end servers, assumed by multiple running server machines, and executed.

Figure 25-25: System configuration example of standby-less system switchover (effects distributed)



Explanation

1. When an error occurs in Unit 1, BES1 executes its processing in Unit 2 as a guest BES, and BES2 executes its processing in Unit 3 as a guest BES.
2. If an error occurs in Unit 2 while Unit 1 is still in error status, BES1, BES2, BES3, and BES4 all execute their processing in Unit 3 as guest BESs.

(2) Determining the switching destination for each server

When you use the standby-less system switchover (effects distributed) facility, HiRDB takes the following issues into consideration in determining the switching destinations for the individual servers:

- Balancing among the units the number of running guest BESs
- Balancing among the units the number of servers that share each global buffer

When standby-less system switchover (effects distributed) is used, HiRDB uses the following procedure to determine switching destinations:

Procedure

1. Determines the switching-destination unit for the server that has the highest

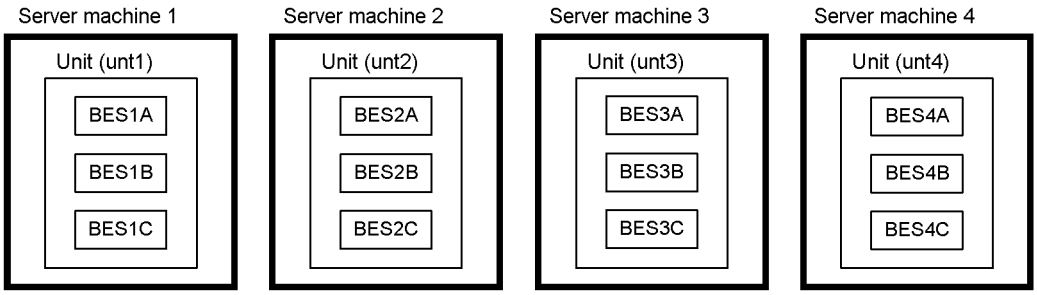
priority among all servers. The following are the determination criteria:

- Balancing the number of servers to be switched from one unit to other units
- Balancing among the units at the switching destination the number of servers that will share a global buffer

2. From i = 1 to [*number-of-units-in-HA-group* - 2], steps 3 through 9 are repeated.
3. Selects a server with priority ranking of i + 1 whose switching destination has not been determined.
4. If there is an applicable server, the process proceeds to step 5. If there is no applicable server, i = i + 1 is set and the process returns to step 3.
5. Assumes that switching destinations 1 through i must be determined for the selected servers, and that the unit to which these servers belong is the erroneous unit.
6. Units other than the erroneous unit are selected as switching destination candidates.
7. From the servers belonging to the erroneous unit, all servers that have switching destination candidates as defined switching destinations are extracted, and it is assumed that guest BESs for the server in question are allocated to the switching-destination unit with the highest priority.
8. From the servers belonging to the erroneous unit, all servers that do not have switching destination candidates as defined switching destinations are extracted, and the i +1-th switching-destination unit is determined for each server. The following are the determination criteria:
   - The unit with the smallest number of guest BESs is selected.
   - Units in which fewer guest BESs share a global buffer with the server are selected.
9. Returns to step 3.

(a) Configuration with four units

The standby-less system switchover (effects distributed) facility is applied to the following configuration consisting of four units:



In this case, a global buffer is shared in each of the following groups:

- Group A: BES1A, BES2A, BES3A, and BES4A
- Group B: BES1B, BES2B, BES3B, and BES4B
- Group C: BES1C, BES2C, BES3C, and BES4C

1. Determining the switching-destination unit with the highest priority

   An example is shown that determines the switching-destination unit with the highest priority for BES3B. A switching destination is determined so that it does not coincide with servers that share the same global buffer (BES1B and BES2B) or BES3A. Specifically, unt1 is selected because the switching destination of BES1B (unt3), the switching destination of BES2B (unt4), and the switching destination of BES3A (unt4) cannot be selected.

Server	Host BES	Highest priority	Second priority	Third priority
BES1A	unt1	unt2	Allocated	Allocated
BES1B		unt3	Allocated	Allocated
BES1C		unt4	Allocated	Allocated
BES2A	unt2	unt3	Allocated	Allocated
BES2B		unt4	Allocated	Allocated
BES2C		unt1	Allocated	Allocated
BES3A	unt3	unt4	Allocated	Allocated
BES3B		unt1	Allocated	Allocated
BES3C		Allocated	Allocated	Allocated
BES4A	unt4	Allocated	Allocated	Allocated
BES4B		Allocated	Allocated	Allocated

Server	Host BES	Highest priority	Second priority	Third priority
BES4C		Allocated	Allocated	Allocated

2. Determining the switching-destination unit with the second or subsequent priority

   An example is shown that determines the switching-destination unit with the second priority for BES2B.

   First, unt4, which is a defined switching destination of BES2B, and unt2, to which BES2C belongs, are assumed to be erroneous units (step 5), and unt1 and unt3 are selected as switching-destination candidates (step 6). Next, from the servers belonging to the erroneous units (unt2 and unt4), servers (BES2A, BES2C, BES4A, and BES4C) that have the switching destination candidates as defined switching destinations are extracted, and it is assumed that guest servers are allocated as shown below (step 7).

   BES2A: unt3; BES2C: unt1; BES4A: unt1; and BES4C: unt3

   Next, servers (BES2B and BES4B) that do not contain the switching destination candidates as defined switching destinations are extracted, and the switching-destination unit with the second priority is determined. In this case, the number of guest BESs is the same for unt1 and unt3, and BES2B and BES4B share a global buffer, and therefore, BES2B is switched to unt1 and BES4B is switched to unt3.

Server	Host BES	Highest priority	Second priority	Third priority
BES1A	unt1 Switching-destination candidate unit	unt2	unt4	Allocated
BES1B		unt3	unt2	Allocated
BES1C		unt4	unt2	Allocated
BES2A	unt2 Erroneous unit	unt3	unt1	Allocated
BES2B		unt4	unt1	Allocated
BES2C		unt1	Allocated	Allocated
BES3A	unt3 Switching-destination candidate unit	unt4	Allocated	Allocated
BES3B		unt1	Allocated	Allocated
BES3C		unt2	Allocated	Allocated

Server	Host BES	Highest priority	Second priority	Third priority
BES4A	unt4 Erroneous unit	unt1	Allocated	Allocated
BES4B		unt2	Allocated	Allocated
BES4C		unt3	Allocated	Allocated

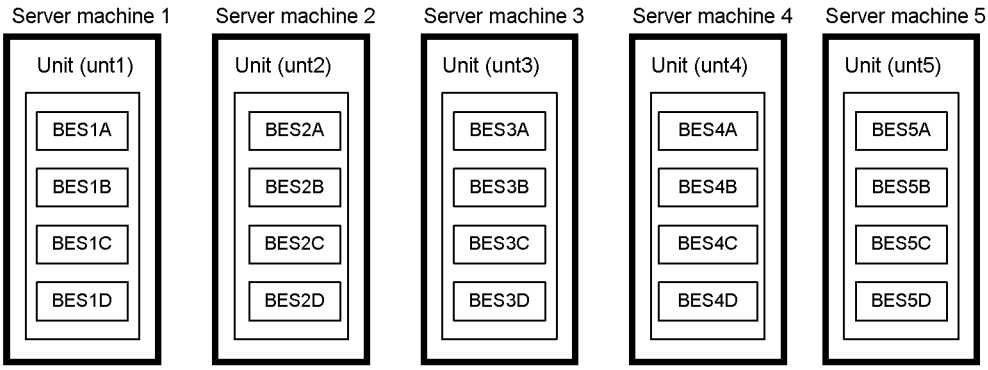
Table 25-11 shows an example of switching destination definition in a 4-unit configuration.

Table 25-11: Switching destination definition example in a 4-unit configuration

Server	Host BES	Highest priority	Second priority	Third priority
BES1A	unt1	unt2	unt4	unt3
BES1B		unt3	unt2	unt4
BES1C		unt4	unt2	unt3
BES2A	unt2	unt3	unt1	unt4
BES2B		unt4	unt1	unt3
BES2C		unt1	unt3	unt4
BES3A	unt3	unt4	unt2	unt1
BES3B		unt1	unt4	unt2
BES3C		unt2	unt4	unt1
BES4A	unt4	unt1	unt3	unt2
BES4B		unt2	unt3	unt1
BES4C		unt3	unt1	unt2

(3) Configuration with five units

The standby-less system switchover (effects distributed) facility is applied to the following configuration consisting of five units:



In this case, a global buffer is shared in each of the following groups:

- Group A: BES1A, BES2A, BES3A, BES4A, and BES5A
- Group B: BES1B, BES2B, BES3B, BES4B, and BES5B
- Group C: BES1C, BES2C, BES3C, BES4C, and BES5C
- Group D: BES1D, BES2D, BES3D, BES4D, and BES5D

Table 25-12 shows an example of switching destination definition in a 5-unit configuration.

Table 25-12: Switching destination definition example in a 5-unit configuration

Server	Host BES	Highest priority	Second priority	Third priority	Fourth priority
BES1A	unt1	unt2	unt5	unt3	unt4
BES1B		unt3	unt4	unt2	unt5
BES1C		unt4	unt3	unt5	unt2
BES1D		unt5	unt2	unt4	unt3
BES2A	unt2	unt3	unt1	unt4	unt5
BES2B		unt4	unt5	unt1	unt3
BES2C		unt5	unt4	unt3	unt1
BES2D		unt1	unt3	unt5	unt4
BES3A	unt3	unt4	unt2	unt1	unt5
BES3B		unt5	unt1	unt2	unt4
BES3C		unt1	unt5	unt4	unt2

Server	Host BES	Highest priority	Second priority	Third priority	Fourth priority
BES3D		unt2	unt4	unt5	unt1
BES4A	unt4	unt5	unt3	unt1	unt2
BES4B		unt1	unt2	unt3	unt5
BES4C		unt2	unt1	unt5	unt3
BES4D		unt3	unt5	unt2	unt1
BES5A	unt5	unt1	unt4	unt2	unt3
BES5B		unt2	unt3	unt1	unt4
BES5C		unt3	unt2	unt4	unt1
BES5D		unt4	unt1	unt3	unt2

25.3 IP address configuration examples

The network configuration and operating procedures depend on whether or not IP addresses are inherited after system switchover. For the applicable procedures, refer to the following sections:

- *25.4 Handling of host names depending on whether or not IP addresses are inherited*
- *25.17 Operating procedures after system switchover*

This section explains the differences between network configurations.

Guidelines

- For the standby system switchover facility, Hitachi recommends strongly that you use a system configuration that inherits IP addresses after system switchover. A system configuration that does not inherit IP addresses after system switchover will be more difficult to operate than a system configuration that does inherit IP addresses.
- Some cluster software will inherit IP addresses after system switchover and other cluster software will not. For details, see the documentation for each cluster software product.
- For units that apply the rapid system switchover facility, use a system configuration that does not inherit IP addresses after system switchover.
- Because a host that is already running takes over processing in a standby-less system switchover facility, IP addresses are not inherited.
- Use of alias IP addresses is recommended.
- When the server mode is used for the Linux version of HiRDB, a system configuration that does not inherit IP addresses is used.

(1) When IP addresses are inherited

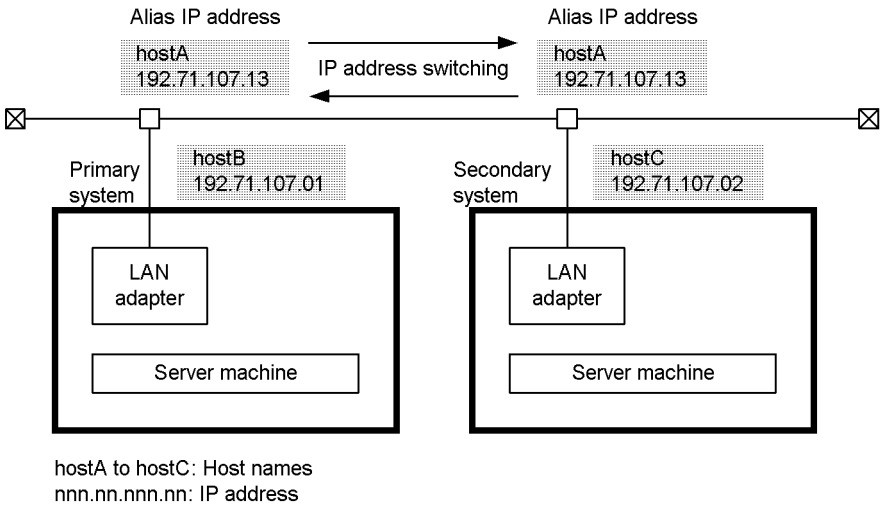
The two methods of inheriting IP addresses described below are as follows:

- Switching the IP address
- Switching LAN adapters

(a) Switching the IP address

You switch the IP address by using an alias IP address (you stop one IP address and start the other). In this way, you use the same IP address and host name for the primary system and the secondary system. Figure 25-26 shows a network configuration example of inheriting IP addresses (switching the IP address).

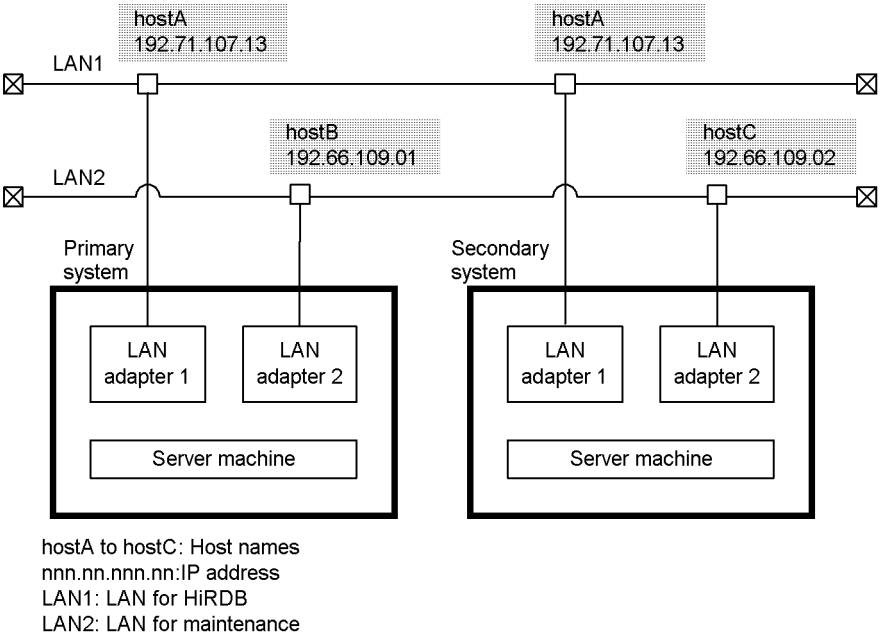
Figure 25-26: Network configuration example when inheriting IP addresses (switching the IP address)



(b) Switching LAN adapters

You prepare a maintenance LAN adapter in addition to the LAN adapter that HiRDB uses and you switch the LAN adapter that is being used (stop one LAN adapter and start the other one). Then, you use the same IP address and host name for the primary system and the secondary system. Figure 25-27 shows a network configuration example of inheriting IP addresses (switching LAN adapters).

Figure 25-27: Network configuration example when inheriting IP addresses (switching LAN adapters)



(2) When IP addresses are not inherited

Standby-less system switchover (1:1) facility

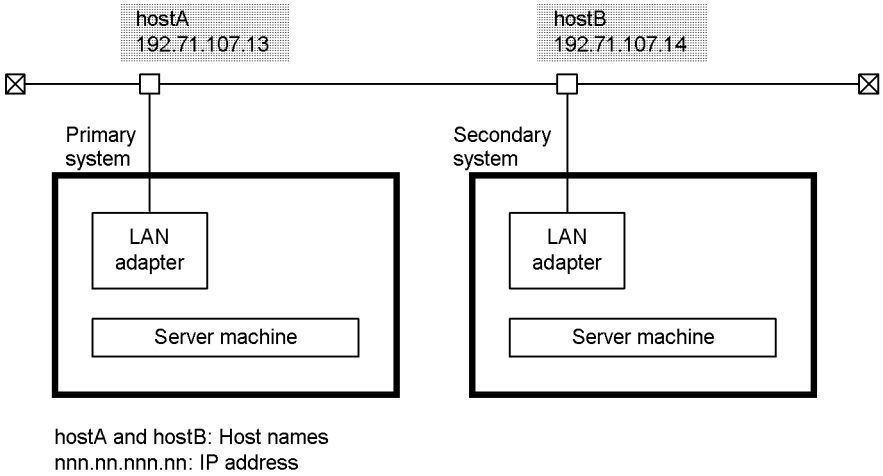
When IP addresses are not inherited, different IP addresses and host names are used for the primary and secondary systems. This means that the host name for the secondary system must be specified in the `pdunit -c` operand in the HiRDB system common definition.

Standby-less system switchover (effects distributed) facility

Different IP addresses and host names are used for the regular unit and the accepting unit. Specifying the `pdunit -c` operand is not necessary.

Figure 25-28 shows an example of a network configuration when IP addresses are not inherited.

Figure 25-28: Example of network configuration when IP addresses are not inherited



Notes (applicable to the standby system switchover facility only)

- When you use HiRDB/Single Servers, specify the host name of the primary system and secondary system in the PDHOST operand of the client environment definition.
- When a unit in the system manager does not inherit IP addresses after system switchover, specify the host names of the primary system and secondary system in the PDHOST operand of the client environment definition. When a unit in the front-end server does not inherit IP addresses after system switchover, specify the host names of the primary system and secondary system in the PDFESHOST operand of the client environment definition. If you specify the host names in this manner, changing the value specified in the PDHOST or PDFESHOST operand is not necessary even after system switchover occurs.
- If the running system switches from the primary system to the secondary system when the host names of the primary system and secondary system are specified in the PDHOST operand, a UAP will attempt to connect to the primary system (standby system). Because such a UAP connects to the secondary system (running system) after the connection could not be established, the UAP processing time increases by the amount of time it takes to establish this new connection. To resolve this problem, set the host name for the client connection to an alias IP address and make sure that IP addresses are inherited. For details about specifying host names in such cases, see *25.4 Handling of host names depending on whether or not IP addresses are inherited*.

25.4 Handling of host names depending on whether or not IP addresses are inherited

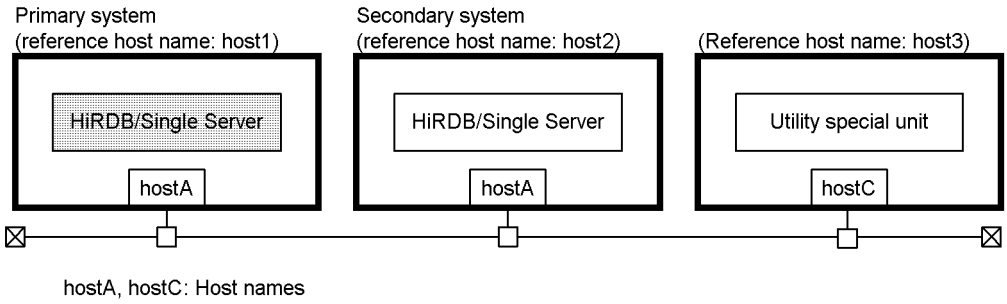
This section explains the procedures for handling (specifying) host names depending on whether or not IP addresses are inherited.

25.4.1 HiRDB/Single Server

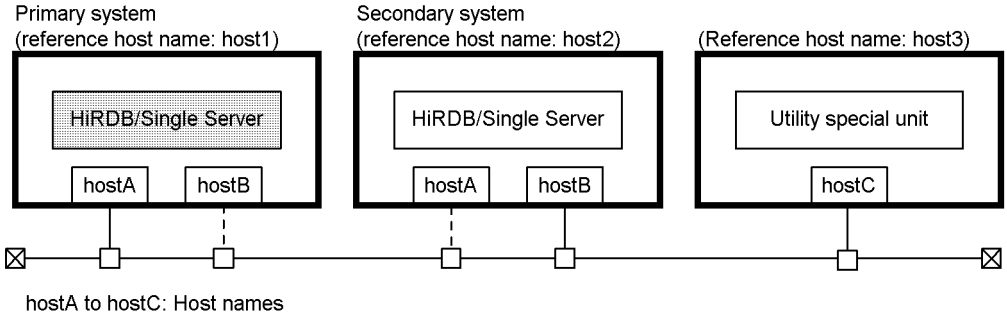
(1) IP addresses inherited

The following explains the procedure for handling (specifying) host names when IP addresses are inherited. The system configuration in this case is a 1-to-1 switchover configuration.

System configuration example (when switching the IP address)



System configuration example (when switching LAN adapters)



Item	Host name to be specified

Host name specified in system common definition	`pdunit -x hostA`
	`pdunit -x hostC`
	`pdstart -x hostA`

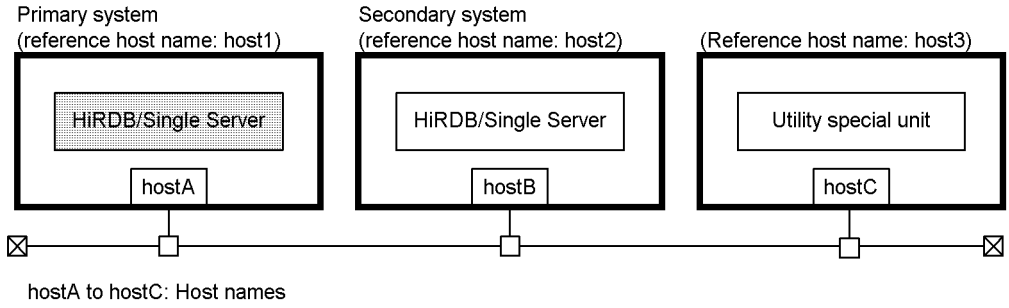
Item		Host name to be specified
Host name specified in unit control information definition		`pd_hostname=host1`
Host name specified during `rlogin`		`hostA`
Host name specified in operation commands*	For single server	`hostA` (host name should be omitted)
	For utility special unit	`hostC`
Host name specified in utilities	For single server	`hostA` (host name should be omitted)
	For utility special unit	`hostC`
Host name displayed in messages		`hostA`
Host name displayed in statistical information		`hostA`
Host name specified in `PDHOST` operand of client environment definition		`hostA`

* The unit identifier may be specified instead of the host name.

(2) IP addresses not inherited

The following explains the procedure for handling (specifying) host names when IP addresses are not inherited. The system configuration in this case is a 1-to-1 switchover configuration.

System configuration example



Hint:

If the default host names are the same (`host1` = `host2`), HiRDB cannot recognize a system switchover. Therefore, use different default host names.

Item		Host name to be specified

Host name specified in system common definition		`pdunit -x hostA -c hostB`
		`pdunit -x hostC`
		`pdstart -x hostA`

Host name specified in unit control information definition		`pd_hostname=host1`
Host name specified during `rlogin`		`hostA` or `hostB` (specify the host name of the running system)
Host name specified in operation commands[1]	For single server	`hostA` (host name should be omitted)
	For utility special unit	`hostC`
Host name specified in utilities	For single server	`hostA` (host name should be omitted)
	For utility special unit	`hostC`
Host name displayed in messages		`hostA`
Host name displayed in statistical information		`hostA`
Host name specified in `PDHOST` operand of client environment definition		`hostA` and `hostB`[2]

[1] The unit identifier may be specified instead of the host name.

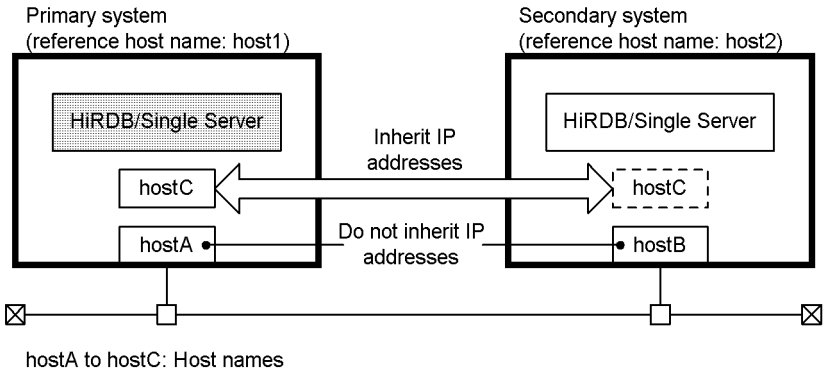
[2] Specify the host names of the primary system and secondary system in the `PDHOST` operand of the client environment definition. If you specify the host names in this manner, changing the value specified in the `PDHOST` operand will not be necessary, even after a system switchover. However, when the running system switches from the primary system to the secondary system, a UAP attempts to connect to the primary system (standby system). Because the UAP connects to the secondary system (running system) after the connection could not be established, the UAP processing time increases by the amount of time it takes to establish this new connection. To resolve this problem, set the host name for the client connection to an alias IP address and make sure that IP addresses are inherited. For details about specifying host names in such cases, see *(3) IP addresses not inherited (only IP addresses for client connection inherited)*.

(3) IP addresses not inherited (only IP addresses for client connection inherited)

The following explains the procedure for handling (specifying) host names when the IP addresses HiRDB/Single Servers use are not inherited, but IP addresses for client

connections are inherited. The system configuration in this case is a 1-to-1 switchover configuration.

System configuration example



Explanation

- `hostA` and `hostB` are the host names the HiRDB/Single Server uses. These hosts do not inherit IP addresses.
- `hostC` is the host name (alias IP address) used for client connection. This host inherits IP addresses.

Hint:

If the default host names are the same (`host1` = `host2`), HiRDB cannot recognize a system switchover. Therefore, use different default host names.

Item	Host name to be specified
Host name specified in system common definition	`pdunit -x hostA -c hostB` `pdstart -x hostA`
Host name specified in unit control information definition	`pd_hostname=host1`
Host name specified during `rlogin`	`hostA` or `hostB` (specify the host name of the running system)
Host name specified in operation commands*	`hostA` (host name should be omitted)
Host name specified in utilities	`hostA` (host name should be omitted)
Host name displayed in messages	`hostA`

Item	Host name to be specified
Host name displayed in statistical information	hostA
Host name specified in PDHOST operand of client environment definition	hostC

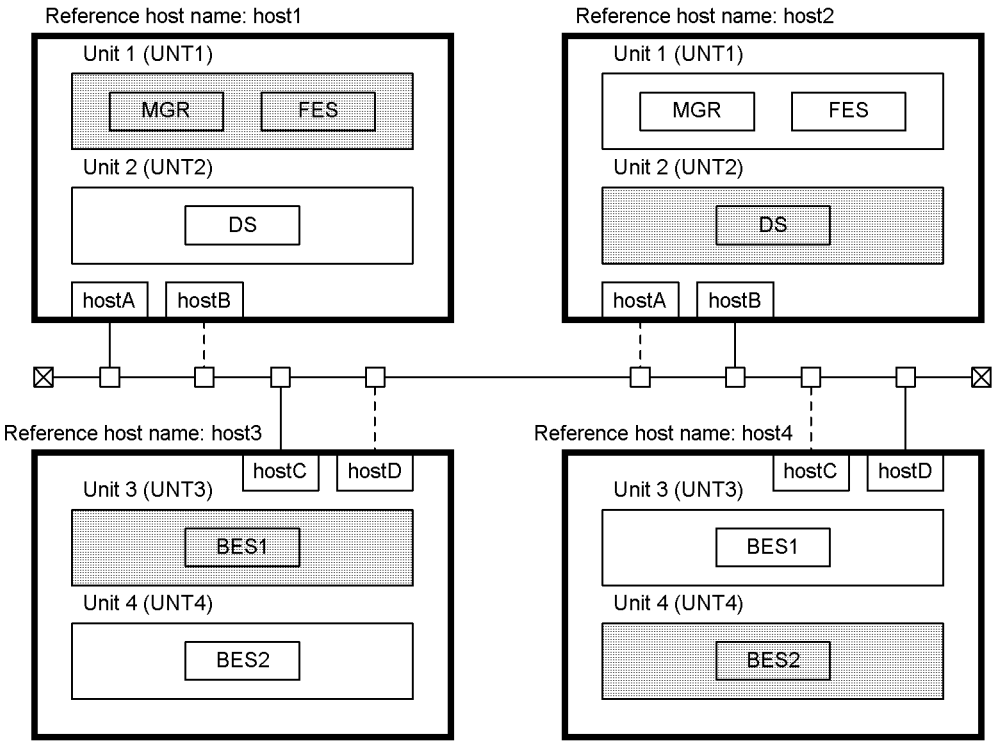
* The unit identifier may be specified instead of the host name.

25.4.2 HiRDB/Parallel Server

(1) IP addresses inherited

The following explains the procedure for handling (specifying) host names when IP addresses are inherited. The system configuration in this case is a mutual system switchover configuration.

System configuration example



hostA to hostD: Host names
UNT1 to UNT4: Unit names
Note: Shaded units are in the primary system.

Item		Host names to be specified

Host names specified in system common definition		`pdunit -x hostA -u UNT1`
		`pdunit -x hostB -u UNT2`
		`pdunit -x hostC -u UNT3`
		`pdunit -x hostD -u UNT4`

Host names specified in the unit control information definition	Unit 1	`pd_hostname=host1`
	Unit 2	`pd_hostname=host2`
	Unit 3	`pd_hostname=host3`
	Unit 4	`pd_hostname=host4`

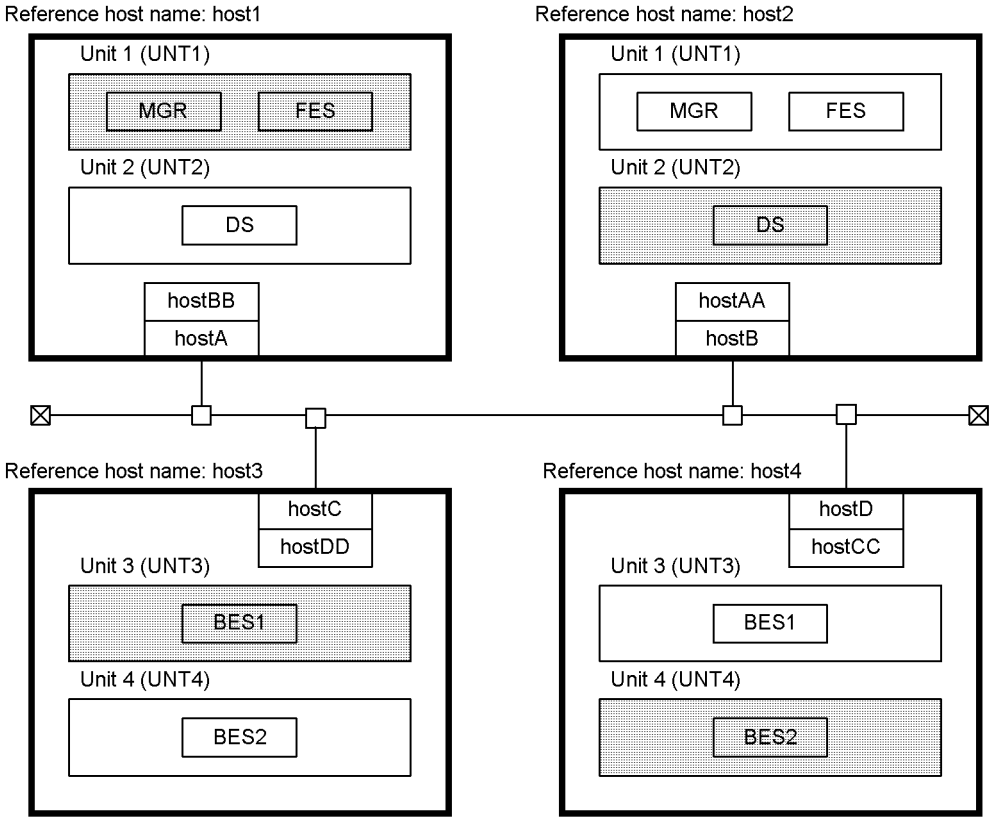
Item		Host names to be specified
Host name specified during `rlogin`	`rlogin` in server machine where `BES1` is located	`hostC`
	`rlogin` in server machine where `BES2` is located	`hostD`
Host name specified in operation commands*	Execution of operation commands in `BES1`	`hostC`
	Execution of operation commands in `BES2`	`hostD`
Host name specified in utilities	Execution of utilities in `BES1`	`hostC`
	Execution of utilities in `BES2`	`hostD`
Host name displayed in messages	Messages for `BES1`	`hostC`
	Messages for `BES2`	`hostD`
Host name displayed in statistical information	Statistical information about `BES1`	`hostC`
	Statistical information about `BES2`	`hostD`
Host name specified in the following client environment definitions: • `PDHOST` • `PDFESHOST`		`hostA`

* The unit identifier may be specified instead of the host name.

(2) IP addresses not inherited

The following explains the procedure for handling (specifying) host names when IP addresses are not inherited. The system configuration in this case is a mutual system switchover configuration.

System configuration example



hostA to hostD: Host names used by the primary systems
hostAA to hostDD: Host names used by the secondary systems
UNT1 to UNT4: Unit names
Note: The shaded units are in the primary system.

Hint:

If the standard host names are the same (`host1` = `host2`, `host3` = `host4`), HiRDB cannot recognize a system switchover. Therefore, use different default host names.

Item		Host names to be specified
Host names specified in system common definition		pdunit -x hostA -u UNT1 -c hostAA pdunit -x hostB -u UNT2 -c hostBB pdunit -x hostC -u UNT3 -c hostCC pdunit -x hostD -u UNT4 -c hostDD
Host names specified in the unit control information definition	Unit 1	pd_hostname=host1
	Unit 2	pd_hostname=host2
	Unit 3	pd_hostname=host3
	Unit 4	pd_hostname=host4
Host name specified during rlogin	rlogin in server machine where BES1 is located	hostC or hostD (specify the host name of the running system)
	rlogin in server machine where BES2 is located	hostC or hostD (specify the host name of the running system)
Host name specified in operation commands[1]	Execution of operation commands in BES1	hostC
	Execution of operation commands in BES2	hostD
Host name specified in utilities	Execution of utilities in BES1	hostC
	Execution of utilities in BES2	hostD
Host name displayed in messages	Messages for BES1	hostC
	Messages for BES2	hostD
Host name displayed in statistical information	Statistical information about BES1	hostC
	Statistical information about BES2	hostD
Host name specified in the following client environment definitions: • PDHOST • PDFESHOST		hostA and hostB[2]

[1] The unit identifier may be specified instead of the host name.

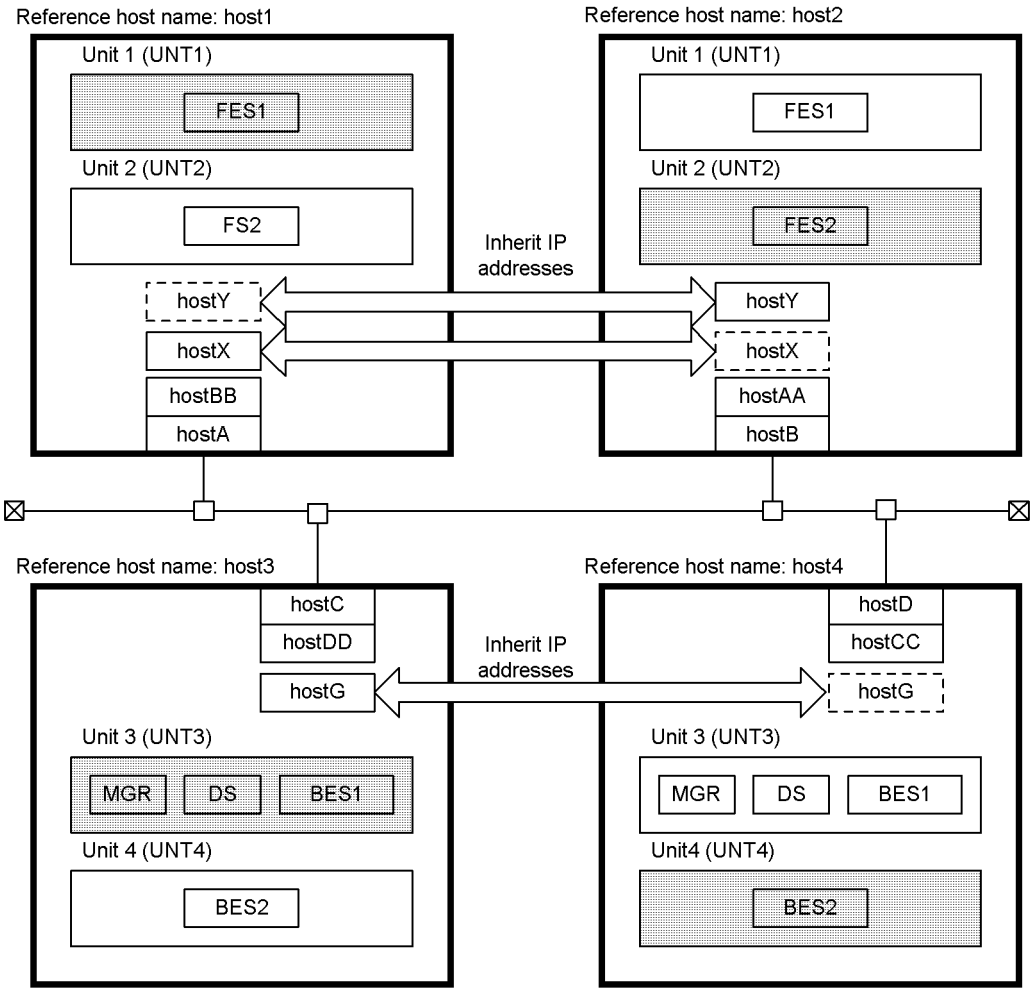
[2] When units in the system manager do not inherit IP addresses, specify the host names of the primary system and secondary system in the PDHOST operand of the client environment definition. When units in the front-end server do not inherit IP addresses, specify the host names of the primary system and secondary system in the

PDFHESHOST operand of the client environment definition. If you specify the host names in this manner, changing the value specified in the PDHOST or PDFESHOST operand is not necessary even after system switchover occurs. However, if the running system switches from the primary system to the secondary system, a UAP attempts to connect to the primary system (standby system). Because the UAP connects to the secondary system (running system) after the connection could not be established, the UAP processing time increases by the amount of time it takes to establish this new connection. To resolve this problem, set the host name for the client connection to an alias IP address and make sure that IP addresses are inherited. For details about specifying host names in such cases, see *(3) IP addresses not inherited (only IP addresses for client connection inherited)*.

(3) IP addresses not inherited (only IP addresses for client connection inherited)

The following explains the procedure for handling (specifying) host names when IP addresses HiRDB/Parallel Servers use are not inherited, but IP addresses for client connections are inherited. The system configuration in this case is a 1:1 mutual system switchover configuration.

System configuration example



hostA to hostD: Host names used by the primary systems
hostAA to hostDD: Host names used by the secondary systems
hostX to hostZ: Host names used by the client connection hosts
UNT1 to UN4: Unit names
Note: Shaded units are in the primary system.

Explanation

- hostA is the host name used by Unit 1 (primary system). hostA does not inherit IP addresses. hostAA is the host name used by Unit 1 (secondary system).
- hostB is the host name used by Unit 2 (primary system). hostB does not inherit IP addresses. hostBB is the host name used by Unit 2 (secondary system).
- hostC is the host name used by Unit 3 (primary system). hostC does not inherit IP addresses. hostCC is the host name used by Unit 3 (secondary system).
- hostD is the host name used by Unit 4 (primary system). hostD does not inherit IP addresses. hostDD is the host name used by Unit 4 (secondary system).
- hostX to hostZ are the host names (alias IP addresses) used for client connections. These hosts inherit IP addresses.

Hint:

If the standard host names are the same (host1 = host2, host3 = host4), HiRDB cannot recognize a system switchover. Therefore, use different default host names.

Item		Host names to be specified
Host names specified in system common definition		pdunit -x hostA -u UNT1 -c hostAA pdunit -x hostB -u UNT2 -c hostBB pdunit -x hostC -u UNT3 -c hostCC pdunit -x hostD -u UNT4 -c hostDD
Host names specified in the unit control information definition	Unit 1	pd_hostname=host1
	Unit 2	pd_hostname=host2
	Unit 3	pd_hostname=host3
	Unit 4	pd_hostname=host4
Host name specified during rlogin	rlogin in server machine where BES1 is located	hostC or hostD (specify the host name of the running system)
	rlogin in server machine where BES2 is located	hostC or hostD (specify the host name of the running system)

Item		Host names to be specified
Host name specified in operation commands*	Execution of operation commands in BES1	hostC
	Execution of operation commands in BES2	hostD
Host name specified in utilities	Execution of utilities in BES1	hostC
	Execution of utilities in BES2	hostD
Host name displayed in messages	Messages for BES1	hostC
	Messages for BES2	hostD
Host name displayed in statistical information	Statistical information about BES1	hostC
	Statistical information about BES2	hostD
Host name specified in client environment definition	PDHOST	hostG
	PDFESHOST	hostX or hostY (specify the host name of the connected front-end server)

* The unit identifier may be specified instead of the host name.

25.5 HiRDB preparations

Executor: HiRDB administrator

This section explains the procedures for preparing HiRDB. This section covers the following topics:

- Conditions and notes
- Preparing a shared disk unit
- Creating HiRDB system definitions
- Client environment definition specification
- Specification examples of host names in HiRDB system definitions and client environment definitions
- RDAREA creation
- Definition of global buffers (standby-less system switchover (1:1) facility only)
- Definition of global buffers (standby-less system switchover (effects distributed) facility only)
- Using audit trail files

25.5.1 Conditions and notes

(1) Standby system switchover facility

(a) Information needed in the primary and secondary systems

It is essential to check that the following information is the same in both systems (primary system and secondary system):

- Versions of HiRDB and related programs
- HiRDB administrator's environment (user ID, group ID, and environment variables)
- Absolute path names of the HiRDB directories
- HiRDB system definitions
- HiRDB file definition formats
- Users' execute-form programs

(b) Notes on environment setup

- Set up the HiRDB environment in both the primary system and the secondary system. Use the same version of HiRDB in the primary system and the secondary

system. Also, when upgrading HiRDB, be sure to upgrade both the primary system and the secondary system.

- Do not create the HiRDB directory on a shared disk.
- When a DNS server is not used, register a re-allocatable IP address in the `hosts` file.
- To operate a HiRDB/Parallel Server in the server mode, install the following products and set up the environment:
- Hitachi HA Toolkit Extension (for the primary and secondary system machines) (not necessary when the cluster software used is HA monitor)
- The system switchover facility is not applicable to a recovery-unnecessary front-end server unit.

Notes on using ClusterPerfect

Use the following command to stop the ClusterPerfect daemon before setting up the HiRDB environment; to execute this command, you must have root privileges:

```
# /etc/rc.d/init.d/dncware_daemon stop
```

Execute the following command to start the ClusterPerfect daemon:

```
# /etc/rc.d/init.d/dncware_daemon start
```

(2) Standby-less system switchover (1:1) facility

- Install and set up the environment for the following products:
  - HiRDB Advanced High Availability (for all server machines)
  - Hitachi HA Toolkit Extension (for normal BES and alternate BES units) (not necessary when the cluster software used is HA monitor)
- Define only back-end servers on the normal BES units and alternate BES units. The standby-less system switchover (1:1) facility cannot be used on units that include servers other than back-end servers.
- There is no need to provide a HiRDB directory for an alternate BES.

  Because the standby-less system switchover (1:1) facility uses the HiRDB directory of an alternate BES unit, there is no need to provide HiRDB directories specifically for the alternate BESs. That is, the `pdsetup` command is not necessary for an alternate BES.
- Allocate system definition files.

  In each unit comprising the group, allocate a back-end server definition file to each back-end server. The parameters to be set in the unit control information definition as the default values for back-end server definition must be defined in

the system common definition or a back-end server definition file.

(3) Standby-less system switchover (effects distributed) facility

- Install and set up the environment for the following products:
  - HiRDB Advanced High Availability (for all server machines)
  - Hitachi HA Toolkit Extension (for the regular units and accepting units) (not necessary when the cluster software used is HA monitor)
- Define only back-end servers on regular units and accepting units. Units that include servers other than back-end servers cannot be used with the standby-less system switchover (effects distributed) facility.
- There is no need to provide a HiRDB directory for a guest BES.

  Because the standby-less system switchover (effects distributed) facility uses the HiRDB directory of an accepting unit, there is no need to provide HiRDB directories specifically for the guest BESs. That is, the `pdsetup` command is not necessary for a guest BES.
- Allocate system definition files.

  In each unit comprising the HA group, allocate a back-end server definition file to each back-end server. The parameters to be set in the unit control information definition as the default values for back-end server definitions must be defined in the system common definition or a back-end server definition file.

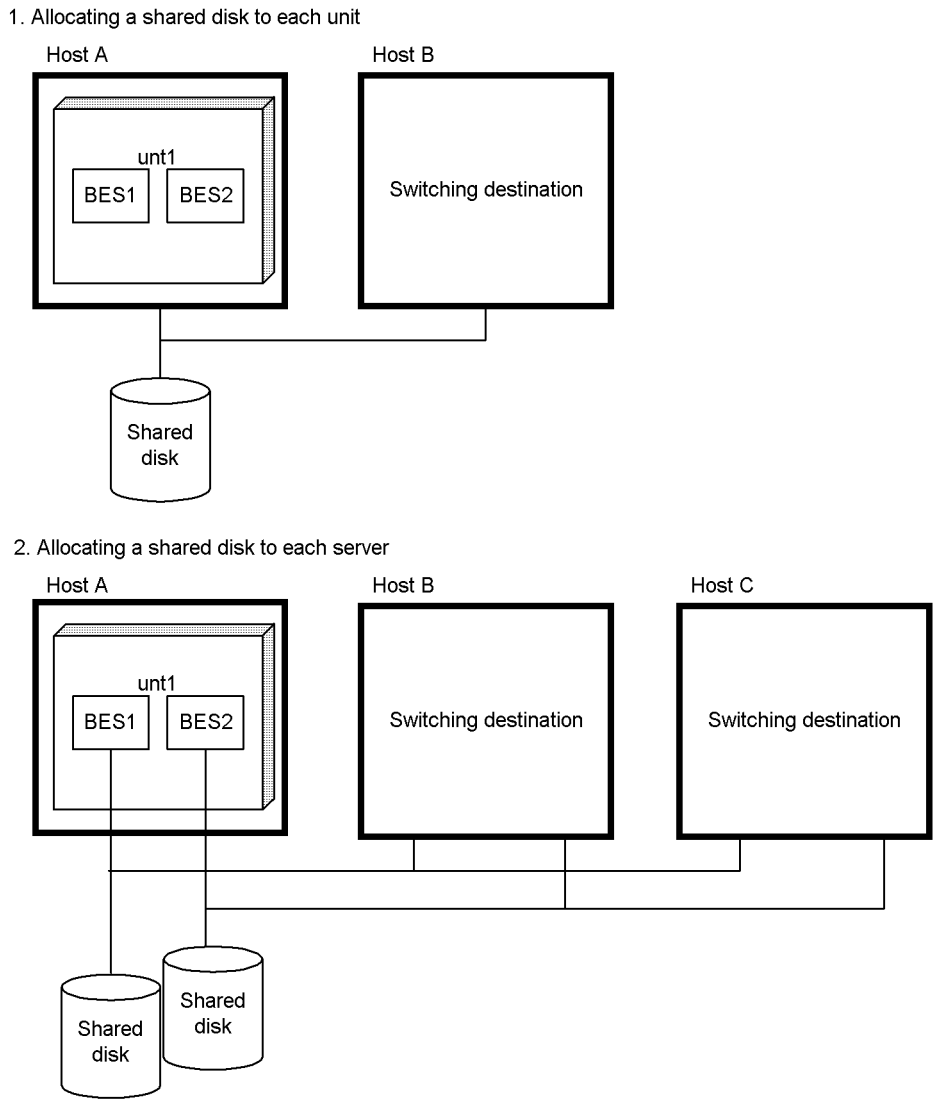
25.5.2 Preparing a shared disk unit

There must be an external hard disk that can be shared between the primary system and the secondary system (or between the normal BES and the alternate BES in the case of the standby-less system switchover facility). This hard disk is called a *shared disk unit*.

(1) Shared disk allocation

Figure 25-29 shows shared disk allocation.

Figure 25-29: Shared disk allocation



Explanation

1. When you use the standby system switchover facility or the standby-less system switchover (1:1) facility, allocate a shared disk to each unit because system switchover occurs on a unit-by-unit basis.
2. When you use the standby-less system switchover (effects distributed) facility, allocate a shared disk to each server because system switchover

occurs on a server-by-server basis. You cannot store information from multiple servers in a single shared disk.

The following HiRDB file system areas are created in a shared disk unit:

- HiRDB file system area for RDAREAs
- HiRDB file system area for system files
- HiRDB file system area for backup files
- HiRDB file system area for unload log files

Notes

- Set up these HiRDB file system areas so that the HiRDBs of both the primary system and the secondary system can reference the shared disk unit using the same path name. When you use the standby-less system switchover (1:1) facility, set up these HiRDB file system areas so that both the normal BES unit and the alternate BES unit can reference the shared disk unit using the same path name. When you use the standby-less system switchover (effects distributed) facility, set up these HiRDB file system areas so that all units within the HA group can reference the shared disk unit using the same path name. However, for the standby-less system switchover (effects distributed) facility, create the unit status file in an independent, non-shared disk that is different from those used for server status files, system log files, and synchronization point dump files.
- The shared disk in which HiRDB file system areas for shared RDAREAs are created must be activated in the write mode from all units. For this reason, the disk must not be deactivated or activated in conjunction with system switchover.
- Only HiRDB file system areas created in character special files can be shared. HiRDB file system areas in regular files cannot be shared.
- Do not use regular files on a shared disk. If system switchover occurs when regular files are in a status that does not apply to shared disks (for example, data remains in the operating system cache even though HiRDB finished writing it to the target files), any updates made to the files may be lost.

(2) Shared disk access control

When the system switchover source and target both attempt to access the shared disk at the same time while the system switchover facility is being used, the database may become corrupted. For this reason, accesses from the system to the shared disk must be controlled. This shared disk access control is performed by either the cluster software or HiRDB.

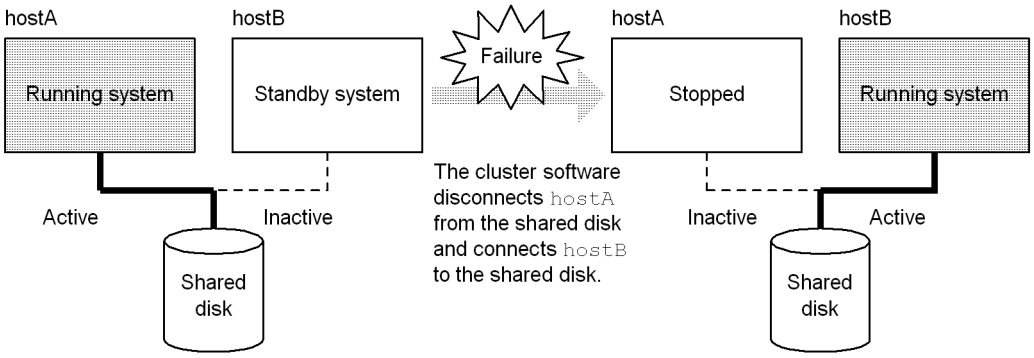
Normally, the method described in *(a) Shared disk access control by the cluster software* is used to perform access control on the shared disk. To use the method

described in *(b) Shared disk access control by HiRDB*, HA monitor 01-08 or later is required.

(a) Shared disk access control by the cluster software

The cluster software can perform access control on the shared disk. It exercises controls so that the running system is active and the standby and stopped systems are inactive, which means that only the running system can access the shared disk. Figure 25-30 shows how the cluster software exercises control over shared disk access.

Figure 25-30: Shared disk access control by the cluster software



Explanation

Because an inactive system cannot access the shared disk, only the running system is capable of accessing the shared disk.

For details about the shared disk switching method (between active and inactive), see the cluster software documentation.

If you are using HA monitor, you must specify the `disk` operand in the HA monitor's `servers` definition statement.

(b) Shared disk access control by HiRDB

To use HiRDB to perform access control on the shared disk, HA monitor 01-08 or later is required.

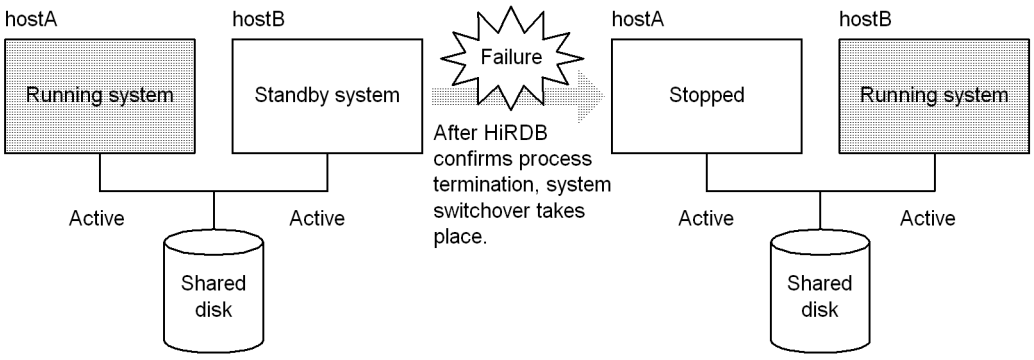
HiRDB can perform access control on the shared disk. In such a case, shared disk switching (between active and inactive) is not performed. System switchover takes place in the following sequence:

1. A failure that results in system switchover occurs.
2. HiRDB confirms that all processes (HiRDB processes) have terminated in the source system.
3. System switchover takes place.

4. The target system starts accessing the shared disk.

Figure 25-31 shows the shared disk access control that is provided by HiRDB.

Figure 25-31: Shared disk access control by HiRDB



■ Criteria

You should use HiRDB to perform shared disk access control in the following cases:

- When the Linux OS in being used

  Character special files on LVM are not supported in the Linux version, but LVM is required by a shared disk whose access is to be controlled by the HA monitor. For this reason, the method described in *(a) Shared disk access control by the cluster software* cannot be used.

- When shared RDAREAs are used

  When shared RDAREAs are used, the shared disk that contains a shared RDAREA must be made active from all server machines where a back-end server is located. If an updatable back-end server and a reference-only back-end server are both located in the same server machine, the updatable back-end server becomes the target of system switchover. If shared disk switching occurs in such a case, the shared RDAREA can no longer be referenced from the reference-only back-end server. Therefore, the method described in *(a) Shared disk access control by the cluster software* cannot be used.

- When Real Time SAN Replication is used in the log-only synchronous method

  When Real Time SAN Replication is used in the log-only synchronous method, TrueCopy is used at the log application site to copy system files from a remote location. When TrueCopy is used, LVM cannot be used, but LVM is required for a shared disk whose access is to be controlled by the HA

monitor. For this reason, the method described in *(a) Shared disk access control by the cluster software* cannot be used.

- HiRDB environment settings

  You must specify the following operands in the HiRDB system definition:

  - `pd_ha_prc_cleanup_check = Y`

    When `Y` is specified in this operand, system switchover takes place only after all processes have terminated in the unit. In the case of the standby-less system switchover (effects distributed) facility, system switchover is executed after all processes have terminated at the back-end servers.

  - `pd_ha_switch_timeout = Y`

    System switchover may be delayed for a reason such as ongoing disk I/O processing. When `Y` is specified in this operand, the HA monitor treats it as a server (HiRDB) slowdown and resets the system, so that system switchover can take place.

- HA monitor environment settings

  The following are the operand specification requirements in the HA monitor's `servers` definition statement:

  - `pairdown`

    You must specify `use:serv_slow` in this operand.

    Termination of processes may not be confirmed, such as when a process does not terminate at the source system or when HiRDB slows down. Such an event prevents system switchover from being executed. When this operand is specified, the system is reset to allow execution of system switchover when termination of processes cannot be confirmed for a reason such as a slowdown.

  - `disk`

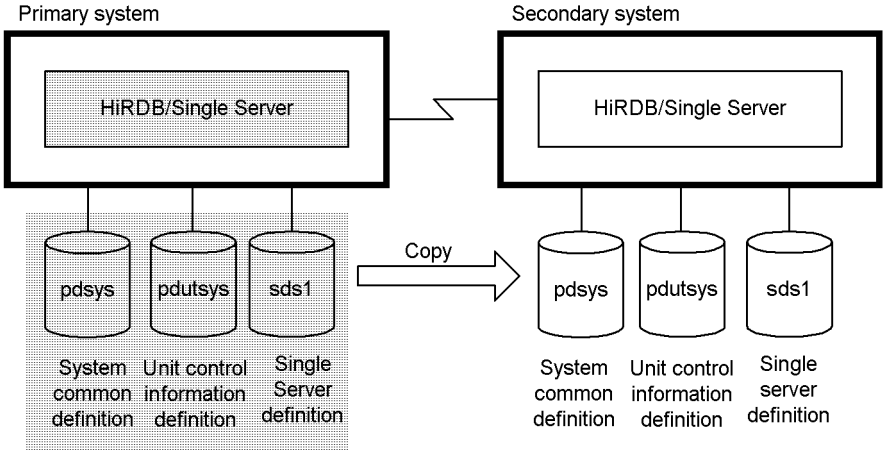
    Omit this operand because the HA monitor is not used to perform access control on the shared disk.

25.5.3 Creating HiRDB system definitions

(1) Standby system switchover facility

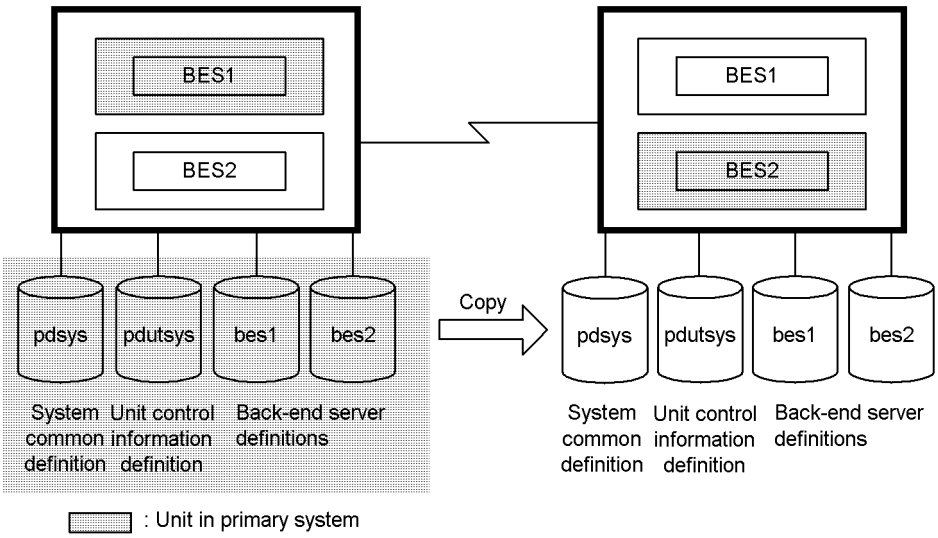
Use the same HiRDB system definitions in the primary system and the secondary system. Create the HiRDB system definitions for the primary system, then copy those HiRDB system definitions to the secondary system. Figures 25-32 and 25-33 show configuration examples of the HiRDB system definition files.

Figure 25-32: Configuration example of HiRDB system definition files when using the standby system switchover facility (for a HiRDB/Single Server)



Note: This is an example of a 1:1 system switchover configuration.

Figure 25-33: Configuration example of HiRDB system definition files when using the standby system switchover facility (for a HiRDB/Parallel Server)



Note: This is an example of a mutual system switchover configuration.

(2) Standby-less system switchover (1:1) facility

Copy the unit control information definition file and back-end server definition file of the normal BES unit to the alternate BES unit. Change the name of the unit control

information definition file as shown below:

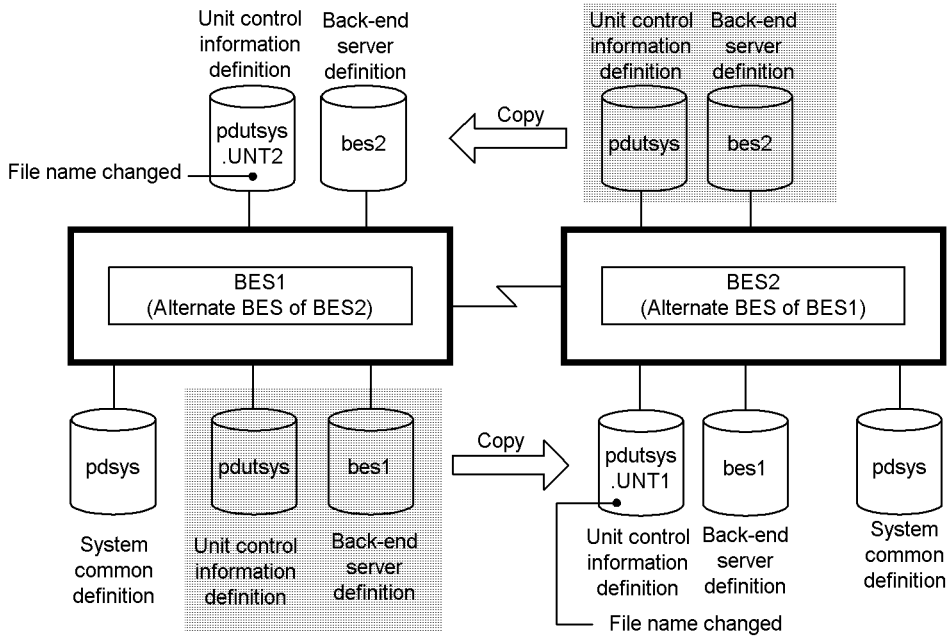
pdutsys.*unit-identifier-of-normal-BES-unit*

Of the operands specified in this definition file, those whose settings become effective during alternation are listed below. For all other operands (other than those listed below), the values that are set in the unit control information definition file of the alternate BES unit are effective.

- pd_syssts_file_name_1 to 7
- pd_syssts_singleoperation
- pd_syssts_initial_error
- pd_syssts_last_active_file
- pd_syssts_last_active_side
- pd_audit
- pd_aud_file_name
- pd_aud_max_generation_size
- pd_aud_max_generation_num
- pd_aud_async_buff_size
- pd_aud_async_buff_count
- pd_ha_switch_timeout
- pd_rpl_hdepath

Figure 25-34 shows a configuration example of the HiRDB system definition files when using the standby-less system switchover facility (mutual alternating configuration).

Figure 25-34: Configuration example of HiRDB system definition files when using the standby-less system switchover facility (mutual alternating configuration)



Note: This is an example of a mutual alternating configuration.

Explanation

- Copy the unit control information definition file and back-end server definition file of the normal BES unit (`BES1`) to the alternate BES unit (`BES2`). Then change the name of the unit control information definition file to `pdutsys.UNT1`.
- Copy the unit control information definition file and back-end server definition file of the normal BES unit (`BES2`) to the alternate BES unit (`BES1`). Then change the name of the unit control information definition file to `pdutsys.UNT2`.

(3) Standby-less system switchover (effects distributed) facility

Table 25-13 shows how system definition files are used when standby-less system switchover (effects distributed) is used.

Table 25-13: Use of system definition files when standby-less system switchover (effects distributed) is used

Definition type	Use of definition files
System common definition	Copy files to all units within the system. Specify in the system common definition the parameters that are to be set as the default values for back-end server definitions.
Unit control information definition	Specify only the following operands (operands that cannot be specified in the system common definition): • pd_unit_id • pd_hostname • pd_ha_unit • pd_rpl_hdepath • pd_ha_restart_failure • pd_ha_acttype • pd_ha_server_process_standby • pd_ha_agent • pd_ha_max_act_guest_servers • pd_ha_max_server_process • pd_ha_resource_act_wait_time • pd_ha_process_count • pd_syssts_file_name_1 ～ 7 • pd_syssts_initial_error • pd_syssts_last_active_file • pd_syssts_last_active_side • pd_syssts_singleoperation Specify all other operands in the system common definition or in back-end server definitions. If any operand other than those listed above is specified, an error occurs (and the KFPS05618-E message is output).
Server common definition	Copy files to all units within the HA group.
Back-end server definition	Copy files to all units within the HA group.

(4) HiRDB system definition operands related to the system switchover facility

Table 25-14 explains the HiRDB system definition operands that relate to the system switchover facility.

Table 25-14: HiRDB system definition operands related to the system switchover facility

Operand name	Explanation and Notes
pd_ha	Specifies that the system switchover facility is to be used.

Operand name	Explanation and Notes
pd_ha_ipaddr_inherit	Specifies whether or not IP addresses are to be inherited after a system switchover. Specify N for units using the rapid system switchover facility. Omit this operand for units using the standby-less system switchover facility. Y: Inherit IP addresses after system switchover. N: Do not inherit IP addresses after system switchover.
pd_ha_unit	Specify nouse for any unit that is not to use the system switchover facility. You must specify nouse for a recovery-unnecessary front-end server.
pd_ha_acttype	Specifies whether the system switchover facility is to be used in the monitor mode or the server mode. The server mode cannot be used when the system switchover facility uses Sun Cluster, HACMP, or ClusterPerfect. monitor: Operate the system switchover facility in the monitor mode. server: Operate the system switchover facility in the server mode.
pd_ha_restart_failure	When operating the system switchover facility in the monitor mode, specifies a command to be executed if the restarting HiRDB fails. This operand has no effect when you use the server mode.
pd_ha_server_process_standby	Specifies whether or not user server hot standby is to be used. Y: Use user server hot standby. N: Do not use user server hot standby.
pd_ha_agent	Specifies the system switchover facility to be used: standbyunit: Rapid system switchover facility server: Standby-less system switchover (1:1) facility activeunits: Standby-less system switchover (effects distributed) facility
pd_ha_transaction pd_ha_trn_queuing_wait_time pd_ha_trn_restart_retry_time	• Specify these operands when you use the transaction queuing facility. • If you specify queuing in the pd_ha_transaction operand and the maximum number of concurrent connections (value of the pd_max_users operand) is exceeded, the HiRDB client will make retries to connect to the HiRDB server for only the amount of time that is equal to pd_ha_trn_queuing_wait_time + pd_ha_trn_restart_retry_time.

Operand name	Explanation and Notes
pd_ha_switch_timeout	This operand can be specified when the server mode is used. This operand is invalid if it is specified in the monitor mode. This operand specifies whether or not system switchover is to performed without waiting for HiRDB termination processing when termination processing of HiRDB (or the unit for a HiRDB/Parallel Server) during system switchover exceeds the *server failure monitoring time*. Server failure monitoring time refers to the time specified in the patrol operand of HA monitor or Hitachi HA Toolkit Extension. Y: Switch systems without waiting for HiRDB termination processing when HiRDB termination processing during system switchover exceeds the server failure monitoring time. N: Do not switch systems until HiRDB termination processing during system switchover terminates.
pd_ha_prc_cleanup_check	Specifies whether or not system switchover processing is to be placed on hold until HiRDB processes have terminated. For details, see *25.5.2(2) Shared disk access control*.
pd_mode_conf	This operand is related to HiRDB (or unit) startup. Specify this operand as follows: When the monitor mode is used, specify MANUAL1. When the server mode is used, specify one of the following: • MANUAL2 if switch is specified in the switchtype operand of the servers definition of Hitachi HA Toolkit Extension. • MANUAL1 if restart or manual is specified in the switchtype operand of the servers definition of Hitachi HA Toolkit Extension.
pd_hostname	Specifies the standard host name of the primary system. When using the standby-less system switchover facility, specifies the unit's standard host name. (This is the same as when not using the system switchover facility.)

Operand name		Explanation and Notes
pdunit	-x	Specifies the host name of the primary system. When using the standby-less system switchover facility, specifies the unit's host name. (This is the same as when not using the system switchover facility.)
	-u	Specifies the unit identifier.
	-d	Specifies the HiRDB directory name. When using the standby-less system switchover (1:1) facility, specify the same directory name for the normal BES unit and the alternate BES unit. When using the standby-less system switchover (effects distributed) facility, specify the same directory name for all units within the HA group.
	-c	Specifies the host name of the secondary system. Specify this option when not inheriting IP addresses after system switchover. Omit this option when using the standby-less system switchover facility.
	-p	Specifies the port number. Specify this option when using a utility special unit or HiRDB/Parallel Server. When using the standby-less system switchover (1:1) facility, specify the same port number for the normal BES unit and the alternate BES unit. When using the standby-less system switchover (effects distributed) facility, specify the same port number for all units within the HA group.
pdstart	-c	Specifies the alternate BES name. Specify this option when using the standby-less system switchover (1:1) facility.
	-g	When using the standby-less system switchover (effects distributed) facility, specify the identifier of the HA group that constitutes the set of units that become server switching destinations.
pdbuffer	-c	Specify this option when allocating global buffers that the alternate portion uses when alternating units. Specify this option when using a standby-less system switchover facility. For details when using the standby-less system switchover (1:1) facility, see *25.5.7 Definition of global buffers (standby-less system switchover (1:1) facility only)*; for details when using the standby-less system switchover (effects distributed) facility, see *25.5.8 Definition of global buffers (standby-less system switchover (effects distributed) facility only)*.
pdhagroup	-g	To use the standby-less system switchover (effects distributed) facility, you define an HA group that will constitute the set of units that will become server switching destinations. Specify an identifier that will uniquely identify this HA group within the system.
	-u	Specifies the unit identifiers of the units that are to comprise the HA group.
pd_ha_max_act_guest_servers		When using the standby-less system switchover (effects distributed) facility, specifies the maximum number of guest BESs that will be permitted to run concurrently in a unit.

Operand name	Explanation and Notes
pd_ha_max_server_process	When using the standby-less system switchover (effects distributed) facility, specifies the maximum permissible number of active user server processes in a unit.
pd_ha_resource_act_wait_time	When the standby-less system switchover (effects distributed) facility is used, specifies the maximum time to wait until the running server's resources are activated when the unit is started.
pd_service_port	Care must be exercised in specifying this operand in a server machine configuration that includes multiple units (including a mutual system switchover configuration). For such a configuration (including a mutual system switchover configuration), use this operand to specify a separate port number for each unit in its unit control information definition. If either of the following specifications is made, system switchover to one of the units fails: • The pd_service_port operand of the system common definition is specified (when the pd_service_port operand of the unit control information definition is not specified). • A port number that is specified in the pd_service_port operand of another unit control information definition is specified in the pd_service_port operand of the unit control information definition.
pd_redo_allpage_put	When Y is specified in this operand, all pages that have been updated since a synchronization point are written into the database during full recovery processing that occurs when HiRDB is restarted. This can eliminate inconsistencies between the original and duplicate volumes that occurred during system switchover. For details about how to handle inconsistencies between the original and duplicate volume, see *18.24 Actions to take when a mismatch occurs between the original and the mirror duplicate*.
pd_ha_mgr_rerun	When notwait is specified in this operand, HiRDB does not wait to receive a processing startup completion notice from each unit when switching system manager units (when starting processing at the switching destination). As a result, system manager units can be switched even when some units are stopped. For details about the operation method, see *25.21 Actions to take when a stopped unit prevents switching of the system manager unit*.

(5) Specifying the switching destination (standby-less system switchover (effects distributed) facility only)

When the standby-less system switchover (effects distributed) facility is used, the method of determining the switching destination differs significantly from when the other system switchover facilities are used.

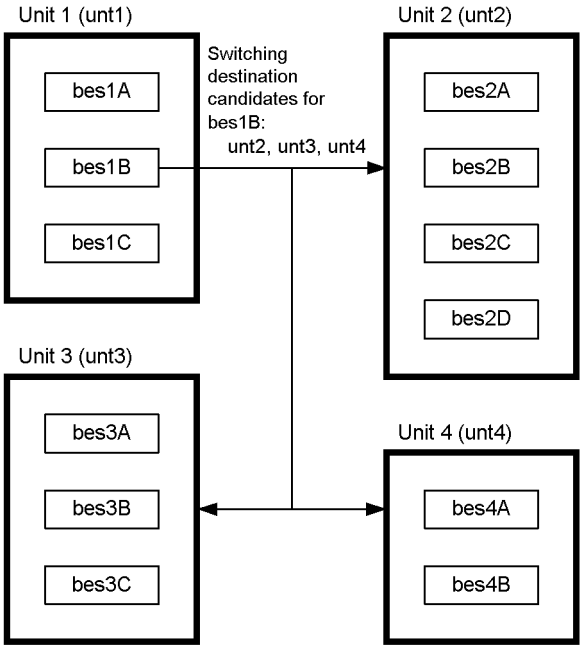
(a) Accepting unit

Because the standby-less system switchover (effects distributed) facility switches systems on a server-by-server basis, a switching destination must be specified for each server. You may specify multiple accepting units for a server. Multiple accepting units are defined as an HA group, you must specify an HA group as the switching destination for each server.

When you use the standby-less system switchover (effects distributed) facility, you can also specify the maximum number of guest BESs that will be permitted to run concurrently in each unit (`pd_ha_max_act_guest_servers`).

Figure 25-35 shows an example of an HA group configuration.

Figure 25-35: HA group configuration example



```
pdhagroup -g hag1 -u unt1,unt2,unt3,unt4

pdstart -t BES -s bes1A -u unt1 -g hag1
pdstart -t BES -s bes1B -u unt1 -g hag1
pdstart -t BES -s bes1C -u unt1 -g hag1
pdstart -t BES -s bes2A -u unt2 -g hag1
pdstart -t BES -s bes2B -u unt2 -g hag1
pdstart -t BES -s bes2C -u unt2 -g hag1
pdstart -t BES -s bes2D -u unt2 -g hag1
pdstart -t BES -s bes3A -u unt3 -g hag1
pdstart -t BES -s bes3B -u unt3 -g hag1
pdstart -t BES -s bes3C -u unt3 -g hag1
pdstart -t BES -s bes4A -u unt4 -g hag1
pdstart -t BES -s bes4B -u unt4 -g hag1
```

(b) HA group definition

You use the HiRDB system definition to define an HA group. Specify a name for the HA group in the `-g` option of the `pdhagroup` operand, and specify in the `-u` option the unit identifiers of the units that will comprise the HA group.

You can specify only one HA group in each system definition.

Example: `pdhagroup -g hag1 -u unt1,unt2,unt3,unt4`

Defines an HA group named `hag1` that consists of `unt1`, `unt2`, and `unt3`.

The following restrictions apply to HA groups:

- A maximum of 32 units can be defined in an HA group.
- All units in an HA group must be allocated in the same network segment.
- The maximum total number of host BESs and guest BES areas (= maximum active guest BESs) that can be defined in a unit within an HA group is 34.

Each unit comprising an HA group must satisfy all the following conditions:

1. Because a unit that contains no host BES (an accepting-only unit) cannot belong to an HA group, each unit belonging to an HA group must contain at least one host BES.
2. All servers that comprise a unit belonging to an HA group must be back-end servers; an HA group unit cannot contain any server whose server type is other than `BES`.
3. The only type of system switchover that can be used for units belonging to an HA group is standby-less system switchover (effects distributed). This means that for units belonging to an HA group, the only value that can be specified in the `pd_ha_agent` operand is `activeunits`.

(c) Specifying an accepting unit

In the HiRDB system definition, you specify in the `-g` option of the `pdstart` command the HA group to which an accepting unit belongs.

You must specify the `-g` option for all servers that belong to a unit to which standby-less system switchover (effects distributed) is applicable.

Example: `pdstart -t BES -s bes1A -u unt1 -g hag1`

When `unt1` or `bes1` terminates abnormally, processing for `bes1` can be accepted by a unit belonging to the HA group named `hag1`.

You should note the following about specifying the `-g` option:

1. Both the regular unit and the accepting unit must be comprised exclusively of back-end servers.
   - `BES` must be specified in the `-t` option.
   - Each unit belonging to the HA group specified by the `-g` option must not contain any server whose server type is not `BES`.
2. The number of servers comprising a regular unit need not be the same as the number of servers comprising an accepting unit.
   - The number of servers in the unit specified by the `-u` option (regular unit) need not be the same as the number of servers in the unit belonging to the HA group specified by the `-g` option (accepting unit).

(d) Specifying the maximum number of concurrently running guest BESs

You can specify in the `pd_ha_max_act_guest_servers` operand of the unit control information definition the maximum number of guest BESs that will be permitted to operate concurrently as running systems in a unit. The purpose of this specification is to reduce the amount of resources required by guest BESs. It can also prevent excessive increases in workload.

Example: `pd_ha_max_act_guest_servers = 2`

The maximum value that can be specified in the `pd_ha_max_act_guest_servers` operand is the number obtained by subtracting the number of servers in the local unit from the number of servers in the HA group. If you specify a value greater than this maximum, the maximum value will be set in the `pd_ha_max_act_guest_servers` operand. The number of host BESs plus the value of the `pd_ha_max_act_guest_servers` operand cannot exceed 34.

The number of guest BESs that are in accepting status in a unit is not restricted. However, when the number of guest BESs that are operating as running systems in a unit reaches the value specified in the `pd_ha_max_act_guest_servers` operand, acceptability is cancelled for all the non-active guest BESs.

Once the number of erroneous BESs in an HA group exceeds the combined total number of free guest areas in the running units in the HA group, any subsequent error will cause some servers to stop and their processing will be suspended.

(6) Allocating server processes following system switchover

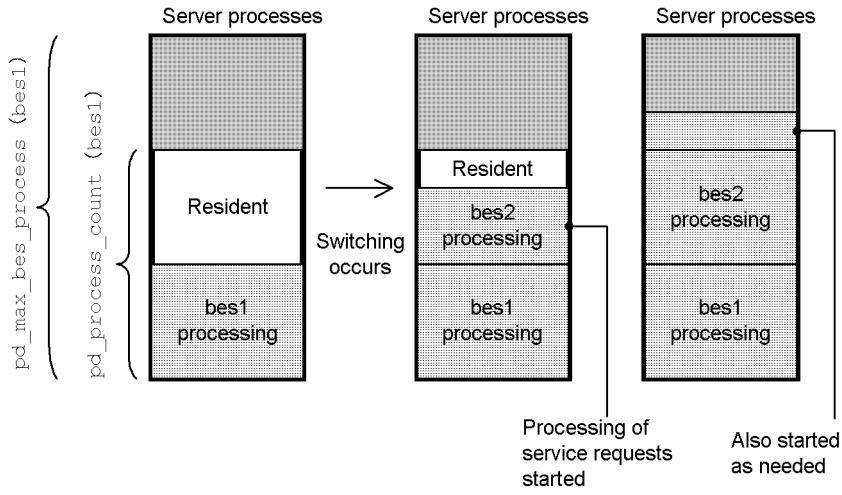
(a) Standby-less system switchover (1:1)

Once standby-less system switchover (1:1) occurs, the alternate BES unit both executes its own processes and assumes the alternate BES's processes. For this to occur, server processes are allocated to the alternate BES's original processes as well as to the normal BES's processes. The number of server processes executing the alternate BES's original processes and assuming the normal BES's processes varies according to need. However, the maximum number of active alternate BES processes (value of the `pd_max_bes_process` operand) is also the maximum for the combined total of the number of processes for both BESs. This prevents an excessive increase in workload at the alternate BES after system switchover. On the other hand, however, you need to be aware that the maximum number of service requests that can be processed concurrently after system switchover is limited to one half of the original. For this reason, when you specify the `pd_max_bes_process` operand for the alternate BES, you should take into consideration both the increase in the unit's workload and the number of service requests that can be processed concurrently.

If a safety margin has been built into the number of resident processes before system switchover (value of the `pd_process_count` operand), and if processes that are not actually processing service requests are resident, you have these resident processes that are not processing service requests available to assume the normal BES's processing after system switchover. As a result, processing performance after system switchover improves.

Figure 25-36 shows allocation of server processes following standby-less system switchover (1:1) (Part 1).

Figure 25-36: Allocation of server processes following standby-less system switchover (1:1) (Part 1)



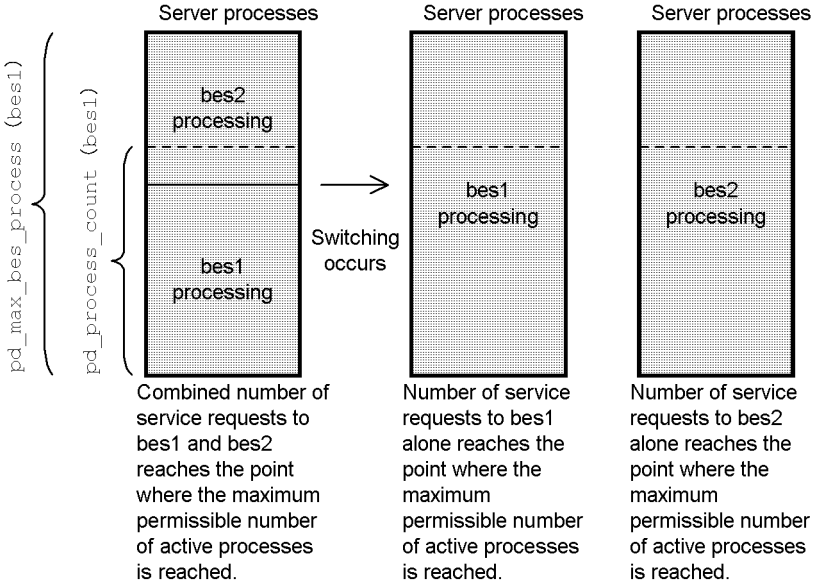
Before system switchover occurs, the maximum number of processes that can be processed concurrently equals the value of the `pd_max_bes_process` operand specified for the alternate BES (`bes1`). Additionally, as many server processes as the value of the `pd_process_count` operand for the alternate BES (`bes1`) can be kept resident.

When system switchover occurs, processing for the normal BES (`bes2`) begins using available resident processes of the alternate BES (`bes1`). Therefore, there is no need to start processes for the normal BES (`bes2`) and the processing of the normal BES (`bes2`) resumes immediately following system switchover. Moreover, there is no need to start standby processes for the normal BES (`bes2`) before system switchover.

Once all resident processes are being used, additional processes are started as needed. However, the total number of processes is limited to the value of the `pd_max_bes_process` operand for the alternate BES (`bes1`).

Figure 25-37 shows allocation of server processes following standby-less system switchover (1:1) (Part 2).

Figure 25-37: Allocation of server processes following standby-less system switchover (1:1) (Part 2)



After system switchover, while the alternate BES (`bes1`) is handling the processes of the normal BES (`bes2`), processes that are started as needed within the value of the `pd_max_bes_process` operand of the alternate BES are allocated to handle the processes of the alternate BES (`bes1`) as well as of the normal BES (`bes2`).

Where there are processing requests only for the alternate BES (`bes1`), the number of processes up to the value of the `pd_max_bes_process` operand for the alternate BES (`bes1`) can be executed concurrently for the alternate BES (`bes1`).

Where there are processing requests only for the normal BES (`bes2`), the number of processes up to the value of the `pd_max_bes_process` operand for the alternate BES (`bes1`) can be executed concurrently for the alternate BES (`bes2`).

(b) Standby-less system switchover (effects distributed) facility

Even though standby-less system switchover (effects distributed) has occurred, an accepting unit can continue to accept guest servers until the number of running guest servers reaches the value of the `pd_ha_max_act_guest_servers` operand.

At an accepting unit, the host BESs and guest BESs individually start server processes up to the maximum number of processes that can be started (value of the `pd_max_bes_process` operand). However, the total number of server processes in a unit is limited to the value of the `pd_ha_max_server_process` operand. This prevents an excessive increase in workload at the accepting unit. However, you should

be aware that there may be an upper limit to the number of service requests that can be processed concurrently after system switchover. For this reason, when you specify the pd_ha_max_server_process operand, you should take into consideration both the increase in the unit's workload following system switchover and the number of service requests that can be processed concurrently.

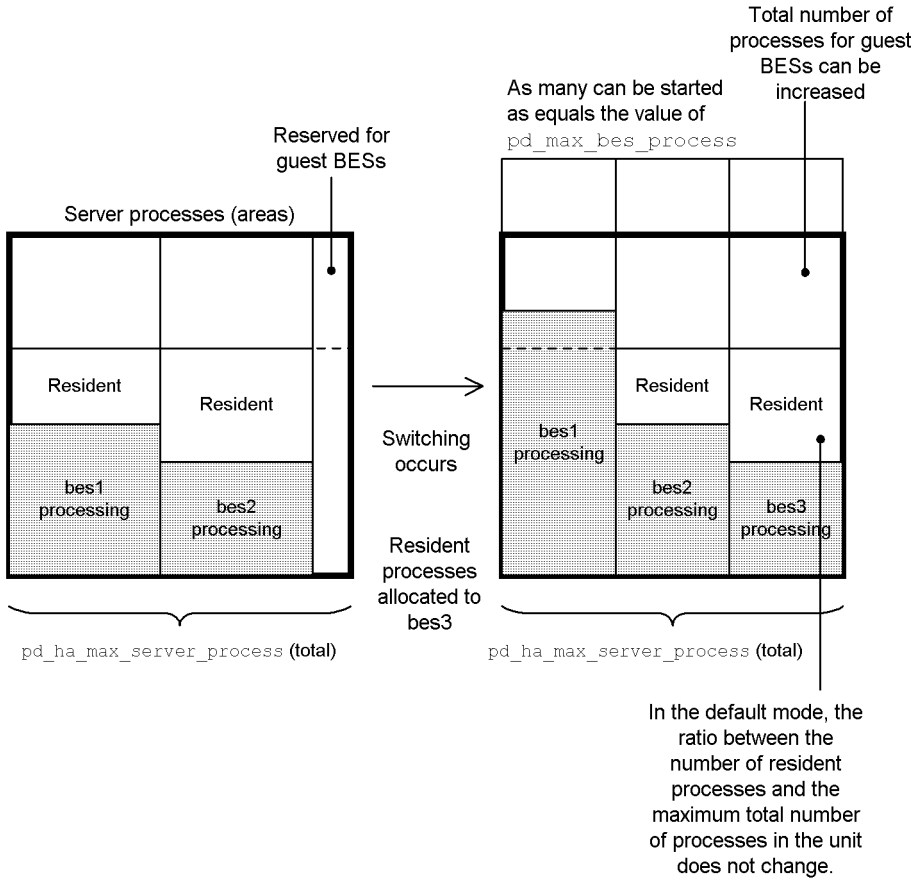
If a safety margin has been built into the number of resident processes before system switchover (value of the pd_process_count operand) and if processes that are not actually processing service requests are resident, you have these resident processes that are not processing service requests available to assume the normal BES's processing after system switchover. As a result, processing performance after system switchover improves. On the other hand, when the number of resident processes is set too large, processes that are not processing service requests may cause the number of processes to reach the value of the pd_ha_max_server_process operand. As a result, it may not be possible to process additional service requests even though the number of processes that have been started by other servers has not reached the value of the pd_max_bes_process operand. It is advisable to set the ratio between the total number of resident processes in units and the total of the maximum number of running processes to remain the same before and after guest servers are accepted. In this way, the total number of resident processes in units after guest servers are accepted is restricted by the pd_ha_process_count operand. The actual number of resident processes is either the number obtained by allocating proportionally the value of the pd_ha_process_count operand based on the values of the pd_process_count operands of the servers that are running in the unit, or the actual value of the pd_process_count operand, whichever is smaller.

The meanings of the operands related to number of processes are explained below:

- pd_ha_max_act_guest_servers: Number of guest BESs that can be accepted (maximum number in accepting status)
- pd_ha_max_server_process: Maximum number of running processes, for both guest BESs and host BESs
- pd_ha_process_count: Maximum number of resident processes, for both guest BESs and host BESs

Figure 25-38 shows allocation of server processes following standby-less system switchover (effects distributed) (Part 1)

Figure 25-38: Allocation of server processes following standby-less system switchover (effects distributed) (Part 1)



Before system switchover occurs, each host BES (`bes1` and `bes2`) can execute concurrently as many processes as the value of its `pd_max_bes_process` operand. For each, as many server processes as the value of its `pd_process_count` operand can also be made resident.

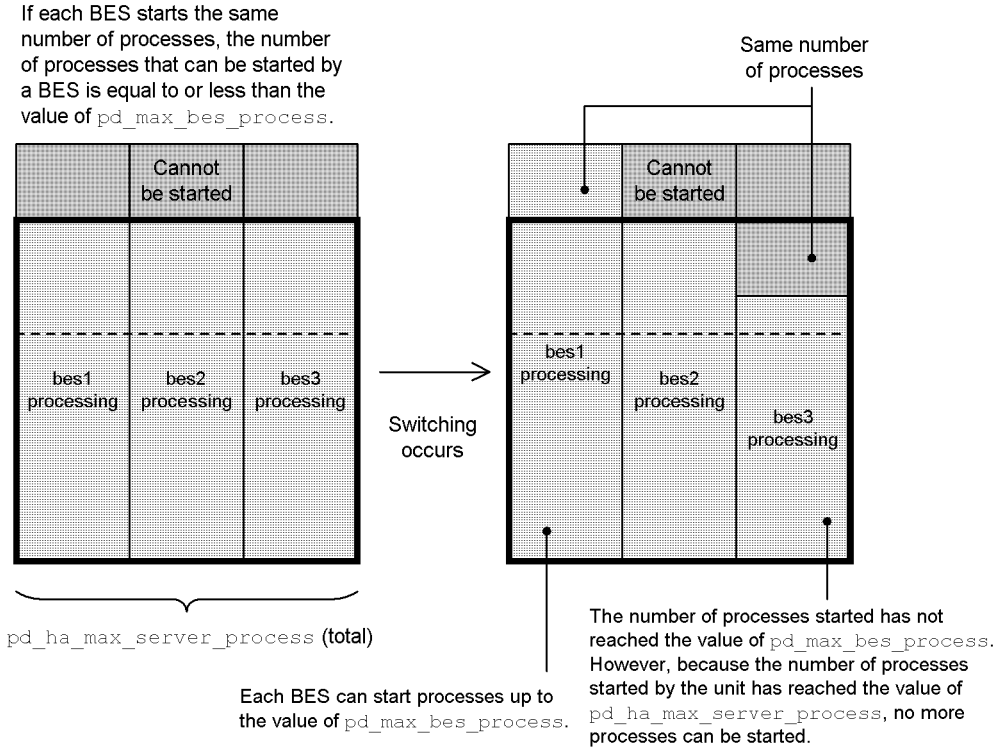
When system switchover occurs, resident processes in the host BESs (`bes1` and `bes2`) are used to provide processes for the guest BES (`bes3`). Therefore, there is no need to start processes for the guest BES (`bes3`) and processing of the guest BES (`bes3`) can begin immediately following system switchover. Moreover, there is no need to start standby processes for the guest BES (`bes3`) before system switchover.

Each server starts processes as needed up to the value of its `pd_max_bes_process` operand, but the combined total number of server processes in the unit is limited by the value of the `pd_ha_max_server_process` operand.

Also, the number of resident processes in each server is adjusted so that the combined total number of resident processes in the units equals the value of the `pd_ha_process_count` operand. The value of the `pd_ha_process_count` operand is allocated among the servers so that the number of resident processes for each server after adjustment maintains the ratio determined by each server's `pd_process_count` operand value.

Figure 25-39 shows allocation of server processes following standby-less system switchover (effects distributed) (Part 2).

Figure 25-39: Allocation of server processes following standby-less system switchover (effects distributed) (Part 2)



After system switchover and once the guest BES (`bes3`) has been accepted, the processes of the host BESs (`bes1` and `bes2`) and the processes of the guest BES (`bes3`) are started, as long as the number of processes in the unit does not exceed the value of the `pd_ha_max_server_process` operand.

If the number of processing requests to a particular host BES (`bes1`, for example) is especially large, processes can be executed concurrently up to the value of the `pd_max_bes_process` operand of that host BES (`bes1`). However, the number of

processing requests that can be handled by other servers (bes3, for example) decreases accordingly.

25.5.4 Client environment definition specification

Pay attention to the following notes about specifying a client environment definition when using the standby system switchover facility.

(1) HiRDB/Single Server

When inheriting IP addresses after a system switchover, specify the host name of the primary system only in the PDHOST operand of the client environment definition. When not inheriting IP addresses, specify the host names of the primary system and the secondary system.

(2) HiRDB/Parallel Server

When a unit in the system manager inherits IP addresses after system switchover, specify the host name of the primary system only (host name of the system manager) in the PDHOST operand of the client environment definition. When not inheriting IP addresses after system switchover, specify the host names of the primary system and the secondary system.

When a unit in the front-end server inherits IP addresses after system switchover, specify the host name of the primary system only (host name in the front-end server) in the PDFESHOST operand. When the unit does not inherit IP addresses after system switchover, specify the host names of the primary system and the secondary system. For details about the PDHOST and PDFESHOST operands, see the manual *HiRDB Version 8 UAP Development Guide*.

25.5.5 Specification examples of host names in HiRDB system definitions and client environment definitions

Examples of specifying host names when using the standby system switchover facility are explained below.

(1) When inheriting IP addresses

Examples of specifying the HiRDB system definition and client environment definition (host names) are shown below.

- System common definition

```
pdunit -x pkghost -u UNT1 -d /hirdb_x            1
```

- Unit control information definition

```
set pd_hostname = mainhost                          2
```

■ Client environment definition

```
export PDHOST = pkghost                             3
```

Explanation

1. Specify the host name* of a re-allocatable IP address in the `-x` option of the `pdunit` operand.
2. Specify the host name of the primary system in the `pd_hostname` operand.
3. Specify the host name* of a re-allocatable IP address (the host name of the system manager for a HiRDB/Parallel Server) in the `PDHOST` operand of the client environment definition.

   In the case of multiple front-end servers, specify the host names* of re-allocatable IP addresses (host names of front-end servers) in the `PDFESHOST` operand.

* In the case of VERITAS Cluster Server or ClusterPerfect, the host name of a logical IP address (the host name of the IP address set in the IP type resource) is specified. In the case of SunCluster, the logical host name registered in SunCluster is specified.

(2) When not inheriting IP addresses

Examples of specifying the HiRDB system definition and client environment definition (host names) are shown below.

System common definition

```
pdunit -x mainhost -u UNT1 -d /hirdb/pddir_s -c reservedhost      1
```

Unit control information definition

```
set pd_hostname = mainhost                                        2
```

Client environment definition

```
export PDHOST = mainhost,reservedhost                                    3
```

Explanation

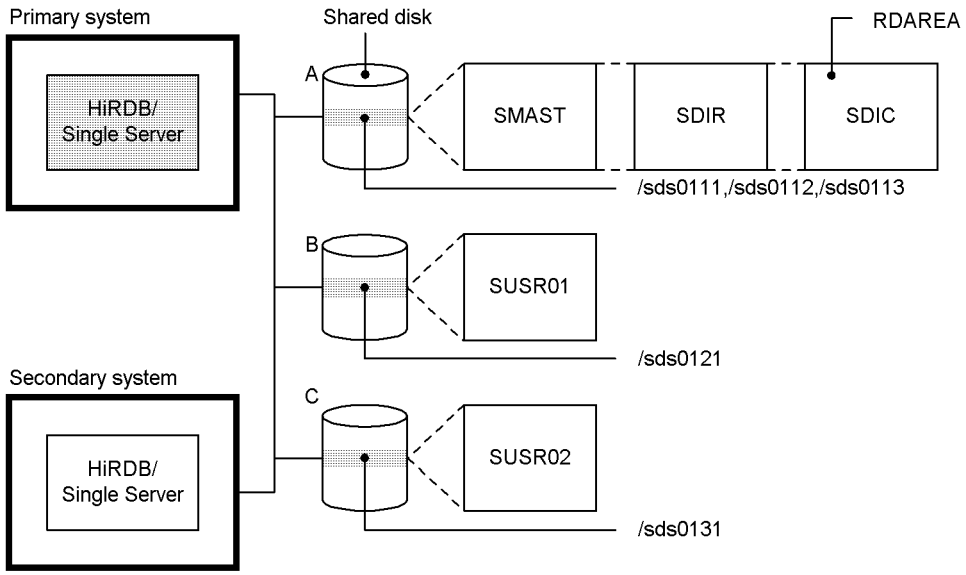
1. Specify the host name of the primary system in the `-x` option of the `pdunit` operand. Specify the host name of the secondary system in the `-c` option.
2. Specify the host name of the primary system in the `pd_hostname` operand.
3. Specify the host name of the primary system and the secondary system (host name of the system manager for a HiRDB/Parallel Server) in the `PDHOST` operand of the client environment definition.

   In the case of multiple front-end servers, specify the host names of the primary system and secondary system (host names of front-end servers) in the `PDFESHOST` operand.

25.5.6 RDAREA creation

Define RDAREAs in HiRDB file system areas for RDAREAs. Figures 25-40 and 25-41 show definition examples of creating user RDAREAs and system RDAREAs in different HiRDB file system areas on shared disks. Each system configuration is described following the figure. When you use the standby-less system switchover (effects distributed) facility, you define RDAREAs in HiRDB file system areas created on a different shared disk for each server.

Figure 25-40: HiRDB/Single Server system configuration example



Example of create rdarea statement specification

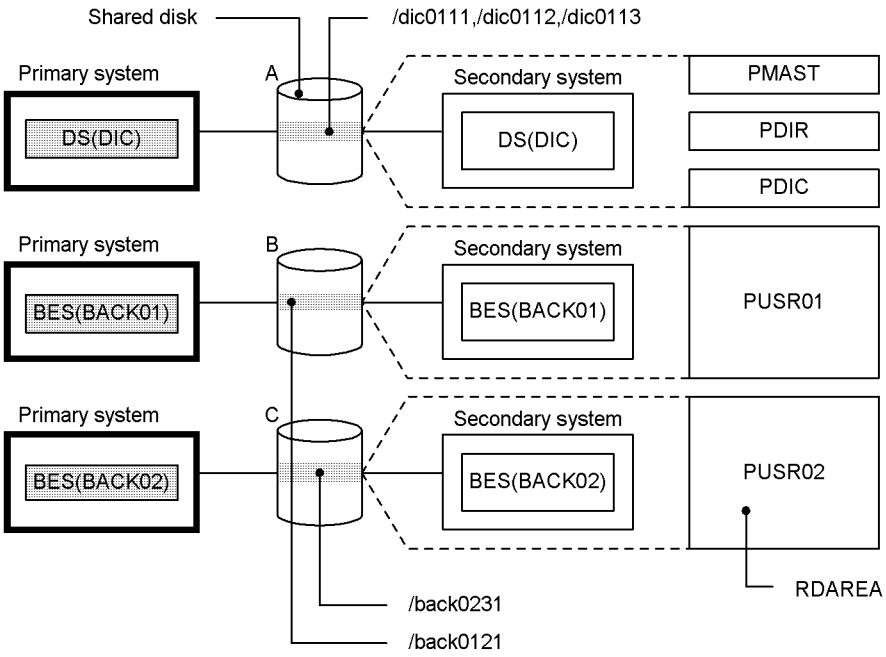
```
create rdarea SMAST for masterdirectory                  1
  file name "/sds0111/srd01" initial 10 segments;
create rdarea SDIR for datadirectory                     2
  file name "/sds0112/srd02" initial 5 segments;
create rdarea SDIC for datadictionary                    3
  file name "/sds0113/srd03" initial 20 segments;
create rdarea SUSR01 for user used by PUBLIC             4
  file name "/sds0121/srd04" initial 500 segments;
create rdarea SUSR02 for user used by PUBLIC             5
  file name "/sds0131/srd05" initial 500 segments;
```

Explanation

1. Creates the `SMAST` master directory RDAREA in the HiRDB file system area on shared disk A.
2. Creates the `SDIR` data directory RDAREA in the HiRDB file system area on shared disk A.
3. Creates the `SDIC` data dictionary RDAREA in the HiRDB file system area on shared disk A.
4. Creates the `SUSR01` user RDAREA in the HiRDB file system area on shared disk B.

5. Creates the SUSR02 user RDAREA in the HiRDB file system area on shared disk C.

Figure 25-41: HiRDB/Parallel Server system configuration example



Example of create rdarea statement specification

```
create rdarea PMAST for masterdirectory                    1
    server name DIC file name "/dic0111/prd01"
    initial 10 segments;
create rdarea PDIR for datadirectory                       2
    server name DIC file name "/dic0112/prd02"
    initial 5 segments;
create rdarea PDIC for datadictionary                      3
    server name DIC file name "/dic0113/prd03"
    initial 20 segments;
create rdarea PUSR01 for user used by PUBLIC               4
    server name BACK01 file name "/back0121/prd04"
    initial 500 segments;
create rdarea PUSR02 for user used by PUBLIC               5
    server name BACK02 file name "/back0231/prd05"
    initial 500 segments;
```

Explanation

1. Creates the PMAST master directory RDAREA in the HiRDB file system area

on shared disk A.

2. Creates the PDIR data directory RDAREA in the HiRDB file system area on shared disk A.
3. Creates the PDIC data dictionary RDAREA in the HiRDB file system area on shared disk A.
4. Creates the PUSR01 user RDAREA in the HiRDB file system area on shared disk B.
5. Creates the PUSR02 user RDAREA in the HiRDB file system area on shared disk C.

25.5.7 Definition of global buffers (standby-less system switchover (1:1) facility only)

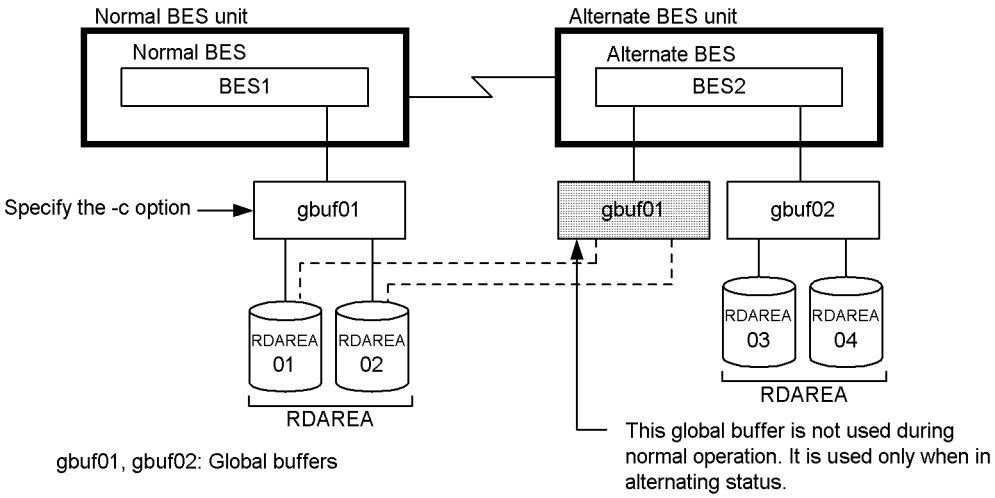
When you define global buffers to be used by an RDAREA in the normal BES, specifying the -c option in the pdbuffer operand allows you to also secure global buffers for use by the alternating portion when units are alternated.

If the -c and -o options of the pdbuffer operand are both omitted, the alternate BES unit cannot be started.

(1) Global buffers for data (-r option global buffers)

The system configuration examples shown below are used to explain how to allocate global buffers for data when alternating units.

System configuration example 1

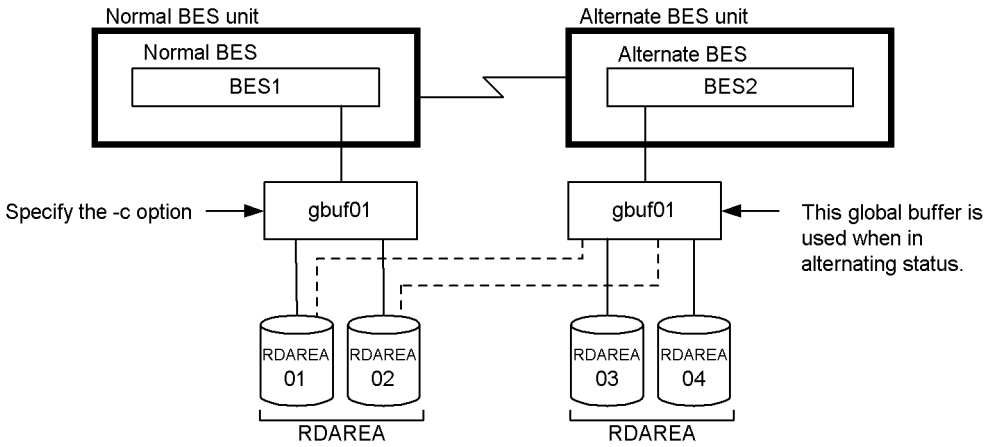


```
pdbuffer -a gbuf01 -r RDAREA01,RDAREA02 -n 1000 -c
pdbuffer -a gbuf02 -r RDAREA03,RDAREA04 -n 1000
```

Explanation

- This configuration does not use global buffers with the same names in the normal BES and the alternate BES.
- When the `-c` option is specified, `gbuf01` will also be secured in the alternate BES. This global buffer is used when alternating units. However, this global buffer is not used during normal operation.
- When estimating shared memory that the global buffers for the alternate BES unit will use, be sure to add enough memory for `gbuf01`.
- The buffer hit rate will not decrease when alternating units.

System configuration example 2



gbuf01: Global buffer

```
pdbuffer -a gbuf01 -r RDAREA01,RDAREA02,RDAREA03,RDAREA04 -n 1000 -c
```

Explanation

- This configuration uses global buffers with the same names in the normal BES and the alternate BES.
- With the `-c` option specified, `gbuf01` of the alternate BES will be used when

alternating units.

- The buffer hit rate may decrease when alternating units.
- The buffer size of `gbuf01` in the alternate BES will be the maximum page length of the RDAREAs in the normal BES and alternate BES.

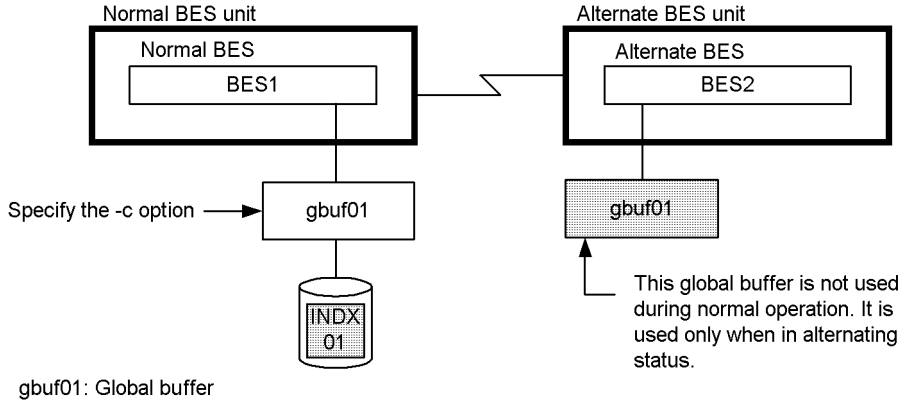
When the -c option is omitted

When the `-c` option is omitted, global buffers specified by the `-o` option are used when alternating units.

(2) Global buffers for index (-i option global buffer)

The system configuration examples shown below are used to explain how to allocate global buffers for index when alternating units.

System configuration example

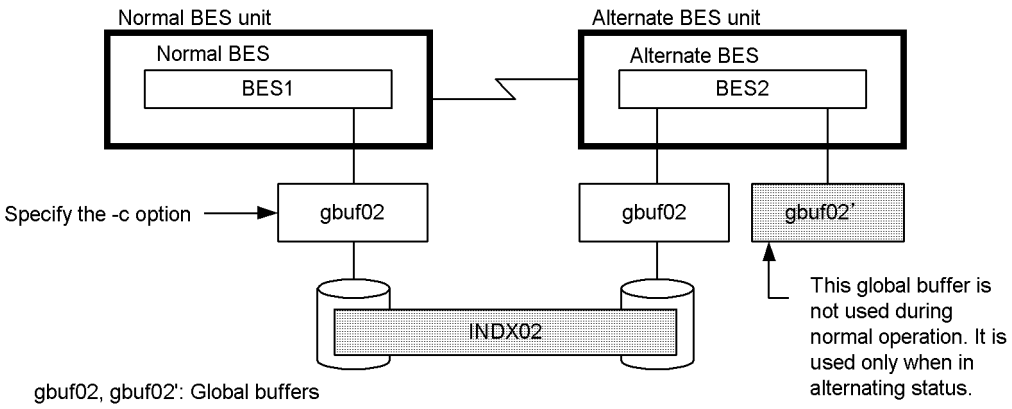



```
pdbuffer -a gbuf01 -i USER01.INDX01 -n 1000 -c
```

Explanation

- When the `-c` option is specified for a global buffer that uses row nonpartitioned index `INDX01`, `gbuf01` will also be secured in the alternate BES. This global buffer is used when alternating units. However, this global buffer is not used during normal operation.
- When estimating shared memory that the global buffers for the alternate BES unit will use, be sure to add enough memory for `gbuf01`.

System configuration example




```
pdbuffer -a gbuf02 -i USER01.INDX02 -n 1000 -c
```

Explanation

- When the `-c` option is specified for a global buffer that uses row partitioned index `INDX02`, `gbuf02` will also be secured in the alternate BES. This global buffer is used when alternating units, so no access conflict with `INDX02` occurs on BES2. However, this global buffer is not used during normal operation.
- When estimating shared memory that the global buffers for the alternate BES unit will use, be sure to have twice the size of `gbuf02` available.

When the -c option is omitted

When the `-c` option is omitted, global buffers are used when alternating units according to the priorities explained below.

1. Allocate a global buffer to the index RDAREA. If the `-c` option is specified for the global buffer, allocate the global buffer using the procedure explained in (1).
2. The global buffer specified by the `-o` option is used.

(3) Global buffers for LOB (-b option global buffer)

The procedure for allocating global buffers for LOB when alternating units is the same as the procedure for allocating global buffers for data. However, if the `-c` option is omitted, data will be written to or read from the RDAREA directly without using a global buffer.

(4) -o option global buffers

An `-o` option global buffer in the alternate BES also uses the RDAREA in the normal BES when alternating units. The buffer size of the global buffer is the maximum page length of the RDAREA in the normal BES and alternate BES.

(5) Design guidelines for global buffers

Basically, specify the `-c` option for global buffers used by RDAREAs, indexes, and LOB RDAREAs in normal BES units.

- When the `-c` option is specified, the global buffer used when an alternating unit is created in the alternate BES. Therefore, be sure there is enough extra shared memory for use by the global buffer in the alternate BES unit. If there is not enough extra shared memory, do not specify the `-c` option.
- When using a global buffer for the alternate BES also in the RDAREA in the normal BES, the buffer size of this global buffer is the maximum page length of the RDAREA in the normal BES and alternate BES. Therefore, be sure to consider the page length of the RDAREAs when specifying the `-c` option.

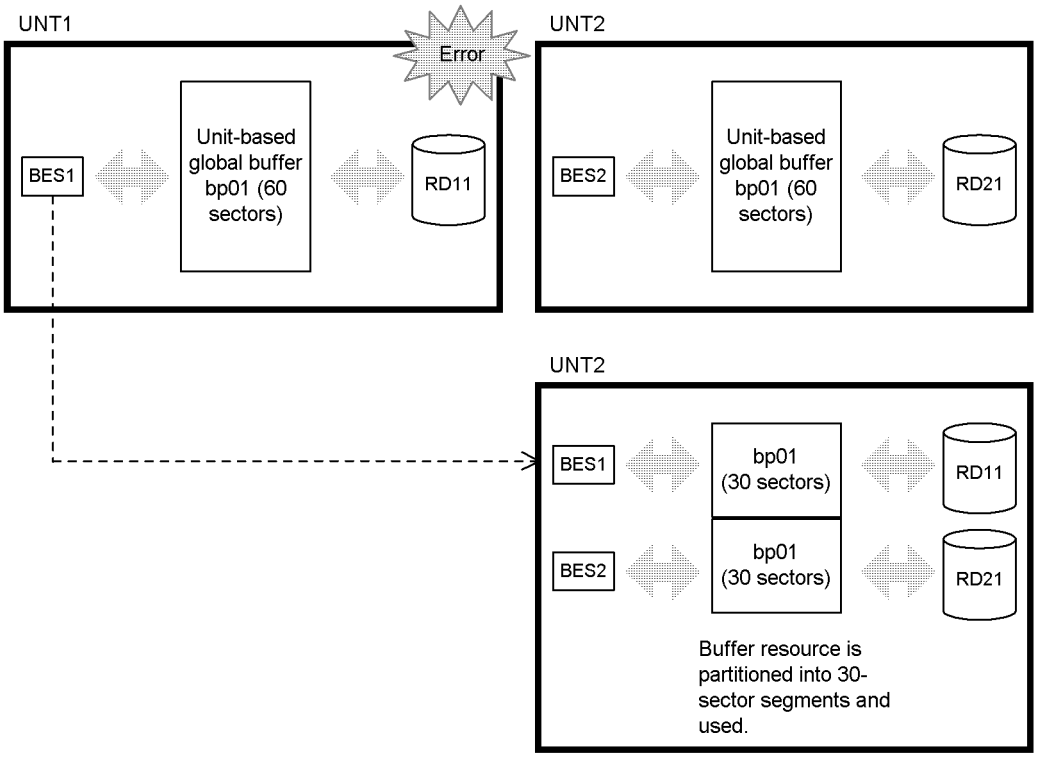
25.5.8 Definition of global buffers (standby-less system switchover (effects distributed) facility only)

When you use the standby-less system switchover (effects distributed) facility, you can allocate global buffers on a unit-by-unit basis.

To allocate a global buffer to RDAREAs or indexes located in back-end servers to which the standby-less system switchover (effects distributed) facility is applicable, specify the `-c` option (sharing option) in the `pdbuffer` operand. A global buffer allocated by specifying the `-c` option is called a *unit-based global buffer*. The unit-based global buffers have the following characteristics:

- Unit-based global buffers must be allocated to all units comprising an HA group.
- If the RDAREAs and indexes to which the same unit-based global buffer is to be allocated are located in multiple back-end servers, its global buffer resources are divided and used equitably among the multiple back-end servers. If these resources are located in a single back-end server, they are used exclusively by that back-end server. If a global buffer defined with the `-o` option specified is allocated to RDAREAs located in multiple back-end servers, the global buffer's resources are also divided and used equitably among the multiple back-end servers. Figure 25-42 shows an example in which a unit-based global buffer is shared. In this figure, the same unit-based global buffer is allocated to servers in different units.

Figure 25-42: Sharing of a unit-based global buffer




```
pdbuffer -a bp01 -a bp01 -r RD11,RD12 -n 60 -c
```

(1) Design concepts for a unit-based global buffer

First, select whether or not the global buffer is to be shared. You can design a global buffer for reduced operation after system switchover occurs based on either of the following concepts:

1. Sharing design

   During reduced operation, memory is used efficiently by means of sharing the accepting unit's global buffer. This is called a *sharing design*; it has the following characteristics:

   - Advantage: Because the accepting unit's global buffer is shared during reduced operation, memory usage efficiency is high.
   - Disadvantage: During reduced operation, the buffer hit rate is reduced proportionately to the number of servers that share the global buffer.

2. Non-sharing design

   Global buffer resources to be used only during reduced operation are allocated to each accepting unit and go into service when system switchover occurs. This is called a *non-sharing design*. It has the following characteristics:

   - Advantage: Because the same amount of buffer resources is available for use before and during reduced operation, the hit rate can be maintained.
   - Disadvantage: Because global buffer resources to be used only during reduced operation are allocated to an accepting unit, memory usage efficiency is low.

Because the objective of the standby-less system switchover (effects distributed) facility is to share resources and distribute the workload, it is more advantageous to use the sharing design (1 above); the reason for this is that memory is used more efficiently.

In the case of non-sharing design, global buffers dedicated to servers are allocated to all accepting units. Therefore, for an entire HA group, you can estimate the required amount of shared memory for global buffers by multiplying the estimated amount of shared memory for normal global buffers by the number of units comprising the HA group. If the amount of shared memory available satisfies the estimate, performance can be maintained even during reduced operation, enabling you to use non-sharing design (2 above).

(a) Procedure for sharing design

This section explains how to create a sharing design in which a global buffer is shared.

1. Determining the RDAREAs that share the same buffer pool

   - Sharing among RDAREAs of the same type

     Use the `-r` option of the `pdbuffer` operand to specify RDAREAs of the same type, such as RDAREAs for data, index RDAREAs, or LOB RDAREAs, so that they cam share a global buffer. When a global buffer is shared by RDAREAs of the same size or same access frequency, memory efficiency is high.

   - Sharing among RDAREAs for row-partitioned tables or indexes

     Use the `-r` option of the `pdbuffer` operand to specify RDAREAs for row-partitioned tables or RDAREAs for row-partitioned indexes so that they share a global buffer. If the row-partitioned tables and indexes are stored in the same RDAREA, specify the `-i` option of the `pdbuffer` operand to allocate a buffer exclusively for indexes.

   Also, depending on the server and unit allocation of the RDAREAs that share a global buffer, the characteristics described below can be obtained. Use these characteristics for reference when selecting the RDAREAs that will share a global

buffer.

- When sharing of RDAREAs of servers located in different units occurs:

  Because global buffers can be used exclusively before reduced operation occurs, allocation can be made that places more importance on performance before reduced operation occurs. However, global buffer resource allocation during reduced operation becomes unbalanced among units.

- When sharing occurs between RDAREAs of servers located within the same unit with an RDAREA of a server located in a different unit:

  Buffer allocation during reduced operation can be kept balanced among units.

2. Determining the buffer sector count of a global buffer to be shared

   The number of buffer sectors specified with the `-n` option are divided equally and used among the sharing serves within the HA group. For this reason, you must specify a buffer sector count that is appropriate to the number of sharing servers so that a shortage will not occur. Use the following formula for estimating an appropriate buffer sector count:

number-of-sectors-needed-by-each-server × (*number-of-host-BESs* + *number-of-BESs-that-can-be-accepted*)

If there is an RDAREA that requires the same level of performance before and during reduced operation, allocate to that RDAREA alone a server-specific global buffer; for details, see *(b) Procedure for non-sharing design*.

(b) Procedure for non-sharing design

This section explains how to create a non-sharing design that does not share global buffers.

You can allocate a server-specific global buffer either for a single RDAREA or for multiple RDAREAs belonging to the same server.

- Allocating exclusively to a single RDAREA

  Specify the RDAREA in the `-r` option of the `pdbuffer` operand. Because there is no competition from other RDAREAs, a global buffer can be allocated to maximize performance. You can also allocate an index-specific buffer by specifying a non-partitioning index in the `-i` option of the `pdbuffer` operand.

- Allocating to multiple RDAREAs belonging to the same server

  Specify in the `-r` option of the `pdbuffer` operand multiple RDAREAs belonging to the same server. Specify RDAREAs of the same type, such as RDAREAs for

data, index RDAREAs, or LOB RDAREAs.

(2) Allocating global buffers for RDAREAs and LOB global buffers (-r or -b option specified)

Allocation of global buffers for RDAREAs and LOB global buffers on a unit-by-unit basis can be classified into four types depending on the combination of the specified RDAREAs. Table 25-15 shows the recommended conditions for global buffer sharing modes (specification of the `-r` or `-b` option).

Table 25-15: Recommended conditions for global buffer sharing modes (-r or -b option specified)

Specification method (combination of RDAREAs specified with -r or -b)			Buffer sharing mode	Benefit	Recommended condition
RDAREAs in different servers	RDAREAs in the same unit	RDAREAs in different units			
None	None	None	Non-shared	Because global buffers are not shared with other servers, normal buffer performance can be maintained even when multiple errors occur.	Because buffers are allocated to all accepting units, this mode is recommended for RDAREAs for which buffer performance should be maintained before and during reduced operation in an environment with ample memory capacity.
Yes	Yes	None	Sharing by servers within a unit	Normal buffer performance can be maintained even when multiple errors occur.	As with the non-shared mode, this mode can maintain buffer performance before and during reduced operation. However, because memory usage efficiency is low when switching first occurs, the non-shared mode is recommended.

Specification method (combination of RDAREAs specified with -r or -b)			Buffer sharing mode	Benefit	Recommended condition
RDAREAs in different servers	RDAREAs in the same unit	RDAREAs in different units			
Yes	None	Yes	Sharing by servers in different units	• Back-end servers of the primary system can use all of the specified buffer sectors. • During reduced operation, the accepting unit's resources are shared, resulting in high memory efficiency.	This mode is recommended when performance during normal operation is important and buffer resources should be shared during reduced operation.
Yes	Yes	Yes	Sharing by servers within a unit and in different units	• During reduced operation, the accepting unit's resources are shared, resulting in high memory usage efficiency. • Because the number of sharing servers can be balanced, workload during reduced operation can be balanced.	This mode is recommended when buffer resources should be shared and the workload should be balanced during reduced operation.

Select the appropriate sharing mode by considering the buffer design guidelines for the `-r` or `-b` specification explained below. Because the objective of the standby-less system switchover (effects distributed) facility is to share resources and distribute the workload, it is preferable to use *Sharing by servers in different units* or *Sharing by servers within a unit and in different units*, both of which involve sharing between units. If it is important to maintain performance even during reduced operation, select *Non-shared* or *Sharing by servers within a unit*.

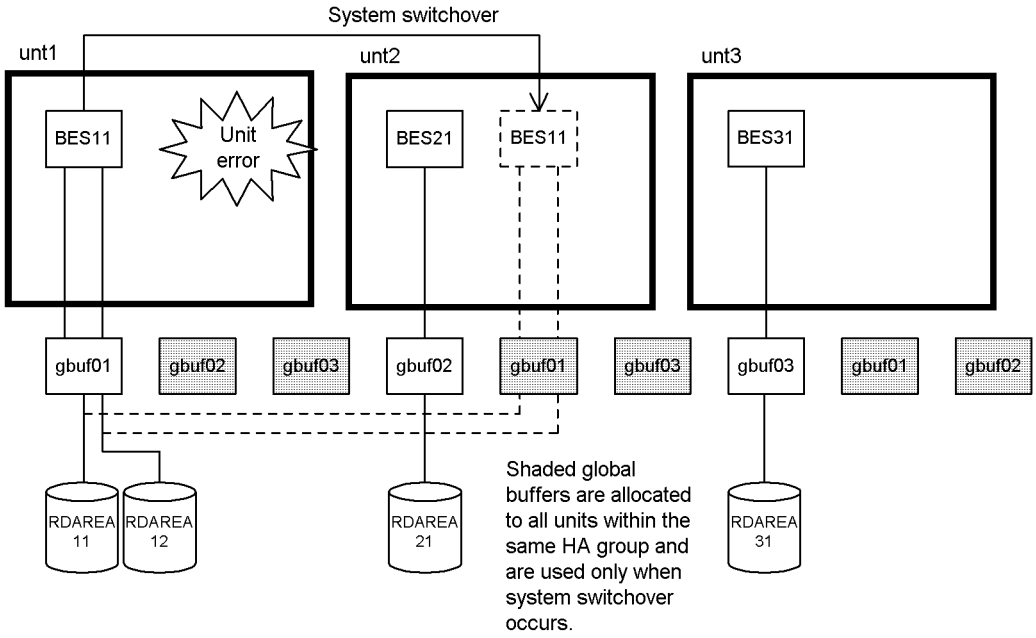
Buffer design guideline for -r or -b specification

1. For an RDAREA for which the buffer hit rate should be maintained even during reduced operation in an environment with ample memory: Select *Non-shared* or *Sharing by servers within a unit*.
2. To have a particular server use the buffer resources exclusively during normal operation and allow sharing with other servers during reduced operation: Select *Sharing by servers in different units*.
3. Cases other than 1 or 2: Select *Sharing by servers within a unit and in different units*.

(a) Allocating non-shared global buffers

Specify only RDAREAs belonging to a single back-end server in the `-r` or `-b` option of each `pdbuffer` operand. A system configuration example follows:

■ System configuration example



■ Global buffer definitions

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA12 -n 2000 -c
pdbuffer -a gbuf02 -r RDAREA21 -n 1000 -c
pdbuffer -a gbuf03 -r RDAREA31 -n 1000 -c
```

Explanation

Only RDAREAs belonging to a single back-end server are specified in the `-r` or `-b` option of each `pdbuffer` operand.

Non-shared buffers `gbuf01`, `gbuf02`, and `gbuf03` are allocated to `BES11`, `BES21`, and `BES31`, respectively.

Even after system switchover, `gbuf01` is used exclusively by `BES11` and thus the buffer hit rate does not decline. However, because a buffer to be used after system switchover must be allocated to each accepting unit, a large amount of memory is used.

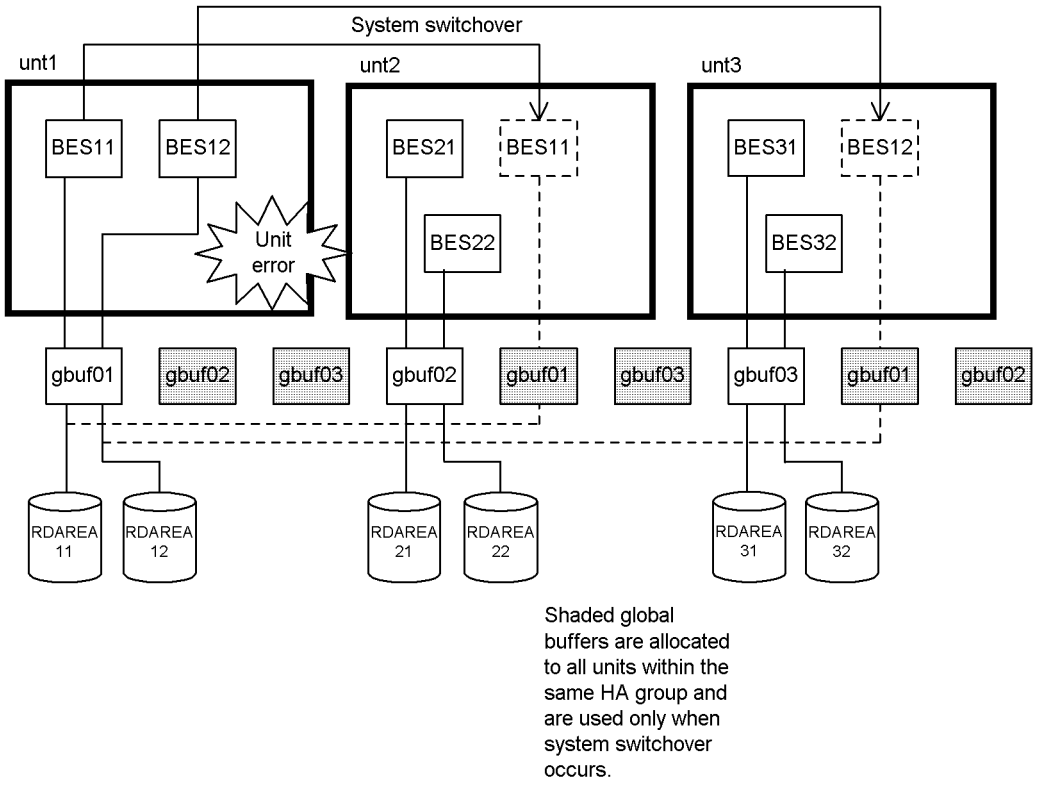
Notes

- The same global buffer is created for all accepting units. This global buffer is not used until system switchover occurs.
- To improve memory efficiency, specify RDAREAs of the same page size.

(b) Allocating global buffers to be shared by servers within a unit

Specify this option if you wish to maintain the same buffer performance even when multi-point errors occur during reduced operation in an environment with ample memory. Specify RDAREAs allocated to back-end servers within the same unit in the `-r` or `-b` option of each pdbuffer operand.

■ System configuration example



■ Global buffer definitions

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA12 -n 1000 -c
pdbuffer -a gbuf02 -r RDAREA21,RDAREA22 -n 1000 -c
pdbuffer -a gbuf03 -r RDAREA31,RDAREA32 -n 1000 -c
```

Explanation

RDAREAs allocated to back-end servers within the same unit are specified in the `-r` or `-b` option of each `pdbuffer` operand.

Global buffers `gbuf01`, `gbuf02`, and `gbuf03` that are to be shared by servers within each unit are allocated.

Because `gbuf01` is used even after system switchover, the buffer hit rate does not decline. However, because a buffer to be used after system switchover must be allocated to each accepting unit, a large amount of memory is used.

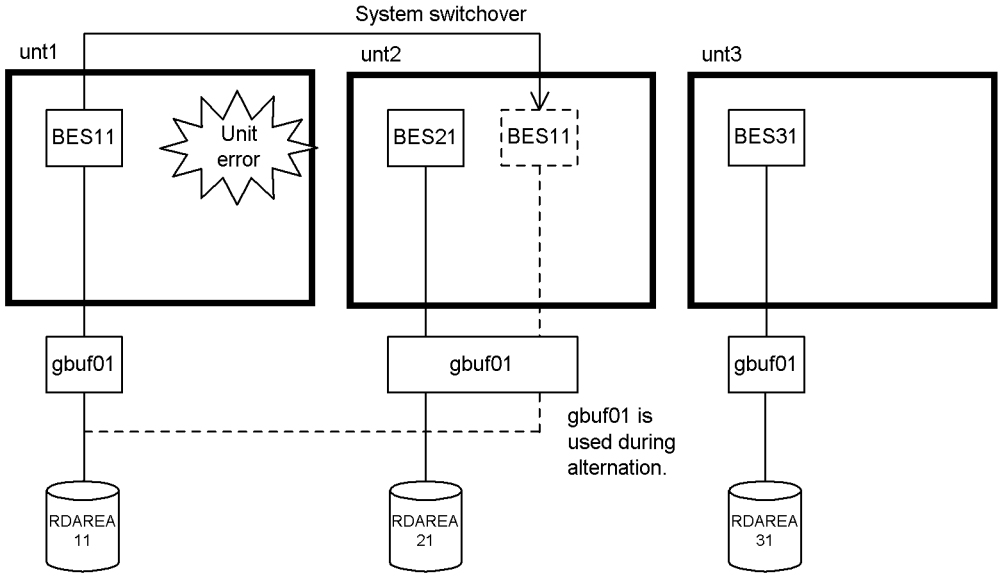
Notes

- The same global buffer is created for all accepting units. This global buffer is not used until system switchover occurs.
- A global buffer with this specification is shared among multiple servers.
- The buffer size when the `-l` option of the `pdbuffer` operand is omitted is the maximum page size of the specified RDAREAs.

(c) Allocating global buffers to be shared by servers in multiple units

Rather than specifying RDAREAs allocated to servers within the same unit, you may specify RDAREAs allocated to servers in different units in the `-r` or `-b` option of the pdbuffer operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA21,RDAREA31 -n 1000 -c
```

Explanation

RDAREAs allocated to servers located in different units are specified.

A global buffer is not shared exclusively among RDAREAs allocated to servers within the same unit. Instead, a global buffer can be shared by RDAREAs

allocated to servers located in different units.

Before reduced operation, the resource (buffer) can be used exclusively by `BES11`. During reduced operation, because the resource must be shared between `BES21` and `BES11`, the amount of the resource (buffer) that can be used by each back-end server is halved.

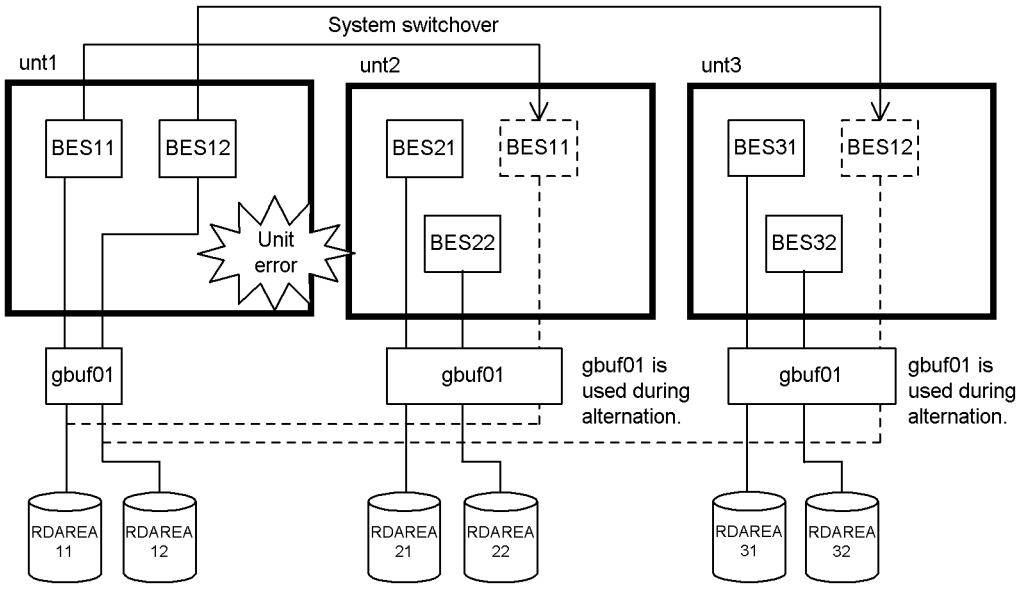
Notes

- Because the switching destination is a single unit, the workload on the global buffer of that unit alone becomes high. Therefore, define multiple buffers to be shared by servers at multiple units so that the workload among individual units is balanced.
- The same global buffer is created for all accepting units.
- A global buffer with this specification is used exclusively by a single server during normal operation and is shared among multiple servers during reduced operation.
- The buffer size when the `-l` option of the `pdbuffer` operand is omitted is the maximum page size of the specified RDAREAs.

(d) Allocating global buffers to be shared by servers within a unit and in different units

You can specify RDAREAs to be shared by servers within a unit and in different units in the `-r` or `-b` option of the pdbuffer operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA12,RDAREA21,RDAREA22,RDAREA31,RDAREA32 -n 1000 -c
```

Explanation

The RDAREAs to be shared by servers within the unit and in different units are specified in the `-r` or `-b` option of the pdbuffer operand.

Because the global buffer is shared among RDAREAs that are shared by servers in different units, this method of buffer allocation results in a balanced workload during reduced operation. During normal operation, the back-end servers in each unit share that unit's `gbuf01`, and the amount of each buffer's resources allocated to each back-end server is one half of the total. During reduced operation, the three back-end servers share `gbuf01`, so the amount of buffer resources allocated to the back-end servers in each unit is one-third of the total.

Notes

- Design the configuration so that the number of servers to be supported during reduced operation is equalized among the individual accepting units.
- The same global buffer is created for all accepting units.

- A global buffer with this specification is shared by multiple servers.
- The buffer size when the `-l` option of the `pdbuffer` operand is omitted is the maximum page size of the specified RDAREAs.

(3) Allocating a global buffer to an index (-i option specified)

To buffer the index pages of a particular index, allocate a global buffer for the index. If an index RDAREA contains only an index, you can also achieve the same effect by allocating a global buffer for an RDAREA (`-r` option specified) to that RDAREA. Allocation of global buffers to indexes on a unit-by-unit basis can be classified into four types depending on the allocation of the index RDAREA for the specified index. Table 25-16 shows the recommended conditions for global buffer sharing modes (specification of the `-i` option).

Table 25-16: Recommended conditions for global buffer sharing modes (-i option specified)

Specification method (index RDAREA specified by the -i option)			Index partitioning mode	Buffer sharing mode	Benefit	Recommended condition
RDAREAs in different servers	RDAREAs in the same unit	RDAREAs in different units				
None	None	None	Non-partitioning and partitioning within the same server	Non-shared	Because global buffers are not shared with other servers, normal buffer performance can be maintained even when multiple errors occur.	Because buffers are allocated to all accepting units, this mode is recommended for indexes for which buffer performance should be maintained before and during reduced operation in an environment with ample memory capacity.

Specification method (index RDAREA specified by the -i option)			Index partitioning mode	Buffer sharing mode	Benefit	Recom-mended condition
RDAREAs in different servers	RDAREAs in the same unit	RDAREAs in different units				
Yes	Yes	None	Partitioning within the same unit	Sharing by servers within a unit	Normal buffer performance can be maintained even when multiple errors occur.	Because buffers are allocated to all accepting units, this mode is recommended for indexes for which buffer performance should be maintained before and during reduced operation in an environment with ample memory capacity.
Yes	None	Yes	Partitioning between units but not within each unit	Sharing by servers in different units	• Back-end servers of the primary system can use all the specified buffer sectors. • During reduced operation, the accepting unit's resources are shared, resulting in high memory efficiency.	This mode is recommended when performance during normal operation is important and the buffer resources should be shared during reduced operation.

Specification method (index RDAREA specified by the -i option)			Index partitioning mode	Buffer sharing mode	Benefit	Recom-mended condition
RDAREAs in different servers	RDAREAs in the same unit	RDAREAs in different units				
Yes	Yes	Yes	Partitioning between units and within each unit	Sharing by servers within a unit and in different units	• During reduced operation, the accepting unit's resources are shared, resulting in high memory efficiency. • Because the number of sharing servers can be balanced, workload during reduced operation can be balanced.	This mode is recommended when the buffer resources should be shared and the workload should be balanced during reduced operation.

You should determine whether or not to allocate buffers dedicated to indexes by considering the buffer design guidelines for the `-i` specification explained below.

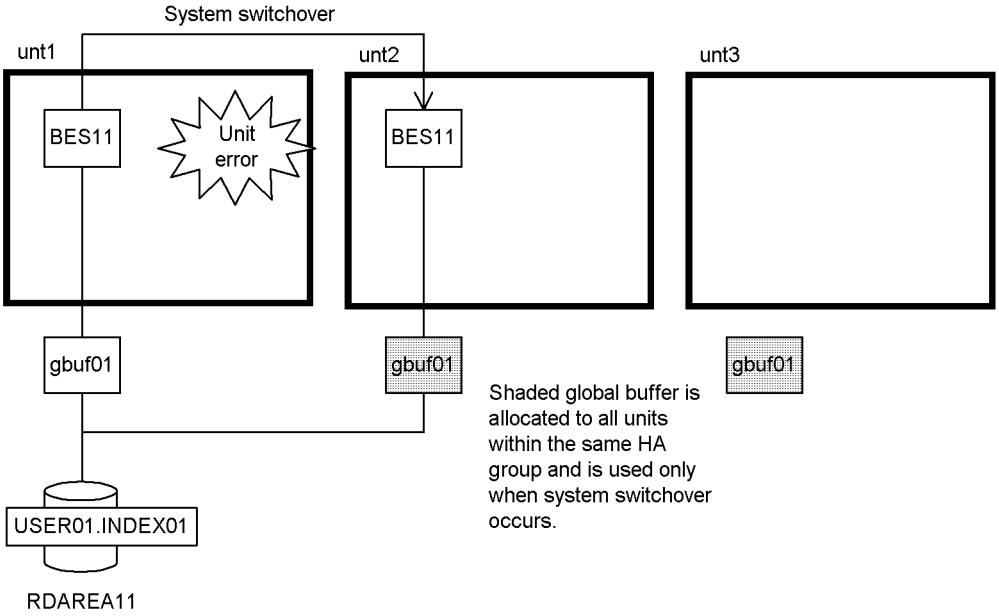
Buffer design guidelines for -i option specification

1. For a non-partitioning index, partitioning index within the same server, or partitioning index within the same unit: Define a dedicated buffer if each unit within the HA group has sufficient memory capacity to allocate a global buffer.
2. Cases other than 1: Define a global buffer for index. Allocate a number of buffer sectors that is appropriate for the sharing servers.

(a) Allocating a global buffer to a non-partitioning index

Specify a non-partitioning index in the `-i` option of the `pdbuffer` operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 200 -c
```

Explanation

Non-partitioning index `USER01.INDEX01` is specified.

This example allocates a shared global buffer to a non-partitioning index. The global buffer is used exclusively and is not shared with other servers. Even after system switchover, `gbuf01` is used exclusively by `BES11` and thus the buffer hit rate does not decline. However, because a buffer to be used after system switchover must be allocated to each accepting unit, a large amount of memory is used.

Note

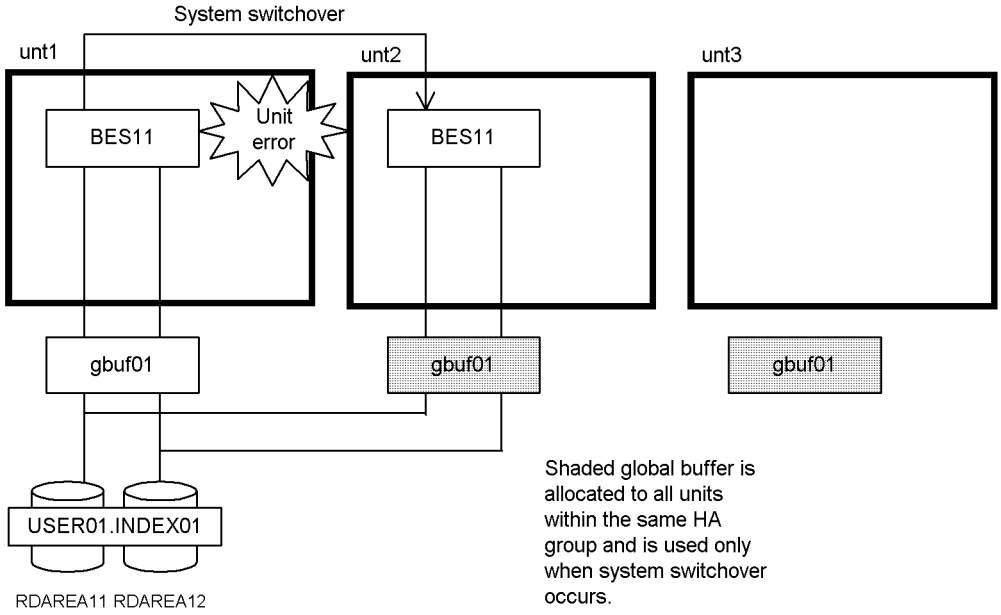
The same global buffer is created for all accepting units. This global buffer is not used until system switchover occurs.

(b) Allocating a global buffer to a partitioning index within the same server

Specify an index that is partitioned within the same server in the `-i` option of the

pdbuffer operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

Explanation

USER01.INDEX01, an index with row partitioning within a server, is specified.

This example allocates a shared global buffer to an index partitioned within the same server. The global buffer is used exclusively and is not shared with other servers. Even after system switchover, gbuf01 is used exclusively by BES11 and thus the buffer hit rate does not decline. However, because a buffer to be used after system switchover must be allocated to each accepting unit, a large amount of memory is used.

Notes

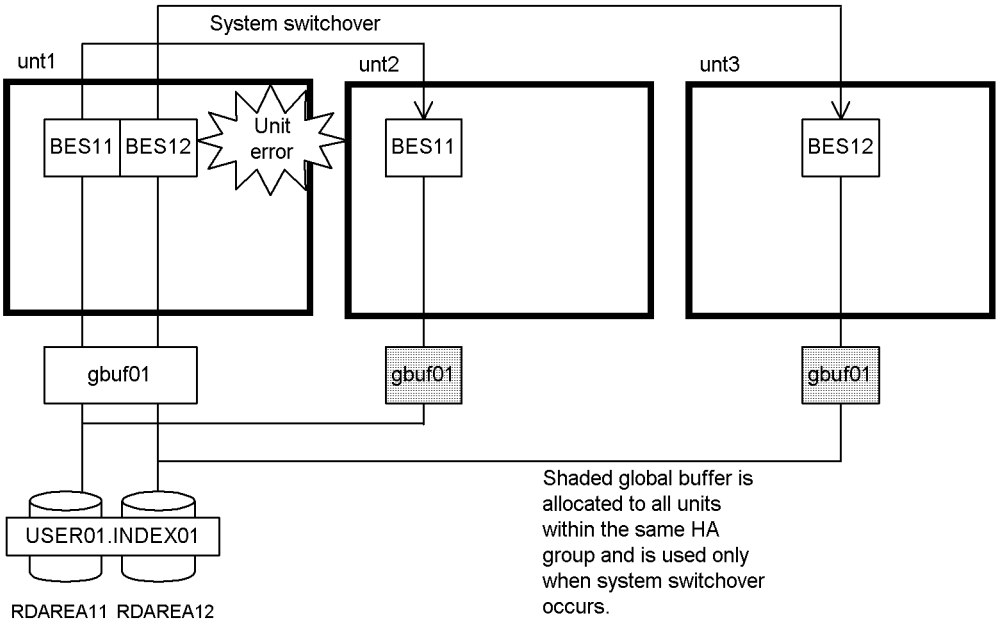
- The same global buffer is created for all accepting units. This global buffer is not used until system switchover occurs.
- To improve memory efficiency, specify index RDAREAs of the same page

size.

(c) Allocating a global buffer to a partitioning index within the same unit

Specify an index that is partitioned within the same unit in the `-i` option of the `pdbuffer` operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

Explanation

This example allocates a shared global buffer to an index partitioned within the same unit. Even after system switchover, `gbuf01` is used and thus the buffer hit rate does not decline. However, because a global buffer to be used after system switchover must be allocated to each accepting unit, a large amount of memory is used.

Notes

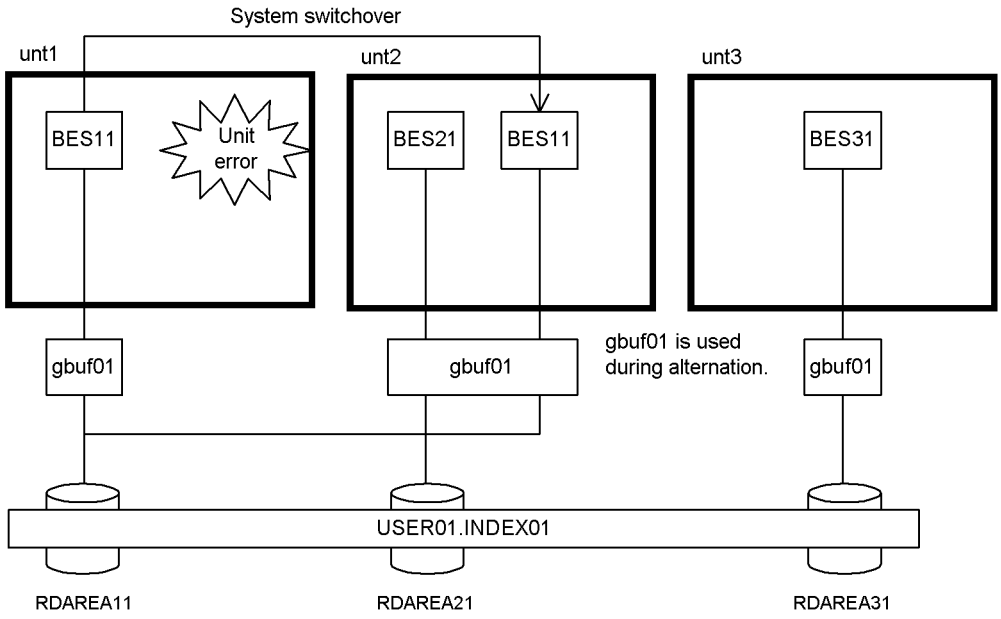
- The same global buffer is created for all accepting units. This global buffer is not used until system switchover occurs.

- A global buffer with this specification is shared by multiple servers.
- To improve memory efficiency, specify index RDAREAs of the same page size.
- The buffer size when the `-l` option of the `pdbuffer` operand is omitted is the maximum page size of the specified index RDAREAs.

(d) Allocating a global buffer to an index partitioned between units but not within each unit

Specify an index that is partitioned between units but not within each unit in the `-i` option of the `pdbuffer` operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

Explanation

`USER01.INDEX01`, an index that is row partitioning between different units, is specified.

This example allocates a global buffer to an index partitioned between units but

not within each unit. Before reduced operation, the `gbuf01` resource (buffer) is used exclusively by `BES11`. During reduced operation, because the resource is shared between `BES21` and `BES11`, the amount of the resource (buffer) that can be used by each back-end server is halved.

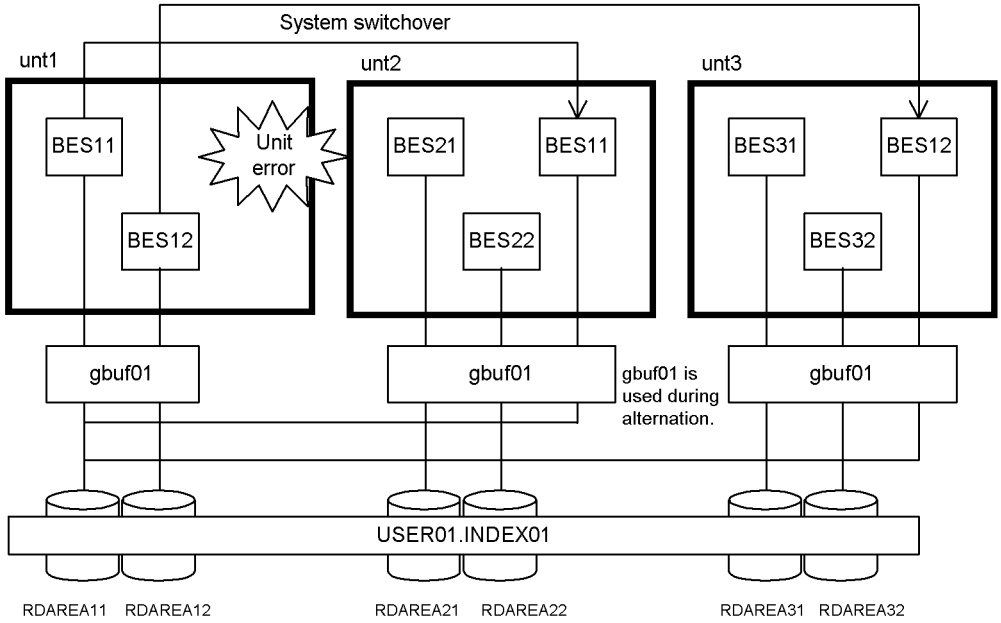
Notes

- Because the switching destination is a single unit, the workload on the global buffer of that unit alone becomes high. Therefore, you should define multiple buffers to be shared by servers at multiple units so that the workload among individual units is balanced.
- The same global buffer is created for all accepting units.
- A global buffer with this specification is used exclusively by a single server during normal operation and is shared among multiple servers during reduced operation.
- To improve memory efficiency, specify index RDAREAs of the same page size.
- The buffer size when the `-l` option of the `pdbuffer` operand is omitted is the maximum page size of the specified index RDAREAs.

(e) Allocating a global buffer to an index partitioned between units and within each unit

Specify an index that is partitioned between units and within each unit in the `-i` option of the `pdbuffer` operand.

■ System configuration example



■ Global buffer definition

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

Explanation

`USER01.INDEX01`, an index with row partitioning within each unit and between servers, is specified.

This example allocates a shared global buffer to an index partitioned between units and within each unit. During normal operation, back-end servers in the units share each `gbuf01`, and the amount of the buffer resources allocated to each back-end server is one half of the total. During reduced operation, the three back-end servers share `gbuf01`, so the amount of the buffer resources allocated to the back-end servers in each unit is one-third of the total.

Notes

- Design the configuration so that the number of servers to be supported during reduced operation is equalized among the individual accepting units.
- The same global buffer is created for all accepting units.

- A global buffer with this specification is shared by multiple servers.
- To improve memory efficiency, specify index RDAREAs of the same page size.
- The buffer size when the `-l` option of the `pdbuffer` operand is omitted is the maximum page size of the specified index RDAREAs.

(4) Allocating a global buffer for OTHER (-o option specified)

One global buffer for `OTHER` can be allocated to all units to which the standby-less system switchover (effects distributed) facility is applied. The allocation method is explained below:

- Within a system, you can define only one buffer for `OTHER` with the `-c` option of the `pdbuffer` operand. If you use the `-c` option to define multiple buffers for `OTHER`, the first one defined in the system common definition is valid.
- Out of the RDAREAs allocated to the host BESs and guest BESs that run in the units to which the standby-less system switchover (effects distributed) facility is applied, the buffer for `OTHER` is allocated to those RDAREAs to which no global buffer has been allocated with the `-r` option.
- When a buffer size is omitted in the `-l` option of the `pdbuffer` operand, if there are RDAREAs to which no global buffer has been allocated with the `-c` or `-r` option within the same HA group, the maximum page size of these RDAREAs is used as the buffer size for the buffer for `OTHER`. If global buffers have been allocated to all RDAREAs within the HA group with the `-c` or `-r` option specified, the maximum page size of the specified RDAREAs in the HA group is used.
- You can simultaneously define a global buffer for `OTHER` with the `-c` option of the `pdbuffer` operand specified and another without the `-c` option specified. Table 25-17 shows the relationship between global buffers for `OTHER` that are specified in duplicate.

  Table 25-17: Relationship between global buffers for OTHER that are specified in duplicate

Definition of a global buffer for OTHER with -c option specified	Definition of a global buffer for OTHER without -c option specified	Units to which the standby-less system switchover (effects distributed) facility is applied	Units to which the standby-less system switchover (effects distributed) facility is not applied
Yes	Yes	Allocates the buffer defined with the `-c` option specified.	Allocates the buffer defined without the `-c` option specified.

Definition of a global buffer for OTHER with -c option specified	Definition of a global buffer for OTHER without -c option specified	Units to which the standby-less system switchover (effects distributed) facility is applied	Units to which the standby-less system switchover (effects distributed) facility is not applied
Yes	No	Allocates the buffer defined with the `-c` option specified.	Allocates the buffer defined with the `-c` option specified.
No	Yes	Does not allocate a buffer for `OTHER`.	Allocates the buffer defined without the `-c` option specified.
No	No	Does not allocate a buffer for `OTHER`.	Does not allocate a buffer for `OTHER`.

(a) Recommended conditions for a global buffer for OTHER

- System in which RDAREAs are added on an online basis
- RDAREAs with a low access frequency
- RDAREAs with a small number of accessed pages
- RDAREAs storing an extremely large number of pages (RDAREAs for which buffer hits are not expected)

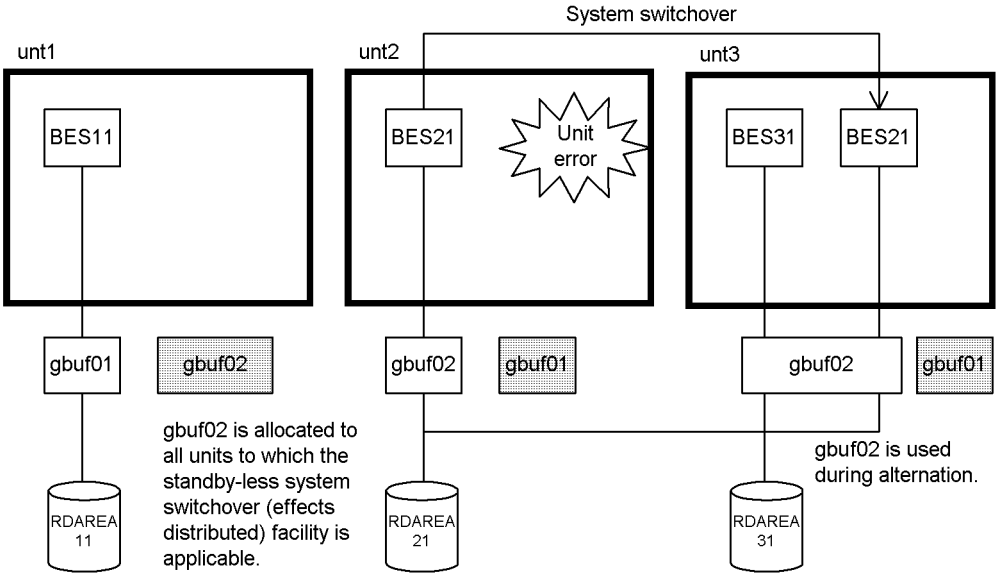
(b) Notes about a global buffer for OTHER

- The resources of a global buffer for `OTHER` allocated to units are divided and used equitably among these units. Therefore, specify in the `-n` option of the `pdbuffer` operand a buffer sectors count that is appropriate to the number of servers used.
- In a system in which RDAREAs are added on an online basis, specify a buffer size in the `-l` option of the `pdbuffer` operand by taking into consideration the page sizes of the RDAREAs that will be added in the future.

(c) Allocation example of a global buffer for OTHER

Specify the `-o` option of the `pdbuffer` operand.

■ System configuration example



■ Global buffer definitions

```
pdbuffer -a gbuf01 -r RDAREA11 -n 500 -c
pdbuffer -a gbuf02 -o -n 1000 -c
```

Explanation

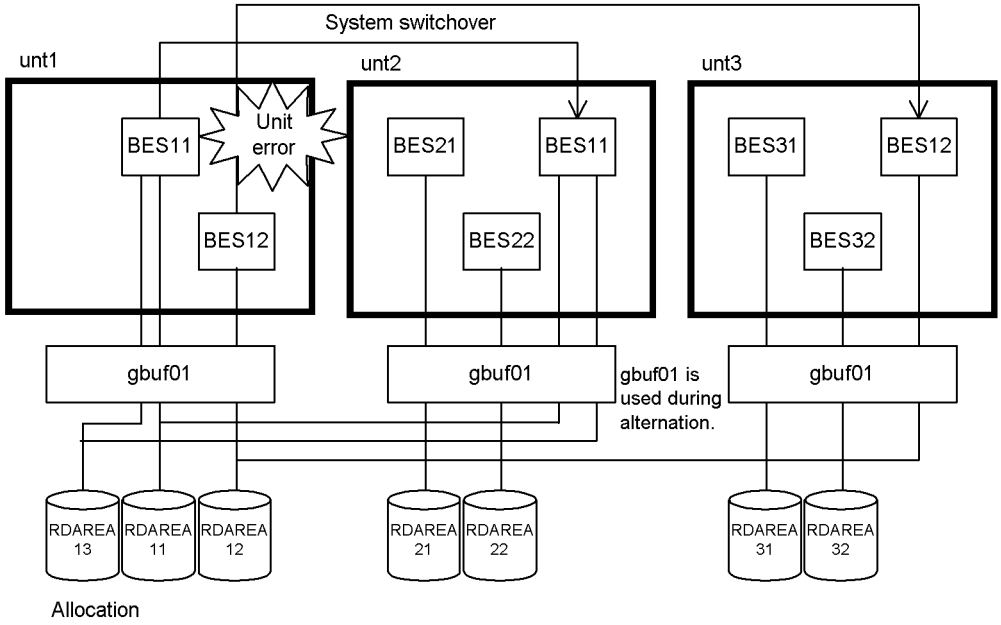
The `-o` and `-c` options of the `pdbuffer` operand are specified.

A dedicated buffer is allocated to `RDAREA11` and a global buffer for `OTHER` is allocated to all other RDAREAs. The global buffer for `OTHER` is created for all units to which the standby-less system switchover (effects distributed) facility is applicable.

(5) Allocating a global buffer during a configuration change (database reorganization utility)

Specify the name of an existing global buffer in the `globalbuffer` operand of the control statement of the database reorganization utility. You can use the `pdbufls` command to check the existing global buffers.

■ System configuration example



■ Configuration changed definition

```
create rdarea RDAREA13 globalbuffer gbuf01 server name BES11
                :
```

Explanation

Shared global buffer `gbuf01` is allocated to an added RDAREA. The added RDAREA uses `gbuf01` even after system switchover.

Design guidelines

- During both system switchover and system reactivation, allocate the global buffer specified with the `globalbuffer` operand.
- You cannot allocate an index global buffer or a LOB global buffer.
- The size of the global buffer to be specified must be greater than the page size of any RDAREA to be added. You can use the `pdbufls` command to check the global buffer size.
- The global buffer allocation specified here becomes invalid when the server terminates normally (normal or planned termination of the HiRDB system,

normal termination of the unit, or normal termination of a server itself). Therefore, before the server starts normally the next time, you must use the `pdbuffer` operand in the system common definition to allocate global buffers. However, if a global buffer with the `-o` option specification has been allocated, that global buffer is allocated again, and therefore there is no need to modify the system common definition.

- When HiRDB fails to allocate a global buffer, no RDAREAs can be added.

25.5.9 Using audit trail files

Executor: HiRDB administrator or auditor

When using the standby system switchover facility or the standby-less system switchover (1:1) facility:

The HiRDB administrator creates audit trail files on a shared disk. The HiRDB administrator and the auditor can use the audit trail files on the shared disk.

When using the standby-less system switchover (effects distributed) facility:

The HiRDB administrator creates audit trail files on a shared disk of the regular unit. During this process, the HiRDB administrator must select a destination disk that is different from the individual servers' shared disks (disks that store individual servers' system log files, synchronization point dump files, and server status files).

At the system switchover destination, the audit trail files of the accepting unit are shared.

The HiRDB administrator and the auditor can use the audit trail files of both the regular unit and the accepting unit.

(1) Creating audit trail files

(a) When using the standby system switchover facility or standby-less system switchover (1:1) facility

The HiRDB administrator creates audit trail files on a shared disk.

(b) When using the standby-less system switchover (effects distributed) facility

The HiRDB administrator creates audit trail files on a shared disk of the regular unit. During this process, the HiRDB administrator must select a destination disk that is different from the individual servers' shared disks (disks that store individual servers' system log files, synchronization point dump files, and server status files).

If audit trail files are created on a shared disk that corresponds to individual servers, the disk's host is switched when system switchover occurs. Consequently, other running servers within the unit can no longer output audit trails. At the system

switchover destination, the audit trail files of the accepting unit are shared.

(2) Using audit trail files

(a) When using the standby system switchover facility or standby-less system switchover (1:1) facility

When system switchover occurs, HiRDB records monitored events in an audit trail file on the shared disk. For details about using audit trail files related to recording of monitored events, see *22.6 Operation of audit trail files*.

(b) When using the standby-less system switchover (effects distributed) facility

When system switchover occurs, HiRDB records monitored events in the audit trail file being used by the accepting unit at the switching destination. In this case, operation of audit trail files related to monitored event records is managed centrally by the accepting unit.

For a system that uses the standby-less system switchover (effects distributed) facility, audit trails must be collected at all units.

(3) Collecting audit trails

(a) When using the standby system switchover facility or standby-less system switchover (1:1) facility

When system switchover occurs, how the audit trail collection status is inherited depends on whether or not the switched unit stops. If the system at the switching destination is restarted, the status before system switchover occurred is inherited. If the system at the switching destination is started normally, the specification in the `pd_audit` operand is used.

(b) When using the standby-less system switchover (effects distributed) facility

When system switchover occurs, whether an audit trail is collected depends on the accepting unit's status. Table 25-18 shows whether an audit trail is collected when the standby-less system switchover (effects distributed) facility is used.

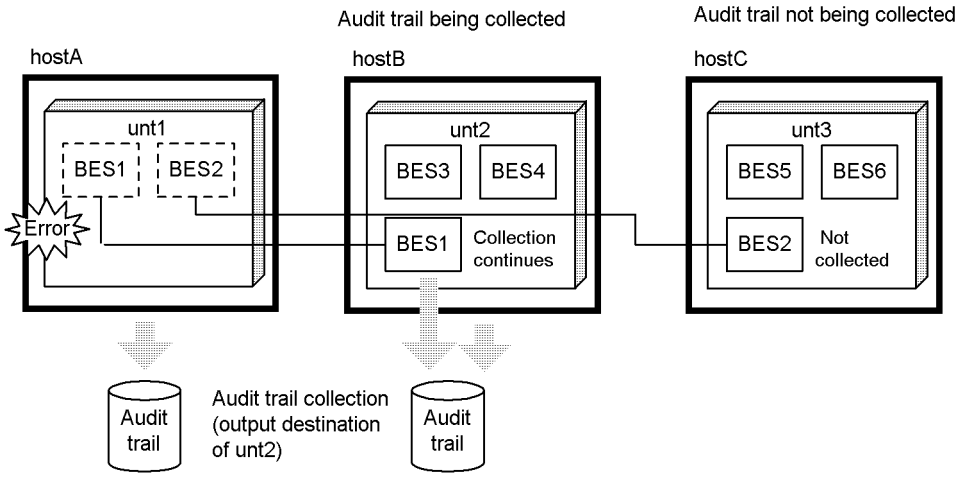
Table 25-18: Collection of audit trails when the standby-less system switchover (effects distributed) facility is used

Unit type	Unit status	Accepting unit	
		Collecting	Not collecting
Regular unit	Collecting	Collects	Does not collect
	Not collecting	Collects	Does not collect

Figure 25-43 shows an example of audit trail collection when the standby-less system

switchover (effects distributed) facility is used.

Figure 25-43: Audit trail collection example when the standby-less system switchover (effects distributed) facility is used



(4) Executing the pdload command

(a) When using the standby system switchover facility or standby-less system switchover (1:1) facility

The auditor executes the `pdload` command using an audit trail file as the input information. However, if a factor such as an error caused system switchover, HiRDB will not have correctly collected the audited events that occurred immediately before the system switchover. For this reason, even if the `pdload` command is executed, it may not be possible to collect the data that existed immediately before system switchover.

(b) When using the standby-less system switchover (effects distributed) facility

The auditor executes the `pdload` command using the audit trail files of the regular unit and the accepting unit as the input information. The audit trails of a server that has been switched are processed as server information belonging to the accepting unit.

When a factor such as an error caused system switchover, HiRDB will not have correctly collected the audited events that occurred immediately before the system switchover. For this reason, even if the `pdload` command is executed, it may not be possible to collect the data that existed immediately before system switchover.

Operation during an error: When an error occurs, load the audit log as follows:

1. At the running host, manually activate the disk storing the audit trail files collected before system switchover.

2. Using the audit trail files of the regular unit and the accepting unit as the input information, execute the `pdload` command.

Operation after error recovery: Load the audit log using the same method as used before the error occurred.

25.6 HA monitor preparations

You should read this section when you use HA monitor in the cluster software. This section provides guidelines for the values to be specified in operands of the following HA monitor definition statements that relate to HiRDB:

- `sysdef` definition statement
- `server` definition statement

Reference note:

- Note that the path names of the files storing the `sysdef` and `server` definition statements depend on the operating system.
- When you use the standby system switchover facility or standby-less system switchover (1:1) facility, you must set up an operating environment for each unit; when you use the standby-less system switchover (effects distributed) facility, you must set up an operating environment for each server.

When using the monitor mode

When you use the monitor mode, you must set up the environment by referring to the explanation in this section.

When using the server mode

When you use the server mode, you must set up the environment by referencing the following:

- Explanation in this section
- Explanation in *25.12 Hitachi HA Toolkit Extension preparations (server mode only)*

25.6.1 sysdef definition statement

(1) servmax operand

You can specify this operand when the version of HA monitor is 01-08 or later.

For the servers to be switched using HA monitor, specify `16` or `64` as the maximum number of servers that can be started concurrently as running servers or standby servers on a single server machine.

`16`: Sets 16 as the maximum number of servers that can be started concurrently.

`64`: Sets 64 as the maximum number of servers that can be started concurrently.

Specify `64` when the number of servers, which are the units for switching on a single server machine, exceeds 16.

The number of servers to be switched using HA monitor also includes products other than HiRDB that are to be switched. HiRDB computes the number of servers to be switched as follows:

- Total number of running units and standby units that are operating on a single server machine and that are subject to standby system switchover
- Total number of normal BES units and alternate BES units that are running on a single server machine and that are subject to standby-less system switchover (1:1)
- Total number of host BESs and guest BESs that are running on a single server machine and that are subject to standby-less system switchover (effects distributed)

(2) multistandby operand

You can specify this operand when the version of HA monitor is 01-08 or later.

This operand specifies whether the multi-standby function, which defines multiple standby systems for a single running system, is to be used:

- `use`: Use the multi-standby function
- `nouse`: Do not use the multi-standby function

Table 25-19 provides specification guidelines for the `multistandby` operand.

Table 25-19: Specification guidelines for the multistandby operand

Environment of system switchover facility		multistandby operand specification
Standby system switchover facility	When IP addresses are not inherited.	Can be omitted.
	When IP addresses are inherited.	Specify `use` if multiple standby systems are defined.
Standby-less system switchover facility	Standby-less system switchover (1:1) facility	Can be omitted.
	Standby-less system switchover (effects distributed) facility	Specify `use` if the number of units belonging to the HA group is 3 or more.

Note

When `use` is specified, you must use the `standbypri` operand of the `servers` definition statement to specify priorities for the standby systems.

25.6.2 server definition statement

(1) acttype operand

To operate the system switchover facility in the server mode, specify `server` in this operand. To operate in the monitor mode, specify `monitor` in this operand.

(2) switchtype operand (applicable to the server mode only)

Consider specifying this operand when operating the system switchover facility in the server mode. You cannot specify this operand when operating in the monitor mode. Specify in this operand the processing to be performed when a server failure is detected.

`switch`:

When HiRDB (or unit for a HiRDB/Parallel Server) terminates abnormally, system switchover is to be performed and HiRDB is to restart on the switchover destination system.

For the standby-less system switchover (1:1) facility, it is recommended that you specify `switch` in the `server` definition statement for the alternate portion created in the alternate BES unit. When `switch` is specified, the system is switched from the alternate portion to the normal BES unit, if an error occurs in the alternate BES unit while alternating units; as a result, the load on the alternate BES unit can be reduced after it restarts.

For the standby-less system switchover (effects distributed) facility, it is advisable to specify `switch` in the `server` definition statement for the guest BES. When `switch` is specified, the guest BES switches to another unit if an error occurs in the unit in which the guest BES is running; as a result, the load on the unit can be distributed after it restarts.

`restart`:

HiRDB (or unit for a HiRDB/Parallel Server) is to be restarted on the system resulting in an error. Only if HiRDB cannot be restarted on the system resulting in the error is system switchover to occur and HiRDB to be restarted on the switchover destination system (system switchover is reported by output of the `KFPS00715-E` message).

For the standby-less system switchover (1:1) facility, it is not recommended that you not specify `restart` in the `server` definition statement for the alternate portion created in the alternate BES unit. When `restart` is specified, the alternate BES unit continues to assume the alternate processes of the normal BES unit when an error occurs in the alternate BES unit while alternating units; as a result, the load on the alternate BES unit is not reduced after it restarts.

For the standby-less system switchover (effects distributed) facility, it is not recommended that you specify `restart` in the `server` definition statement for

the guest BES. When `restart` is specified, the guest BES continues processing, if an error occurs in the unit in which the guest BES is running after the unit restarts; as a result, the load on the unit cannot be distributed after it restarts.

`manual`:

The systems are not to be switched automatically even if HiRDB (or unit for a HiRDB/Parallel Server) cannot be restarted.

Hint:

When you use the standby-less system switchover (1:1) facility with a mutual alternating configuration, specify the same value in the `switchtype` operand of the alternate BES unit and the alternate portion.

When you use the standby-less system switchover (effects distributed) facility, specify the same value in the `switchtype` operand of all servers in the HA group.

(3) name operand

Server mode

- In the case of the standby system switchover facility, specify the HiRDB directory name using an absolute path name. In the case of a HiRDB/Parallel Server, specify the HiRDB directory name of the applicable unit using an absolute path name.
- In the case of the standby-less system switchover (1:1) facility, specify a HiRDB identifier (value specified in the `pd_system_id` operand) and unit identifier (value specified in the `pd_unit_id` operand) of a normal BES unit, separated by a slash (/). Specify as shown below if the HiRDB identifier is `DB01` and the unit identifier of the normal BES unit is `UNT1`:

  ```
  "DB01/UNT1"
  ```

- In the case of the standby-less system switchover (effects distributed) facility, specify a HiRDB identifier and server identifier, separated by a slash (/). Specify as shown below if the HiRDB identifier is `DB01` and the server identifier is `BES1`:

  ```
  "DB01/BES1"
  ```

Monitor mode

Specify a command (shell) for starting HiRDB (or unit for a HiRDB/Parallel Server) using an absolute path name. If the environment variables inherited from HA monitor are not appropriate for use by the HiRDB start command, HiRDB will not function properly. Use a user command (shell) to change the execution environment, then issue the following commands:

- `$PDDIR/bin/pdstart` (for a HiRDB/Single Server)
- `$PDDIR/bin/pdstart -q`* (for a HiRDB/Parallel Server)

* This command starts a unit in the HiRDB/Parallel Server when using the system switchover facility.

The following example shows how to create a user command (shell) when using a HiRDB/Single Server.

Example:

```
PDDIR=/HiRDB_S
PATH=/bin:/usr/bin:/usr/bin/ucb:/$PDDIR/bin
PDCONFPATH=$PDDIR/conf
SHLIB_PATH=$PDDIR/lib
export PATH PDDIR PDCONFPATH SHLIB_PATH
$PDDIR/bin/pdstart
```

(4) termcommand operand (applicable to the monitor mode only)

Consider specifying this operand when operating the system switchover facility in the monitor mode. You cannot specify this operand when operating in the server mode. In the following cases, specify a command (shell) for terminating HiRDB (or unit for a HiRDB/Parallel Server) using an absolute path name:

- When performing grouped system switchover
- When terminating* HiRDB by issuing only the `monend` command of HA monitor or performing a planned system switchover of HiRDB by issuing only the `monswap` command

* This applies when forcible termination is used. For normal termination, use the `pdstop` command to terminate HiRDB and then execute the `monend` command.

If the environment variables inherited from HA monitor are not appropriate for use by the HiRDB termination command, HiRDB will not function properly. Use a user command (shell) to change the execution environment, then issue the following commands:

- `$PDDIR/bin/pdstop` or `$PDDIR/bin/pdstop -f -q`: For a HiRDB/Single Server
- `$PDDIR/bin/pdstop -z -q`: For a HiRDB/Parallel Server

When this command is executed, RDAREAs on a shared disk may go into error shutdown. If this occurs, recover the RDAREAs on the shared disk by executing the database recovery utility (`pdcopy` command).

The following example shows how to create a user command (shell) when using a HiRDB/Single Server:

Example:

```
PDDIR=/HiRDB_S
PATH=/bin:/usr/bin:/usr/ucb:/$PDDIR/bin
PDCONFPATH=$PDDIR/conf
SHLIB_PATH=$PDDIR/lib
export PATH PDDIR PDCONFPATH SHLIB_PATH
$PDDIR/bin/pdstop
```

Reference note:

- If this operand is specified and HA monitor's `monswap` command is entered during HiRDB termination processing, the `pdstop` command results in an error (`pdstop` was entered while HiRDB was stopping). No error message is output at this time, but this is an actual error.
- Even if you define a command (in the `pd_ha_restart_failure` operand) that is to execute when restart fails, the `pdstop` command results in an error similar to the above case.

(5) alias operand

Specify an identifying name that is unique within the system to which HA monitor is applied. You must specify the same identifying name in the primary and secondary systems.

For the standby-less system switchover facility, it is recommended that the unit identifier be specified. For the standby-less system switchover (1:1) facility, it is recommended that the unit identifier of the normal BES unit be specified. For the standby-less system switchover (effects distributed) facility, it is recommended that the server identifier of the server subject to system switchover be specified.

(6) disk operand

Specify the name of the disk area (volume group and partition) in which a HiRDB file system area was created.

(7) lan_updown operand

When inheriting IP addresses:

To turn the power to the network on or off, you can specify the IP address to be inherited in the `.up` file or the server identifier's `.down` file as the `alias` operand value. To do this, you must also specify `use` in the `lan_updown` operand.

When not inheriting IP addresses:

Before starting the running system and standby system HiRDB (or unit), you can activate the IP addresses specified in the `-x` and `-c` options of the `pdunit` operand. Do not specify an IP address specified in the `-x` or `-c` option of the

pdunit operand in the alias operand value's .up or .down file for HA monitor. If IP addresses for client connection are to be inherited, specify these IP addresses. If there are no IP addresses to be inherited, such as IP addresses for client connection, specify nouse in the lan_updown operand of the server definition statement of HA monitor, or delete the alias operand value's .up and .down files.

Standby-less system switchover facility

For the standby-less system switchover (1:1) facility, activate the IP addresses specified in the -x and -c options of the pdunit operand before starting the normal BES unit and the alternate BES unit. For the standby-less system switchover (effects distributed) facility, activate the IP addresses specified in the -x and -c options of the pdunit operand before starting the unit with the HA group name.

Do not specify an IP address specified in the -x or -c option of the pdunit operand in the alias operand value's .up or .down file for HA monitor. If IP addresses for client connection are to be inherited, specify these IP addresses. If there are no IP addresses to be inherited, such as IP addresses for client connection, specify nouse in the lan_updown operand of the server definition statement of HA monitor, or delete the alias operand value's .up and .down files.

(8) group operand

Specify this operand in order to perform grouped system switchover. Specifying this operand is not necessary if the only server to be switched is HiRDB. You specify a server group name in this operand. The examples below provide guidelines for specifying server group names:

- When performing OpenTP1 and grouped system switchover, specify the same value as the definition file that set up the environment for the specified server in Open TP1 of the same system.
- When using HiRDB Datareplicator and performing grouped system switchover, specify the same value as the definition file that set up the environment for the specified server in HiRDB Datareplicator.

(9) initial operand

When you use the standby system switchover facility, specify online for the primary system and standby for the secondary system.

When you use the standby-less system switchover (1:1) facility, specify online for the normal BES unit and standby for the alternate BES unit.

When you use the standby-less system switchover (effects distributed) facility, specify online for the host BES and standby for the guest BES.

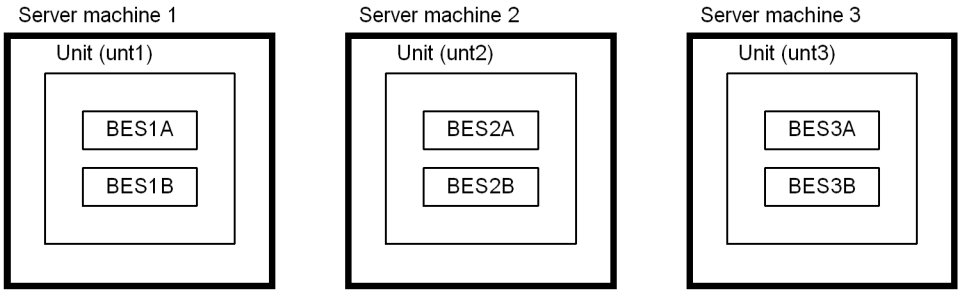
(10) standbypri operand

Specification of this operand is applicable when the version of HA monitor is 01-08 or later.

You use this operand to specify priorities when you use the multi-standby function of HA monitor (`use` must be specified in the `multistandby` operand of the `sysdef` definition statement).

(11) Example of specifying the server definition statements

For this example, the standby-less system switchover (effects distributed) facility is applied to the following 3-unit configuration:



Also, each of the following groups shares a global buffer:

- Group A: BES1A, BES2A, and BES3A
- Group B: BES1B, BES2B, and BES3B

The following table shows the priorities assigned to each unit:

Server	online		Highest priority		Second priority	
BES1A	unt1	[1]	unt2	[2]	unt3	[3]
BES1B		[4]	unt3	[5]	unt2	[6]
BES2A	unt2	[7]	unt3	[8]	unt1	[9]
BES2B		[10]	unt1	[11]	unt3	[12]
BES3A	unt3	[13]	unt2	[14]	unt1	[15]
BES3B		[16]	unt1	[17]	unt2	[18]

A specification example of the `server` definition statements follows:

■ Server definition statements for server machine 1

```
server  name     PDB1/bes1A,
        alias    bes1A,
        patrol   10,
        disk     /dev/vg01,
        initial  online;        ....[1]
server  name     PDB1/bes1B,
        alias    bes1B,
        patrol   10,
        disk     /dev/vg02,
        initial  online;        ....[4]
server  name     PDB1/bes2A,
        alias    bes2A,
        patrol   10,
        disk     /dev/vg03,
        initial  standby,
        standbypri 2;           ...[9]
server  name     PDB1/bes2B,
        alias    bes2B,
        patrol   10,
        disk     /dev/vg04,
        initial  standby,
        standbypri 1;           ...[11]
server  name     PDB1/bes3A,
        alias    bes3A,
        patrol   10,
        disk     /dev/vg05,
        initial   standby,
        standbypri 2;           ...[15]
server  name     PDB1/bes3B,
        alias    bes3B,
        patrol   10,
        disk     /dev/vg06,
        initial  standby,
        standbypri 1;           ...[17]
```

■ Server definition statements for server machine 2

```
server  name     PDB1/bes1A,
        alias    bes1A,
        patrol   10,
        disk     /dev/vg01,
        initial  standby,
        standbypri 1;          ....[2]
server  name     PDB1/bes1B,
        alias    bes1B,
        patrol   10,
        disk     /dev/vg02,
        initial  standby,
        standbypri 2;          ...[6]
server  name     PDB1/bes2A,
        alias    bes2A,
        patrol   10,
        disk     /dev/vg03,
        initial  online;       ...[7]
server  name     PDB1/bes2B,
        alias    bes2B,
        patrol   10,
        disk     /dev/vg04,
        initial  online;       ...[10]
server  name     PDB1/bes3A,
        alias    bes3A,
        patrol   10,
        disk     /dev/vg05,
        initial  standby,
        standbypri 1;          ...[14]
server  name     PDB1/bes3B,
        alias    bes3B,
        patrol   10,
        disk     /dev/vg06,
        initial  standby,
        standbypri 2;          ...[18]
```

■ Server definition statements for server machine 3

```
server  name      PDB1/bes1A,
        alias     bes1A,
        patrol    10,
        disk      /dev/vg01,
        initial   standby,
        standbypri 2;              ...[3]
server  name      PDB1/bes1B,
        alias     bes1B,
        patrol    10,
        disk      /dev/vg02,
        initial   standby,
        standbypri 1;              ...[5]
server  name      PDB1/bes2A,
        alias     bes2A,
        patrol    10,
        disk      /dev/vg03,
        initial   standby,
        standbypri 1;              ...[8]
server  name      PDB1/bes2B,
        alias     bes2B,
        patrol    10,
        disk      /dev/vg04,
        initial   standby,
        standbypri 2;              ...[12]
server  name      PDB1/bes3A,
        alias     bes3A,
        patrol    10,
        disk      /dev/vg05,
        initial   online;          ...[13]
server  name      PDB1/bes3B,
        alias     bes3B,
        patrol    10,
        disk      /dev/vg06,
        initial   online;          ...[16]
```

25.7 MC/ServiceGuard preparations

Read this section when using MC/ServiceGuard in the cluster software. This section covers the following topics:

- Package
- Shell script for starting HiRDB
- Shell script for terminating HiRDB
- Shell script for generating a dummy process
- Package IP address
- Example of grouped MC/ServiceGuard and HiRDB configuration

When using the monitor mode:

When you use the monitor mode, set up the environment by referencing the explanation in this section and the MC/ServiceGuard documentation. For details about setting up the environment for MC/ServiceGuard, see the MC/ServiceGuard documentation.

When using the server mode:

When you use the server mode, set up the environment by referencing the following:

- Explanation in *25.12 Hitachi HA Toolkit Extension preparations (server mode only)*
- MC/ServiceGuard documentation

For details about setting up the environment for MC/ServiceGuard, see the MC/ServiceGuard documentation.

25.7.1 Package

The unit MC/ServiceGuard uses for switching systems is called a *package*. A package refers to a set of the following resources necessary for applications to run:

- Volume group
- Network address
- Service (application)
- Operation for startup or stop (script)

Products that group MC/ServiceGuard with HiRDB are usually handled as a single package, which MC/ServiceGuard uses for switching systems. Figure 25-44 provides

an overview of packaging. Figure 25-45 shows the flow of package processing by MC/ServiceGuard.

Figure 25-44: Package overview

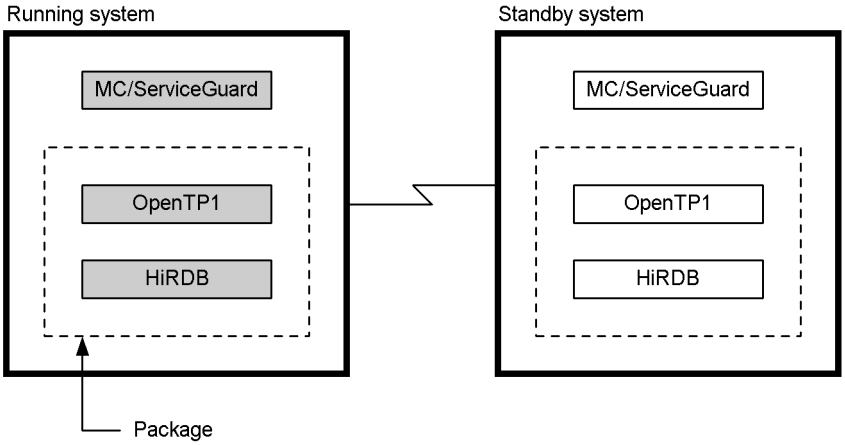
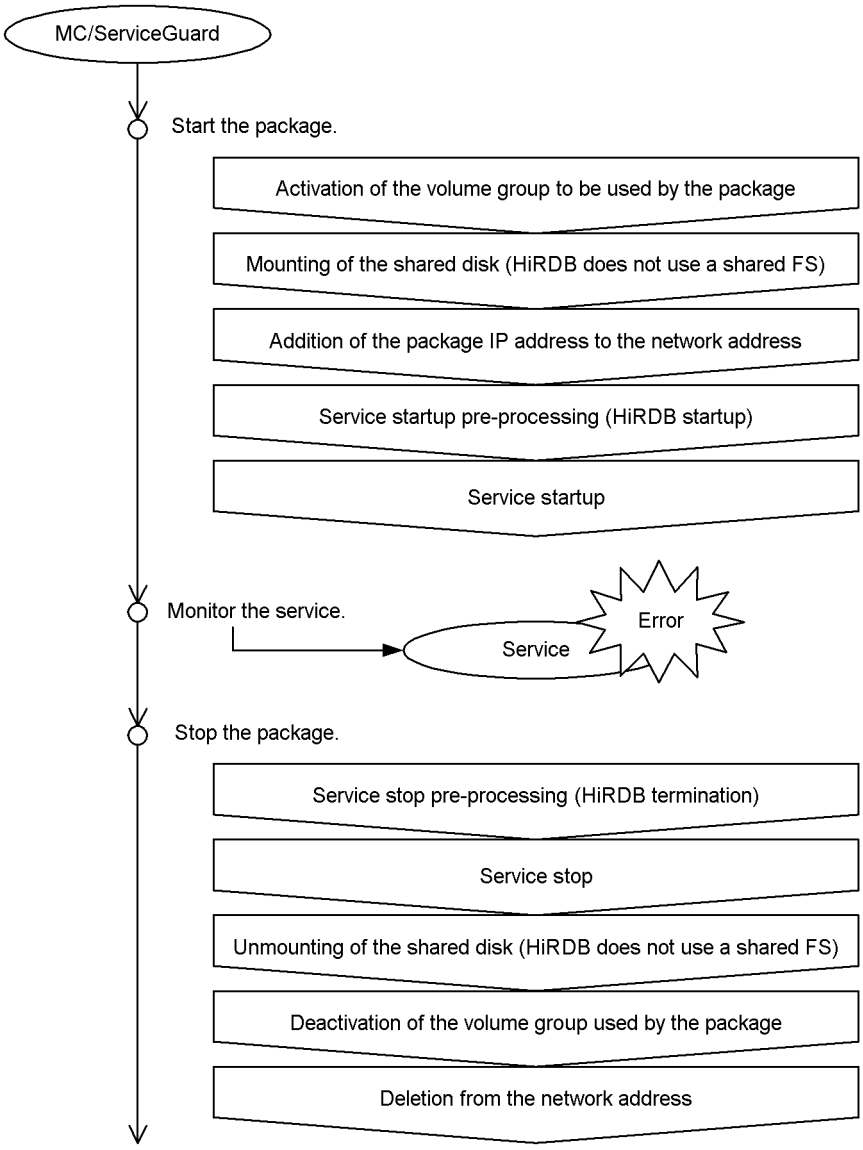


Figure 25-45: Flow of package processing by MC/ServiceGuard

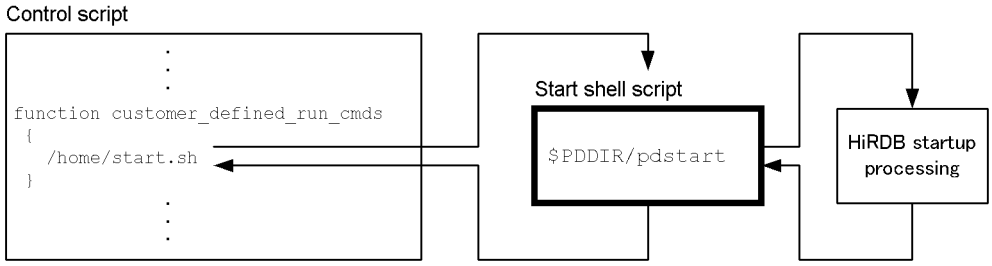


25.7.2 Shell script for starting HiRDB

Create a shell script for starting HiRDB (execute `$PDDIR/bin/pdstart`). Be sure to execute that shell script when you start the package (during package startup preprocessing). Set up the shell script for starting HiRDB so that the package control script for MC/ServiceGuard executes this shell script. Figure 25-46 shows the flow of

HiRDB startup processing.

Figure 25-46: HiRDB startup processing flow (MC/ServiceGuard)



(1) Examples of a shell script for starting HiRDB

Examples of a shell script for starting HiRDB are shown below.

HiRDB/Single Server

```
#!/bin/sh
PDDIR=/HiRDB_S
PDCONFPATH=${PDDIR}/conf
SHLIB_PATH=${PDDIR}/lib
PATH=${PATH}:${PDDIR}/bin
export PDDIR PDCONFPATH SHLIB_PATH PATH
${PDDIR}/bin/pdstart>/dev/null 2>&1
```

HiRDB/Parallel Server

```
#!/bin/sh
PDDIR=/HiRDB_P
PDCONFPATH=${PDDIR}/conf
SHLIB_PATH=${PDDIR}/lib
PATH=${PATH}:${PDDIR}/bin
export PDDIR PDCONFPATH SHLIB_PATH PATH
${PDDIR}/bin/pdstart -q>/dev/null 2>&1
```

Note

Modify the `PDDIR` directory on the second line as appropriate for the environment.

The `pdstart -q` command starts the units in a HiRDB/Parallel Server when the system switchover facility is used.

(2) When the pdstart command fails and HiRDB cannot be started

The following are possible causes of a `pdstart` command failure:

1. HiRDB is being started
2. HiRDB is running
3. HiRDB is being terminated normally or planned termination is underway
4. HiRDB is terminating abnormally
5. HiRDB's environment setup is incorrect

Items 1-4 can occur if the package is started before HiRDB has stopped. There is no means to accurately determine this timing. In such a case, the `pdstart` command error may be ignored, because it does not affect any other operations.

In the case of item 5, HiRDB cannot be started because the HiRDB environment setup is incorrect; correct the environment setup.

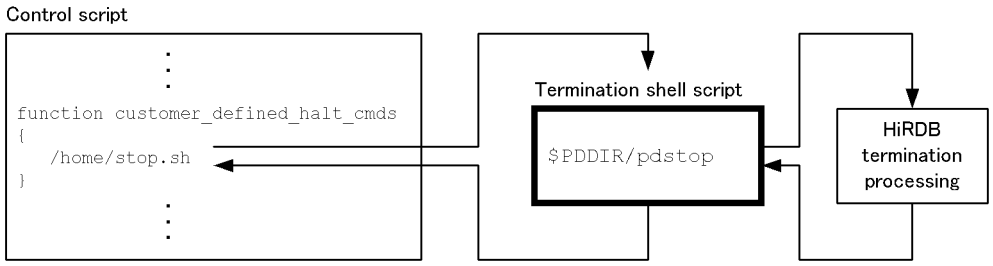
(3) Note

If no system RDAREAs have been created, an instruction to create system RDAREAs (execute the `pdinit` command) will be displayed during HiRDB startup (during execution of the `pdstart` command). The `pdstart` command started from MC/ServiceGuard is executed in the background, so it cannot respond to this instruction. Therefore, before starting operation of MC/ServiceGuard, execute the `pdstart` command and ensure that system RDAREAs have been created.

25.7.3 Shell script for terminating HiRDB

Create a shell script for terminating HiRDB (executing `$PDDIR/bin/pdstop`) and start this shell script when the package stops. To do this, set a shell script that terminates HiRDB inside the package control script of MC/ServiceGuard. Figure 25-47 shows the HiRDB termination processing flow.

Figure 25-47: HiRDB termination processing flow (MC/ServiceGuard)



(1) Examples of a shell script for terminating HiRDB

Examples of a shell script for terminating HiRDB are shown below.

HiRDB/Single Server

```
#!/bin/sh
PDDIR=/HiRDB_S
PDCONFPATH=${PDDIR}/conf
SHLIB_PATH=${PDDIR}/lib
PATH=${PATH}:${PDDIR}/bin
export PDDIR PDCONFPATH SHLIB_PATH PATH
${PDDIR}/bin/pdstop -f -q>/dev/null 2>&1
```

Note

Specify the `pdstop -f -q` command to terminate HiRDB forcibly.

Executing this command may cause an error shutdown in the RDAREAs on the shared disk. If this happens, use the database recovery utility to recover the RDAREAs on the shared disk.

HiRDB/Parallel Server

```
#!/bin/sh
PDDIR=/HiRDB_P
PDCONFPATH=${PDDIR}/conf
SHLIB_PATH=${PDDIR}/lib
PATH=${PATH}:${PDDIR}/bin
export PDDIR PDCONFPATH SHLIB_PATH PATH
${PDDIR}/bin/pdstop -z -q>/dev/null 2>&1
```

Note

Specify the `pdstop -z -q` command to terminate HiRDB forcibly.

Executing this command may cause an error shutdown in the RDAREAs on the shared disk. If this happens, use the database recovery utility to recover the RDAREAs on the shared disk.

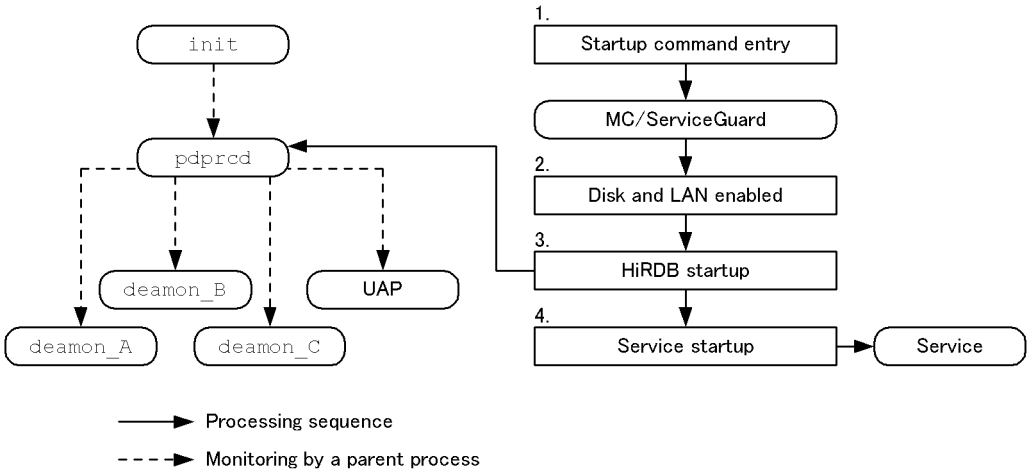
(2) Notes

- The package is also stopped when system switchover is being performed erroneously. In such a case, HiRDB is terminated forcibly so that the systems can be switched immediately and the standby system can restart HiRDB in order to resume operations.
- A termination mode, such as normal termination or planned termination, cannot be specified in a shell script.
- To terminate HiRDB normally, first use the `pdstop` command to terminate HiRDB, and then deactivate the package.

25.7.4 Shell script for generating a dummy process (services monitored by MC/ServiceGuard) (monitor mode only)

MC/ServiceGuard monitors services (applications). It executes a service as a sub-process, and when it detects service termination, it determines that a package error has occurred. The only interface between HiRDB and MC/ServiceGuard, except for HiRDB startup and termination processing, is service termination (exit). All HiRDB processes are monitored by `pdprcd` and are restarted by HiRDB facilities after an abnormal termination. Therefore, there is no need to have MC/ServiceGuard monitor HiRDB processes. Figure 25-48 shows the relationship between process startup and monitoring.

Figure 25-48: Relationship between process startup and monitoring (MC/ServiceGuard)



To maintain the status in which HiRDB is running as a package, a dummy service (dummy process) is necessary. This dummy process must satisfy all the following conditions:

Conditions

1. The process must be resident.
2. Because MC/ServiceGuard issues SIGTERM when it issues an instruction to terminate the package, the process must be terminated when SIGTERM is received.
3. There is no need to notify MC/ServiceGuard of HiRDB abnormal termination.

The following is an example of a shell script for generating a dummy process:

```
#!/bin/sh
trap exit SIGTERM
while true
do
    sleep 5
done
exit
```

Explanation

This example is coded in the Bourne shell. sleep is executed in an infinite loop until `SIGTERM (=15)` is received. Other implementation methods are also possible.

Specify this shell script in `SERVICE_CMD` in the package control script. To start a resident process, infinite startup should be used by specifying `SERVICE RESTART ="-R"` in the package control script. Otherwise, unexpected termination will be treated as abnormal termination of the HiRDB server.

25.7.5 Package IP address

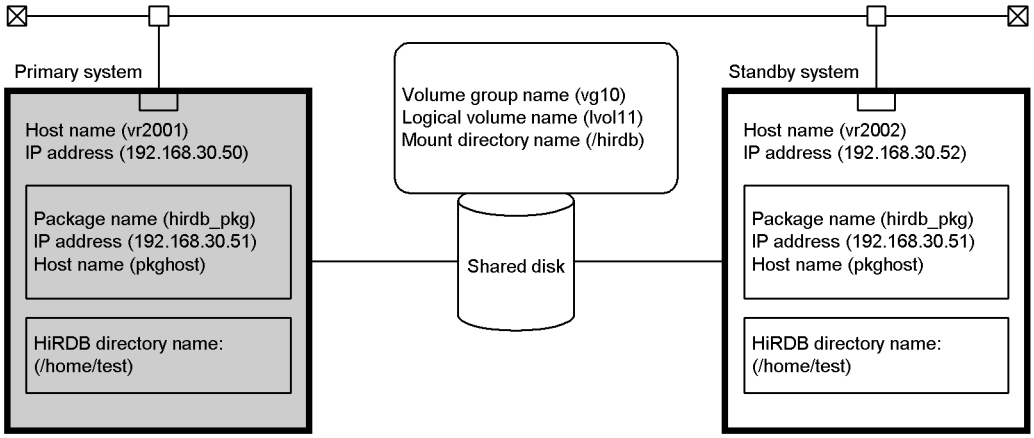
Specify the package's IP address (repositionable IP address) in the package control script.

`IP[0] =` *Package-IP-address* `(`*repositionable-IP-address*`)`

25.7.6 Example of grouped MC/ServiceGuard and HiRDB configuration

This section provides and explains an example of a grouped MC/ServiceGuard and HiRDB configuration. Figure 25-49 shows the system configuration. In this case, HiRDB uses a HiRDB/Single Server.

Figure 25-49: Example of grouped MC/ServiceGuard and HiRDB configuration



(1) Package control script

Definitions related to system configuration

```
VGCHANGE="vgchange -a e"
VG[0]=vg10
LV[0]=/dev/vg10/lvol11; FS[0]=/hirdb
IP[0]=192.168.30.51
SUBNET[0]=192.168.30.0
```

MC/ServiceGuard service registration (definitions for monitoring HiRDB startup and termination)

```
SERVICE_NAME[0]=hirdb

#Service = specifies the process to be monitored [required]
#MC/ServiceGuard starts and monitors for process termination.
SERVICE_CMD[0]=/etc/cmcluster/hirdb_pkg/monitor.sh          1
SERVICE_RESTART[0]="-R"

#Service startup preprocessing: Start HiRDB here.
function customer_defined_run_cmds
{
/etc/cmcluster/hirdb_pkg/run.sh                             2
test_return 51
}

#Service stop preprocessing: Stop HiRDB here.
function customer_defined_halt_cmds
{
/etc/cmcluster/hirdb_pkg/halt.sh                            3
test_return 52
}
```

Explanation

1. Specifies the shell script for generating a dummy script.
2. Specifies the shell script for starting HiRDB.
3. Specifies the shell script for terminating HiRDB.

(2) Shell script

Shell script for generating a dummy script (monitor.sh)

```
!/bin/ksh
trap exit SIGTERM
while true
do
   sleep 5
done
exit
```

Shell script for starting HiRDB (run.sh)

```
#!/bin/sh
PDDIR=/home/test
PDCONFPATH=${PDDIR}/conf
SHLIB_PATH=${PDDIR}/lib
PATH=${PATH}:${PDDIR}/bin
export PDDIR PDCONFPATH SHLIB_PATH PATH
/home/test/bin/pdstart > /dev/null 2>&1
```

Shell script for terminating HiRDB (halt.sh)

```
#!/bin/sh
export PDDIR=/home/test
export PDCONFPATH=${PDDIR}/conf
export SHLIB_PATH=${PDDIR}/lib
export PATH=${PATH}:${PDDIR}/bin
/home/test/bin/pdstop -f -q> /dev/null 2>&1
```

(3) HiRDB system definitions

System common definitions

```
set pd_mode_conf = MANUAL1
set pd_ha_ipaddr_inherit = Y
set pd_ha   = use
pdunit -x pkghost -u puid -d /home/test
pdstart -t SDS -s sds -x pkghost
```

Note

Specify the host name that corresponds to the repositionable IP address for accessing the package.

Unit information definition

```
set pd_hostname = vr2001
```

25.8 VERITAS Cluster Server preparations

Read this section when using VERITAS Cluster Server in the cluster software. This section covers the following topics:

- Groups and resources
- HiRDB resource type definition
- Agent definition pre-preparation
- Agent definition
- Environment setup file creation

When using the monitor mode:

When you use the monitor mode, set up the environment by referencing the explanation in this section and the VERITAS Cluster Server documentation. For details about setting up the environment for VERITAS Cluster Server, see the VERITAS Cluster Server documentation.

When using the server mode:

When you use the server mode, set up the environment by referencing the following:

- Explanation in *25.8.1 Groups and resources*
- Explanation in *25.12 Hitachi HA Toolkit Extension preparations (server mode only)*
- VERITAS Cluster Server documentation

For details about setting up the environment for VERITAS Cluster Server, see the VERITAS Cluster Server documentation.

25.8.1 Groups and resources

The unit VERITAS Cluster Server uses for switching systems between nodes is called a *group*. The applications comprising a group operate using what are referred to as *resources*. The following types of resources are available:

- Disk group (shared disk)
- Network address (logical IP address)
- Network interface card (NIC)
- Service (application)

(1) Setting up groups and resources

Provide NICs, logical IP addresses, and a shared disk, and set them up as VERITAS Cluster Server resources to form a group. In the explanation in this manual, VERITAS Volume Manager is used for setting up a shared disk. When these resources are set up, resource types that have already been defined (such as NIC type, IP type, and DiskGroup) can be used.

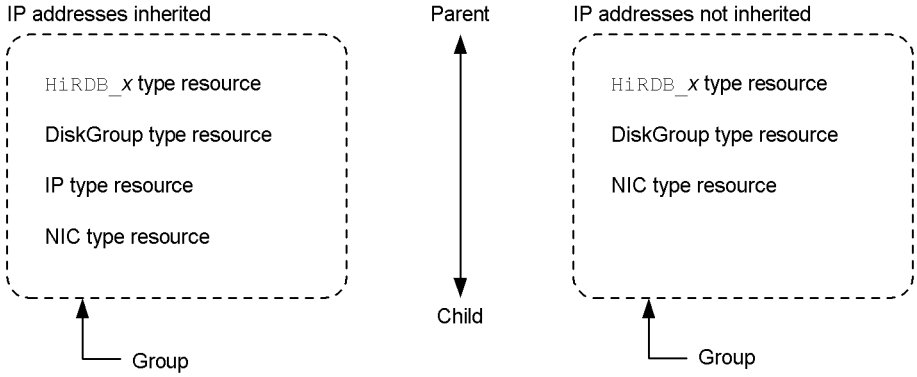
It is also necessary to set up HiRDB as a resource within the group so that it can be managed by VERITAS Cluster Server. For this purpose, a new resource type is defined for HiRDB. For details on how to define a resource type, see *25.8.2 HiRDB resource type definition*.

The resource type name is `HiRDB_S` in the case of a HiRDB/Single Server and `HiRDB_P` in the case of a HiRDB/Parallel Server. This manual uses the generic notation `HiRDB_`*x* for resource type names; substitute `HiRDB_S` or `HiRDB_P` as appropriate.

(2) Defining the parent-child relationships of resources

Parent-child relationships are defined for the resources defined within a group. For HiRDB to run, logical IP addresses (when IP addresses are inherited) and the shared disk must be enabled. Therefore, `HiRDB_`*x* type resources must be defined as parent resources to the IP type and DiskGroup type resources. Figure 25-50 shows a group configuration.

Figure 25-50: Group configuration



(3) Dummy file (applicable to the monitor mode only)

If HiRDB terminates abnormally when operating the system switchover facility in the monitor mode, HiRDB will restart. For this reason, it is not necessary to monitor the operating status of HiRDB. However, you must create a dummy file for notifying VERITAS Cluster Server that HiRDB is currently active as a resource. This dummy file must satisfy all the conditions listed below.

Conditions

- Dummy file is created when HiRDB is started by VERITAS Cluster Server.
- Dummy file is deleted when HiRDB is terminated by VERITAS Cluster Server.
- While dummy file is in existence, the HiRDB resource is considered to be running.

Hint:

If the dummy file is deleted inadvertently, VERITAS Cluster Server assumes that an error has occurred in the resource and switches systems. To prevent this, specify `Critical=0` as the resource attribute value for the `HiRDB_`*x* type resource. Also, to prevent the dummy file from being deleted inadvertently, create it using the name `$PDDIR/.pdveritas`.

(4) Notes

- The standard output messages of HiRDB commands executed from the agent script of a `HiRDB_`*x* type resource are output to VERITAS Cluster Server's log file (`/var/VRTSvcs/log/engine.log_A`).
- Do not delete the dummy file for the `HiRDB_`*x* type resource. Otherwise, system monitoring cannot be performed correctly.

25.8.2 HiRDB resource type definition

To set up HiRDB as a resource, resource type `HiRDB_`*x* for HiRDB must be defined. When a new resource type is created, an agent for monitoring the resource must also be defined. Agent definition is explained in Section *25.8.3 Agent definition pre-preparation*.

(1) HiRDB/Single Server

The following is an example of a resource type definition for a HiRDB/Single Server:

```
type HiRDB_S (
     static str ArgList[] = { PdDir, PdConfPath, Ld_Library_Path, DummyFilePath }
     str PdDir
     str PdConfPath
     str Ld_Library_Path
     str DummyFilePath
)
```

Create a file with this contents under the name `/etc/VRTSvcs/conf/config/HiRDB_STypes.cf`.

(2) HiRDB/Parallel Server

The following is an example of a resource type definition for a HiRDB/Parallel Server:

```
type HiRDB_P (
     static str ArgList[] = { PdDir, PdConfPath, Ld_Library_Path, DummyFilePath }
     str PdDir
     str PdConfPath
     str Ld_Library_Path
     str DummyFilePath
)
```

Create a file with this contents under the name `/etc/VRTSvcs/conf/config/HiRDB_PTypes.cf`.

25.8.3 Agent definition pre-preparation

Define an agent for the newly-created resource type `HiRDB_`*x*. This section explains how to define an agent using a shell script. Before defining an agent, take the pre-preparation steps described below.

- HiRDB/Single Server

  Copy `/opt/VRTSvcs/bin/ScriptAgent` under the name `/opt/VRTSvcs/bin/HiRDB_S/HiRDB-SAgent`.

- HiRDB/Parallel Server

  Copy `/opt/VRTSvcs/bin/ScriptAgent` under the name `/opt/VRTSvcs/bin/HiRDB_P/HiRDB-PAgent`.

25.8.4 Agent definition

Define the action details of the agents shown in Table 25-20.

Table 25-20: Agent action definition items and file names

Agent action	HiRDB type	Script file name
Bringing a resource online	HiRDB/Single Server	`/opt/VRTSvcs/bin/HiRDB_S/online`
	HiRDB/Parallel Server	`/opt/VRTSvcs/bin/HiRDB_P/online`
Taking a resource offline	HiRDB/Single Server	`/opt/VRTSvcs/bin/HiRDB_S/offline`
	HiRDB/Parallel Server	`/opt/VRTSvcs/bin/HiRDB_P/offline`
Monitoring a resource	HiRDB/Single Server	`/opt/VRTSvcs/bin/HiRDB_S/monitor`
	HiRDB/Parallel Server	`/opt/VRTSvcs/bin/HiRDB_P/monitor`

(1) Online script

An online script explains the details of the processing to be performed when an agent brings a resource online. The following processing is required:

- Setting of the environment variables necessary for executing `$PDDIR/bin/pdstart`
- HiRDB startup
- Dummy file creation
- Dummy file mode change

(a) HiRDB/Single Server

The following is an example of an online script for a HiRDB/Single Server:

```
#!/bin/sh
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin:/opt/VRTSvcs/bin:"$2"/bin
export PATH
PDDIR="$2"
PDCONFPATH="$3"
LD_LIBRARY_PATH="$4"
export PDDIR PDCONFPATH LD_LIBRARY_PATH
$PDDIR/bin/pdstart
/bin/touch "$5"
/bin/chmod 0400 "$5"
```

(b) HiRDB/Parallel Server

The following is an example of an online script for a HiRDB/Parallel Server:

```
#!/bin/sh
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin:/opt/VRTSvcs/bin:"$2"/bin
export PATH
PDDIR="$2"
PDCONFPATH="$3"
LD_LIBRARY_PATH="$4"
export PDDIR PDCONFPATH LD_LIBRARY_PATH
$PDDIR/bin/pdstart -q
/bin/touch "$5"
/bin/chmod 0400 "$5"
```

Note

The `pdstart -q` command starts the units in a HiRDB/Parallel Server during use of the system switchover facility.

(2) Offline script

An offline script explains the details of the processing to be performed when an agent

takes a resource offline. The following processing is required:

- Setting of the environment variables necessary for executing `$PDDIR/bin/pdstop`
- HiRDB termination
- Dummy file deletion

(a) HiRDB/Single Server

Following is an example of an offline script for a HiRDB/Single Server:

```
#!/bin/sh
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin:/opt/VRTSvcs/bin:"$2"/bin
export PATH
PDDIR="$2"
PDCONFPATH="$3"
LD_LIBRARY_PATH="$4"
export PDDIR PDCONFPATH LD_LIBRARY_PATH
$PDDIR/bin/pdstop -f -q
/bin/rm -f "$5"
```

Note

Specify the `pdstop -f -q` command to terminate HiRDB forcibly. Executing this command may cause an error shutdown in the RDAREAs on the shared disk. If this happens, use the database recovery utility to recover the RDAREAs on the shared disk.

The offline script is executed during system switchover. At such a time, HiRDB is terminated forcibly so that the systems can be switched immediately and the standby system can restart HiRDB in order to resume operations.

(b) HiRDB/Parallel Server

The following is an example of an offline script for a HiRDB/Parallel Server follows.

```
#!/bin/sh
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin:/opt/VRTSvcs/bin:"$2"/bin
export PATH
PDDIR="$2"
PDCONFPATH="$3"
LD_LIBRARY_PATH="$4"
export PDDIR PDCONFPATH LD_LIBRARY_PATH
$PDDIR/bin/pdstop -z -q
/bin/rm -f "$5"
```

Note

Specify the `pdstop -z -q` command to terminate HiRDB forcibly. Executing this

command may cause an error shutdown in the RDAREAs on the shared disk. If this happens, use the database recovery utility to recover the RDAREAs on the shared disk.

The offline script is executed during system switchover. At such a time, HiRDB is terminated forcibly so that the systems can be switched immediately and the standby system can restart HiRDB in order to resume operations.

(3) Monitor script

A monitor script explains the details of the processing to be performed when an agent monitors a resource (for checking whether or not a resource is online). The following processing is required:

- Check of whether or not a dummy file exists
- Return value setting

The following is an example of a monitor script:

```
#!/bin/sh
if /bin/test -f "$5"
then
    exit 110
else
    exit 100
fi
```

The values of the environment variables and dummy file path names, which will be needed in each script, can be transferred during script execution. For the arguments and return values to be transferred to each script, see the VERITAS Cluster Server manual.

25.8.5 Environment setup file creation

Create the environment setup file for VERITAS Cluster Server (`/etc/VRTSvcs/conf/config/main.cf`) and set up groups and resources.

(1) Resource attribute setting value

Table 25-21 shows the values to be specified for resource attributes. For details on the individual items, see the VERITAS Cluster Server manual.

Table 25-21: Values to be specified for resource attributes

Resource	Attribute	Value to be specified
HiRDB_*x* type resource	Critical	Specify 0.
	PdDir	Specify the HiRDB directory name (`$PDDIR`).
	PdConfPath	Specify the name of the directory storing the HiRDB system definition file (`$PDCONFPATH`).
	`Ld_Library_Path`	Specify the name of the directory storing the HiRDB library (`$LD_LIBRARY_PATH = $PDDIR/lib`).
	DummyFilePath	Specify the dummy file name (`$PDDIR/.pdveritas`).
DiskGroup type resource	DiskGroup	Specify the name of the VERITAS Volume Manager's disk group to be used as a shared disk by HiRDB (monitor mode only).
IP type resource	Device	Specify the name of the NIC device related to the logical IP address to be used by HiRDB.
	Address	Specify the logical IP address to be used by HiRDB.
NIC type resource	Device	Specify the device name of the NIC connected to the network to be used by HiRDB.
	NetworkHosts	Specify the IP address of the host on the network, to be used by HiRDB. This attribute is not required.

(2) Environment setup file example (IP addresses inherited)

```
include "types.cf"
include "HiRDB_STypes.cf"

cluster vcs (
      UserNames = { vcsadm = cDi1yyJLOgPWY }
      CounterInterval = 5
      Factor = { runque = 5, memory = 1, disk = 10, cpu = 25, network = 5 }
      MaxFactor = { runque = 100, memory = 10, disk = 100, cpu = 100, network = 100 }
      )

system mainhost

system reservedhost

snmp vcs (
        TrapList = { 1 = "A new system has joined the VCS Cluster",
                 2 = "An existing system has changed its state",
                 3 = "A service group has changed its state",
                 4 = "One or more heartbeat links has gone down",
                 5 = "An HA service has done a manual restart",
                 6 = "An HA service has been manually idled",
                 7 = "An HA service has been successfully started" }
        )

group gr1 (
        SystemList = { mainhost, reservedhost }
        AutoStartList = { mainhost }
        )

        HiRDB_S gr1_HiRDB_S_UNT1 (
                Critical = 0
                PdDir = "/hirdb/pddir_s"
                PdConfPath = "/hirdb/pddir_s/conf"
                Ld_Library_Path = "/hirdb/pddir_s/lib"
                DummyFilePath = "/hirdb/pddir_s/.pdveritas"
                )

        DiskGroup gr1_DiskGroup_sharedg1 (
                DiskGroup = sharedg1
                )

        IP gr1_IP_logicalhost (
                Device = hme0
                Address = "172.16.161.177"
                )

        NIC gr1_NIC_hme0 (
                Device = hme0
                NetworkHosts = { "172.16.161.1" }
                )

        gr1_HiRDB_S_UNT1 requires gr1_DiskGroup_sharedg1
        gr1_DiskGroup_sharedg1 requires gr1_IP_logicalhost
        gr1_IP_logicalhost requires gr1_NIC_hme0
```

(3) Environment setup file example (IP addresses not inherited)

```
include "types.cf"
include "HiRDB_STypes.cf"

cluster vcs (
      UserNames = { vcsadm = cDi1yyJLOgPWY }
      CounterInterval = 5
      Factor = { runque = 5, memory = 1, disk = 10, cpu = 25, network = 5 }
      MaxFactor = { runque = 100, memory = 10, disk = 100, cpu = 100, network = 100 }
      )

system mainhost

system reservedhost

snmp vcs (
        TrapList = { 1 = "A new system has joined the VCS Cluster",
                  2 = "An existing system has changed its state",
                  3 = "A service group has changed its state",
                  4 = "One or more heartbeat links has gone down",
                  5 = "An HA service has done a manual restart",
                  6 = "An HA service has been manually idled",
                  7 = "An HA service has been successfully started" }
        )

group gr1 (
        SystemList = { mainhost, reservedhost }
        AutoStartList = { mainhost }
        )

        HiRDB_S gr1_HiRDB_S_UNT1 (
                Critical = 0
                PdDir = "/hirdb/pddir_s"
                PdConfPath = "/hirdb/pddir_s/conf"
                Ld_Library_Path = "/hirdb/pddir_s/lib"
                DummyFilePath = "/hirdb/pddir_s/.pdveritas"
                )

        DiskGroup gr1_DiskGroup_sharedg1 (
                DiskGroup = sharedg1
                )

        NIC gr1_NIC_hme0 (
                Device = hme0
                NetworkHosts = { "172.16.161.1" }
                )

        gr1_HiRDB_S_UNT1 requires gr1_DiskGroup_sharedg1
        gr1_DiskGroup_sharedg1 requires gr1_NIC_hme0
```

25.9 Sun Cluster preparations

Read this section when using Sun Cluster in the cluster software. This section covers the following topics:

- Cluster startup
- Shared disk setup (disk group creation)
- Network setup (PNM setup)
- Logical host creation
- Service creation and registration

For details about setting up the environment for Sun Cluster, see the Sun Cluster manual.

25.9.1 Cluster startup

This section explains how to start Sun Cluster, based on the system configuration described below.

Configuration example

- Cluster name: `sun_cluster`
- Configuration node: `sc-node0sc-node1`

Sun Cluster does not have a command for starting the entire cluster. The cluster at the first node is started, then other nodes are added to that cluster. First, scadmin(1M) is used to start the cluster of the master node.

```
[sc-node0] # scadmin startcluster  sc-node0  sun_cluster
Node specified is sc-node0
Cluster specified is sun_cluster
========================== WARNING ================================
=                      Creating a new cluster                      =
===================================================================
You are attempting to start up the cluster node 'sc-node0' as the
only node in a new cluster. It is important that no other cluster
nodes be active at this time. If this node hears from other cluster
nodes, this node will abort. Other nodes may only join after this
command has completed successfully. Data corruption may occur if
more than one cluster is active.

Do you want to continue [y,n,?]   y
```

Next, confirm that the master node has started, then add other nodes to the cluster.

```
[sc-node1] # scadmin startnode
```

When all nodes have been started, use hastat(1M) to check the status of the entire cluster.

```
[sc-node1] # hastat
Getting Information from all the nodes ......
              HIGH AVAILABILITY CONFIGURATION AND STATUS
              -----------------------------------------

LIST OF NODES CONFIGURED IN <sun_cluster> CLUSTER
      sc-node0 sc-node1

CURRENT MEMBERS OF THE CLUSTER
     sc-node0 is a cluster member        <-- Node has started.
     sc-node1 not a cluster member       <-- Node has not started.
```

If startup fails, use `/var/adm/messages` and `/var/opt/SUNWcluster/scadmin.log` to check for errors.

25.9.2 Shared disk setup (disk group creation)

Create a disk group that is to be used as a shared disk. Area setup and formatting for the created disk group are specified when the HiRDB environment for the master node is set. The explanation in this section uses Sun Enterprise Volume Manager.

25.9.3 Network setup (PNM setup)

The network interface to be used by the logical host must be set to PNM. If a dual network interface is used, it must be set in a NAFO group.

25.9.4 Logical host creation

Once the shared disk and network have been set up, create a logical host. The following configuration is assumed here:

Configuration of logical host to be created:

- Logical host name: `sc-lnode0` (IP address is `172.16.170.100`)
- Default master node: `sc-node0`
- Default network interface: `qfe1` (qfe1 and qfe2 have been set up in the NAFO group)
- Disk group: `dg0`
- File system name: `/shdsk/node0`

(1) Registering the host name of the logical host

When a DNS server is not used, register the host name of the logical host in the `hosts` file. This must be set up in both nodes in the cluster.

```
172.16.170.100  sc-lnode0              # Sun Cluster  logical host 0
```

(2) Creating a logical host

After confirming that the cluster has been started, create a logical host. This is performed in one node within the cluster.

```
[sc-node0] # hastat                                          <-- Cluster status check
[sc-node0] # scconf sun_cluster  -L sc-lnode0   \            <-- Logical host name
>               -n  sc-node0, sc-node1          \            <-- Node
>               -g  dg0                         \            <-- Disk group
>               -i  qfe1, qfe1,  sc-lnode0      \            <-- Network
>               -m                                           <-- Automatic changeback disabled
/etc/opt/SUNWcluster/conf/sun_cluster.cdb
Checking node status...
[sc-node0] #
```

Explanation

`-L`: Specifies the logical host name.

`-n`: Specifies the nodes comprising the logical host (separated by commas).

The order in which these nodes are specified corresponds to their priority order. The host specified first becomes the master node. In Sun Cluster, when a high-priority node starts in a logical host without the `-m` specification, changeback occurs automatically.

`-g`: Specifies the disk group to be used by the logical host.

The shared disk specified here is imported and mounted automatically when the logical host starts.

`/etc/opt/SUNWcluster/conf/hanfs/`*vfstab.logical-host-name* must be set up.

`-i`: Specifies the network interfaces to be used by the logical host.

Specify network interfaces for the nodes specified in `-n` in the order specified, and specify the host name of the logical host at the end. If a NAFO group is set up, specify the primary interface.

`-m`: Specifies that automatic changeback is to be disabled.

If `-m` is not specified, changeback occurs automatically when a

higher-priority node, as specified in -n, starts.

(3) Creating the logical host's management file system

Use scconf -F to create a management file system for storing the configuration information of the logical host. A management file system must be created for all nodes that use the logical host.

```
[sc-node0] # scconf sun_cluster -F sc-lnode0
/etc/opt/SUNWcluster/conf/sun_cluster.cdb
Checking node status...
[sc-node0] #
```

When scconf -F terminates normally, the logical host starts in the master node (the first node specified in -n), and the shared disk and logical IP addresses are allocated.

Use vxprint to confirm that a management file system has been created. The logical volume indicated by *disk-group-name-stat* is the management file system.

```
[sc-node0] # vxprint

Disk group: dg0

Y NAME          ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg dg0          dg0          -        -        -        -        -       -
dm dg001        c2t0d0s2     -        17678493 -        -        -       -
v  dg0-stat     fsgen        ENABLED  4096     -        ACTIVE   -       -
pl dg0-stat-01  dg0-stat     ENABLED  7182     -        ACTIVE   -       -
sd dg001-01     dg0-stat-01  ENABLED  7182     0        -        -       -
```

(4) Confirming logical host startup

The steps taken so far should enable the logical host to start. Start the logical host at each node and confirm its operation.

```
[sc-node0] # haswitch  -m sc-lnode0               <-- Stops once.
[sc-node0] # haswitch  sc-node0  sc-lnode0        <-- Starts the logical host.
[sc-node0] # netstat -in                          <-- Verifies logical IP address.
Name   Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
Qfe1:1 1500 172.16.170.0  172.16.170.100 0      0     0      0     0      0
```

Check these items at all nodes.

```
[sc-node1] # haswitch  sc-node1  sc-lnode0        <-- Starts the logical host.
[sc-node1] # netstat -in                          <-- Verifies logical IP address.
```

If the shared disk and logical IP addresses have been assigned correctly, the logical host has been constructed normally.

25.9.5 Service creation and registration

This section provides an overview of Sun Cluster's service control and explains how to register HiRDB as a data service in the Sun Cluster environment using a Sun Cluster control script.

(1) Sun Cluster's service control specification

Use HA-API to register HiRDB as a data service in the Sun Cluster environment. Available APIs include data service registration (`hareg`), cluster status check (`haget`), etc. For details, see the Sun Cluster documentation.

(2) Methods

A method refers to a call-out to a data service that occurs in each step during reconfiguration. The following methods are available:

- Start (`START`, `START_NET`)
- Stop (`STOP`, `STOP_NET`)
- Abnormal termination (`ABORT`, `ABORT_NET`)
- Monitoring control (`FM_INIT`, `FM_START`, `FM_STOP`, `FM_CHECK`)

For the HiRDB data service, register the control scripts that will be called when the Start (`START`, `START_NET`) and Stop (`STOP`, `STOP_NET`) methods occur.

(3) HiRDB data service control script (Sun Cluster control script)

To set up the HiRDB environment as a data service in the Sun Cluster environment, execution control using Sun Cluster HA-API must be carried out. This section explains an example of creating Sun Cluster control scripts for controlling the calling of methods from Sun Cluster. The following names are used for the Sun Cluster control scripts to be created:

- Name of script file to be executed by `START_NET` method:
  `hirdb00-start_net.sh`
- Name of script file to be executed by `STOP_NET` method:
  `hirdb00-stop_net.sh`

Script to be executed by START_NET

```
#!/bin/sh
## ********************************************************************
##  HiRDB START_NET Control Script (for Sun Cluster)
## ********************************************************************
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin:/opt/SUNWcluster/bin:/HiRDB_S1/bin
export PATH
PDDIR=/HiRDB_S1
PDCONFPATH=/HiRDB_S1/conf
LD_LIBRARY_PATH=/HiRDB_S1/lib
export PDDIR PDCONFPATH LD_LIBRARY_PATH

if [ "$1" != "sc-lnode0" ]
then
   exit 0
fi

$PDDIR/bin/pdstart
```

Script to be executed by STOP_NET

```
#!/bin/sh
## ********************************************************************
##  HiRDB STOP_NET Control Script (for Sun Cluster)
## ********************************************************************
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin:/opt/SUNWcluster/bin:/HiRDB_S1/bin
export PATH
PDDIR=/HiRDB_S1
PDCONFPATH=/HiRDB_S1/conf
LD_LIBRARY_PATH=/HiRDB_S1/lib

export PDDIR PDCONFPATH LD_LIBRARY_PATH

if [ "$2" != "sc-lnode0" ]
then
   exit 0
fi

MASTER_HOST=`haget -f master -h sc-lnode0`
if [ $MASTER_HOST != `hostname` ]
then
    exit 0
fi

$PDDIR/bin/pdstop -f -q*
```

Note

The control scripts created above must be stored in the same directory as the HiRDB system definition files (`$PDDIR/conf`). For nodes other than the master node as well, the same files must be stored in the same directory.

* Executing this command may cause an error shutdown in the RDAREAs on the shared disk. If this happens, use the database recovery utility to recover the RDAREAs on the shared disk.

(4) HiRDB data service registration

Use Sun Cluster's `hareg` command to register the HiRDB data service `hirdb00` into the logical host.

```
hareg -r hirdb00 -b "/HiRDB_S1/conf"
  -m START_NET="hirdb00-start_net.sh" -t START_NET=1800
  -m STOP_NET="hirdb00-stop_net.sh"   -t STOP_NET=300
  -h sc-lnode0 -a 1
```

Explanation

`-r`: Specifies the data service name; this name must be unique within the cluster.

`-b`: Specifies the directory for storing the control script file.

`-m`: Specifies `START_NET="`*HiRDB-startup-control-script-file-name*`"`.

`-t`:`START_NET=1800`

Specifies in seconds the amount of time for terminating the action of the HiRDB startup control script. Specify at least 1800 seconds.

`-m`: Specifies `STOP_NET="`*HiRDB-stop-control-script-file-name*`"`.

`-t`: `STOP_NET=300`

Specifies in seconds the amount of time for terminating the action of the HiRDB stop control script. Specify around 300 seconds.

`-h`: Specifies the name of the logical host.

`-a`: Specifies the HA-API version (specify 1).

A data service must be registered or deleted while the logical host is stopped. Sun Cluster can control independently startup and stop of the logical host and startup and stop of data services. Therefore, another service can be added while a service is running in the logical host.

However, if the configuration is changed while a service is running, a problem due to cluster configuration change or a problem caused by the added service, etc., might cause a fail over. Therefore, to ensure stable service operations, do not add a service while another service is running.

25.10 HACMP preparations

For details about setting up the environment for HACMP, see the HACMP documentation.

25.11 ClusterPerfect preparations

Read this section when using ClusterPerfect in the cluster software. This section covers the following topics:

- System configurations unable to perform system switchover
- Network configuration examples
- Scenario preparations
- Shells used when setting HiRDB scenarios

For details about setting up the environment for ClusterPerfect, see the ClusterPerfect manual.

25.11.1 System configurations unable to perform system switchover

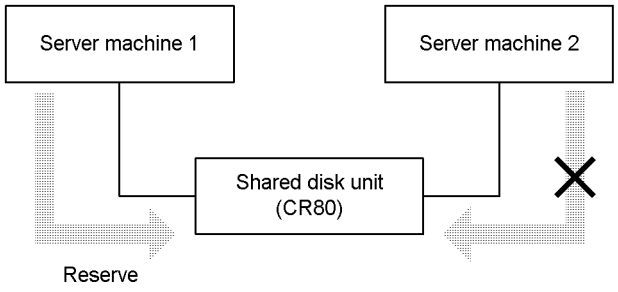
The conditions for the system switchover facility when ClusterPerfect is used are listed below:

- There must be at least two connectable fiber optic cables for the shared disk unit (CR80).
- Only a 1-to-1 switchover configuration can be used; a mutual system switchover configuration cannot be used.

(1) System configuration able to perform system switchover

Figure 25-51 shows a system configuration that can perform system switchover.

Figure 25-51: System configuration able to perform system switchover



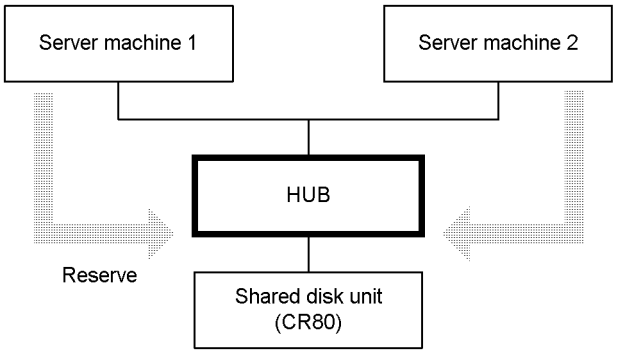
Explanation

The fiber optic cable of each server machine is connected to the two fiber optic cables that connect to the shared disk unit (CR80). Locks operate correctly with this system configuration.

(2) System configurations unable to perform system switchover

Figures 25-52 and 25-53 show system configurations that cannot perform system switchover.

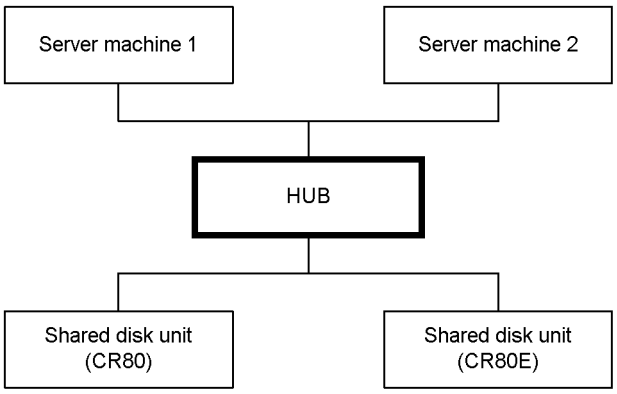
Figure 25-52: System configuration unable to perform system switchover (1)



Explanation

If an optical HUB is placed between CR80 and the server machines, even if server machine 1 is in reserved status, its reserved status is released when server machine 2 restarts. This occurs because server machine 2 generates a LIP that affects server machine 1.

Figure 25-53: System configuration unable to perform system switchover (2)



Explanation

- Because a HUB is being used, the LIP that is generated when the system restarts releases the lock and does not operate properly.
- In the case of SWITCH, LIP is performed independently on each server so it operates properly.

- When using a Fiber-HUB, LIP causes an error in the lock.

Note:

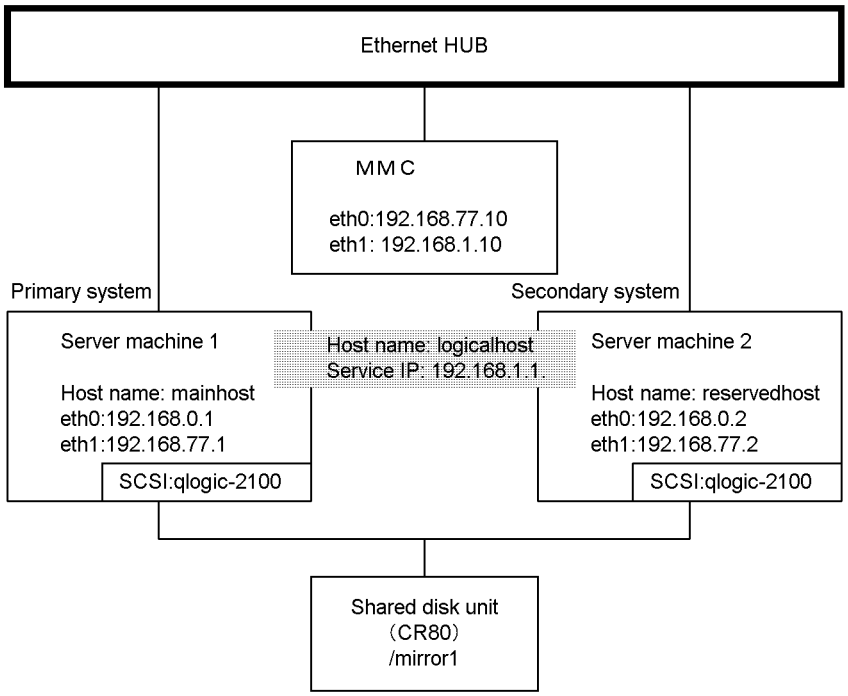
LIP is valid on all devices that are connected to the loop. Therefore, the lock may not be performed properly when two CR80 units are connected or multiple server machines and one CR80 unit are connected via a HUB.

25.11.2 Network configuration examples

(1) When inheriting IP addresses

When inheriting IP addresses, set a logical IP address and host name that correspond to the logical IP address (host name of the IP address that is set in the IP type resource). Also, set up this logical IP address so it can be replaced on multiple server machines using the system switchover facility. Figure 25-54 shows a network configuration example for when inheriting IP addresses.

Figure 25-54: Network configuration example when inheriting IP addresses (using ClusterPerfect)



Explanation

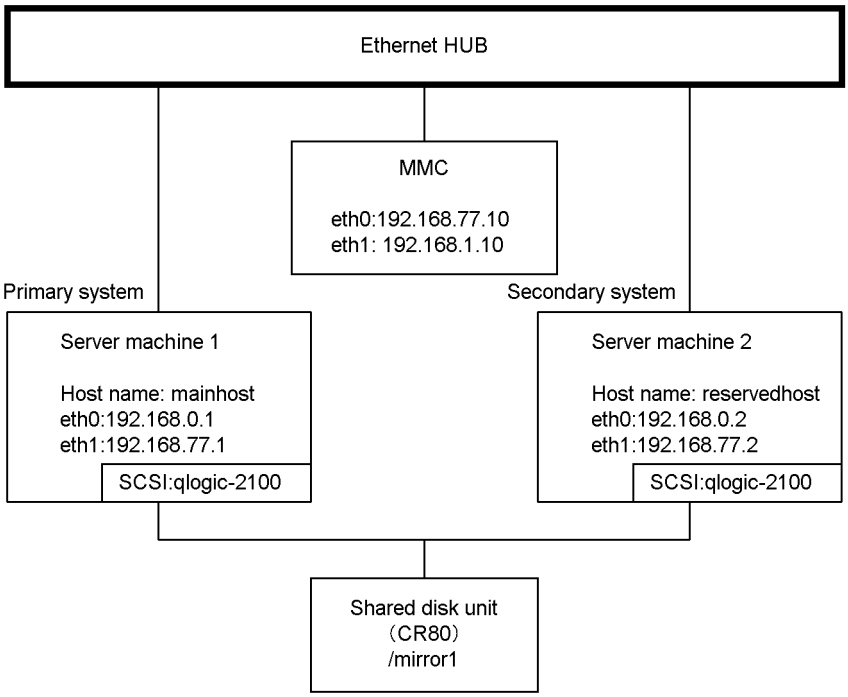
- Set a logical IP address (Service IP: 192.168.1.1) and the host name (`logicalhost`) that correspond to this logical IP address.

- HiRDB is installed on both server machine 1 and server machine 2.
- ClusterPerfect, which monitors server machine 1 and server machine 2, is installed on the MMC machine.
- `/mirror1` is the mount point.
- Create HiRDB system files on the shared disk unit (`/mirror1`).

(2) When not inheriting IP addresses

When not inheriting IP addresses, set a different IP address and host name in the primary system and secondary system. Therefore, you must specify the host name of the secondary system in the `pdunit -c` operand of the HiRDB system common definition. Figure 25-55 shows a network configuration example for when IP addresses are not inherited.

Figure 25-55: Network configuration example when not inheriting IP addresses (using ClusterPerfect)



Explanation

- HiRDB is installed in server machine 1 and server machine 2.
- ClusterPerfect, which monitors server machine 1 and server machine 2, is installed on the MMC machine.

- /mirror1 is the mount point.
- Create HiRDB system files on the shared disk unit (/mirror1).

25.11.3 Scenario preparations

Use the DNCWARE design support system in ClusterPerfect to create and register abnormal termination, startup, termination, and takeover scenarios. For details about creating and registering scenarios, see the ClusterPerfrect manual. The content of each scenario is described below.

(1) Abnormal termination scenario

The contents of the abnormal termination scenario are outlined below:

```
1: Asynchronous calling of process 4 (enddb)
2: Disconnecting of disk 1 (CR80)
3: Disconnecting of service IP1 (serviceIP)
```

(2) Startup scenario

The contents of the startup scenario are outlined below:

```
1: Embedding of service IP1 (serviceIP)
2: Embedding of disk 1 (CR80)
3: Asynchronous calling of process 3 (startdb)
```

(3) Termination scenario

The contents of the termination scenario are outlined below:

```
1: Asynchronous calling of process 4 (enddb)
2: Disconnecting of disk 1 (CR80)
3: Disconnecting of service IP1 (serviceIP)
```

(4) Takeover scenario

The contents of the takeover scenario are outlined below:

```
1: Embedding of service IP1 (serviceIP)
2: Asynchronous calling of process 10 (mount1)
3: Asynchronous calling of process 3 (startdb)
```

25.11.4 Shells used when setting HiRDB scenarios

This section explains examples of shells used to set up HiRDB scenarios.

(1) HiRDB/Single Server

The following examples show shells used to set up HiRDB/Single Server scenarios:

Start

```
#!bin/sh
PDDIR=/hirdb_x
PATH=:/bin:/usr/bin:/usr/bin/ucb:/$PDDIR/bin
PDCONFPATH=$PDDIR/conf
SHLIB_PATH=$PDDIR/lib
LD_LIBRARY_PATH=$PDDIR/lib
export PATH PDDIR PDCONFPATH SHLIB_PATH LD_LIBRARY_PATH
# single
$PDDIR/bin/pdstart
exit 0
```

Terminate

```
#!bin/sh
PDDIR=/hirdb_x
PATH=:/bin:/usr/bin:/usr/bin/ucb:/$PDDIR/bin
PDCONFPATH=$PDDIR/conf
SHLIB_PATH=$PDDIR/lib
LD_LIBRARY_PATH=$PDDIR/lib
export PATH PDDIR PDCONFPATH SHLIB_PATH LD_LIBRARY_PATH
# single
$PDDIR/bin/pdstop -f
exit 0
```

Mount the shared disk

```
#!/bin/sh
/usr/local/DNCWARE/bin/genresv /dev/sdb
sleep 10
/usr/local/DNCWARE/bin/hadkresv /dev/sdb
mount /mirror1
```

(2) HiRDB/Parallel Server

The following examples show shells used to set up HiRDB/Parallel Server scenarios:

Start

```
#!bin/sh
PDDIR=/hirdb_x
PATH=:/bin:/usr/bin:/usr/bin/ucb:/$PDDIR/bin
PDCONFPATH=$PDDIR/conf
SHLIB_PATH=$PDDIR/lib
LD_LIBRARY_PATH=$PDDIR/lib
export PATH PDDIR PDCONFPATH SHLIB_PATH LD_LIBRARY_PATH
# parallel
$PDDIR/bin/pdstart -q
exit 0
```

Terminate

```
#!bin/sh
PDDIR=/hirdb_x
PATH=:/bin:/usr/bin:/usr/bin/ucb:/$PDDIR/bin
PDCONFPATH=$PDDIR/conf
SHLIB_PATH=$PDDIR/lib
LD_LIBRARY_PATH=$PDDIR/lib
export PATH PDDIR PDCONFPATH SHLIB_PATH LD_LIBRARY_PATH
# parallel
$PDDIR/bin/pdstop -z
exit 0
```

Mount the shared disk

```
#!/bin/sh
/usr/local/DNCWARE/bin/genresv /dev/sdb
sleep 10
/usr/local/DNCWARE/bin/hadkresv /dev/sdb
mount /mirror1
```

25.12 Hitachi HA Toolkit Extension preparations (server mode only)

You should read this section when you use Hitachi HA Toolkit Extension. This section provides guidelines for the values to be specified in operands of Hitachi HA Toolkit Extension's `sysdef` and `server` definition statements that relate to HiRDB.

When you use the standby system switchover facility or the standby-less system switchover (1:1) facility, you must set up an operating environment for each unit. When you use the standby-less system switchover (effects distributed) facility, you must set up an operating environment for each server.

25.12.1 sysdef definition statement

(1) servmax operand

You can specify this operand when the version of Hitachi HA Toolkit Extension is 01-04 or later.

For the servers to be switched using Hitachi HA Toolkit Extension, specify 16 or 64 as the maximum number of servers that can be started concurrently as running servers or standby servers on a single server machine.

`16`: Sets 16 as the maximum number of servers that can be started concurrently.

`64`: Sets 64 as the maximum number of servers that can be started concurrently.

Specify `64` when the number of servers, which are the units for switching on a single server machine, exceeds 16.

The number of servers to be switched using Hitachi HA Toolkit Extension also includes products other than HiRDB that are to be switched. HiRDB computes the number of servers to be switched as follows:

- Total number of running units and standby units that are running on a single server machine and that are subject to standby system switchover
- Total number of normal BES units and alternate BES units that are running on a single server machine and that are subject to standby-less system switchover (1:1)
- Total number of host BESs and guest BESs that are running on a single server machine and that are subject to standby-less system switchover (effects distributed)

25.12.2 server definition statement

(1) switchtype operand

Specify the processing to be performed when a server failure is detected.

`switch:`

When HiRDB (or unit for a HiRDB/Parallel Server) terminates abnormally, system switchover is to be performed and HiRDB is to restart on the switchover destination system.

For the standby-less system switchover (1:1) facility, it is recommended that you specify `switch` in the `server` definition statement for the alternate portion created in the alternate BES unit. When `switch` is specified, the system is switched from the alternate portion to the normal BES unit if an error occurs in the alternate BES unit while alternating units; as a result, the load on the alternate BES unit can be reduced after it restarts.

For the standby-less system switchover (effects distributed) facility, it is recommended that you specify `switch` in the `server` definition statement for the guest BES. When `switch` is specified, the guest BES switches to another unit if an error occurs in the unit in which the guest BES is running; as a result, the load on the unit can be distributed after it restarts.

`restart:`

HiRDB (or unit for a HiRDB/Parallel Server) is to be restarted on the system on which the error occurred. Only if HiRDB cannot be restarted on the system on which the error occurred is system switchover to occur and HiRDB to be restarted on the switchover destination system (system switchover is reported by output of the `KFPS00715-E` message).

For the standby-less system switchover (1:1) facility, it is not recommended that you specify `restart` in the `server` definition statement for the alternate portion created in the alternate BES unit. When `restart` is specified, the alternate BES unit continues to assume the alternate processes of the normal BES unit when an error occurs in the alternate BES unit while alternating units; as a result, the load on the alternate BES unit is not reduced after it restarts.

For the standby-less system switchover (effects distributed) facility, it is not recommended that you specify `restart` in the `server` definition statement for the guest BES. When `restart` is specified, the guest BES continues processing if an error occurs in the unit in which the guest BES is running after the unit restarts; as a result, the load on the unit cannot be distributed after it restarts.

`manual:`

The systems are not to be switched automatically even if HiRDB (or unit for a HiRDB/Parallel Server) cannot be restarted.

Hint:

When you use the standby-less system switchover (1:1) facility with a mutual alternating configuration, specify the same value in the `switchtype` operand of the alternate BES unit and the alternate portion.

When you use the standby-less system switchover (effects distributed) facility, specify the same value in the `switchtype` operand of all servers in the HA group.

(2) actcommand operand

Specify a command to be executed at the time of package startup or failover. Hitachi recommends that you not specify this operand in the server mode, because this is an extension issued by system switchover and the command must be issued to the standby system unit.

When a unit in the system manager is subject to system switchover, specifying the `pdstart` command (`pdstart -q` for a HiRDB/Parallel Server) in the `actcommand` operand makes it possible to group this command with the package startup command and start HiRDB. However, because the `actcommand` operand is also executed during system switchover, the `pdstart` command is issued to the standby system HiRDB that is already performing the startup processing and an error is output.

For the standby-less system switchover (1:1) facility, specify the `pdstart -q -c` command in the startup batch file if the `actcommand` operand must be specified for the standby system (alternate portion).

For the standby-less system switchover (effects distributed) facility, you cannot specify the `actcommand` operand.

(3) termcommand operand

Hitachi recommends specifying the `pdstop -f -q` command (`pdstop -z -q` command for a HiRDB/Parallel Server) for two reasons. First, specifying the forced termination option is guaranteed to stop the unit. Second, during planned system switchover, the standby system must inherit IP addresses when it restarts. Also, if planned system switchover is performed when the running system unit has not started, the `pdstop` command specified in the `termcommand` operand will result in an error. If the `termcommand` operand is not specified, the `pdstop` command must be used to terminate forcibly the running unit before system switchover occurs. If a planned system switchover is performed while the running system unit is still active, there will be two running units. If an error occurs, both systems may stop.

For the standby-less system switchover (1:1) facility, specify the `pdstop -z -c` command in the `termcommand` operand of the alternate portion.

For the standby-less system switchover (effects distributed) facility, specify the `pdstop -z -s` command in the `termcommand` operand of all servers within the HA group.

25.13 Differences in the HiRDB operating procedures

The operating procedures for the items listed below depend on whether or not you are using the system switchover facility. The differences in each procedure are described in this section.

- Starting HiRDB (in the server mode)
- Starting HiRDB (in the monitor mode)
- Terminating HiRDB (in the server mode)
- Terminating HiRDB (in the monitor mode)
- Checking status
- Handling of statistics log files
- Notes on operations
- Notes on using the standby-less system switchover facility

25.13.1 Starting HiRDB (in the server mode)

(1) HiRDB/Single Server

The procedure for starting a HiRDB/Single Server when using the system switchover facility is explained below.

Procedure:

1. Use the `pdstart` command to start HiRDB on the running system.
2. Use the `pdstart` command to start HiRDB on the standby system.

(2) HiRDB/Parallel Server (in the case of the standby system switchover facility)

Use the `pdstart` command to start HiRDB on the running system and the standby system.

Inheriting IP addresses

- Starting HiRDB on the running system

  When starting HiRDB on the running system without allocating the IP address in advance, directly log onto the server machine of each unit and execute the `pdstart -q` command.

  If you allocate an IP address for each server machine and execute the `pdstart` command, you can start all units on the running system. When using HiRDB on the running system to start one unit at a time, be sure to allocate the IP address of the unit you will start first.

- Starting HiRDB on the standby system

  Directly log onto a server machine that has standby system units and execute the `pdstart -q` command.

Not inheriting IP addresses

- Starting HiRDB on the running system

  Directly log onto a server machine that has running system units and execute the `pdstart -q` command. Another method is to execute the `pdstart` command to start all units on the running system.

- Starting HiRDB on the standby system

  Directly log onto a server machine that has standby system units and execute the `pdstart -q` command.

(3) Standby-less system switchover (1:1) facility

Table 25-22 lists the methods of starting the normal BES unit and alternate BES unit.

Table 25-22: HiRDB startup methods when using the standby-less system switchover (1:1) facility

Objective	Command to execute	Remarks
Starting the normal BES unit	`pdstart -q`	If this command is executed when a normal BES unit has stopped when alternating units, the normal BES unit is placed in waiting status.[1]
Starting the alternate BES unit	`pdstart -q`	The alternate portion in the alternate BES unit is also started. The alternate portion is placed in waiting status[2] if it was in normal status.
Starting the alternate portion	`pdstart -q -c`	Not necessary because HiRDB starts automatically. The command needs to be executed only when the alternate portion is stopped or is to be reactivated.

Note: Startup cannot be performed at the server level.

[1] The system will only be switched back to the normal BES unit if the normal BES unit is in waiting status (that is, it will not be switched back to normal status from alternating status). If the normal BES unit is in waiting status, the system status is displayed as `SBY` in the execution results of the `pdls -d ha` command.

[2] System switching only occurs when the alternating portion is in waiting status. If the alternate BES unit is in waiting status, the system status is displayed as `SBY` in the execution results of the `pdls -d ha` command.

(4) Standby-less system switchover (effects distributed) facility

Table 25-23 shows how to start the regular unit and a guest BES unit.

Table 25-23: HiRDB startup methods when using the standby-less system switchover (effects distributed) facility

Objective	Command to execute	Remarks
Starting the regular unit	`pdstart -q`	Started when the regular unit is stopped.
Starting the accepting unit	`pdstart -q`	Guest BESs in the accepting unit are also started at the same time.

Objective	Command to execute	Remarks
Starting the host BES or starting the guest BES	`pdstart -u -s`	If a guest BES is stopped, it is placed in accepting status. In general, it is not necessary to use this command because HiRDB starts automatically; this command is needed only when a back-end server or its standby server was stopped explicitly.

(a) Starting the entire system

Table 25-24 shows how to start the entire system.

Table 25-24: Startup method for the entire system

Input location	Command	Operation
Unit where the system manager is defined	`pdstart`	Starts the entire system.

The process of starting the entire system is explained below.

- Server startup: Execute server startup for all host BESs and guest BESs of each unit.

Figure 25-56 shows an example of starting the entire system when the standby-less system switchover (effects distributed) facility is used.

To start the entire system, enter the `pdstart` command from the unit where the system manager is defined, as is the case when standby system switchover is used. The guest BESs in each unit are placed in accepting status automatically.

Figure 25-56: Example of starting the entire system when the standby-less system switchover (effects distributed) facility is used

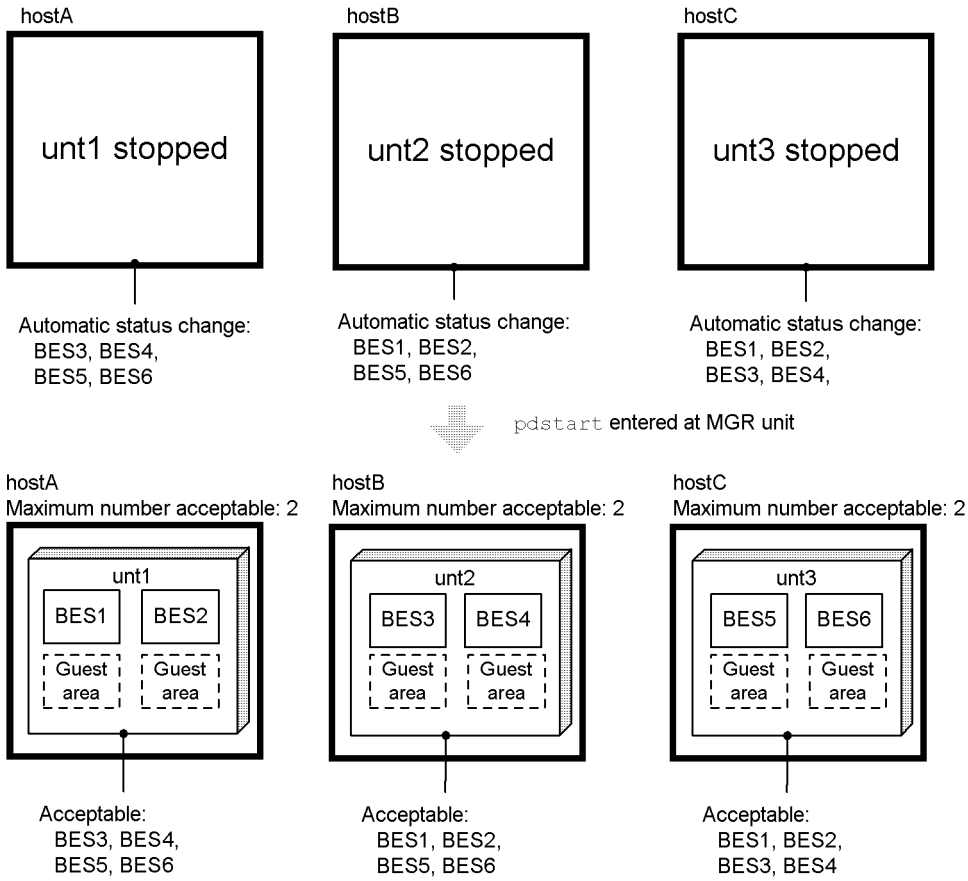


Table 25-25 shows the system startup operations. When system startup is completed, the `KFPS05210-I` message is output.

Table 25-25: System startup operations

Start type	If standby-less system switchover (effects distributed) facility is applicable to the unit	
	No	Yes
System startup	Starts all units (All servers start).	Starts all servers (Not all units start).

Start type	If standby-less system switchover (effects distributed) facility is applicable to the unit	
	No	Yes
Solo-startup of the unit where the system manager is defined (after abnormal termination of the unit where the system manager is defined)	Starts one front-end server, one dictionary server, and one back-end server.	Starts one front-end server, one dictionary server, and one back-end server.
Solo-startup of a unit in an HA group	—	—
Solo-startup of other units	—	—

Legend:

—: Not applicable

(b) Starting a unit

Table 25-26 shows how to start a unit.

Table 25-26: Unit startup method

Input location	Command	Option		Operation
		-q	-u	
Unit where the system manager is defined	`pdstart`	No	Yes	Starts the target unit.
		Yes	No	
Target unit	`pdstart`	Yes	No	

■ Startup modes for units

Table 25-27 shows the unit startup modes. Whether a unit is to be started normally or restarted is determined exclusively by the unit's previous termination mode; it is not affected by the presence or absence of a host BES or guest BES that starts, or by normal start or restart of the server that starts.

Table 25-27: Unit startup modes

Unit's previous termination mode	Host BES or guest BES		Unit startup mode
	Startup of start server	Startup of restart server during startup of start server	
Normal stop	No	No	Normal start

Unit's previous termination mode	Host BES or guest BES		Unit startup mode
	Startup of start server	Startup of restart server during startup of start server	
	Yes	No	
	Yes	Yes	
Planned stop, forced termination, or abnormal termination	No	No	Restart
	Yes	No	
	Yes	Yes	

When the standby-less system switchover (effects distributed) facility is used, determination and processing of restart or normal start is executed for each user server. Therefore, the unit can start normally even when another server within the unit has been terminated forcibly or abnormally or is running on another unit in the HA group.

Also when the standby-less system switchover (effects distributed) facility is used, a unit is restarted after it has been terminated forcibly or abnormally. Because restart processing is executed for each user server, there is no resource (database) to be restored for the unit itself. However, as usual, the operations during unit restart are different from those during a normal start, as shown in Table 25-28.

Table 25-28: Significance of unit restart

Item	Content
Configuration modification check	Checks for definition modifications that increase system resources (such as shared memory) to eliminate the risk of a restart failure (the number of guest BESs that can be accepted can be changed).
System startup completion check	If there is one front-end server, one dictionary server, and one back-end server available during the restart of the unit where the system manager is defined, system startup is considered complete and service begins. During normal startup of the unit where the system manager is defined, the system waits for all servers to be completely started. However, during a restart of the unit where the system manager is defined, the number of servers to wait for is reduced, resulting in an earlier resumption of service acceptance.
Unit run ID inheritance	Unit run ID is inherited before the unit stops. It is used as the reference for output of the `KFPS01826-I` message during unit startup and for regular and automatic execution of `pdcspool`.

■ Unit startup example

Figure 25-57 shows an example of unit start when the standby-less system switchover

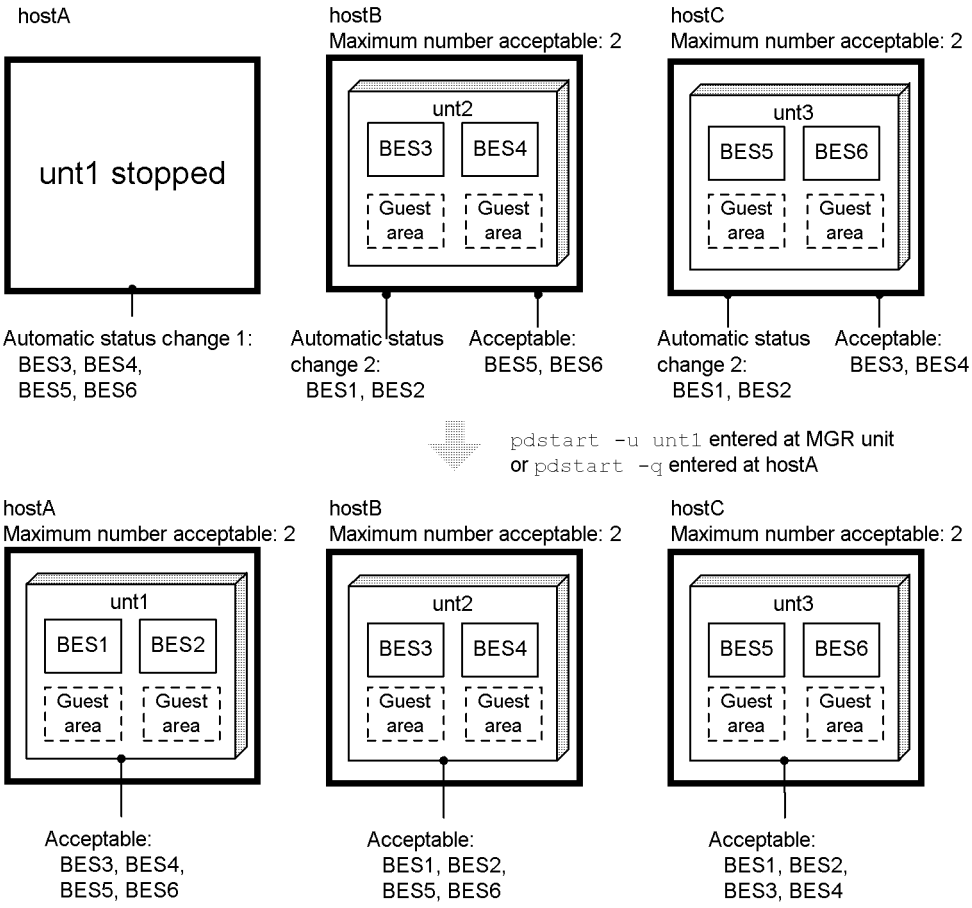
(effects distributed) facility is used. Start the unit as follows:

- Enter the `pdstart -u` command from the unit where the system manager is defined.
- Log onto the host where the unit is located and enter the `pdstart -q` command.

As the unit starts, the system changes the status of the following guest BESs:

- Cancels the accepting status for guest BESs in the unit (Automatic status change 1 in the figure)
- Guest BESs corresponding to host BESs in the unit (Automatic status change 2 in the figure)

Figure 25-57: Unit startup example when the standby-less system switchover (effects distributed) facility is used



■ Unit startup example when there is no running system back-end server

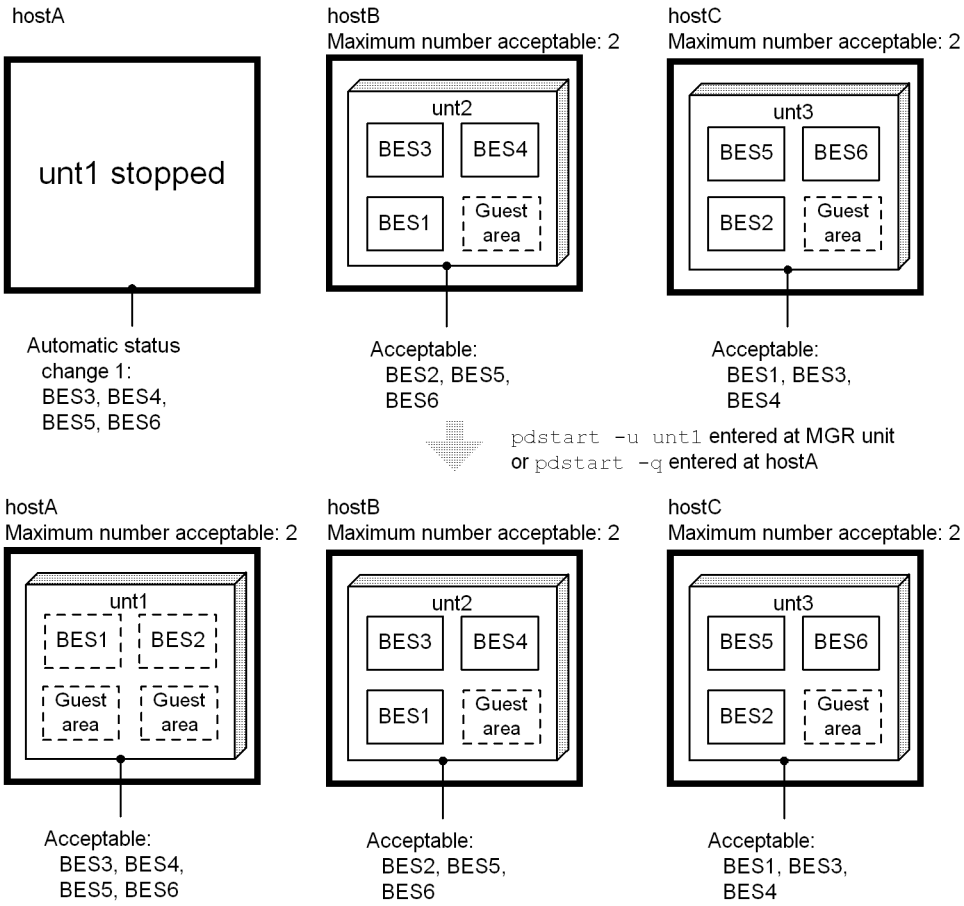
Figure 25-58 shows an example of starting a unit that has no running system back-end server when the standby-less system switchover (effects distributed) facility is used.

The following methods are available for starting the unit:

- Entering the `pdstart -u` command from the unit where the system manager is defined
- Logging onto the host where the unit is located and entering the `pdstart -q` command

As the unit starts, the system changes the status of the guest BESs in the unit (Automatic status change 1 in the figure).

Figure 25-58: Example of starting a unit that has no running system back-end server when the standby-less system switchover (effects distributed) facility is used



You can determine unit startup completion based on the output of the `KFPS05110-I` message:

- For a unit that does not use the standby-less system switchover (effects distributed) facility, the message is output when startup of all servers in the unit has been completed.
- For a unit that uses the standby-less system switchover (effects distributed) facility, the message is output when startup of all running system servers in the unit has been completed.

(c) Starting a server

Table 25-29 shows how to start a server.

Table 25-29: Startup method for a server

Input location	Command	Option			Operation
		-q	-u	-s	
Unit where the system manager is defined	`pdstart`	No	No	Yes	Starts the target servers in all active units in an HA group[1]
		Yes	No	Yes	Starts the host BES (`-u` may be omitted)[2]

[1] The back-end servers in one of the units that are active in the HA group become the running servers and the other units are placed in accepting status.

[2] When the target back-end servers are started successfully as running servers, the active units in the HA group are placed in accepting status automatically.

Table 25-30 shows the processing results during server startup depending on the cluster software used.

Table 25-30: Processing results during server startup

Cluster software	Server type	Other units while the applicable server is active	Activation on the applicable host[3]	Server startup result
HA monitor	Host BES	No	—	Active
		Yes	—	Accepting
	Guest BES	No	—	Waits for running system to start[1]
		Yes	—	Accepting
Hitachi HA Toolkit Extension	—	No	Yes	Active[2]
			No	Accepting
		Yes	Yes	—
			No	Accepting

Legend:

—: Not applicable

[1] HA monitor does not include the concepts of the running server and the standby server once a server has stopped. When the server starts again, the running system and the standby system are determined anew. In such a case, a secondary system (standby system in the default mode) waits for the running system to start when there is no running system. Therefore, if an active guest BES is stopped and then restarted at the same unit, it waits for the running system to start and does not return to active status until the `monact` command is executed.

[2] Even after it stops a server, Hitachi HA Toolkit Extension makes the server appear to the cluster software to remain active. Therefore, when active servers, including guest BESs, are stopped and started on the same unit, they are placed on active status again.

[3] This is equivalent to package activation in MC/ServiceGuard or group start in VERITAS Cluster Server.

■ Server startup example

Figure 25-59 shows an example of starting a running system server when the standby-less system switchover (effects distributed) facility is used.

The following method is available for starting the running system server:

- Entering the `pdstart -s` command from the unit where the system manager is defined

As the server starts, the system changes the status of the guest BESs corresponding to the server (Automatic status change in the figure).

To check whether the server is the running system, use the `pdls -d ha` command.

Figure 25-59: Example of starting a running system server when the standby-less system switchover (effects distributed) facility is used

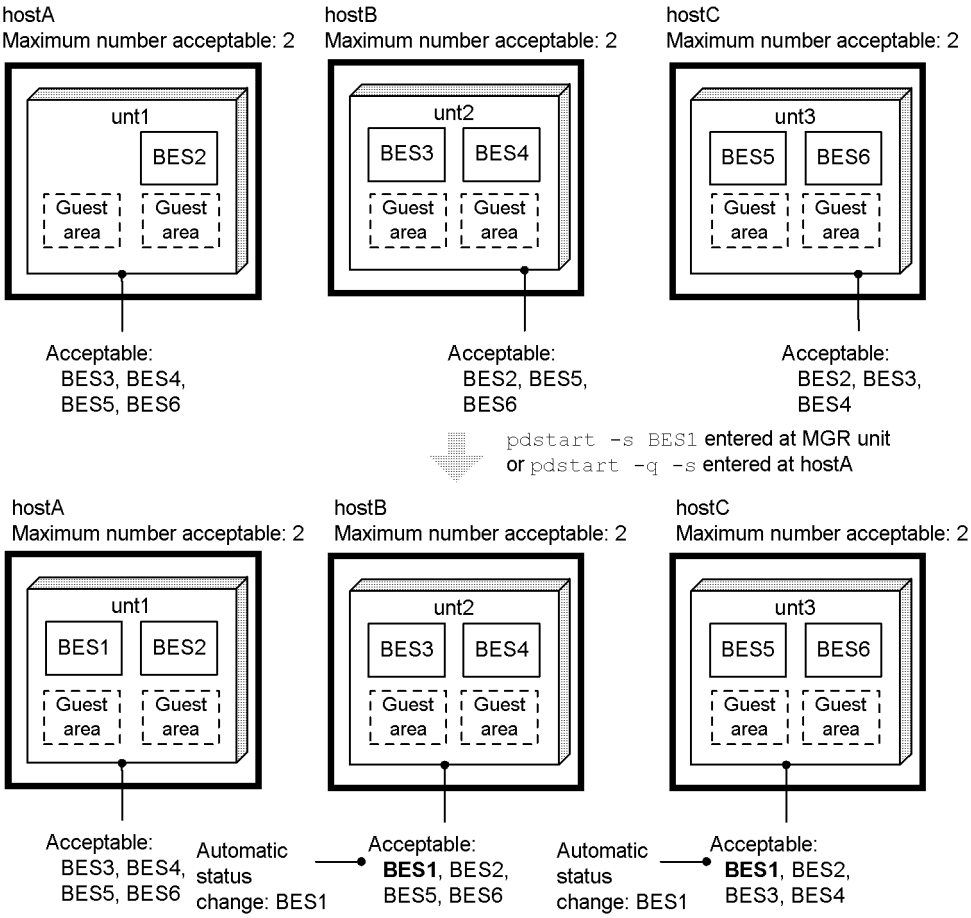


Figure 25-60 shows an example of starting a standby system server when the standby-less system switchover (effects distributed) facility is used.

The following method is available for starting a standby system server:

- Entering the `pdstart -u -s` command from the unit where the system manager is defined

The specified host BES is placed in accepting status. To check whether the server is the running system, use the `pdls -d ha` command.

Figure 25-60: Example of starting a standby system server when the standby-less system switchover (effects distributed) facility is used

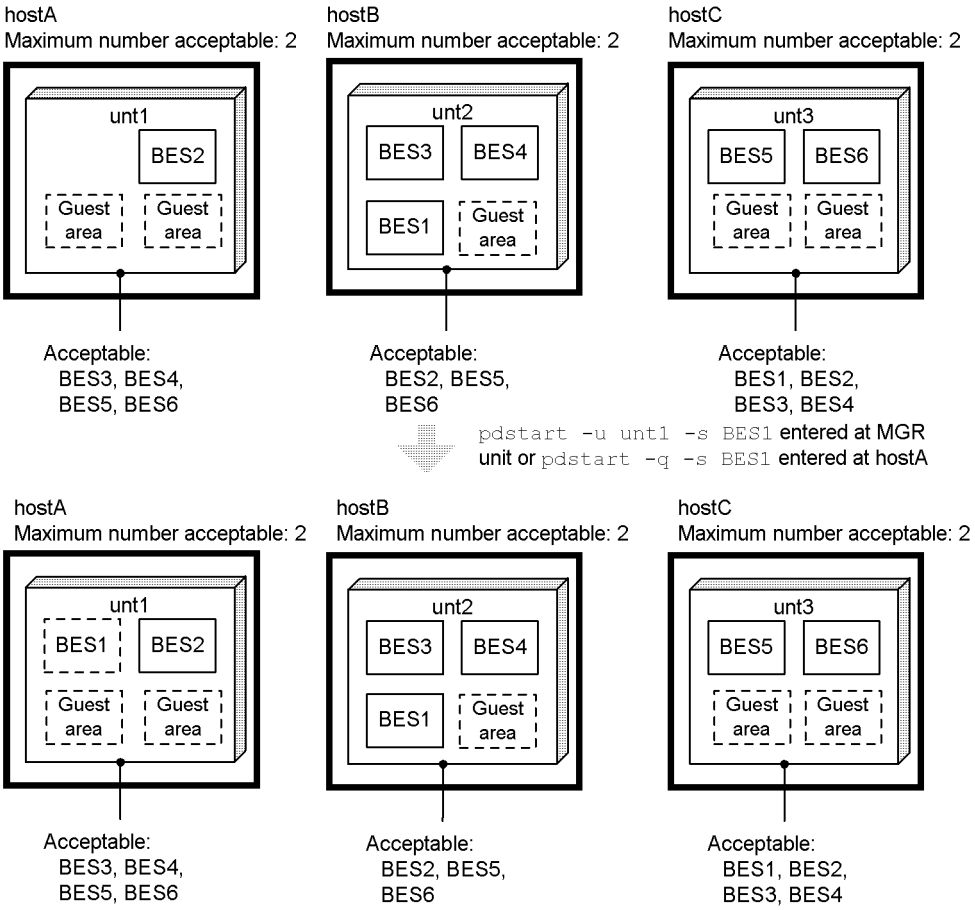


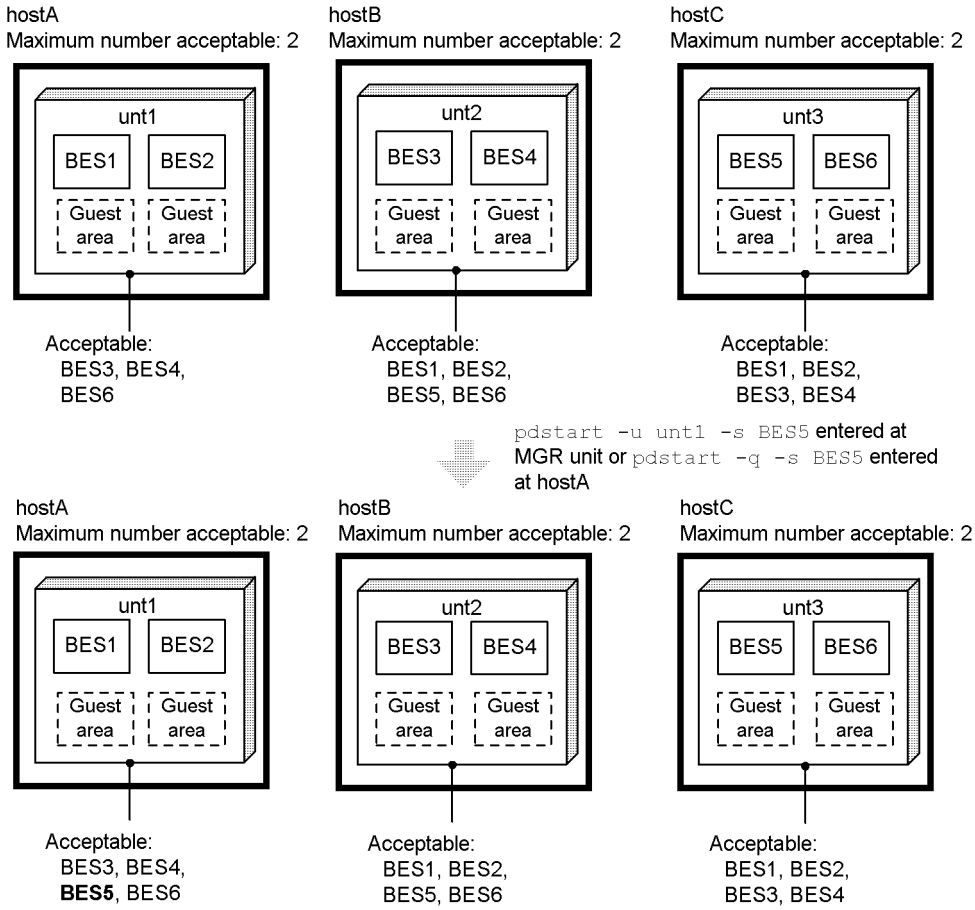
Figure 25-61 shows an example of a guest server status change when the standby-less system switchover (effects distributed) facility is used.

The following method is available to make a guest BES in a unit accepting:

- Entering the `pdstart -u -s` command from the unit where the system manager is defined

The specified guest BES is placed in accepting status.

Figure 25-61: Status change example of a guest server when the standby-less system switchover (effects distributed) facility is used



(5) Notes

Common notes

- When executing the `pdstart -q` command, you must start all units within 20 minutes from the time the first unit starts. If all units cannot be started within 20 minutes, HiRDB startup processing terminates. Note that this value of 20 minutes for the startup time limit can be modified with the `pd_reduced_check_time` operand. The value of 20 minutes is the default for this operand.
- The `-i` option, `-r` option, and `dbdestroy` option cannot be specified when HiRDB on a standby system executes the `pdstart` command.

- Execute the `pdstart -r` command after HiRDB on the running system and standby system terminates. When the `pdstart -r` command is used to start HiRDB, HiRDB is not subject to system switchover. After a process such as database recovery processing terminates, terminate HiRDB then start it on the running system and standby system.

Notes on using the rapid system switchover facility

The notes explained here apply only when you are using Hitachi HA Toolkit Extension.

After startup processing for the running system unit is finished, start the standby system unit that uses the rapid system switchover facility. If the standby system unit is started before the running system unit has started, the standby system unit waits for startup of the running system unit to finish. If the running system unit does not start up within the waiting time limit, the standby system unit outputs abort code `Phi1012` and terminates abnormally.

Notes on using the standby-less system switchover (1:1) facility

The notes explained here apply only when you are using Hitachi HA Toolkit Extension.

After starting either the normal BES unit or the alternate BES unit, start the other idle unit within 20 minutes. If the standby system unit is started before the running system unit has started, the standby system unit waits for startup of the running system unit to finish. If the running system unit does not start within the waiting time limit, the standby system unit outputs abort code `Phi1012` and terminates abnormally.

During normal operation, the normal BES unit becomes the running system and the alternate portion becomes the standby system. When alternating units, the alternate portion becomes the running system and the normal BES unit becomes the standby system.

Notes on using MC/ServiceGuard

- When starting HiRDB, the MC/ServiceGuard package must start normally on the running system. Therefore, before starting HiRDB, confirm that the package has started. Use MC/ServiceGuard commands to confirm that the package has started or to start a package.
- When the running system unit has stopped (including when the unit terminates abnormally), MC/ServiceGuard may recognize that node as one that cannot be switched during a system switchover. In such a case, that node cannot be switched even if HiRDB is waiting. Use an MC/ServiceGuard command to place that node in system switchable status.

Notes on using Hitachi HA Toolkit Extension

If HiRDB is started without activating service processing for Hitachi HA Toolkit Extension, both systems will start as standby systems. In such a case, follow the applicable procedure explained in Table 25-31.

service processing for Hitachi HA Toolkit Extension

Table 25-31: Procedures to perform when HiRDB is started without activating service processing for Hitachi HA Toolkit Extension

Condition	Procedure
User server hot standby is applied to the unit	Message `KFPS01872-I`, which indicates that both systems started as standby systems, is output. This message is output to both systems. The procedure for resolving this problem is explained below. Procedure 1. Use the terminate waiting system command of Hitachi HA Toolkit Extension to terminate both systems. 2. Activate the cluster software on the running system. 3. Start the running system unit. 4. Confirm* that startup processing of the running system unit is finished, then start the standby system unit.
Rapid system switchover facility is applied to the unit	Message `KFPS01854-E` is output and the primary system unit terminates abnormally (abort code: `Psadhfe`). The secondary system unit waits for the primary system unit to start as the running system unit. The procedures for resolving this problem are explained below. Starting the primary system as the running system 1. Activate the cluster software on the primary system. 2. Start the primary system unit as the running system unit. 3. Because the waiting time is elapsing, if the secondary system (standby system) unit terminates abnormally, confirm* that the startup process of the running unit is complete, and then start the standby system unit. Starting the secondary system as the running system 1. Use the `pdstop -z` (`pdstop -f` for a HiRDB/Single Server) command to terminate the secondary unit forcibly. 2. Activate service processing for Hitachi HA Toolkit Extension on the secondary system. 3. Start the secondary system unit as the running system unit. 4. Confirm* that the startup process of the running unit is complete, and then start the standby system unit.

Condition	Procedure
Unit for which the standby-less system switchover (1:1) facility is applicable	Message KFPS01854-E is output and the normal BES unit terminates abnormally (abort code: Psdahfe). The alternate portion waits for the normal BES unit to start. The procedure for resolving this problem is explained below. Procedure 1. Activate service processing for Hitachi HA Toolkit Extension on the normal BES unit. 2. Start the normal BES unit. 3. Because the waiting time is elapsing, if the waiting status of the alternate portion is released, confirm* that the startup process of the normal BES unit is complete, and then place the alternate portion in waiting status.
Unit for which the standby-less system switchover (effects distributed) facility is applicable	Servers in both the running system and the standby system are placed in standby status and can be operated by the user. Neither system is terminated abnormally. In this case, you can activate the cluster software on the host on which the server becomes the running system and complete the startup process (start the service process of Hitachi HA Toolkit Extension).

* You can use the following to confirm that unit startup processing is complete:

- STATUS in the execution results of the pdls command displays ACTIVE.
- The KFPS05210-I or KFPS05110-I message is output.

Notes about using the HA monitor

Before you start the running unit, use HA monitor's monshow command to make sure that the standby unit has stopped. The monshow command does not display any inactive system. If the command displays the status of the standby system, the standby status has not stopped.

An attempt to start the running unit immediately after its termination may result in output of the KFPS01878-I and KFPS00715-E messages, because the standby unit is still engaged in termination processing. If an attempt to start the running unit has failed, follow the procedure below to start the unit:

1. Use HA monitor's monshow command to make sure that the standby unit has stopped.
2. Execute the pdrpause command to restart HiRDB's process service.
3. Use the pdstart command to start the running unit.

25.13.2 Starting HiRDB (in the monitor mode)

Use a command in the cluster software (monbegin command in the case of HA monitor) to start HiRDB on the running system and standby system. If HiRDB cannot be started for either of the following reasons, correct the problem and then use the pdstart (pdstart -q for a HiRDB/Parallel Server) command to start HiRDB:

- An error occurred that requires setup to be performed again with `pdsetup`.
- Restart failed when no user command was specified during a restart attempt.

Notes

- Execute the `pdstart -r` command after HiRDB on the running system and standby system terminates. When the `pdstart -r` command is used to start HiRDB, HiRDB is not subject to system switchover. After a process such as database recovery processing terminates, terminate HiRDB, then start it on the running system and standby system.
- For a HiRDB/Parallel Server, units not subject to system switchover (units specified by `pd_ha_unit=nouse`) cannot be started by a cluster software command. In such a case, log on directly to the server machine on the applicable unit and execute the `pdstart -q` command.

Note on using MC/ServiceGuard

When HiRDB termination terminates abnormally, the status of the MC/ServiceGuard package is `up` or `running`. In such a case, use the `pdstart` command to separately restart HiRDB.

Procedure for starting HiRDB when using ClusterPerfect

To start HiRDB when using ClusterPerfect:

1. Check to see if the `dncware_daemon` on the primary system and secondary system is set to start automatically. You must have root privileges to perform this check.

   ```
   # chkconfig -list

   dncware_daemon 0:off  1:off  2:off  3:off  4:off  5:off
   6:off
   ```

2. If the `dncware_daemon` is off, make the specification to turn it on.

   ```
   # chkconfig dncware_daemon on

   dncware_daemon 0:off  1:off  2:off  3:on  4:on  5:on
   6:off
   ```

3. Restart the server machines for the primary system and the secondary system.
4. Activate the DNCWARE operation support system for MMC.
5. Activate the ClusterPerfect system information window. This window displays the status of the primary system and secondary system.
6. Activate Resource 1 and Resource 2. Start the server (HiRDB) from this window.

For details about activation and the window status, see the ClusterPerfect manual.

25.13.3 Terminating HiRDB (in the server mode)

(1) Standby system switchover facility

Table 25-32 lists the methods of terminating HiRDB when using the standby system switchover facility.

Table 25-32: Terminating HiRDB when using the standby system switchover facility

Condition		Termination method
Cluster software used is HA monitor	Terminating both the running system and the standby system	Use the `pdstop` command to terminate the running system HiRDB. The standby system HiRDB will terminate together with the running system. This situation also applies during planned termination or forced termination. To terminate at the unit level, use the `pdstop -u` command to terminate the running system unit. The standby system unit will terminate together with the running system unit. This situation also applies when the `pdstop -z` command is executed.
	Terminating only the standby system	Use the HA monitor `monsbystp` command to terminate only the standby system.
Cluster software other than HA monitor is used	Terminating both the running system and the standby system*	Use the `pdstop` command to terminate the running system HiRDB, then use the `hatesbystp` command in Hitachi HA Toolkit Extension to terminate HiRDB. Executing the `pdstop` command does not terminate the standby system HiRDB. This situation also applies to planned termination or forced termination. To terminate at the unit level, use the `pdstop -u` command to terminate the running system unit, then use the `hatesbystp` command in Hitachi HA Toolkit Extension to terminate the standby unit. Executing the `pdstop -u` command does not terminate the standby system unit. This situation also applies when the `pdstop -z` command is executed.
	Terminating only the standby system	Use the `hatesbystp` command in Hitachi HA Toolkit Extension to terminate the standby system HiRDB.

*

- When terminating the running system HiRDB (or unit), be sure to terminate the standby system HiRDB also.
- Be sure to terminate the standby system HiRDB before restarting the running system HiRDB (or unit).

(2) Standby-less system switchover (1:1) facility

Table 25-33 lists the methods of terminating HiRDB when you use the standby-less system switchover (1:1) facility.

Table 25-33: Terminating HiRDB when using the standby-less system switchover (1:1) facility

Objective		Command to execute	Remarks (effect on other unit or alternate portion)
Terminating HiRDB		`pdstop`	No specific operation needs to be performed on the alternate BES. Operation does not change even when alternating units.
Terminating the normal BES unit		`pdstop -u`	Waiting status[1] of the alternate portion is released.
Terminating the alternate BES unit	Normal operation	`pdstop -u`	Waiting status of the alternate portion in the alternate BES unit is released.
	Alternating		The alternate portion in the alternate BES unit also stops. Also, when the normal BES unit is in waiting status,[2] the waiting status of the alternate BES unit is released.
Terminating the alternate portion when alternating units		`pdstop -u`	Specify for the unit identifier the unit identifier of the normal BES unit.
Releasing waiting status of the alternate portion		`hatesbystp`[3]	None
Releasing waiting status of the normal BES unit		`hatesbystp`[4]	

Note: Termination at the server level is not possible.

[1] The system is switched to the alternate BES unit only if the alternate portion is in waiting status. If the alternate portion is in waiting status, the system status is displayed as `SBY` in the execution results of the `pdls -d ha` command.

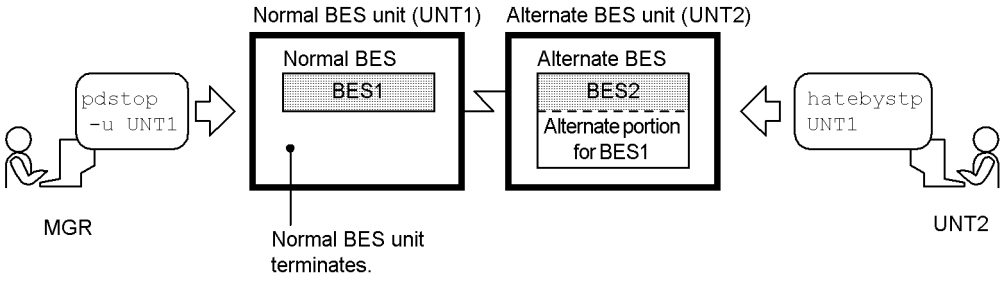
[2] The system will be switched back to the normal BES unit only if the normal BES unit is in waiting status (that is, it will not be switched back to normal status from alternating status). If the normal BES unit is in waiting status, the system status is displayed as `SBY` in the execution results of the `pdls -d ha` command.

[3] Use the `monsbystp` command of HA monitor or the `hatesbystp` command of Hitachi HA Toolkit Extension to release the waiting status of the alternate portion. Specify in the `hatesbystp` command the alias of the normal BES unit that corresponds to the alternate portion to be released from standby status.

[4] Use the `monsbystp` command of HA monitor or the `hatesbystp` command of Hitachi HA Toolkit Extension to release the waiting status of the normal BES unit. Specify in the `hatesbystp` command the unit identifier of the normal BES unit that releases the waiting status.

Examples of terminating an alternate BES unit or normal BES unit are provided below.

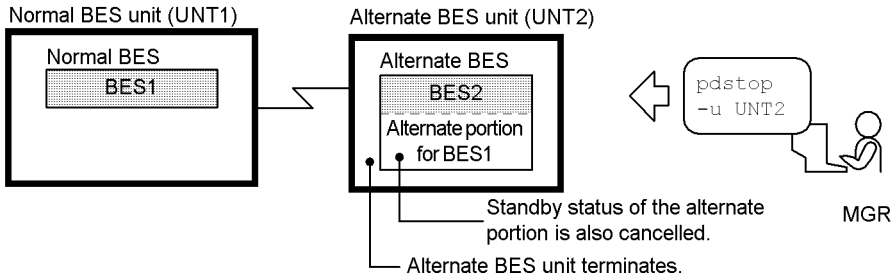
Example 1: Terminating a normal BES unit (normal operation, one-way alternating configuration)



To start the normal BES unit (`UNT1`) that has stopped:

1. Use the `pdstart -q` command to start the normal BES unit (`UNT1`).
2. Use the `pdstart -q -c` command to place the alternate portion of BES1 in waiting status.

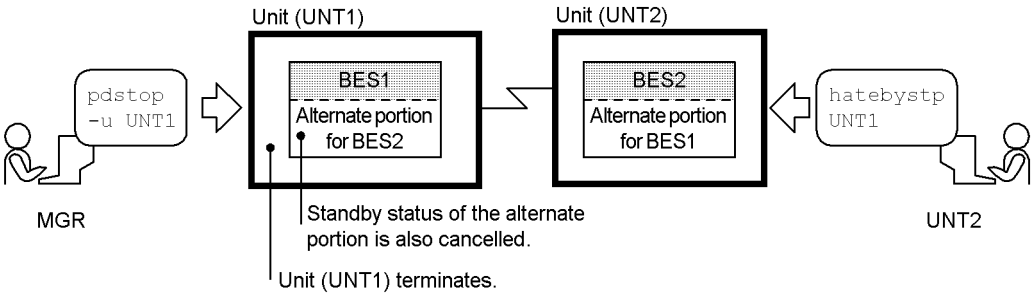
Example 2: Terminating an alternate BES unit (normal operation, one-way alternating configuration)



To start the alternate BES unit (`UNT2`) that has stopped:

1. Use the `pdstart -q` command to start the alternate BES unit (`UNT2`). The alternate portion of BES1 is also placed in waiting status.

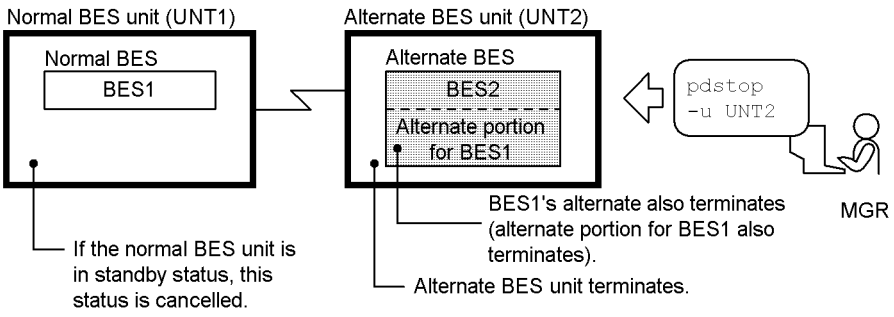
Example 3: Terminating a unit (normal operation, mutual alternating configuration)



To start a unit (`UNT1`) that has stopped:

1. Use the `pdstart -q` command to start the unit (`UNT1`). The alternate portion of BES2 is also placed in waiting status.
2. Use the `pdstart -q -c` command to place the alternate portion of BES1 in waiting status.

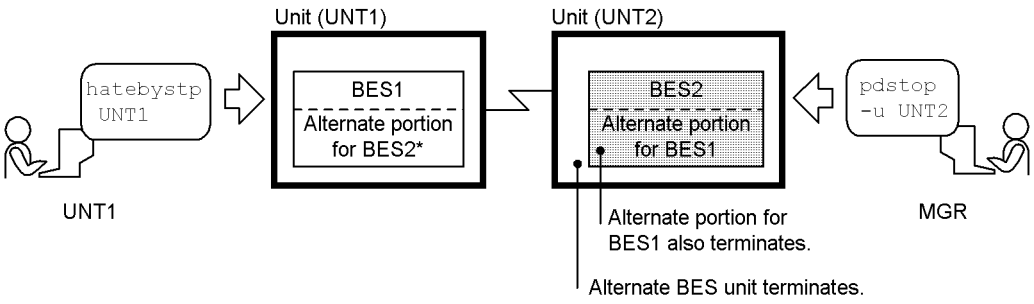
Example 4: Terminating a unit (alternating, one-way alternating configuration)



To return to normal status:

1. Reactivate the package for BES1. This step is necessary only when using Hitachi HA Toolkit Extension.
2. Use the `pdstart -q` command to start the alternate BES unit (`UNT2`).
3. Use the `pdstart -q` command to start the normal BES unit (`UNT1`).

Example 5: Terminating a unit (alternating, mutual alternating configuration)

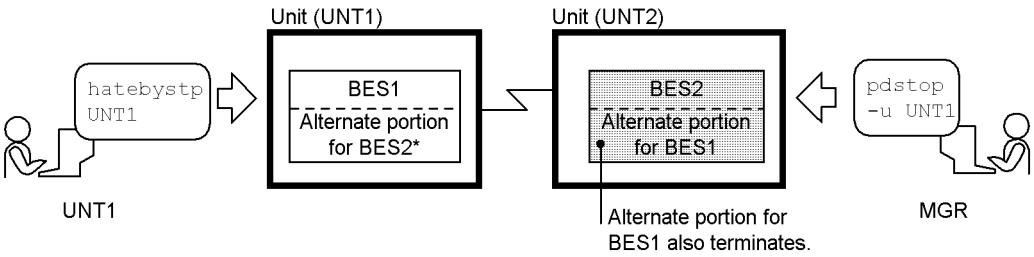


* The alternate portion of BES2 is inactive.

To return to normal status:

1. Reactivate the package for BES1. This step is necessary only when using Hitachi HA Toolkit Extension.
2. Use the `pdstart -q` command to start the unit (`UNT2`).
3. Use the `pdstart -q` command to start the unit (`UNT1`).

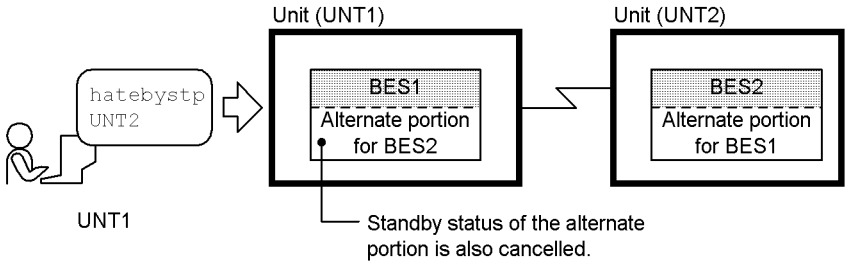
Example 6: Terminating the alternate portion (alternating, one-way alternating configuration)



To return to normal status:

1. Reactivate the package for BES1. This step is necessary only when using Hitachi HA Toolkit Extension.
2. Use the `pdstart -q` command to start the unit (`UNT1`).

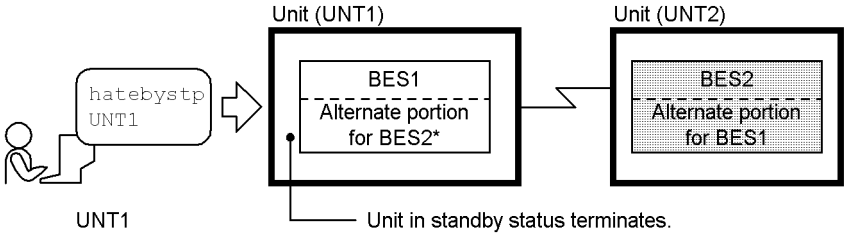
Example 7: Releasing the waiting status of the alternate portion (normal operation, mutual alternating configuration)



To place the alternate portion of BES2, which has stopped, in waiting status:

1. Use the `pdstart -q -c` command to place the alternate portion of BES2 in waiting status.

Example 8: Stopping a unit in waiting status (alternating, mutual alternating configuration)



* The alternate portion of BES2 is inactive.

To place a stopped unit (UNT1) in waiting status:

1. Use the `pdstart -q` command to place the unit (UNT1) in waiting status.

(3) Standby-less system switchover (effects distributed) facility (terminating both the running system and the standby system)

This section explains how to terminate both the running system and the standby system.

(a) Stopping the entire system

Table 25-34 shows how the entire system is stopped when the standby-less system switchover (effects distributed) facility is used.

Table 25-34: Stopping the entire system when the standby-less system switchover (effects distributed) facility is used

Input location	Command	Option	Condition	Operation
		-f	Forcibly/abnormally terminated server?	
Unit where system manager is defined	pdstop	No	Yes[1]	Error (a new message, such as KFPS05063-E or its equivalent, is output)
		No	No[2]	Stops the system.
		Yes	Yes	Stops the system forcibly (some units are already stopped).
		Yes	No	Stops the system forcibly.

[1] Even if a server has been terminated forcibly or abnormally, it is considered that there is no forcibly/abnormally terminated server if this server is in one of the following statuses:

- Restarted at another unit and currently running.
- Restarted at another unit and has already been stopped normally.

[2] *No* under *Forcibly/abnormally terminated server?* means one of the following:

- There are no servers in the system that have been terminated forcibly or abnormally.
- There is a server in the system that was terminated forcibly or abnormally, but it has since been restarted at another unit and is currently running.
- There is a server in the system that was terminated forcibly or abnormally, but it has since been restarted at another unit and has already been stopped normally.

Table 25-35 shows the processing that occurs during system termination.

Table 25-35: Processing that occurs during system termination

Target	Processing detail
Unit	While the system is being stopped, the system manager stops all units (by executing pdstop -u UID (-f) or its equivalent).
Server	When a unit is stopped while the system is being stopped, the system stops all host BESs and guest BESs at that unit (by executing pdstop -q -s *server-name* (-f) or its equivalent).

When the system is stopped, the accepting status for guest BESs at an accepting unit

is cancelled automatically regardless of whether the system is being terminated normally terminated or forcibly. This cancellation occurs even if a guest BES is active. For this reason, you need not take any action with regard to guest BESs.

Table 25-36 shows the processing that occurs for the various back-end servers during system termination when the standby-less system switchover (effects distributed) facility is used.

Table 25-36: Processing that occurs for the various back-end servers during system termination when the standby-less system switchover (effects distributed) facility is used

Back-end server's status		Processing
Host BES	Running	Stops.
	Accepting status	Cancels the accepting status automatically.
Guest BES	Running	Stops automatically.
	Accepting status	Cancels the accepting status automatically.

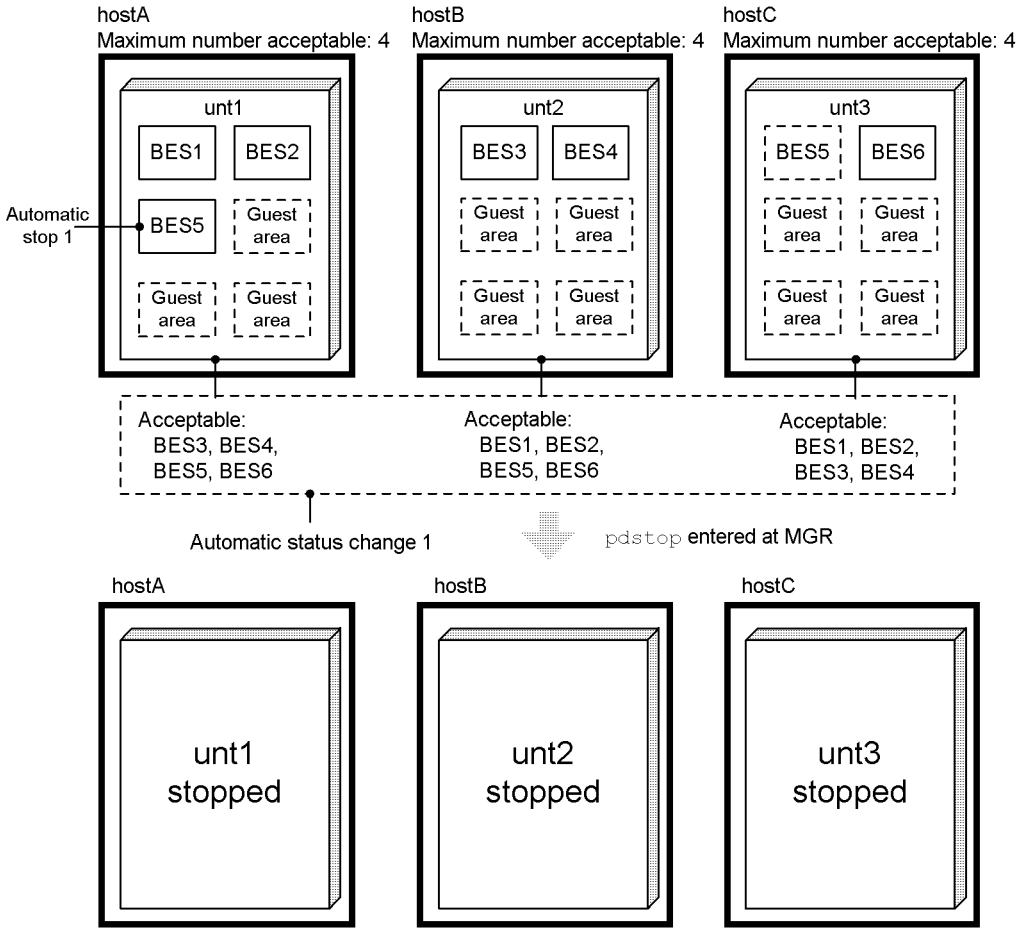
Figure 25-62 shows an example of system termination. In this example, the following servers are stopped when the system is stopped:

- Host BESs
- Guest BES that is running (*Automatic stop 1* in the figure)

Additionally, the accepting status for the following servers is cancelled:

- Guest BESs that are in accepting status (*Automatic status change 1* in the figure)

Figure 25-62: System termination example



(b) Stopping a unit

Table 25-37 shows how a unit is stopped when the standby-less system switchover (effects distributed) facility is used.

Table 25-37: Stopping a unit when the standby-less system switchover (effects distributed) facility is used

Input location	Command	Option			Operation
		-z	-u	-f	
Unit where system manager is defined	pdstop	No	Yes	No	Stops the target unit normally.
				Yes	Stops the target unit forcibly.

Input location	Command	Option			Operation
		-z	-u	-f	
Target unit	`pdstop`	Yes	No	No	Stops the target unit forcibly.

To stop a unit, stop all host BESs and guest BESs at the unit (by executing `pdstop -q -s` *server-name* (`-f`) or its equivalent).

Table 25-38 shows whether a unit can be stopped normally depending on the status of servers in the unit. When the standby-less system switchover (effects distributed) facility is used, a unit can be stopped normally regardless of whether any of its servers, host BESs or guest BESs, have stopped abnormally by themselves or have been stopped forcibly.

Table 25-38: Whether a unit can be stopped normally depending on the status of servers in the unit

Server status (both BESs and guest BESs)			Can the unit be stopped normally?
Starting/stopping[1]	On standby[2]	Stopped[3]	Standby-less system switchover (effects distributed) facility
No	No	No	Yes
		Yes	Yes
	Yes	No	Yes
		Yes	Yes
Yes	No	No	No
		Yes	No
	Yes	No	No
		Yes	No

[1] Includes starting normally, restarting, stopping normally, stopping according to plan, being stopped forcibly, or being stopped abnormally.

[2] Being blocked by `mon_standby()` or `mon_connect()`.

[3] Includes stopped normally, stopped according to plan, stopped forcibly, stopped abnormally, or guest area that has become unallocated after guest BES was stopped.

Table 25-39 shows the processing for the various back-end servers during unit termination when the standby-less system switchover (effects distributed) facility is

used.

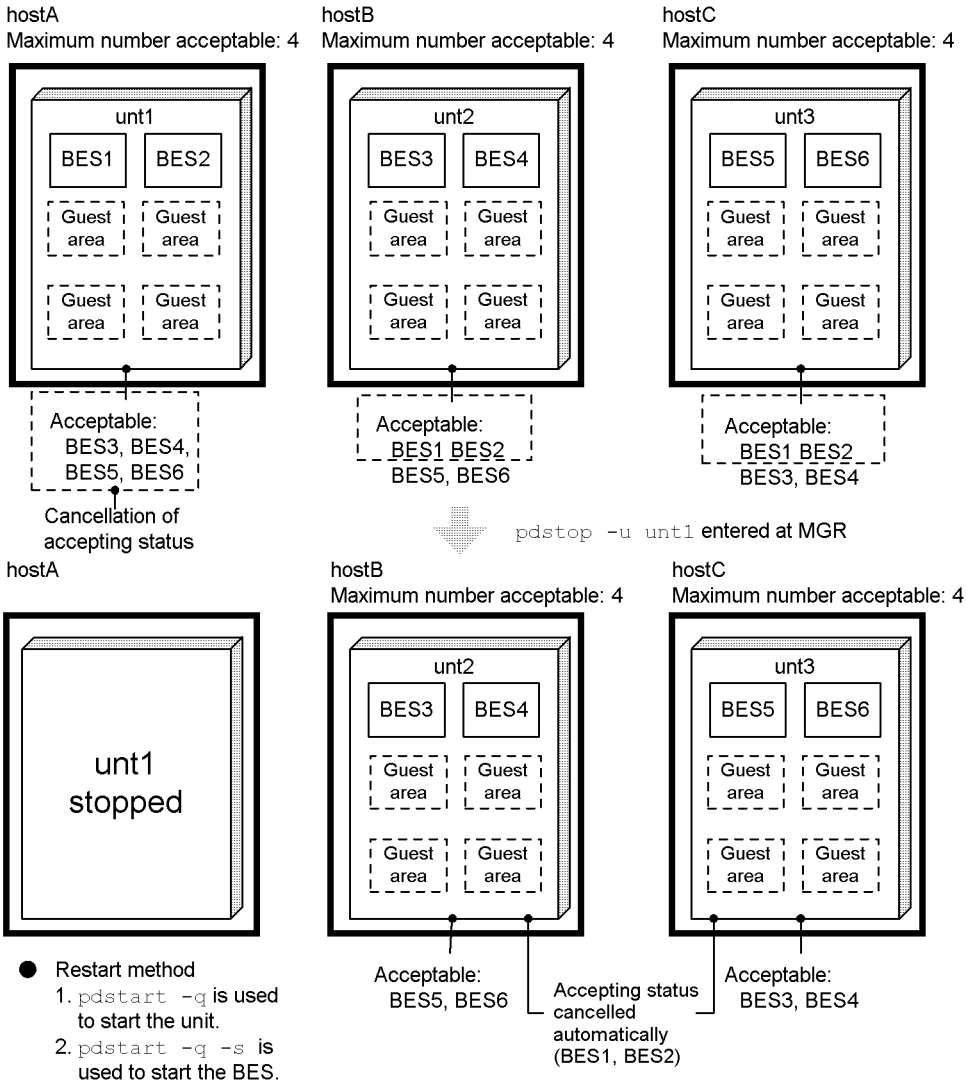
Table 25-39: Processing that occurs for the various back-end servers during unit termination when the standby-less system switchover (effects distributed) facility is used

Running location	Back-end server state	Processing
Unit being stopped	Running	Stops.
	Accepting status	Cancels the accepting status.
Other unit	Running	No change.
	Accepting status	Cancels the accepting status automatically.

Example 1: Stopping a unit during normal operation

Figure 25-63 shows an example of stopping a unit during normal operation.

Figure 25-63: Example of stopping a unit during normal operation



When a unit that has not accepted any guest BES is stopped, the following servers are also stopped:

- Host BESs in the unit

In addition, the accepting status is cancelled for the following servers:

- Guest BESs in the unit that are in accepting status (cancellation of accepting

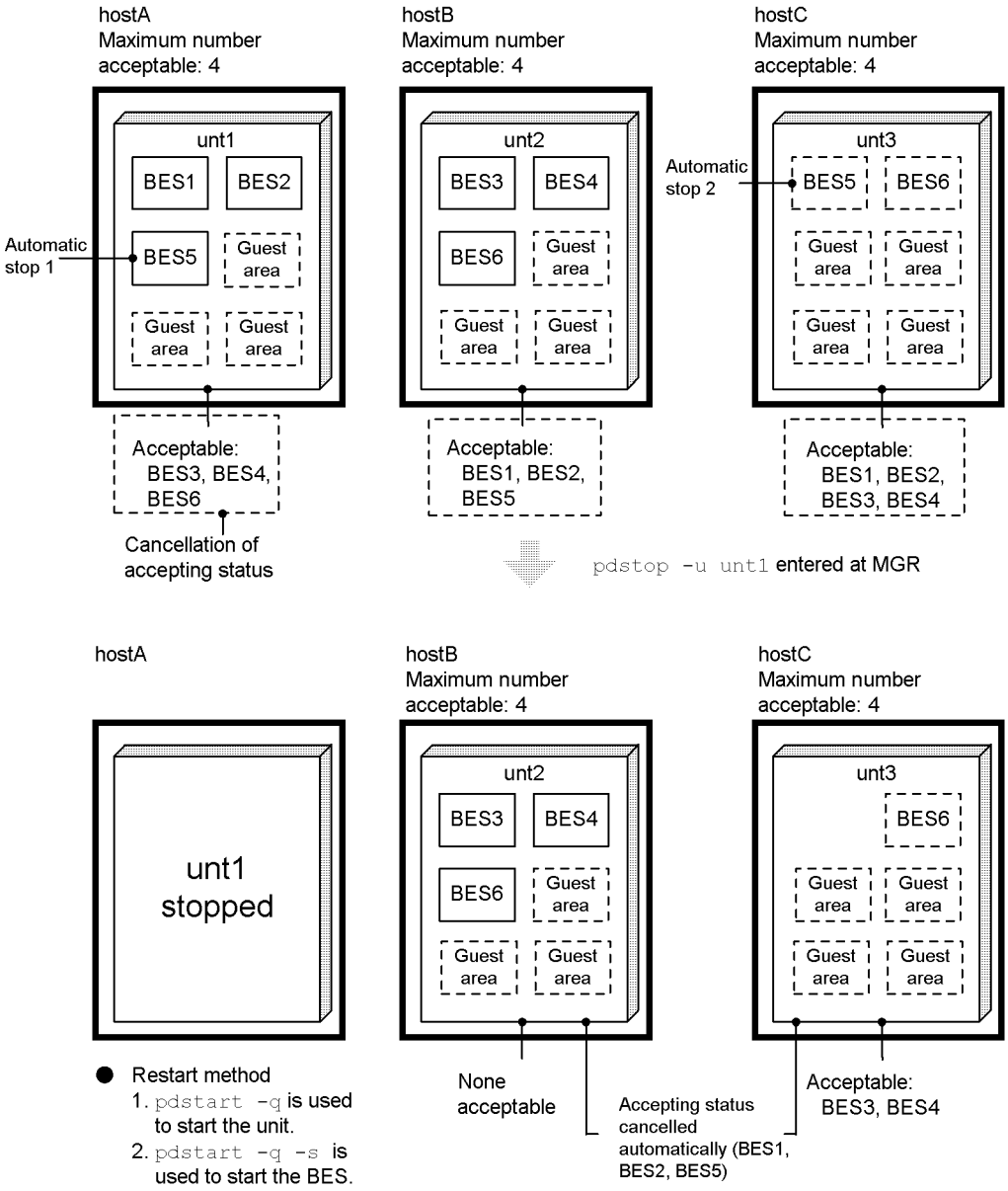
status)

- Guest BESs in another unit that corresponds to the host BES in the unit being stopped (automatic cancellation of accepting status)

Example 2: Stopping a unit that has accepted a guest BES

Figure 25-64 shows an example of stopping a unit that has accepted a guest BES.

Figure 25-64: Example of stopping a unit that has accepted a guest BES



When a unit that has accepted a guest BES is stopped, the following servers are also stopped:

- Host BES in the unit
- Guest BES that is running in the unit (Automatic stop 1)
- Host BES in another unit that corresponds to the guest BES that is running in the unit that is being stopped (Automatic stop 2)

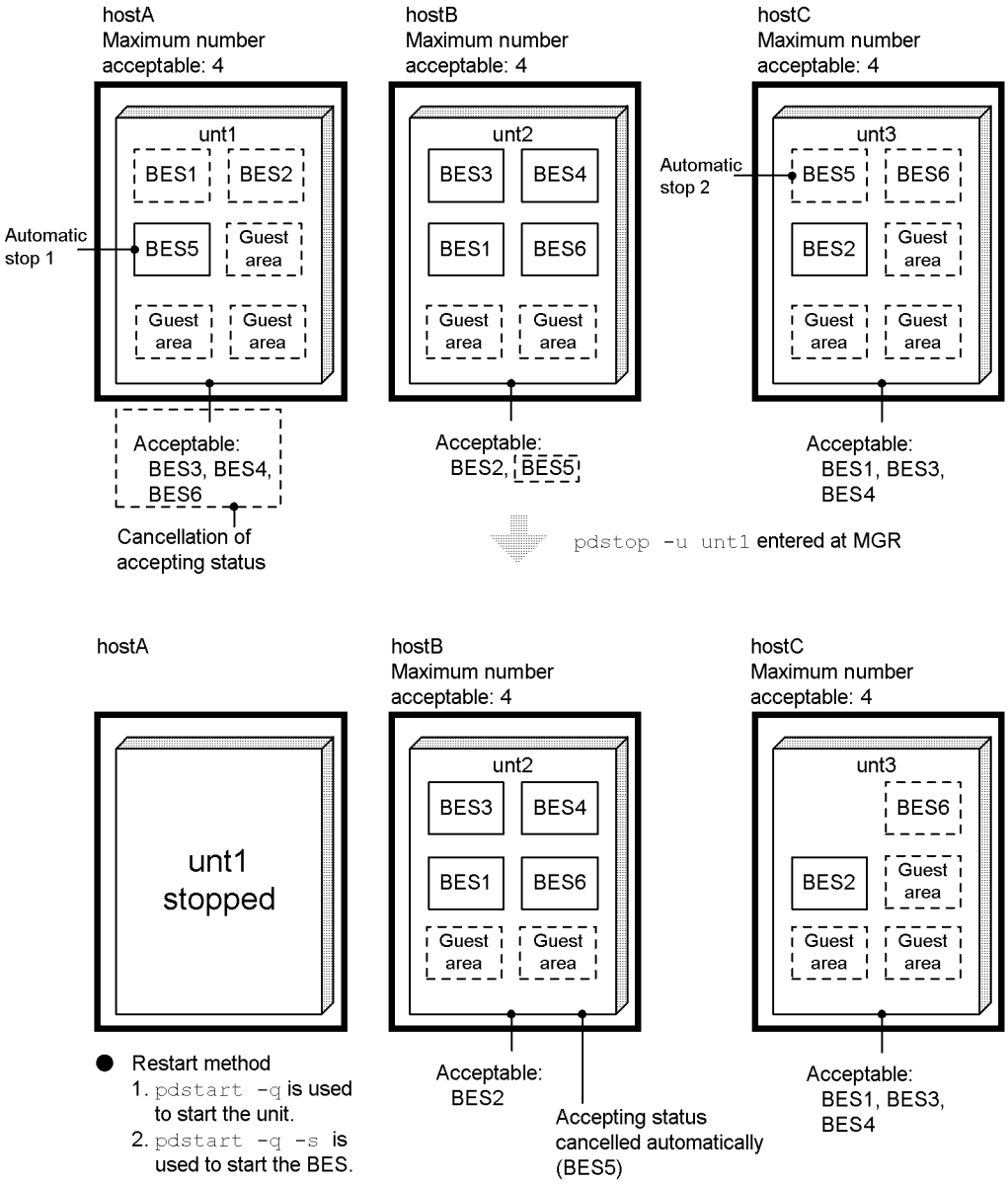
In addition, the accepting status is cancelled for the following servers:

- Guest BESs in the unit that are in accepting status (cancellation of the accepting status)
- Guest BES in another unit that corresponds to the host BES in the unit being stopped (automatic cancellation of accepting status)
- Guest BES in another unit that corresponds to the guest BES that is running in the unit being stopped (automatic cancellation of accepting status)

Example 3: Stopping a unit that has only a guest BES

Figure 25-65 shows an example of stopping a unit that has only a guest BES.

Figure 25-65: Example of stopping a unit that has only a guest BES



When a unit in which only a guest BES is running is stopped, the following servers are also stopped:

- Host BES in the unit

- Guest BES in the unit that is running (Automatic stop 1)
- Host BES in another unit that corresponds to the guest BES that is running in the unit that is being stopped (Automatic stop 2)

In addition, the accepting status is cancelled for the following servers:

- Guest BESs in the unit that are in accepting status (cancellation of accepting status)
- Guest BES in another unit that corresponds to the guest BES that is running in the unit being stopped (automatic cancellation of accepting status)

(c) Stopping a server

Table 25-40 shows how a server is stopped when the standby-less system switchover (effects distributed) facility is used.

Table 25-40: Stopping a server when the standby-less system switchover (effects distributed) facility is used

Input location	Command	Option				Operation
		-z	-u	-s	-f	
Unit where system manager is defined	`pdstop`	No	No	Yes	No	Stops the target server[2]
					Yes	Terminates forcibly the target servers in all active units in the HA group[1].
			Yes	Yes	No	Stops the target server[2, 3].
					Yes	Stops the target server forcibly[2].
Target unit	`pdstop`	Yes	No	Yes	No	Stops the target server (host BES) forcibly[4].

[1] Of all the active units in the HA group, servers in the running units only are stopped; the accepting status is cancelled for other units.

[2] If `pdstop -s(f)` is used to stop a running server, the accepting status is cancelled automatically for all active units in the HA group. For Hitachi HA Toolkit Extension, *ndm* and *rdm* jointly release the standby system (accepting status). For HA monitor, HiRDB and HA monitor both cancel the accepting status.

[3] Table 25-41 shows the results of server termination depending on the server status.

[4] For Hitachi HA Toolkit Extension, even when `pdstop -z -s` is used to stop the running server, the accepting status for the server is not cancelled automatically at other units in the HA group. To cancel the accepting status, enter Hitachi HA Toolkit

Extension's standby stop command (`hatesbystp`) at all units in the HA group.

Table 25-41: Server termination results depending on the server status

Server status	Start result
Waiting for the running system to start	Cancels the wait for the running system to start.
Accepting status	Cancels the accepting status.
Active	Stops the server.

Table 25-42 shows the processing for the various back-end servers during server termination when the standby-less system switchover (effects distributed) facility is used.

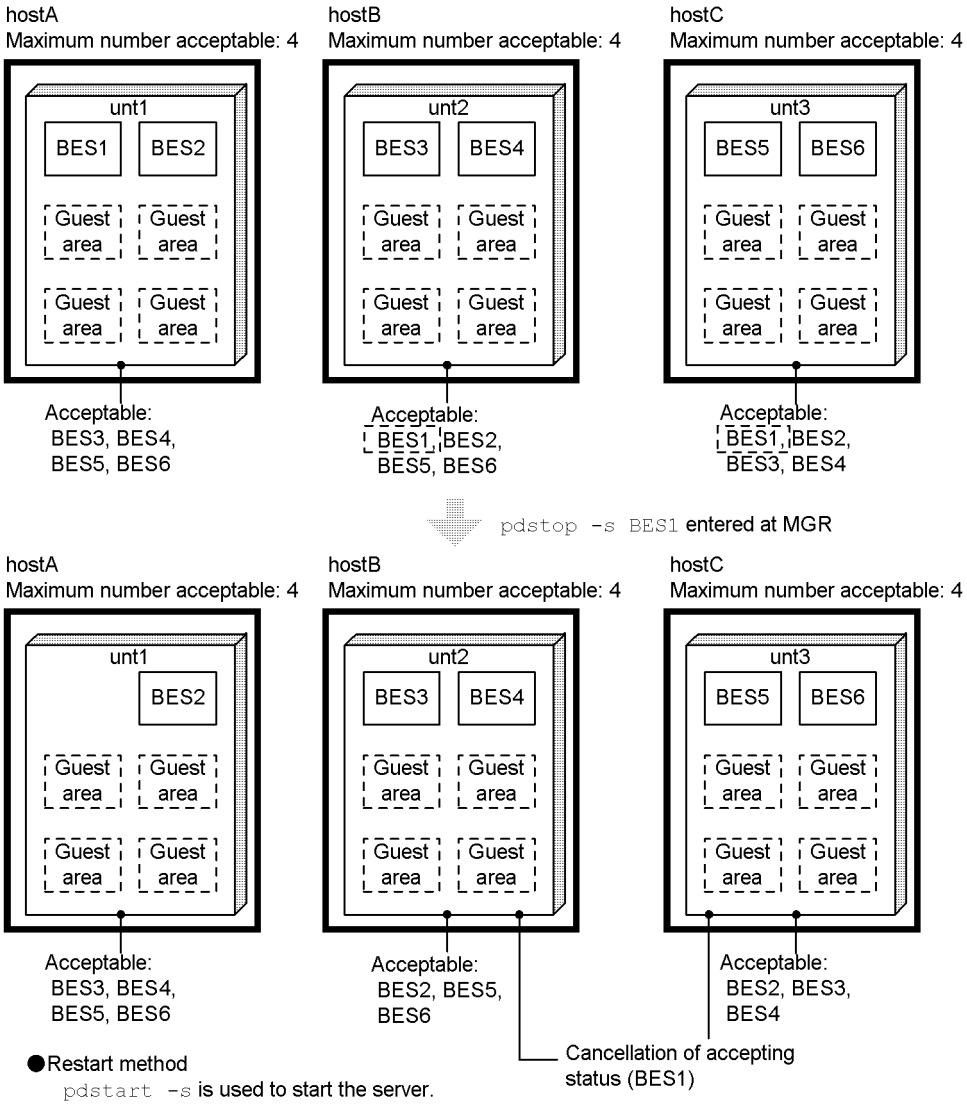
Table 25-42: Processing that occurs for the various back-end servers during server termination when the standby-less system switchover (effects distributed) facility is used

Running location	Back-end server status	Processing
Active unit	Operation target	Stops
Other units	Accepting status	Cancels the accepting status.

Example 1: Stopping a host BES

Figure 25-66 shows an example of stopping a host BES.

Figure 25-66: Example of stopping a host BES



When a host BES is stopped, the following server is also stopped:

- Host BES

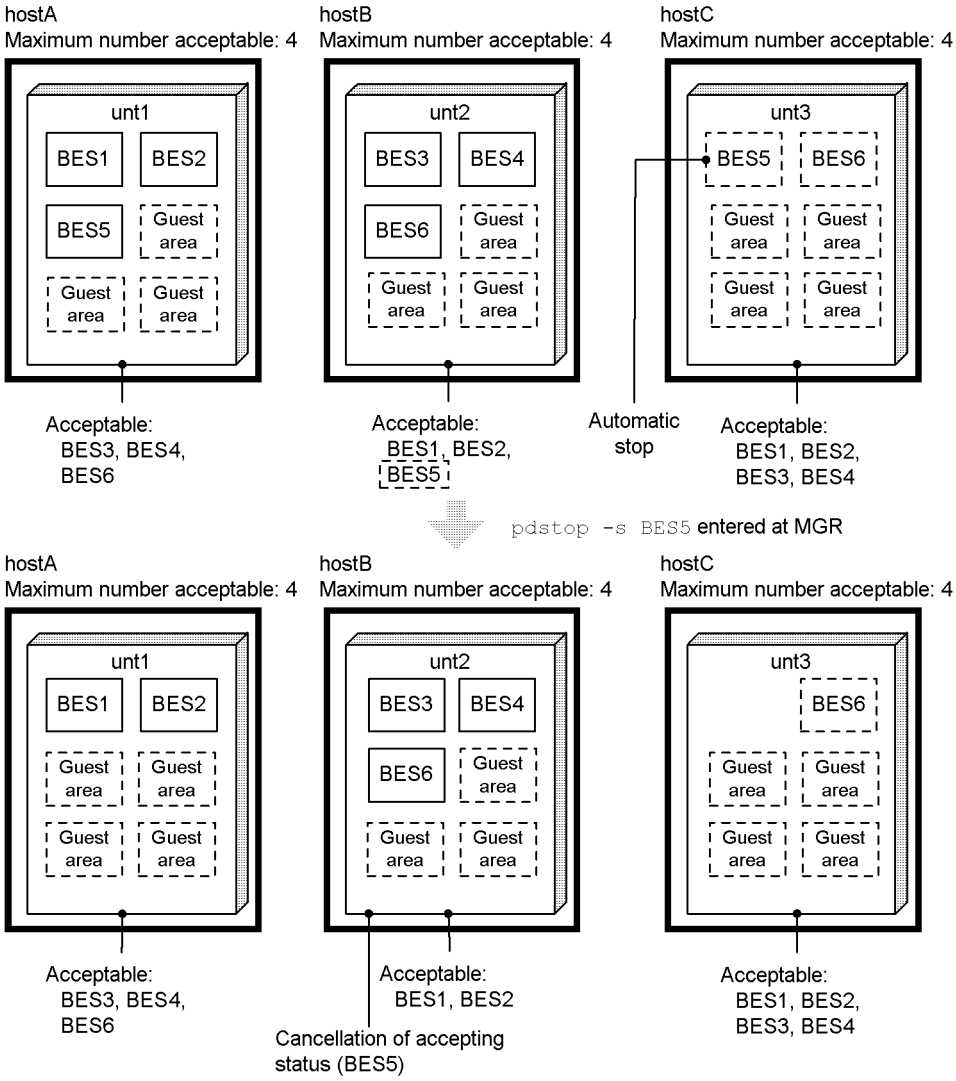
In addition, the accepting status is cancelled for the following servers:

- Guest BES in another unit that corresponds to the host BES in the unit being stopped (cancellation of accepting status)

Example 2: Stopping a guest BES

Figure 25-67 shows an example of stopping a guest BES.

Figure 25-67: Example of stopping a guest BES



- Restart method
  `pdstart -s` is used to start the server.

When a guest BES is stopped, the following servers are also stopped:

- Running guest BES

- Host BES in another unit that corresponds to the running guest BES that is being stopped (automatic stop)

In addition, the accepting status is cancelled for the following servers:

- Guest BES in another unit that corresponds to the running guest BES that is being stopped (cancellation of accepting status)

(4) Standby-less system switchover (effects distributed) facility (terminating only the standby system)

This section explains how to stop only the standby system.

As when using the standby system switchover facility or standby-less system switchover (1:1) facility, the `monsbystp` command of HA monitor can be used to stop the standby system. When the standby-less system switchover (effects distributed) facility is used, you can also perform the operation from the unit where the system manager is defined.

Table 25-43 shows how to terminate the standby system when the standby-less system switchover (effects distributed) facility is used.

Table 25-43: Terminating the standby system when the standby-less system switchover (effects distributed) facility is used

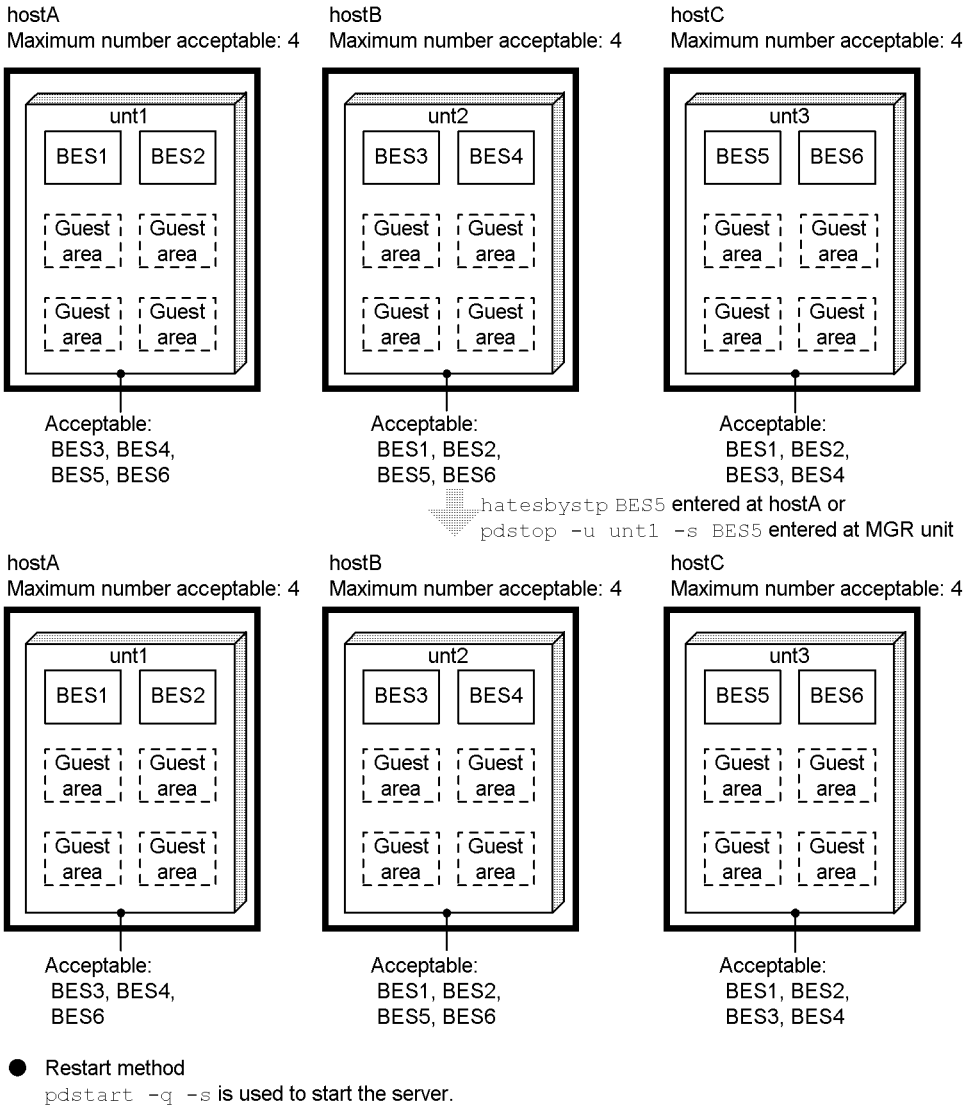
Input location	Command	Operation target	Operation
Host where the operation-target server is located	`monsbystp`*	Back-end server in accepting status	Cancels the accepting status for a guest BES
Unit where system manager is defined	`pdstop -u -s`		

* When you use Hitachi HA Toolkit Extension, use the `hatesbystp` command.

Example 1: Example of cancelling the accepting status for a guest BES

Figure 25-68 shows an example of cancelling the accepting status for a guest BES.

Figure 25-68: Example of cancelling the accepting status for a guest BES

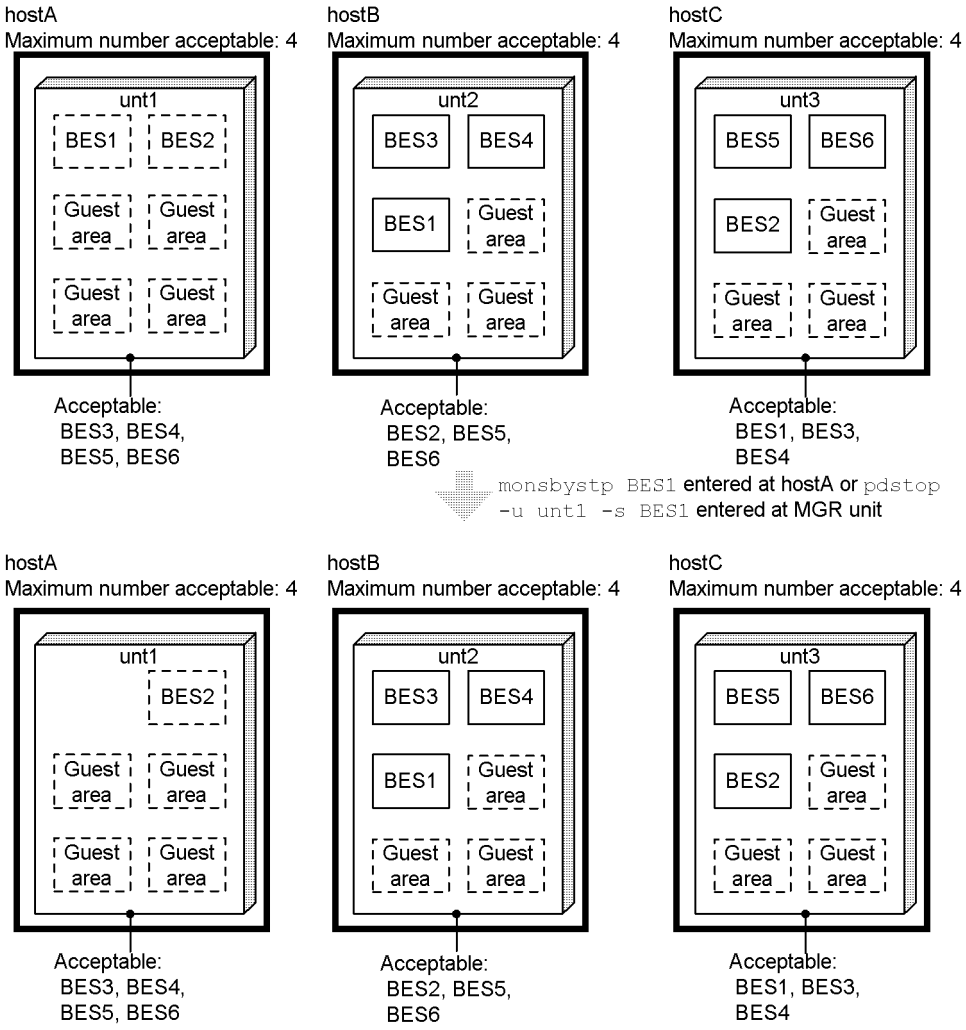


To cancel the accepting status for a guest BES, enter HA monitor's `monsbystp` command. When you use Hitachi HA Toolkit Extension, enter the `hatesbystp` command.

Example 2: Example of stopping a host BES of the standby system

Figure 25-69 shows an example of stopping a host BES of the standby system.

Figure 25-69: Example of stopping a host BES of the standby system



- Restart method
  `pdstart -q -s` is used to start the server.

To stop a host BES of the standby system, enter HA monitor's `monsbystp` command. When you use Hitachi HA Toolkit Extension, enter the `hatesbystp` command.

25.13.4 Terminating HiRDB (in the monitor mode)

Table 25-44 lists the methods of terminating HiRDB (when in the monitor mode).

Table 25-44: Terminating HiRDB (in the monitor mode)

Condition	Termination method
Terminating both the running system and the standby system	1. Use the `pdstop` command to terminate the running system HiRDB. 2. Use a cluster software command (`monend` command in the case of HA monitor) to terminate both the running system and the standby system.
Terminating only the standby system	Use a cluster software command (`monsbystp` command in the case of HA monitor) to terminate the standby system.

Notes on using MC/ServiceGuard

If an MC/ServiceGuard command is used to terminate the package without using the `pdstop` command to terminate HiRDB, HiRDB will be terminated forcibly. Be sure to restart HiRDB the next time the package is started. Note that when HiRDB has been terminated forcibly, database integrity cannot be guaranteed. Therefore, when terminating the package, be sure to use the `pdstop` command to terminate HiRDB (normal termination or planned termination) before terminating the package.

Notes on using VERITAS Cluster Server

When it is not necessary to stop the system immediately, first use the `pdstop` command to terminate HiRDB normally. Then use a VERITAS Cluster Server command to terminate the system. Each resource will disconnect from its parent resource in order and VERITAS Cluster Server will stop. The execution results of the `pdstop` command in the offline script will show an error. This is not a problem because HiRDB has already terminated.

If a VERITAS Cluster Server command is used to terminate the system without first terminating HiRDB normally, the HiRDB termination mode will be force termination because the `pdstop` command in the offline script terminates HiRDB. When HiRDB has been terminated forcibly, database integrity cannot be guaranteed. Therefore, be sure to terminate HiRDB normally before using a VERITAS Cluster Server command.

Terminating HiRDB when using ClusterPerfect

Terminate the server (HiRDB) from the Resource 1 or Resource 2 window.

25.13.5 Monitoring statuses

(1) Unit and server operating statuses

Table 25-45 shows how to check the operating status of units and servers when a system switchover facility is used.

Table 25-45: Checking the operating status of units and servers when a system switchover facility is used

Command	System switchover facility	Output information
`pdls -d svr`	Standby system switchover facility	• Host name (the primary system host name is displayed after system switchover) • Unit operating status • Server operating status
	Standby-less system switchover (1:1) facility	• Host name (the primary system host name (host name of the normal BES) is displayed after system switchover) • Unit operating status (the unit identifier of the normal BES is displayed after system switchover) • Server operating status
	Standby-less system switchover (effects distributed) facility	• Host name (the host name of the accepting unit is displayed after system switchover) • Unit operating status (the unit identifier of the accepting unit is displayed after system switchover) • Server operating status (displayed as a server belonging to the accepting unit after system switchover)

(2) Checking the system status

Table 25-46 shows how to check the system status when a system switchover facility is used.

Table 25-46: Checking the system status when a system switchover facility is used

Command	System switchover facility	Output information
`pdls -d ha`	Standby system switchover facility	• Host name and system status of the primary system (running/standby/stopped) • Host name and system status of the standby system (running/standby/stopped) • This command can check the host name and status of the standby system only when IP addresses are not inherited.[1]
	Standby-less system switchover (1:1) facility	• Host name and status of the normal BES unit (running/standby/stopped) • Host name and status of the alternate BES unit (running/standby/stopped)

Command	System switchover facility	Output information
	Standby-less system switchover (effects distributed) facility	• Name and status of the unit where the back-end server is located (running/standby/stopped/running system in wait status) and defined destination unit name Detailed status of the back-end server is displayed only when the detailed display option (`-a`) is specified.
`monshow` (only when HA monitor is used)	Standby system switchover facility	• Local system's host name and status[2] • Other system's host name and status[3]
	Standby-less system switchover (1:1) facility	
	Standby-less system switchover (effects distributed) facility	
`hateshow` (only when Hitachi HA Toolkit Extension is used)	Standby system switchover facility	• Status of the local system[4]
	Standby-less system switchover (1:1) facility	
	Standby-less system switchover (effects distributed) facility	

[1] When IP addresses are inherited, use a command of the cluster software to check the system status. For details about the information displayed, see the documentation for the cluster software.

[2] Statuses are displayed for the following categories:

Executing, on standby, starting as a running server, starting as a standby server, stopped as the running server, stopped as a standby server, waiting for restart as the running server, waiting for restart as a standby server, waiting for server system switchover, waiting for linked server system switchover

[3] Statuses are displayed for the following categories:

Executing, on standby, starting as a running server, starting as a standby server, stopped as the running server, stopped as a standby server, waiting for restart as the running server

[4] Statuses are displayed for the following categories:

Startup completed for the running server, startup completed for a standby server, running server starting, a standby server starting, the running server being stopped, a standby server being stopped, the running server waiting to be restarted, server not yet started

(3) Checking whether or not a command or utility can be executed

Execute the `pdls -d svr` command on the system manager unit of the primary system.

- When the termination status of the `pdls -d svr` command executed on the primary system is `0`:

  Because the primary system is the running system, execute the command or utility on the primary system.

- When the termination status of the `pdls -d svr` command executed on the primary system is `8` or when the `pdls -d svr` command cannot be executed (for example, remote shell cannot be executed and logon is not possible):

  The secondary system may be the running system. Execute the `pdls -d svr` command on the system manager unit of the secondary system to check whether or not the secondary system is the running system.

- When the termination status of the `pdls -d svr` command executed on the primary system or secondary system is `4`:

  Some units may be stopped or HiRDB is being started or terminated.

  If units are stopped, start them. If units are stopped because of an error, check the message that was output to the event log, correct the error, and then start the stopped units.

  If HiRDB is being started or terminated, execute the `pdls -d svr` command repeatedly at intervals of approximately 5 seconds until the termination status is no longer `4`. Execute the `pdls -d svr` command repeatedly based on the time specified in the `pd_system_complete_wait_time` operand as a guideline.

25.13.6 Handling of statistics log files

When the standby system switchover facility is used, the statistics log files consist of the two files `pdstj1` and `pdstj2`. These files are created automatically for both the primary system and the secondary system, so the HiRDB administrator must prepare a total of four files. Statistics log files cannot be shared between the primary system and the secondary system.

When the standby-less system switchover (1:1) facility or the standby-less system switchover (effects distributed) facility is used, the statistics log files consist of the two files `pdstj1` and `pdstj2`. These files are created as a set for the primary HiRDB system. Because the accepting unit's statistics log files are shared at the switching

destination, no files are created for the secondary system. The HiRDB administrator must prepare files for the regular unit and for the accepting unit.

(1) Creating unload statistics log files

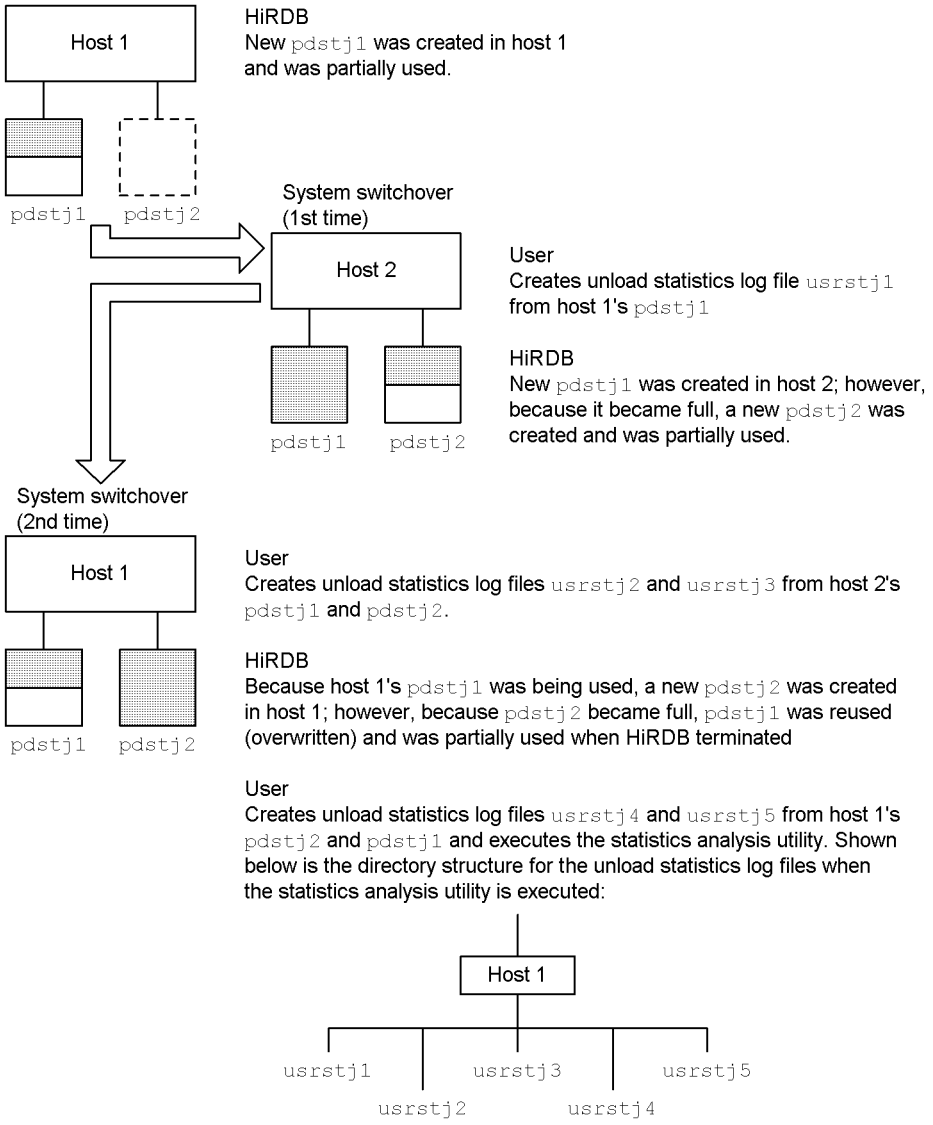
(a) When using the standby system switchover facility

Because statistics log files are distributed to each server machine when system switchover occurs, create unload statistics log files on a specific server machine. Hitachi recommends creating unload statistics log files in the following cases:

- When HiRDB starts
- When statistics log files are swapped
- When system switchover occurs

Figure 25-70 shows examples of unload statistics log files created when a system switchover facility is used.

Figure 25-70: Examples of unload statistics log files created when a system switchover facility is used (Part 1)



Hint:

Each server machine has a statistics log file with the same name. Be sure to create your unload statistics log file with a different name. Even when using the shell script provided by HiRDB (`pdstjacm`), modify the shell script so each unload statistics log file has a different name.

(b) Standby-less system switchover (1:1) facility or standby-less system switchover (effects distributed) facility

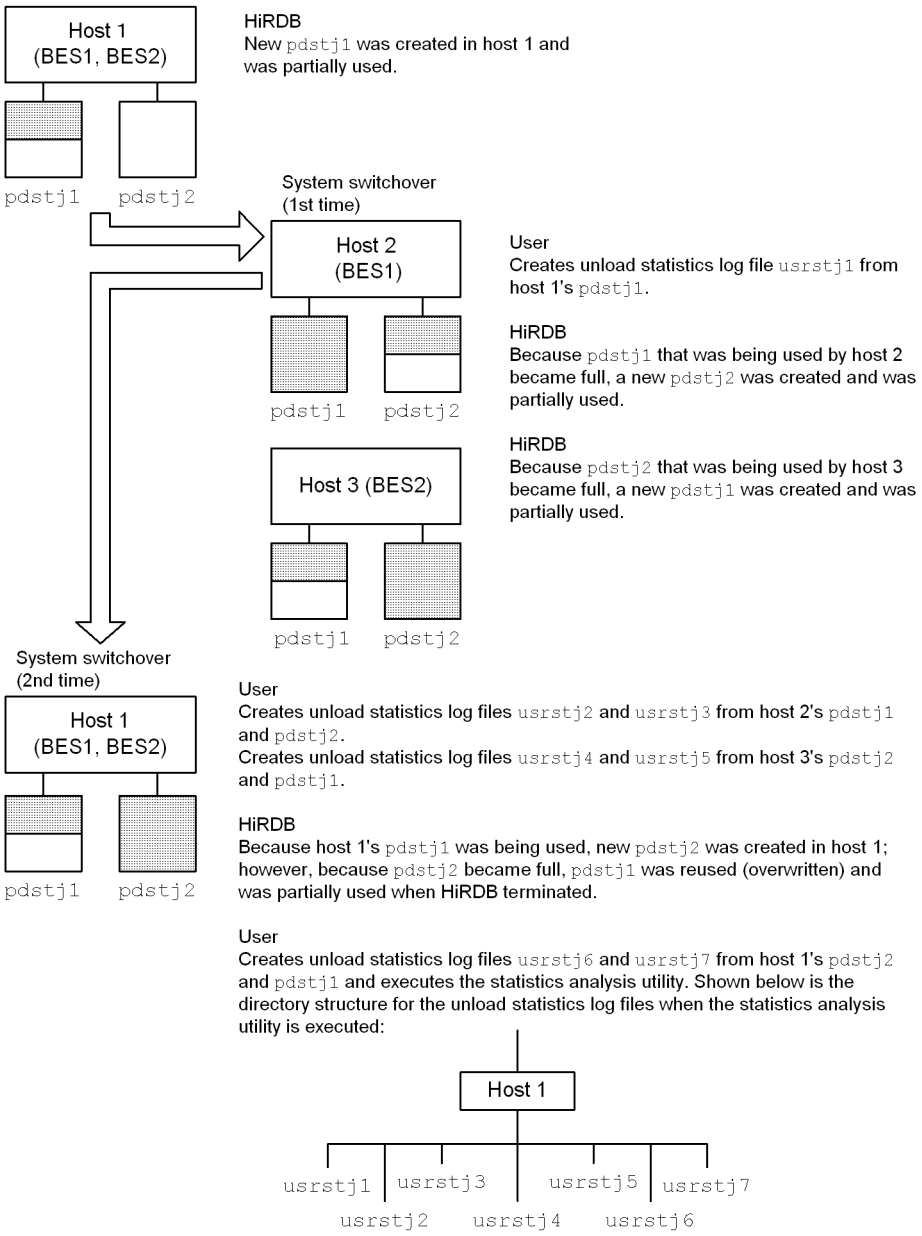
When system switchover occurs, the statistics log files to be used by the switching-destination host are the files being used by the accepting unit at the switching destination. Because statistics log output destination files are distributed to each host, unload statistics log files must be created on a specific server machine.

You should create unload statistics log files at the following times:

- When statistics log files are swapped
- When system switchover occurs

Figure 25-71 shows examples of unload statistics log files created when a system switchover facility is used.

Figure 25-71: Examples of unload statistics log files created when a system switchover facility is used (Part 2)



Hint:

Because statistics log files have identical names on all server machines, do not use the same names when you create the unload statistics log files. Also, when using a shell script (`pdstjacm`) provided by HiRDB, change the shell script so that the same name is not used.

When system switchover occurs, statistics log files are handled by the switching-destination host.

(2) Process for collecting statistical information after a system switchover

(a) Standby system switchover facility

After system switchover occurs, the following operands specify whether or not statistical information is to be collected by the HiRDB on the switchover destination system:

- `pd_statistics`
- `pdstbegin`

If `Y` is specified in the `pd_statistics` operand or the `pdstbegin` operand is specified, statistical information is collected immediately after system switchover occurs.

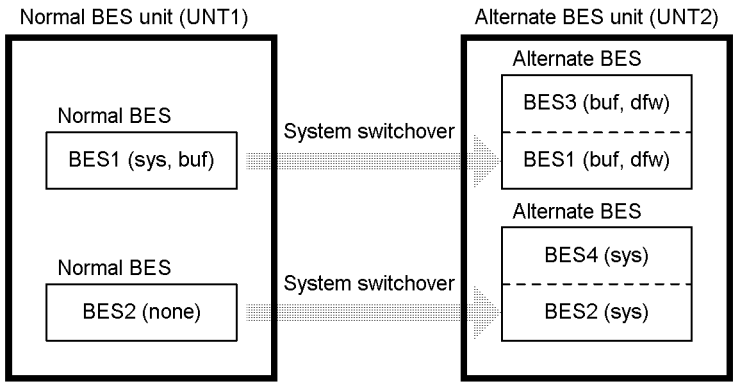
If system switchover occurs when the `pdstbegin` command is being used to start collecting statistical information, HiRDB on the switchover destination system cannot collect statistical information. To collect statistical information in this situation, you must use HiRDB on the switchover destination system to execute the `pdstbegin` command.

Also, HiRDB on the switchover destination system determines whether the `pdstj1` or the `pdstj2` statistics log file will be used. The procedure for determining which statistics log file will be used is the same as when the system switchover facility is not used.

(b) Standby-less system switchover (1:1) facility

The process for collecting statistical information when in alternating status is explained below. The collection status of statistical information on the alternate BES unit determines whether or not collection of statistical information is necessary on a normal BES unit in alternating status. The same types of statistical information are collected on the normal BES unit and on the alternate BES unit. Figure 25-72 shows the process of collecting statistical information in the alternating status.

Figure 25-72: Process of collecting statistical information in alternating status



Note: Parentheses indicate the types of statistical information acquired:
sys: Statistical information relating to system operations
buf: Statistical information relating to the global buffers
dfw: Statistical information relating to deferred write processing

Explanation

The process of collecting statistical information for `BES1` and `BES2` when in alternating status is explained below:

- `BES1`: Collects the same `buf` and `dfw` as the alternate BES unit (`BES3`).
- `BES2`: Collects the same `sys` as the alternate BES unit (`BES4`).

The process for `BES3` and `BES4` is the same.

In addition to the statistical information for BES3 and BES4, the statistical information for `BES1` and `BES2` is output to the statistics log file in the alternate BES unit (`UNT2`).

Remarks

The collection status of statistical information for a normal BES unit and an alternate BES unit in alternating status and the types of statistical information collected are the same. Figure 25-73 is used as an example in the following explanations.

- When the `pdstend` command stops collection of statistical information for `BES3`, the statistical information for `BES1` can no longer be collected. Similarly, when the `pdstend` command stops collection of statistical information for `BES1`, the statistical information for `BES3` can no longer be collected.
- When the `pdstbegin` or `pdstend` command changes the types of statistical

information collected for BES3, the types of statistical information for BES1 are also changed. Similarly, when the pdstbegin or pdstend command changes the types of statistical information collected for BES3, the types of collected information acquired for BES3 are also changed.

When switching a system back

When switching a system back (returning from alternating status to normal status), the operands listed below specify whether or not statistical information of the normal BES unit is to be collected and the types of statistical information to collect:

- pd_statistics
- pdstbegin

Therefore, you must execute the pdstbegin command again when changing whether or not to acquire statistical information or when changing the types of information to be collected.

(c) Standby-less system switchover (effects distributed) facility

When system switchover occurs, the statistics log collection status that existed immediately before system switchover is inherited. That is, if statistics logs were being collected before the system switchover, statistics logs will continue to be collected at the switched server after system switchover, regardless of the value specified in the pd_statistics operand of the system common definition. In this case, the statistics log files are shared with the accepting unit at the switching destination. If statistics logs were not being collected immediately before system switchover, no statistics logs will be collected after system switchover; you can begin collecting statistics logs by entering the pdstbegin command after switchover.

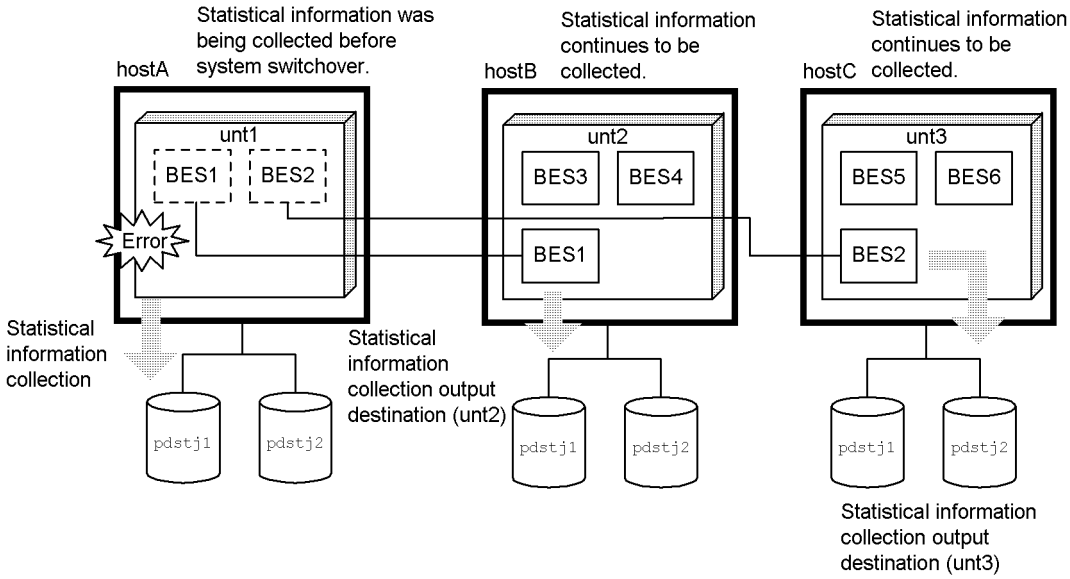
Table 25-47 shows when statistics logs are collected when the standby-less system switchover (effects distributed) facility is used.

Table 25-47: Statistics log collection when the standby-less system switchover (effects distributed) facility is used

Unit type	Being collected	Accepting unit
Regular unit	Was being collected.	Collects.
	Was not being collected.	Does not collect.

Figure 25-73 shows an example of statistics log collection after system switchover when the standby-less system switchover (effects distributed) facility is used.

Figure 25-73: Example of statistics log collection after system switchover when the standby-less system switchover (effects distributed) facility is used



(3) Executing the statistics analysis utility

(a) Standby system switchover facility

The statistics analysis utility is executed using the created unload statistics log files as the input information. If system switchover occurs due to an error, the statistics log information immediately prior to system switchover is not acquired correctly in the file. For this reason, the execution results of the statistics analysis utility may not be accurate when used for tuning, etc.

(b) Standby-less system switchover (1:1) facility or standby-less system switchover (effects distributed) facility

The statistics analysis utility is executed using the unload statistics log files created at the regular unit and the accepting unit as the input information. To manually copy the files that existed prior to system switchover you must use, for example, an OS command. The statistics information on the server that has been switched is processed as information on a server belonging to the accepting unit.

If system switchover occurs because of an error, the statistics log information immediately prior to system switchover is not acquired correctly in the file. For this reason, the execution results of the statistics analysis utility may not be accurate when used for tuning, etc.

25.13.7 Notes on operations

(1) Limitations on execution of operation commands

- To execute an operation command during HiRDB termination processing, the HiRDBs on both systems must be engaged in termination processing (however, this requirement does not apply to the `pdstart` command).
- While HiRDB is running, no operation commands can be executed by the HiRDB on the standby system.

(2) Limitations on execution of operation commands when using the standby-less system switchover (effects distributed) facility

When the situation described below occurs in an environment in which a global buffer for `OTHER` is defined, you must not attempt to terminate the unit alone by executing the `pdstop -u` command from the unit where the system manager is defined. First, either execute the `pdstop` command from the unit where the system manager is defined to terminate HiRDB or use the system reconfiguration command (`pdchgconf` command) to allocate a global buffer to the RDAREA to be added, and then terminate the unit. For details about global buffers for `OTHER`, see *25.5.8 Definition of global buffers (standby-less system switchover (effects distributed) facility only)*.

- In a unit to which the standby-less system switchover (effects distributed) facility is applicable, the `globalbuffer` operand was omitted from the `create rdarea` statement of the database structure modification utility (`pdmod`) and an RDAREA with a page size larger than the size of the global buffer for `OTHER` has been added.

(3) Changing a HiRDB system definition or the HiRDB configuration

If you change a HiRDB system definition or the HiRDB configuration, do not use a cluster software command to terminate HiRDB. Instead, use the `pdstop` command to terminate HiRDB only normally, then change the HiRDB system definition or reconfigure HiRDB. When you are done, you can use the `pdstart` command to start HiRDB only.

(4) When HiRDB files created on a shared disk cannot be accessed

If HiRDB is stopped by the cluster software, it may not be possible to manipulate HiRDB files on the shared disk from either system. In this case, you must use an OS command to activate the shared disk.

(5) Notes on executing the pdsetup command (applicable to the server mode only)

You must be careful about executing the `pdsetup -d` command while terminating HiRDB forcibly or abnormally. Do not enter `Y` at the command's prompt; if you do enter `Y`, it may become impossible to start HiRDB subsequently.

(6) Note on using Hitachi HA Toolkit Extension

When using Hitachi HA Toolkit Extension, the standby system unit will not wait for the running system unit to start. If a command is issued to the running system unit before the standby system unit has started, an error may occur and the standby system unit may abort (`Phi1012`). Therefore, wait for startup of the running system unit to finish before starting the standby system unit. If the standby system unit is started without waiting for startup of the running system to finish, the standby system unit may abort (`Phi1012`).

(7) Notes on using HA monitor

The following points about using HA monitor should be noted:

- During startup of a server,[1] if all systems, including the primary system (system for which `online` is specified in the `init` operand of HA monitor's `servers` definition), wait for the running system server to start,[2] start the primary system[3] as the running system by executing HA monitor's `monact` command.
- If a server is being activated in a unit to which the standby-less system switchover (effects distributed) facility is applicable, do not terminate that unit forcibly. If it is terminated forcibly, the next time a server is started on the primary system, this server may wait for the running system server to start.[2] In this case, start the primary system[3] as the running system by executing HA monitor's `monact` command.

[1] For a unit to which the standby-less system switchover (1:1) facility is applicable, replace *server* with *unit* when you read this sentence.

[2] Status in which `*SBY*` is displayed when the HA monitor's `monshow` command is executed.

[3] If the secondary system (system for which `standby` is specified in the `init` operand of HA monitor's `servers` definition) server is started by the `monact` command as the running system, the messages listed below may be output repeatedly on the primary system until startup of the running system is completed. These messages cease to be output once startup of the running system is completed.

- `KFPS05608-I`
- `KAMN305-E`
- `KAMN222-I`

The `KAMN305-E` and `KAMN222-I` messages are output by HA monitor.

25.13.8 Notes on using the standby-less system switchover facility

(1) Operations requiring a restart of the alternate BES unit or alternate portion (standby-less system switchover (1:1) facility only)

When any of the operations listed below occurs, you must restart the alternate portion. If you do not restart the alternate portion, it will terminate abnormally when system switchover occurs. You must also restart the alternate BES unit after operation 1 or 3 below occurs.

1. The database structure modification utility (`pdmod` command) was executed.
2. A process is performed that updates the master directory RDAREA (for example, executing the database structure modification utility, executing a definition SQL, or executing the `pddbchg` command), and then the server (front-end server, dictionary server, or back-end server) is independently terminated normally and then started normally.
3. The `pdbuffer` operand is modified, and then the server is independently terminated normally and then started up normally.
4. The `pd_max_rdarea_no`, `pd_max_file_no`, `pd_inner_replica_control`, or `pd_index_assurance_no` operand is modified, and then the server is independently terminated normally and then started normally.
5. A foreign server or user mapping is defined, and then the server is independently terminated normally and then started normally.

If the alternate portion terminates abnormally, use the `pdstart -q -c` command to start it.

When performing the operations listed above, terminate the normal BES unit then restart it. If you do not restart the normal BES unit, it will terminate abnormally when system switchover (or system switchback) occurs. If the normal BES unit terminates abnormally, use the `pdstart -q` command or `pdstart -u` command to start the normal BES unit.

(2) RDAREA opening trigger attributes

(a) Standby-less system switchover (1:1) facility

To minimize the time required for system switchover, the standby-less system switchover (1:1) facility opens only the RDAREAs needed for full recovery when system switchover occurs; it does not open other RDAREAs. The RDAREA opening trigger attributes in the normal BES unit are listed below:

- When system switchover occurs, the system can be reactivated only after all processes in the alternate portion have terminated; therefore, the RDAREA opening trigger attribute of the alternate portion is the `SCHEDULE` attribute.

- After recovering from an error and switching the system back to the normal BES unit, the RDAREA opening trigger attribute of the INITIAL or DEFER attribute on the normal BES unit is the DEFER attribute. RDAREAs with the SCHEDULE attribute remain unchanged.

For details about RDAREA opening trigger attributes, see *15.5 Modifying an RDAREA opening trigger attribute (RDAREA modification)*.

(b) Standby-less system switchover (effects distributed) facility

To minimize the time required for system switchover, the standby-less system switchover (effects distributed) facility opens only the RDAREAs needed for full recovery when system switchover occurs; it does not open other RDAREAs. The RDAREA opening trigger attributes are listed below:

- Even if N is specified in the pd_rdarea_open_attribute_use operand, Y is assumed.
- When system switchover occurs, even if the RDAREA opening trigger attribute is INITIAL, DEFER is assumed.

For the RDAREA opening trigger attributes when you use the standby-less system switchover (effects distributed) facility, see the explanations for the pd_rdarea_open_attribute_use and pd_rdarea_open_attribute operands in the manual *HiRDB Version 8 System Definition*.

25.14 Planned system switchover

This section explains the procedures for performing planned system switchover.

(1) Standby system switchover facility

To perform a planned system switchover:

Server mode

1. Use the `pdstop` command to terminate the running system HiRDB (unit or server). Perform this step only when a product other than HA monitor is used in the cluster software.

   Also, if you specify the `pdstop` command in the `termcommand` operand of Hitachi HA Toolkit Extension, you do not have to execute the `pdstop` command at this time, because Hitachi HA Toolkit Extension executes it as an extension of the command in step 2.

2. Use a cluster software command (`monswap` command in the case of HA monitor) to perform a planned system switchover. For details about planned system switchovers, see the documentation for the cluster software product.

3. The HiRDB (or unit) that became the standby system must be placed in waiting status. Execute the `pdstart` command (`pdstart -q` command in the case of a HiRDB/Parallel Server) on the standby system HiRDB.

Monitor mode

1. Use the `pdstop` command to terminate the running system HiRDB (or unit).

2. Use a cluster software command (`monswap` command in the case of HA monitor) to perform a planned system switchover. For details about planned system switchovers, see the documentation for the cluster software product.

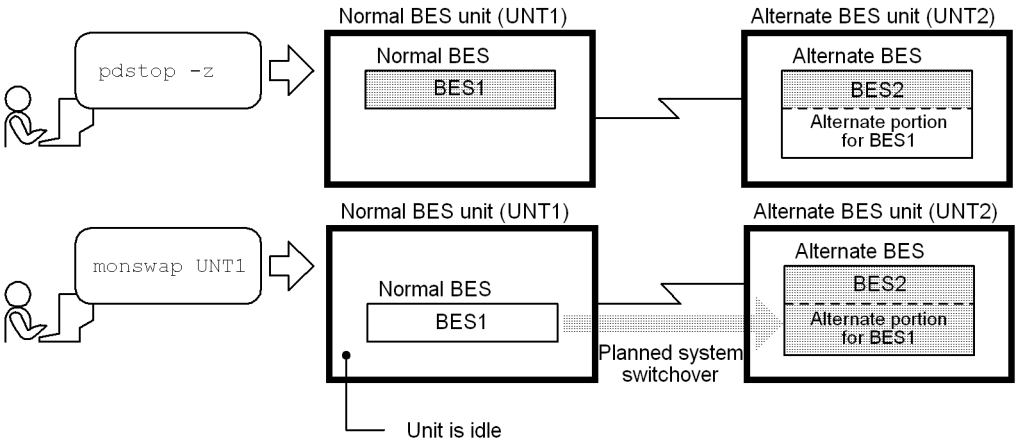
- When the cluster software product used is HA monitor and you specify HiRDB termination in the shell you start with the `termcommand` operand in the server definition statement, you can perform a planned system switchover by simply executing the `monswap` command. For examples of shell creation, see *25.6.2(4) termcommand operand (applicable to the monitor mode only)*.

- To perform a system switchover manually when the cluster software product used is ClusterPerfect, stop the running system then start the standby system. When system switchover occurs, the color of ClusterPerfect's ClusterPerfect system information window changes.

(2) Standby-less system switchover (1:1) facility

When performing a planned system switchover to the alternate BES unit, the alternate portion must be in waiting status. When switching the system back to the normal BES unit, the normal BES unit must be in waiting status. The procedure for performing a planned system switchover is explained below:

Procedure: Switching the system to the alternate BES unit



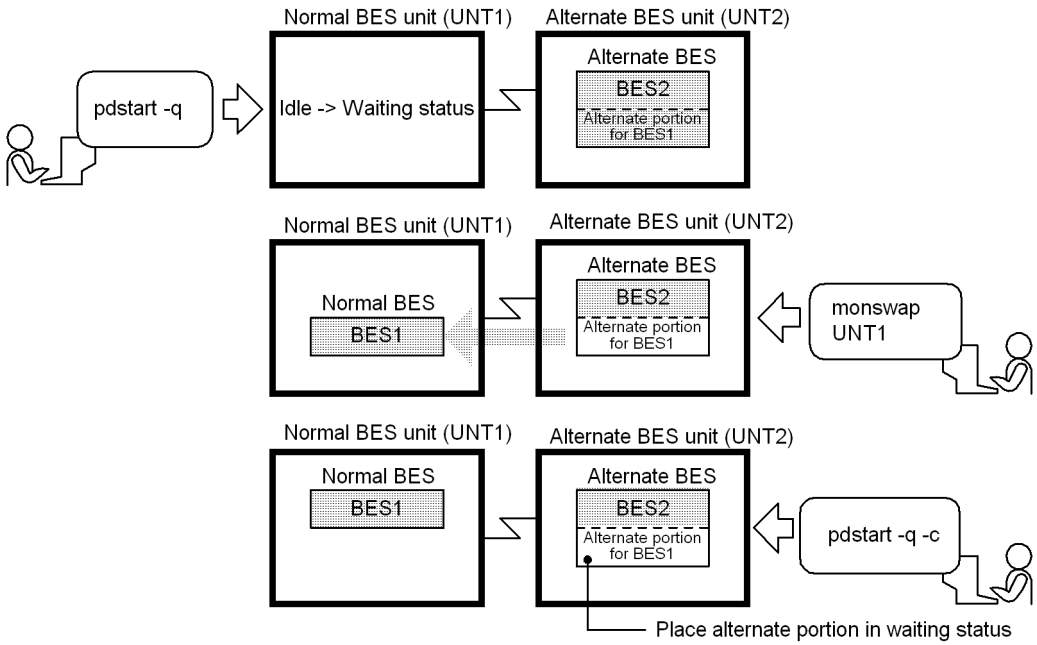
1. Use the `pdstop -z` command to terminate the normal BES unit. Perform this step only when a product other than HA monitor is used in the cluster software.

   Also, when specifying the `pdstop` command in the `termcommand` operand of Hitachi HA Toolkit Extension, you do not have to execute the `pdstop` command at this point, because Hitachi HA Toolkit Extension executes the `pdstop` command as an extension of the command in step 2.

2. Use a cluster software command (`monswap` command in the case of HA monitor) to perform a planned system switchover. For details about planned system switchovers, see the documentation for the cluster software product.

When you use the above procedure to perform a planned system switchover, the alternate BES unit performs normal BES unit processing. The normal BES unit becomes inactive. To switch the system back to the normal BES unit, use the following procedure to perform a planned system switchover again:

Procedure: Switching system back to the normal BES unit



1. Use the `pdstart -q` command to place the normal BES unit in waiting status.
2. Use a cluster software command (the `monswap` command in the case of HA Monitor) to perform a planned system switchover. The waiting status of the alternate portion is released.
3. Use the `pdstart -q` command to place the alternate portion in waiting status.

(3) Standby-less system switchover (effects distributed) facility

For a planned system switchover, execute HA monitor's `monswap` command.

When this command executes, the back-end server at the switching-source unit is placed automatically on standby status.

If you are using Hitachi HA Toolkit Extension, enter the cluster software's planned system switchover command instead of the `monswap` command.

Table 25-48 shows the planned system switchover operation when you use the standby-less system switchover (effects distributed) facility.

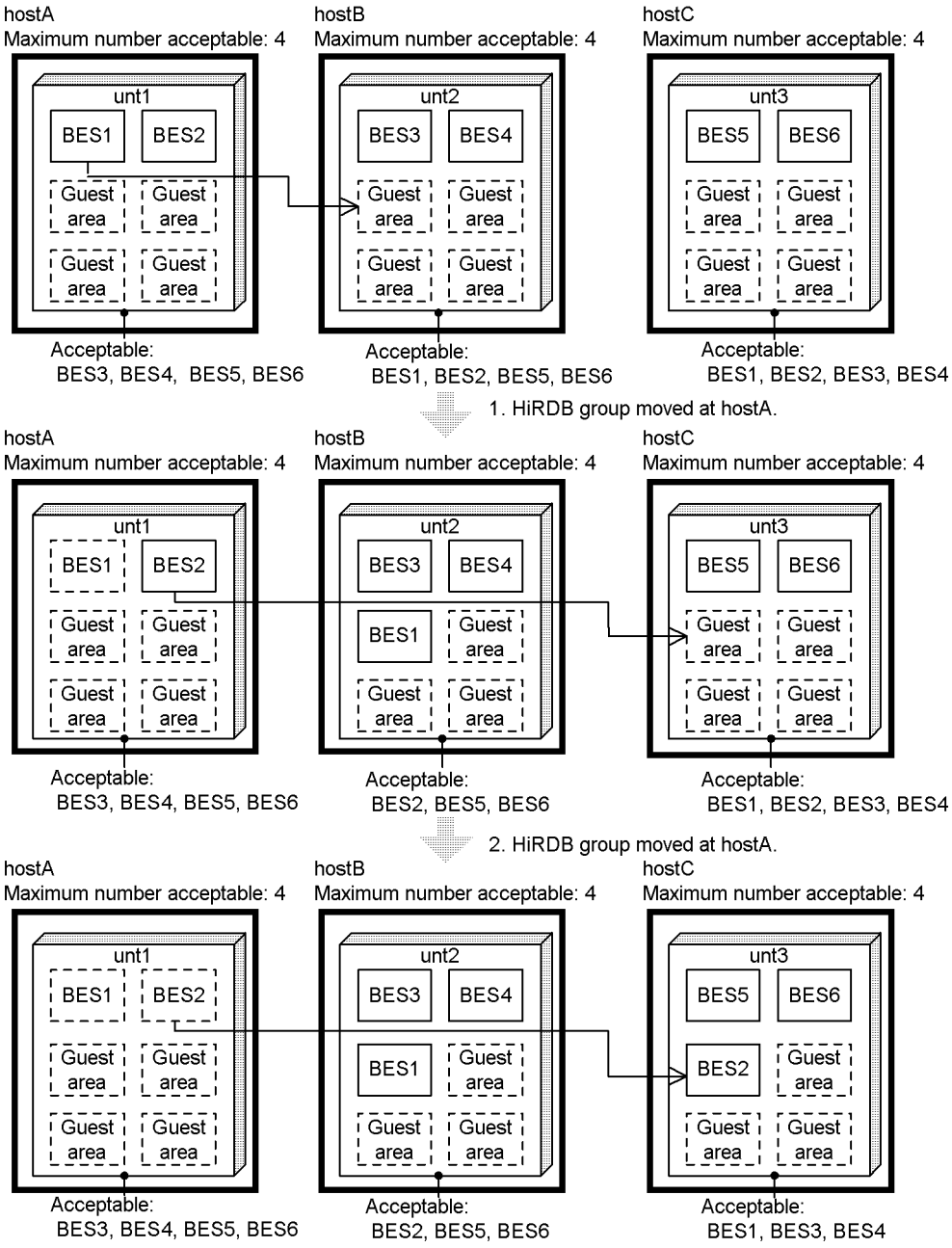
Table 25-48: Planned system switchover operation when the standby-less system switchover (effects distributed) facility is used

Back-end server type	Operation	Move destination
Host BES	Moves to the switching destination with the highest priority; if there is no free space in the guest area, moves to the switching destination with the next highest priority.	Guest BES
Guest BES		Host BES

(a) Planned system switchover for a host BES

Figure 25-74 shows an example of planned system switchover for a host BES. In this example, a server machine (`hostA`) must be stopped for hardware maintenance of the host.

Figure 25-74: Example of planned system switchover for a host BES



In this example, planned system switchover is executed for the current BES on `unt1`.

The procedure is as follows:

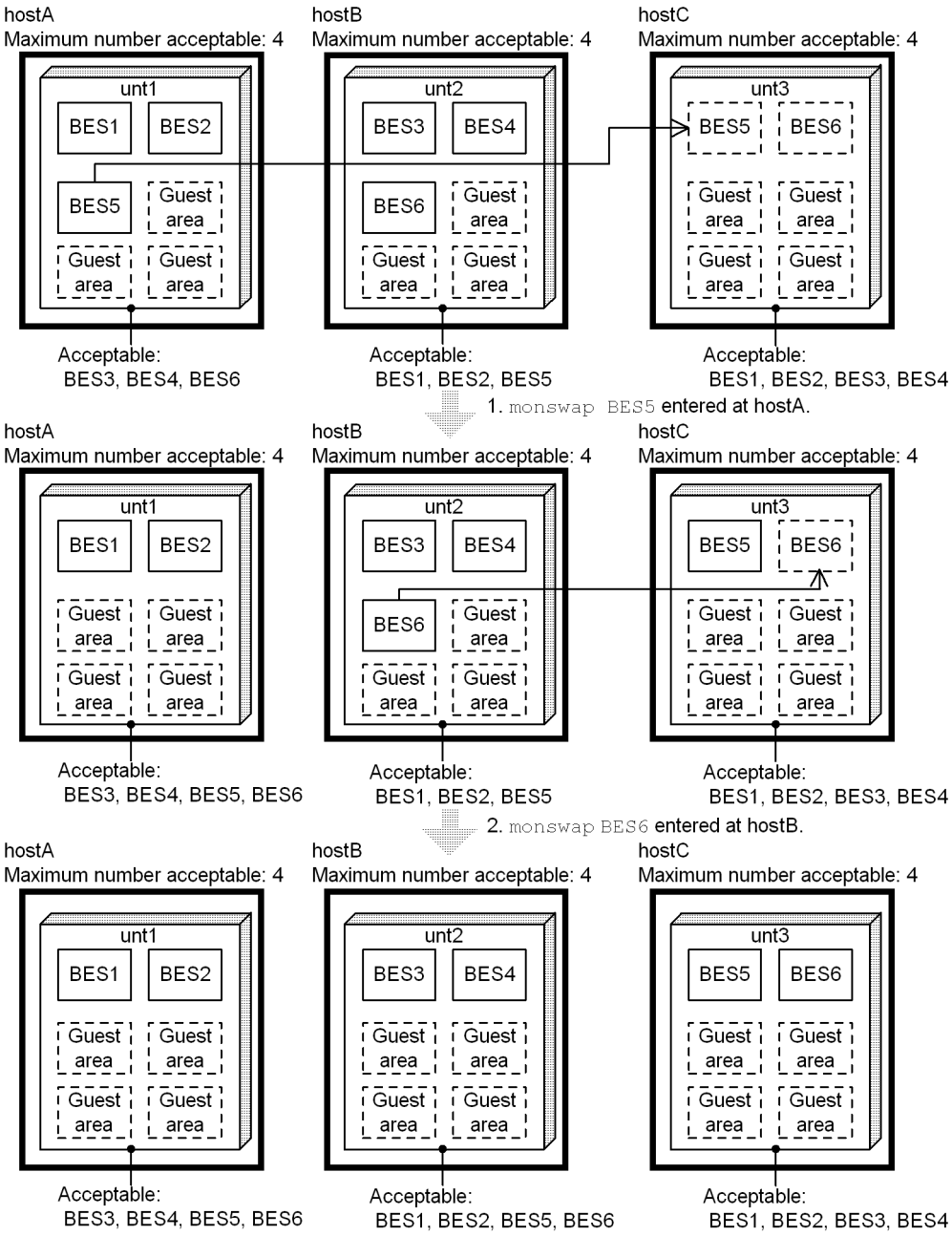
1. In `hostA`, the `monswap` command is entered for `BES1` and `BES1` is switched to `hostB`.
2. In `hostA`, the `monswap` command is entered for `BES2` and `BES2` is switched to `hostC`.

If you are using Hitachi HA Toolkit Extension, enter the cluster software's planned system switchover command.

(b) Planned system switchover for guest BESs (system reactivation)

Figure 25-75 shows an example of planned system switchover for guest BESs (system reactivation). In this example, a unit recovers from an error.

Figure 25-75: Example of planned system switchover for guest BESs (system reactivation)



In this example, BES5 and BES6, which had moved from unt3, are reactivated through planned system switchover. The procedure is as follows:

1. In hostA, the monswap command is entered for BES5 and BES5 is switched to hostC.
2. In hostB, the monswap command is entered for BES6 and BES6 is switched to hostC.

If you are using Hitachi HA Toolkit Extension, enter the cluster software's planned system switchover command.

25.15 Grouped system switchover

This section describes the methods of performing grouped system switchover when using HA monitor in the cluster software. To perform grouped system switchover while using cluster software other than HA monitor, see the documentation for the cluster software product.

(1) Server mode

The following explanation assumes use of grouped system switchover with HiRDB and OpenTP1.

To perform grouped system switchover, you must use the `group` operand of HA monitor to define grouped system switchover. The following explanation assumes that `exchange` is specified in the `group` operand. For details about defining grouped system switchover. Table 25-49 lists the system processing when performing grouped system switchover (in the server mode).

Table 25-49: System processing when performing grouped system switchover (in the server mode)

Error description		System processing
An error occurred that prevents hardware from restarting, or an error occurred when restarting the operating system or HA monitor (a system failure occurred).		Performs system switchover. Also switches over OpenTP1 and HiRDB.
An error occurred in OpenTP1 and OpenTP1 terminated abnormally.	`switchtype=restart`	Restarts OpenTP1 on the running system. Performs system switchover if OpenTP1 cannot be restarted. When system switchover is performed on OpenTP1, grouped system switchover is also performed on HiRDB.
	`switchtype=switch`	Performs system switchover. When system switchover is performed on OpenTP1, grouped system switchover is also performed on HiRDB.
	`switchtype=manual`	Performs system switchover. Restarts OpenTP1 on the running system.

Error description		System processing
An error occurred in HiRDB and HiRDB terminated abnormally.	`switchtype=restart`	Restarts HiRDB on the running system. Switches systems if HiRDB cannot be restarted. When system switchover is performed on HiRDB, grouped system switchover is also performed on OpenTP1.
	`switchtype=switch`	Performs system switchover. When system switchover is performed on HiRDB, grouped system switchover is also performed on OpenTP1.
	`switchtype=manual`	Does not perform system switchover. Restarts HiRDB on the running system.

(2) Monitor mode

The following explanation assumes the use of grouped system switchover with HiRDB and OpenTP1.

To perform grouped system switchover, you must use the `group` operand and `termcommand` operand of HA monitor to define grouped system switchover. The following explanation assumes that `exchange` is specified in the `group` operand. For details about defining grouped system switchover, see the manual. Table 25-50 lists the system processing when performing grouped system switchover (in the monitor mode).

Table 25-50: System processing when performing grouped system switchover (in the monitor mode)

Error description	System processing
An error occurred that prevents hardware from restarting, or an error occurred when restarting the operating system or HA monitor (a system failure occurred).	Performs system switchover. Also performs switchover for OpenTP1 and HiRDB.
An error occurred in OpenTP1 and OpenTP1 terminated abnormally.	Performs system switchover. When system switchover is performed on OpenTP1, grouped system switchover is also performed on HiRDB.
An error occurred in HiRDB and HiRDB terminated abnormally.	Does not perform system switchover. Restarts HiRDB on the running system.

(3) Grouped system switchover with HiRDB Datareplicator

When you use HiRDB Datareplicator, you must perform grouped system switchover with both HiRDB and HiRDB Datareplicator. In such a case, use the `group` operand of HA monitor to define HiRDB and HiRDB Datareplicator in the same server group. For details about defining grouped system switchover. For details about performing system switchover when using HiRDB Datareplicator, see the manual *HiRDB*

Datareplicator Version 8 Description, User's Guide and Operator's Guide.

Conditions

- Perform grouped system switchover with HiRDB Datareplicator in the monitor mode.
- You must describe in the `termcommand` operand of HA monitor a command or shell that terminates HiRDB Datareplicator forcibly.

Automatic system switchover

When an error occurs on the running system, the system switches automatically to the standby system. No action needs to be taken by the HiRDB administrator or HiRDB Datareplicator administrator.

Planned system switchover

You can use the HA monitor's `monswap` command to perform a planned system switchover on HiRDB and HiRDB Datareplicator.

When HiRDB system is switched over

When an error occurs and the HiRDB system is switched over, terminate the running system HiRDB Datareplicator forcibly and switch the HiRDB Datareplicator system.

When an error occurs in HiRDB Datareplicator

When HiRDB Datareplicator independently terminates abnormally, system switchover is not performed because HiRDB Datareplicator is restarted on the running system. When HiRDB Datareplicator is independently terminated abnormally, grouped system switchover cannot be performed on HiRDB and HiRDB Datareplicator.

(4) Relationship to planned system switchover

If you define grouped system switchover in advance, you can use the planned system switchover functionality of HA monitor's `monswap` command to perform grouped system switchovers.

25.16 Actions to be taken by the HiRDB administrator when errors occur

(1) System processing and actions taken by the HiRDB administrator when errors occur

Table 25-51 lists the system processing and the actions to be taken by the HiRDB administrator when errors occur.

Table 25-51: System processing and the HiRDB administrator's actions in the event of an error (using the system switchover facility)

Error		System processing	HiRDB administrator's action
A server failure occurred (only HiRDB terminated abnormally).	Server mode	As specified in the `switchtype` operand[1] of HA monitor or Hitachi HA Toolkit Extension.	Check the cause of the error on the basis of the output message. Refer to *18. Error Handling Procedures*,and take appropriate action.
	Monitor mode	Does not perform system switchover. Restarts HiRDB on the system resulting in the error.	
A system failure occurred.		Performs system switchover and restarts HiRDB on the switchover destination system. However, depending on the type of system failure, system switchover may not be performed. A cluster software specification determines whether or not system switchover will be performed; for details, see the cluster software documentation.	Check the cause of the error on the basis of the output message. Refer to *18. Error Handling Procedures* or the applicable manual, and take appropriate action.
HiRDB restart failed	Server mode	As specified in the `switchtype` operand[1] of HA monitor or Hitachi HA Toolkit Extension.	
	Monitor mode	As specified in the `pd_ha_restart_failure` operand.[2]	
System switchover failed		Terminates the system.	

Note

The information inherited after system switchover is the same as when HiRDB is restarted. For details on the information that is inherited after a HiRDB restart, see *18.1.5 Information inherited during a HiRDB restart*.

[1] Processing depends on the specification of the `switchtype` operand. For details about HA monitor's `switchtype` operand, see *25.6 HA monitor preparations*; for details about the `switchtype` operand of Hitachi HA Toolkit Extension, see *25.12 Hitachi HA Toolkit Extension preparations (server mode only)*.

[2] If you register a command for switching the cluster software system in this operand, the system will be switched over automatically when restart processing fails. The system stops operating if this operand is omitted. In such a case, you must enter a command for switching the cluster software system and then perform system switchover.

(2) Notes on using VERITAS Cluster Server

(a) Actions to take when system switchover occurs

When system switchover occurs, VERITAS Cluster Server takes all resources on the running system offline, starting with the parent resource, and then places the resources on the standby system online starting with the child resources. If an unexpected system switchover occurs, take the following actions as necessary to return the running system to its original server machine:

- When a process on the running system server machine cannot be continued

  Investigate the reason the server machine process cannot continue and eliminate the error. Activate VERITAS Cluster Server.

- When an error occurs in a resource

  Investigate the cause of the resource error and eliminate the error. Use a VERITAS Cluster Server command to release the resource's error status.

(b) Actions to take when HiRDB terminates abnormally

When HiRDB terminates abnormally, restart HiRDB on the running system. Also, when a dummy file for a `HiRDB_`*x* type resource is deleted, the resource is placed in error status and HiRDB terminates forcibly. Actions to take are described below:

Procedure

1. Use a VERITAS Cluster Server command to release the error status of the resource.
2. Use a VERITAS Cluster Server command to bring the resource online; HiRDB restarts.

(3) When the standby-less system switchover (1:1) facility is used

After resolving the error, use a planned system switchover to switch the system back to the normal BES unit (return to normal status from alternating status). For details about planned system switchover, see *25.14(2) Standby-less system switchover (1:1) facility*.

(4) When the standby-less system switchover (effects distributed) facility is used

After resolving the error, use a the planned system switchover to switch the system back to the regular unit (return to normal status from standby status). For details about planned system switchover, see *25.14(3) Standby-less system switchover (effects distributed) facility*.

25.17 Operating procedures after system switchover

When inheriting IP addresses, no action is required after system switchover, because the primary system and the secondary system have the same host name. When not inheriting IP addresses, you must perform the following procedures, because the primary system and the secondary system have different host names.

The following operations are also required when you use the standby-less system switchover (1:1) facility. Note that IP addresses are not inherited when the standby-less system switchover (effects distributed) facility is used. Because the switching destination differs for each server, the host name of the switching destination is used after system switchover; this is why the operations described below are required.

(1) Operation commands and utilities

(a) Standby system switchover facility or standby-less system switchover (1:1) facility

- Even after a system switchover, there is no need to change the host name or unit identifier specified in an operation command or utility. The `-x` option of the `pdunit` operand specifies the host name, which is the host name of the primary system (or normal BES unit).
- Specify a unit identifier as often as possible, and you will not have to be concerned with host names.
- For a HiRDB/Single Server, Hitachi recommends omitting host names to avoid problems that may occur during operation. Only specify host names when manipulating a utility special unit.
- The host name of the primary system is displayed in the processing results of operation commands. For the standby-less system switchover (1:1) facility, the host name of the alternate BES unit is displayed.

(b) Standby system switchover facility or standby-less system switchover (1:1) facility

If system switchover occurs while a server is active, each server defined in the erroneous unit must be operated as a server belonging to the accepting unit. Because the switching destination differs for each server, servers cannot be operated grouped as a regular unit after system switchover occurs.

For this reason, you must specify the values shown in Table 25-52 when you issue operation commands.

Table 25-52: Specification values for operation commands when the standby-less system switchover (effects distributed) facility is used

Type of command	Specification value	
	Before switching	After switching
Host name specification command	Host name of the regular unit	Host name specification command
Unit identifier specification command	Unit identifier of the regular unit	Unit identifier specification command
Server name specification command	Operation target server name	Server name specification command

In operations in which a host name or unit identifier is specified, the processing target differs before and after system switchover. Therefore, when the standby-less system switchover (effects distributed) facility is used, it is advisable to use operations that provide for server name specification.

When a host name is displayed in the processing results of an operation command, it is the host name of the accepting unit that is displayed.

Table 25-53 shows the standby-less system switchover (effects distributed) facility the execution targets when operation command options are specified. Be aware that the execution targets of the `pdstart` and `pdstop` commands are different from those shown in the table; for details about the `pdstart` and `pdstop` commands, see *25.13 Differences in the HiRDB operating procedures*.

Table 25-53: Execution targets when operation command options are specified (for the standby-less system switchover (effects distributed) facility)

Option		Condition		Command operation mode*	Command's execution target
-u	-s	Unit where system manager is defined	Target server		
No	No	—	—	—	System (or all running back-end servers in the system)
	Yes	Offline	—	Offline	Specified server that is a back-end server of the primary system in the regular unit
		Online	Offline	Offline	Specified server that is a back-end server of the primary system in the regular unit

Option		Condition		Command operation mode*	Command's execution target
-u	-s	Unit where system manager is defined	Target server		
			Online	Online	Specified server that is a running back-end server of the running unit
Yes	No	—	—	—	Specified unit (or all running back-end servers in the specified unit)
	Host	Offline	—	Offline	Specified unit that is a host BES in the specified regular unit
		Online	Offline	Offline	Specified unit that is a host BES in the specified regular unit
			Online	Online	Specified server that is a running BES in the running unit (`-u` specification is ignored)
	Guest	Offline	—	Offline	Specified server that is a guest BES in the specified accepting unit
		Online	Offline	Offline	Specified server that is a guest BES in the specified accepting unit
			Online	Online	Specified server that is a running back-end server in the running unit (`-u` specification is ignored)

Legend:

—: Not applicable

* Mode for determining the running system when entering a subcommand on a unit where the system manager is not defined.

(2) Messages

For the standby system switchover facility, the primary system's host name is displayed in messages.

For the standby-less system switchover (1:1) facility, the host name or unit identifier of the normal BES unit is displayed in messages for normal BES unit processing.

For the standby-less system switchover (effects distributed) facility, the host name of the accepting unit is displayed in messages.

(3) Acquisition of statistical information

For details about collecting statistical information after system switchover, see *25.13.6 (2) Process for collecting statistical information after a system switchover*.

(4) Client environment definition (applicable to the standby system switchover facility only)

(a) HiRDB/Single Server

When only the host name of the primary system is specified in the PDHOST operand, you must perform the following procedure. This procedure is not necessary when the host names of the primary system and secondary system are both specified in the PDHOST operand.

Because the host name of the HiRDB to be connected is specified in the PDHOST operand, you must specify in the PDHOST operand after system switchover the host name of the running system. Therefore, after system switchover, the HiRDB administrator must notify all client users that the host name of HiRDB has changed. Client users must change the host name specified in the PDHOST operand to this new host name, then re-execute UAPs. Note that you will not be able to execute UAPs if you do not first change the host name specified in the PDHOST operand. For details about the PDHOST operand, see the manual *HiRDB Version 8 UAP Development Guide*.

(b) HiRDB/Parallel Server

When only the host name of the primary system is specified in the operands listed below, you must perform the following procedure. This procedure is not necessary when the host names of the primary system and secondary system are both specified in these operands.

- PDHOST
- PDFESHOST (applicable when multiple front-end servers are being used)

Because the host name of the HiRDB to be connected is specified in one of these operands, the host name of the running system must be specified after system switchover in the applicable one of these operands. (the host name of the system manager is specified in the PDHOST operand and the host name of a front-end server is specified in the PDFESHOST operand). Therefore, after system switchover, the HiRDB administrator must notify all client users that the host name of HiRDB has changed. Client users must change the host name specified in the applicable operand to this new host name, then re-execute UAPs. Note that you will not be able to execute UAPs if you do not first change the host name specified in the applicable operand. For details about the PDHOST operand and the PDFESHOST operand, see the manual *HiRDB Version 8 UAP Development Guide*.

Remarks

- The host name specified in the PDHOST operand needs to be changed only when system switchover occurs on the system manager unit.
- The host name specified in the PDFESHOST operand needs to be changed when only system switchover occurs on a front-end server unit.
- There is no need to change the PDHOST or PDFESHOST operand specification when system switchover occurs for a unit where no system manager or front-end server is defined.

25.18 Reducing system switchover time (user server hot standby, rapid system switchover facility)

The following functions reduce the amount of time used to perform system switchover; the system switchover facility must be operating in the server mode to use these functions:

- User server hot standby
- Rapid system switchover facility

25.18.1 User server hot standby

When system switchover occurs, the following processes are performed to start HiRDB on the standby system.

- System server activation process
- System file inherit process
- Server process activation process
- Rollforward process

The time required for activating these server processes accounts for a large part of the system switchover time. Because the time required for activating a server process is proportional to the number of resident server processes, the system switchover time increases as the number of resident processes increases. Therefore, server processes of the standby system HiRDB can be activated in advance so that no time is required during switchover for their startup processing. The time required for system switchover is reduced by the amount of time that would have been required for their activation processing. This function is called *user server host hot standby*. For example, activating one server process on a server machine operating at about 100 MIPS requires approximately 1 second. Therefore, eliminating this activation process should reduce system switchover time by approximately 1 second.

To use the user server hot standby function, specify `Y` in the `pd_ha_server_process_standby` operand.

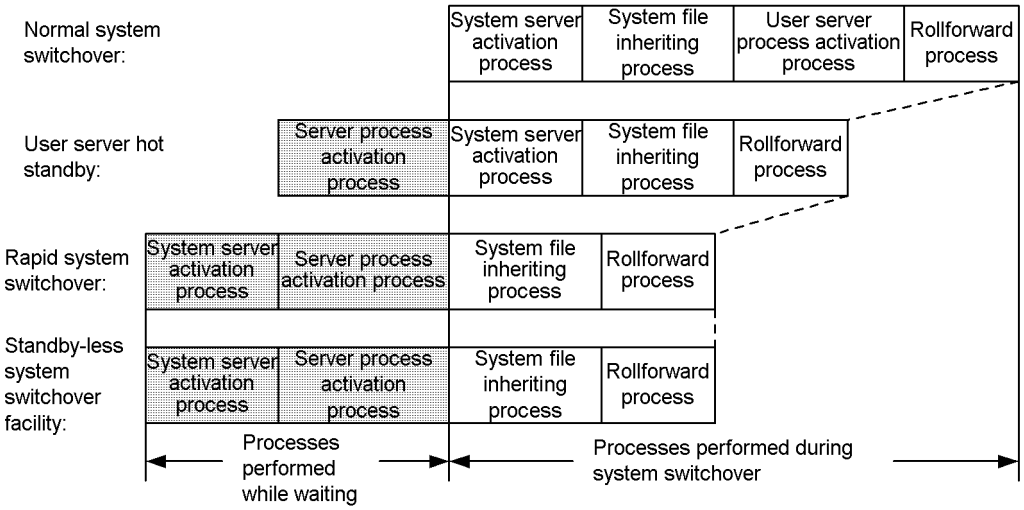
25.18.2 Rapid system switchover facility

Server processes or system servers for HiRDB on the standby system can be activated in advance, not during system switchover. This function is called the *rapid system switchover facility*. The time required for system switchover is reduced by the amount of time that would have been required for the activation processing of the server processes or system servers during system switchover.

The rapid system switchover facility is more effective in reducing the time required for system switchover than user server hot standby (the rapid system switchover facility

includes the functionality of user server hot standby). Figure 25-76 compares the system switchover times of the two system switchover facilities.

Figure 25-76: Comparison of system switchover times



Explanation

Because the processes indicated by shading execute while waiting, these processes are not executed during system switchover. The time required for system switchover is reduced by the amount of time required for these processes.

(1) Conditions for IP addresses

(a) HiRDB/Single Server

The unit that uses the rapid system switchover facility cannot inherit IP addresses. Therefore, in the case of a HiRDB/Single Server, configure the unit so it does not inherit IP addresses.

(b) HiRDB/Parallel Server

Configure the unit that uses the rapid system switchover facility so it does not inherit IP addresses. Specify `N` in the `pd_ha_ipaddr_inherit` operand in the unit control information definition for the applicable unit. Units that do not use the rapid system switchover facility can be configured to inherit IP addresses.

Hitachi recommends allowing use of the rapid system switchover facility on a back-end server unit and not allowing its use on a unit for the system manager or a front-end server. If a unit for the system manager or a front-end server is configured to not inherit IP addresses, operation after system switchover is more difficult than on a unit configured to inherit IP addresses. For system configuration examples, see *25.18.3 System configuration examples when using the rapid system switchover facility*.

(c) When using HA monitor in the cluster software

Before activating the running system and standby system HiRDB (or unit), start the IP addresses specified in the `-x` and `-c` options of the `pdunit` operand. Do not specify an IP address specified in the `-x` or `-c` option of the `pdunit` operand in the `alias` operand value's `.up` or `.down` file for HA monitor. If IP addresses for client connection are to be inherited, specify these IP addresses. If there are no IP addresses to be inherited, such as IP addresses for client connection, either specify `nouse` in the `lan_updown` operand of HA monitor's `server` definition statement or delete the `alias` operand value's `.up` and `.down` files.

(2) Operands specified when using the rapid system switchover facility

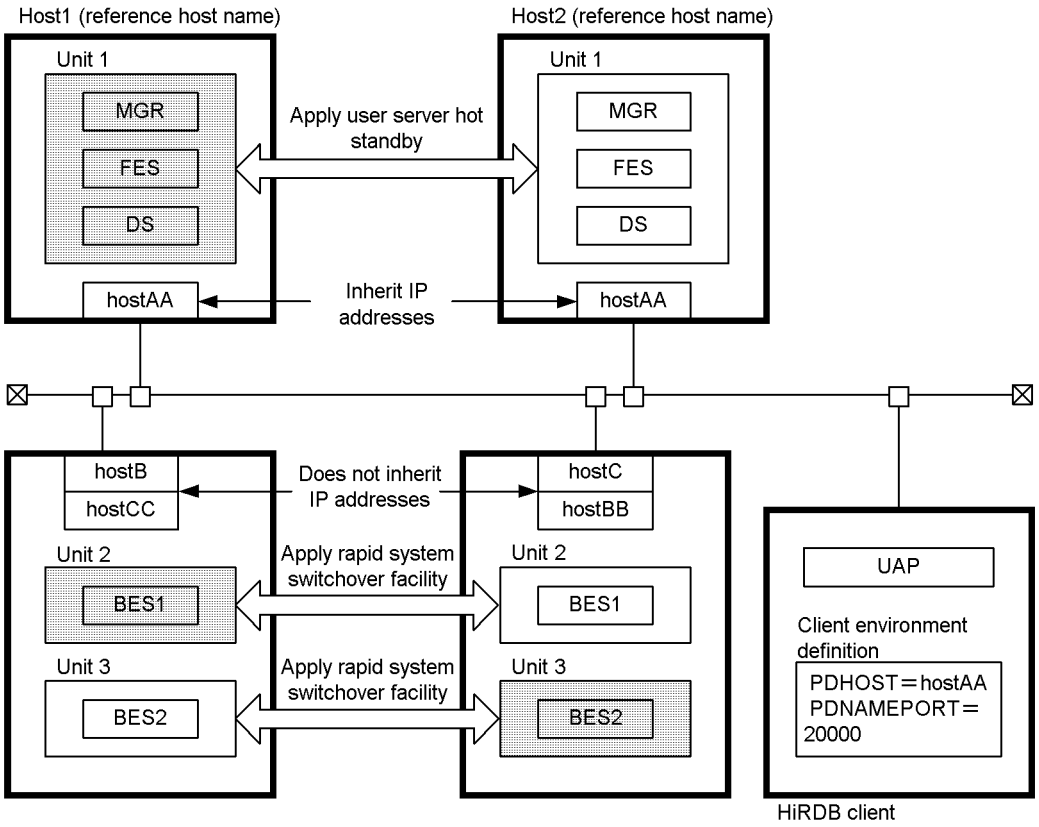
To use the rapid system switchover facility, specify `standbyunit` in the `pd_ha_agent` operand.

In the case of a HiRDB/Parallel Server, consider whether or not to use the transaction queuing facility. When the transaction queuing facility is used, fewer transaction errors will occur during system switchover. For details about the transaction queuing facility, see *25.19 Transaction queuing facility*.

25.18.3 System configuration examples when using the rapid system switchover facility

Figure 25-77 shows a system configuration example when using the rapid system switchover facility.

Figure 25-77: System configuration example when using the rapid system switchover facility



hostB and hostC: Host names
hostAA to hostCC: Host names of alias IP addresses
Note: The shaded units are primary systems.

Explanation

- Because the unit for the system manager and front-end server (unit 1) is configured to inherit IP addresses, it cannot use the rapid system control facility. This unit has a 1-to-1 switchover configuration configuration that uses the user server hot standby function.
- The back-end server units (unit 2 and unit 3) have a mutual system switchover configuration that uses the rapid system switchover facility.

The following examples show the specification of operands in the HiRDB system definition. These definition examples explain only operands related to the system common definition and unit control information definition.

System common definition

```
set pd_ha = use
set pd_name_port = 20000

pdunit -x hostAA -u unt1 -d "/hirdb1" -p 20000                      1
pdunit -x hostB -u unt2 -d "/hirdb2" -c hostBB -p 20001             2
pdunit -x hostC -u unt3 -d "/hirdb3" -c hostCC -p 20002             3

pdstart -t MGR -u unt1
pdstart -t DIC -u unt1 -s DIC
pdstart -t FES -u unt1 -s FES
pdstart -t BES -u unt2 -s BES1
pdstart -t BES -u unt3 -s BES2
```

Explanation

1. This is the definition for unit 1. Because this unit inherits IP addresses after system switchover, the `-c` option is not specified.
2. This is the definition for unit 2. Because this unit does not inherit IP addresses after system switchover, the host name of the secondary system is specified in the `-c` option.
3. This is the definition for unit 3. Because this unit does not inherit IP addresses after system switchover, the host name of the secondary system is specified in the `-c` option.

Unit control information definition (unit 1)

```
set pd_hostaname = host1                            1
set pd_ha_acttype = server                          2
set pd_ha_server_process_standby = Y                3
```

Explanation

1. Specifies the standard host name of the primary system.
2. Executes the system switchover facility in the server mode. The server mode requires that the unit use the user server hot standby function.
3. The user server hot standby function is used on this unit.

Unit control information definition (unit 2)

```
set pd_hostname = hostB                             1
set pd_ha_acttype = server                          2
set pd_ha_ipaddr_inherit = N                        3
set pd_ha_agent = standbyunit                       4
```

Explanation

1. Specifies the host name of the primary system.

2. Executes the system switchover facility in the server mode. The server mode requires that the unit use the user server hot standby function.
3. Specifies that IP addresses are not to be inherited after system switchover. Units using the rapid system switchover facility cannot inherit IP addresses.
4. The rapid system switchover facility is used on this unit.

Unit control information definition (unit 3)

```
set pd_hostname = hostC                          1
set pd_ha_acttype = server                       2
set pd_ha_ipaddr_inherit = N                     3
set pd_ha_agent = standbyunit                    4
```

Explanation

1. Specifies the host name of the primary system.
2. Executes the system switchover facility in the server mode. The server mode requires that the unit use the rapid system switchover facility.
3. Specifies that IP addresses are not to be inherited after system switchovers. Units using the rapid system switchover facility cannot inherit IP addresses.
4. The rapid system switchover facility is used on this unit.

25.18.4 Checking procedure when activation of standby system takes much time

The process for activating the standby system waits for the running system to be activated. When activation of a unit in the standby system using the rapid system switchover facility takes too much time, perform the following checking procedure.

1. Check to see if the running system was activated. Activate the running system if it was not activated.
2. Use the `pdls -d prc -a` command to check to see if the `pdenvcp` command that is issued internally by the HiRDB in the standby system has responded. If the `pdenvcp` command did not respond, check to see if the `_pd0envc` command process remains on the running system. If this command process does remain, terminate it, then reactivate the standby system.
3. Use the `pdls -d rpc` command to see if RPC or a file input/output process stopped the `pdenvcp` command that HiRDB in the running system issues internally. Eliminate the cause of the network or OS error, and then reactivate the standby system.
4. If the process of activating the standby system during system switchover times out, redefine the value of the `pd_system_complete_wait_time` operand (completion wait time for the `pdstart` command). Set a value that takes into

account the activation time of the standby system then reactivate the standby system. If the process for activating the standby system times out, use the `pdls -d prc -a` command to see if the `_pd0envc` command process remains on the running system. If this command process does remain, terminate the process, and then reactivate the standby system.

5. When using a large number of lists, the list initialization process may increase the time required for system switchover. In such a case, consider changing the list initialization time with the `pd_list_initialize_timing` operand. For details about changing the list initialization time, see *13.21(9) Changes when initializing (deleted) lists*.

25.18.5 Notes on using the rapid system switchover facility

(1) Actions requiring restarting HiRDB (or unit) in the standby system

After taking one of the actions listed below, terminate then restart HiRDB on the standby system (unit in the standby system in the case of a HiRDB/Parallel Server). If HiRDB on the standby system is not restarted, it will terminate abnormally when a system switchover occurs.

- HiRDB in the running system executed the database structure modification utility (`pdmod` command)
- After a process was performed that updates the master directory RDAREA (for example, executing the database structure modification utility, executing a definition SQL, executing the `pddbchg` command), the server in the running system (front-end server, dictionary server, or back-end server) independently terminated normally and then was started normally.
- After the `pdbuffer`, `pd_max_rdarea_no`, `pd_max_file_no`, `pd_inner_replica_control`, or `pd_index_assurance_no` `operand` was modified, the server in the running system independently terminated normally and then was started normally.
- After a foreign server or user mapping was defined, the server in the running system independently terminated normally and then was started normally.

If HiRDB in the standby system terminated abnormally, use the `pdstart` command (`pdstart -u` or `pdstart -q` for HiRDB in the standby system) to start HiRDB in the standby system.

(2) RDAREA opening trigger attributes

Units in the standby system that are subject to the rapid system switchover facility do not open any RDAREAs when in waiting status. To minimize the time required for system switchover, the rapid system switchover facility opens only the RDAREAs needed for full recovery when a system switchover occurs. Therefore, the RDAREA opening trigger attribute of the standby system does not become `INITIAL`. The

RDAREAs with the INITIAL attribute change to DEFER.

For details about RDAREA open attributes, see *15.5 Modifying an RDAREA opening trigger attribute (RDAREA modification)*.

(3) Linking with OLTP products

Caution is urged when all of the conditions listed below are present:

- The rapid system switchover facility is being used (in the case of a HiRDB/Parallel Server, units in the system manager are subject to the rapid system switchover facility)
- Products using X/Open-compliant API are linked with OLTP products (such as OpenTP1 or TPBroker)
- The HiRDB client is version 06-02-/A or earlier
- HiRDB_PDHOST is specified in the client environment variables of the OLTP products
- The primary system is in waiting completed status as the standby system

In such a case, if the OLTP products perform recovery processing on an undetermined transaction, the X/Open-compliant API may return an error and not be able to recover the transaction. If this problem occurs, upgrade the HiRDB client to version 06-02-/B or later. If you cannot immediately upgrade the HiRDB client because, for example, you do not wish to stop the current job task, switch the HiRDB (unit) in the primary system from the standby system to the running system. However, this is only a temporary measure. Be sure to upgrade the HiRDB client after the current job task has been completed.

25.19 Transaction queuing facility

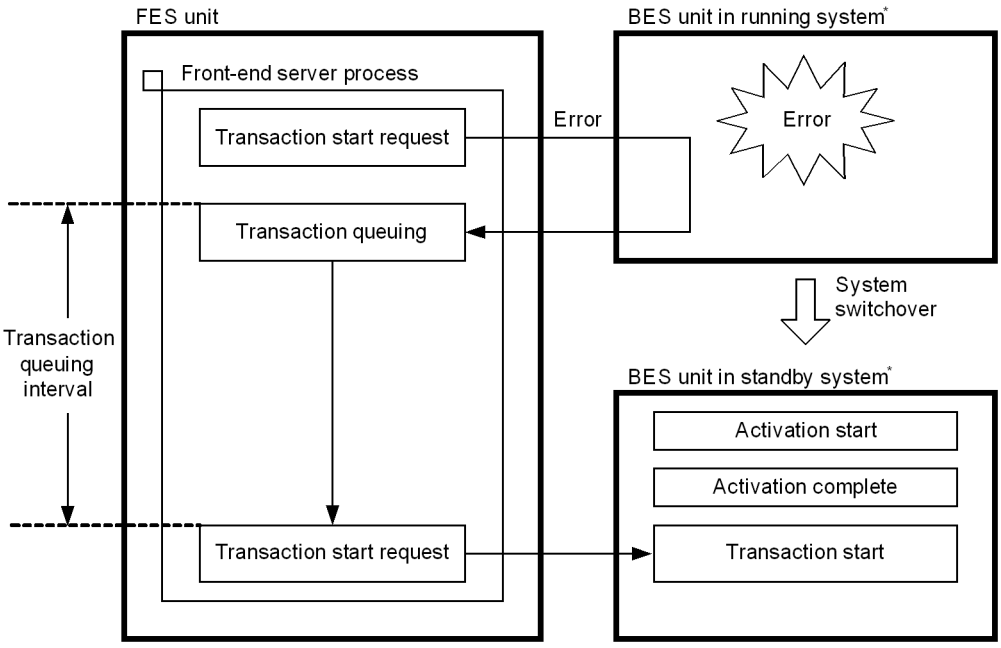
You can use the transaction queuing facility while using the rapid system switchover facility or the standby-less system switchover facility on a HiRDB/Parallel Server.

(1) About the transaction queuing facility

When system switchover occurs in a unit for a back-end server or dictionary server, the back-end server or dictionary server cannot accept any transactions until system switchover is completed. This means that any transaction to be processed by the back-end server or dictionary server while the system switchover is in progress results in an error.

The function called the *transactions queuing facility* queues transactions on the front-end server until system switchover is complete without causing errors for these transactions. This makes it possible to reduce the number of transaction errors occurring during system switchover. Figure 25-78 provides an overview of the transaction queuing facility.

Figure 25-78: Overview of the transaction queuing facility



* In the case of standby-less system switchover:

- During normal operation, the normal BES unit becomes the running system and

the alternate portion becomes the standby system.

- When in alternating status, the alternate portion becomes the running system and the normal BES unit becomes the standby system.

Explanation

An error has occurred in the unit for a back-end server and system switchover has occurred. Transactions are queued until the unit in the standby system has been activated, then transaction processing resumes.

Remarks

- Transactions for execution by a unit that was not switched (did not cause an error) are not queued. These transactions are executed as usual.
- Using multiple front-end servers makes it possible to reduce transaction errors when system switchover occurs for units for front-end servers. In such a case, only the transaction in progress when an error occurs results in an error.

(2) Setting up the environment

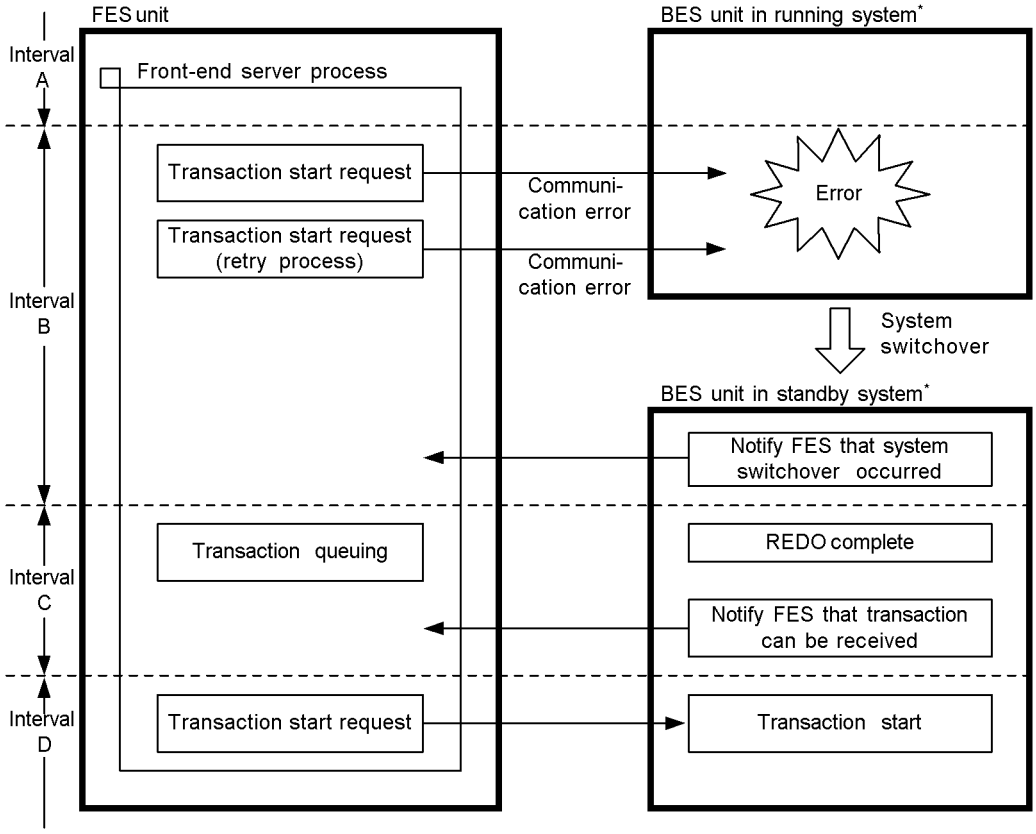
Use of the rapid system switchover facility on a HiRDB/Parallel Server is assumed. The operands explained in Table 25-54 are specified to use the transaction queuing facility.

Table 25-54: Operands specified to use the transaction queuing facility

Operand	Description
`pd_ha_transaction`	Specifies that the transaction queuing facility is to be used. When `NO` is specified in the `PDHATRNQUEUING` operand in the client environment definition, the UAP executed by the HiRDB client is not subject to the transaction queuing facility. For details about the `PDHATRNQUEUING` operand, see the manual *HiRDB Version 8 UAP Development Guide*.
`pd_ha_trn_queuing_wait_time`	Specifies the queuing wait time for transactions. If the unit in the standby system is not activated before the wait time specified in this operand expires, transactions currently being queued result in an error. Any subsequent transactions result in an error and are not queued.

Operand	Description
pd_ha_trn_restart_retry_time	If system switchover occurs while the transaction queuing facility is being used, transactions are queued on the front-end server. However, the front-end server cannot detect system switchover until the unit in the standby system restarts. During the time from when system switchover occurs until the time the unit in the standby system restarts, the front-end server continues to request that the unit in the running system start transactions. However, such transaction startup requests result in errors because the unit in the running system has already terminated abnormally. The front-end server retries the transaction startup requests continuously. A maximum retry time value is specified in this operand. If the unit in the standby system has not restarted by the time the value specified in this operand is reached, transactions currently being retried result in an error. All subsequent transactions result in an error also and are not retried. Figure 25-79 shows the relationship between the pd_ha_trn_queuing_wait_time operand and the pd_ha_trn_restart_retry_time operand.

Figure 25-79: Relationship between the pd_ha_trn_queuing_wait_time operand and the pd_ha_trn_restart_retry_time operand



* For standby-less system switchover:

- During normal operation, the normal BES unit becomes the running system and the alternate portion becomes the standby system
- When in alternating status, the alternate portion becomes the running system and the normal BES unit becomes the standby system.

Explanation

Intervals A and D:

Transactions can be started (normal status).

Interval B:

The back-end server unit is performing system switchover but the front-end server cannot detect that system switchover is in progress. A transaction

startup request is retried until the amount of time specified in the pd_ha_trn_restart_retry_time operand is reached. When the front-end server detects system switchover, the transaction is queued. If system switchover is not detected before the specified amount of time is reached, the transaction results in an error.

Interval C:

The back-end server unit is performing system switchover but the front-end server cannot detect that system switchover is in progress. The transaction is queued until the amount of time specified in the pd_ha_trn_queuing_wait_time operand is reached. If the transaction cannot be started before this amount of time is reached, the transaction results in an error.

(3) Transactions that are subject to queuing

Transactions generated by an SQL extension are subject to queuing. However, transactions generated by definition SQL and transactions using the holdable cursor facility are not subject to queuing. The following transactions not subject to queuing:

- Transactions that issue definition SQL
- Transactions with connections that open holdable cursors
- Transactions that issue the ASSIGN LIST statement
- Transactions generated by operation commands or utilities

However, depending on the timing, some of these transactions may be queued.

(4) Notes

(a) Interval monitoring operands

The transaction queuing time is the maximum value that is equal to the sum of the value of the pd_ha_trn_queuing_wait_time operand (default: 180 seconds) and the value of the pd_ha_trn_restart_retry_time operand (default: 60 seconds). Therefore, be sure to take careful note of the values for the following operands.

- PDCWAITTIME operand of the client environment definition

  Because PDCWAITTIME is also monitored during system switchover, be sure to include the time required for system switchover when determining the value of this operand.

- pd_lck_wait_timeout operand

  Because a lock may be secured on a unit that has been queued and has not caused an error, be sure to include the time required for system switchover when determining the value of this operand.

The time required for system switchover is determined by calculating the difference

between the time the cluster software outputs the system switchover startup message to `syslogfile` and the time it outputs the system switchover completion message to `syslogfile`. One of the messages listed below is output when system switchover starts:

- `KAMN200-I` message (output when a command for switching the cluster software system is entered)
- `KAMN300-E` message (output when HiRDB in the running system terminates abnormally)
- `KAMN301-E` message (output when HiRDB in the running system slows down)

The `KAMN311-I` message is output when system switchover is completed.

(b) Notes on using lists

Caution is urged if system switchover occurs while a search using a list is underway. A list created by a unit before system switchover occurred is deleted when system switchover occurs on a back-end server or a dictionary server. Therefore, a search that uses a list (queued transactions) will result in an error after system switchover occurs. In such a case, either delete or re-create the list.

(c) Maximum number of concurrent connections (value of the pd_max_users operand)

Because a larger number of users than usual is waiting to perform processing while transactions are being queued, the maximum number of concurrent connections (value of the `pd_max_users` operand) may be exceeded. If this occurs, no additional users will be able to connect to the front-end server, and processes for connecting to the front-end server will be retried. These processes will be retried only for the amount of time equal to the sum of the values specified in the `pd_ha_trn_queuing_wait_time` and `pd_ha_trn_restart_retry_time` operands.

(d) UAPs that cannot connect to HiRDB during system switchover

UAPs cannot connect to HiRDB during system switchover in the situations listed below:

- Front-end server allocation cannot be performed if the unit for the system manager is currently being switched. Therefore, a UAP that has not specified a front-end server to connect to (has not specified one in the `PDFESHOST` operand) cannot connect to HiRDB.
- If the unit for the front-end server specified in the `PDFESHOST` operand is currently being switched, the UAP cannot connect to HiRDB.

In these cases, the UAP uses the automatic reconnect facility to retry connecting to HiRDB. If system switchover is completed during the retry interval, the UAP will connect successfully to HiRDB. For details about the automatic reconnect facility, see

the manual *HiRDB Version 8 UAP Development Guide*.

(e) When the BES connection holding facility is used

For notes on using the BES connection holding facility, see *E.1(3)(c) Setting the maximum client wait time*.

25.20 System switchover when errors other than server failures occur

This section explains the procedures for performing system switchover when the following errors occur:

- A large number of server processes has terminated abnormally
- An RDAREA I/O error (path error) has occurred

25.20.1 A large number of server processes has terminated abnormally

If a large number of server processes has terminated abnormally, new services may not be accepted. HiRDB does not terminate abnormally because a server process has terminated abnormally, but HiRDB is essentially in online stopped status. Also, because HiRDB will not have terminated abnormally, system switchover will not be performed. The procedure for performing system switchover when HiRDB is in online stopped status is explained below.

(1) System switchover preparations

(a) Specify the pd_down_watch_proc operand

HiRDB (or applicable unit for a HiRDB/Parallel Server) can be terminated abnormally when the number of server processes terminating abnormally exceeds the value set in the `pd_down_watch_proc` operand during a specified period of time. The facility that terminates HiRDB abnormally in such circumstances is called the *process abnormal termination monitoring facility*. This facility is used to terminate HiRDB abnormally and perform system switchover when HiRDB is in online stopped status. For details about the process abnormal termination monitoring facility, see *8.13 Monitoring the number of times server processes terminate abnormally (abnormal termination monitoring facility)*.

(b) Check the value specified in the pd_mode_conf operand

If system switchover is to be performed when the process abnormal termination monitoring facility terminate HiRDB abnormally, specify `pd_mode_conf=MANUAL2`. If system switchover is not to be performed (if HiRDB is to be restarted on the abnormally terminated system), specify `pd_mode_conf=MANUAL1`.

(c) Specify the switchtype operand for HA monitor or Hitachi HA Toolkit Extension (applicable to the server mode only)

Specify `switch` in the `switchtype` operand for HA monitor or Hitachi HA Toolkit Extension. When `switch` is specified, system switchover will be performed when HiRDB terminates abnormally.

(d) Monitoring of the system switchover time period (applicable to the monitor mode only)

In this case, a system cannot be switched automatically even if a large number of server processes terminate abnormally and HiRDB terminates abnormally. Systems can be switched only by a user operation (such as by executing a system switchover shell script). Example system switchover operations are explained below.

- When HiRDB terminates abnormally, the `KFPS01821-E` message is output to `syslogfile`. Use JP1 or a monitor shell script to monitor this message and perform system switchover when this message is output. In the case of a HiRDB/Parallel Server, this message is output to `syslogfile` on either the server machine running the system manager or on the server machine with the unit that terminated abnormally.
- Execute the UAP regularly and see if the database can be accessed. If the UAP cannot access the database, perform a system switchover.

(2) Mutual system switchover configuration

Performing a system switchover may not be effective and may actually cause traffic to increase because more than one HiRDB is running on the same server machine. When using the process abnormal termination monitoring facility in a mutual system switchover configuration, Hitachi recommends that you not perform system switchover when HiRDB terminates abnormally. Instead, restart HiRDB in the system where it terminated abnormally by specifying `pd_mode_conf=MANUAL1`.

When running in the server mode, specify either `restart` or `manual` in the `switchtype` operand of HA monitor or Hitachi HA Toolkit Extension. When `restart` is specified, HiRDB on the system resulting in an error restarts. When HiRDB cannot be restarted on the system resulting in an error, perform a system switchover and restart HiRDB on the system that was the switchover destination. When `manual` is specified, system switchover will not be performed automatically even if HiRDB cannot be restarted.

(3) Reducing system switchover time

When a large number of server processes terminates abnormally, a large amount of troubleshooting information may be output, requiring much time to perform a system switchover. Specifying the operands listed below suppresses output of troubleshooting information and makes it possible to reduce the system switchover time when many server processes have terminated abnormally:

- `pd_dump_suppress_watch_time`
- `pd_cancel_dump = noput`
- `pd_client_waittime_over_abort = N`

Also, specifying `Y` in the `pd_ha_switch_timeout` operand makes it possible to

perform system switchover without waiting for HiRDB termination processing in the running system if that termination processing (normal BES unit for the standby-less system switchover facility) exceeds the server failure monitoring time when system switchover occurs. Note that this operand can be specified only when operating in the server mode.

Table 25-55 lists the errors that affect the system switchover time.

Table 25-55: Errors that affect the system switchover time

<table>
<tr><th colspan="5" rowspan="2">Error type (cause of system switchover)</th><th colspan="2">Effects?</th></tr>
<tr><th>Monitor mode</th><th>Server mode</th></tr>
<tr><td rowspan="8">Abnormal termination of HiRDB</td><td colspan="4">Abnormal termination of pdprcd</td><td>No</td><td>No</td></tr>
<tr><td colspan="4">Abnormal termination of system server</td><td>Yes</td><td>No</td></tr>
<tr><td rowspan="6">Abnormal termination of user server</td><td colspan="3">Critical</td><td>May</td><td>No</td></tr>
<tr><td rowspan="5">Non-critical[1]</td><td rowspan="2">PDCWAITTIME exceeded</td><td>pd_client_waittime_over_abort=Y (default)</td><td>May</td><td>No</td></tr>
<tr><td>pd_client_waittime_over_abort=N</td><td>No</td><td>No</td></tr>
<tr><td colspan="2">Internal forced termination[2]</td><td>May</td><td>No</td></tr>
<tr><td colspan="2">Abort</td><td>May</td><td>No</td></tr>
<tr><td colspan="2">Rollback occurred in UAP connected to XA</td><td>May</td><td>No</td></tr>
<tr><td colspan="2">Other than the above</td><td>May</td><td>No</td></tr>
<tr><td colspan="3">Slowdown of HiRDB</td><td>No response from pdprcd</td><td>Yes</td><td>No</td></tr>
<tr><td colspan="4">System failure</td><td>No</td><td>No</td></tr>
<tr><td colspan="4">Planned system switchover</td><td>No</td><td>No</td></tr>
</table>

Legend:

No: Has no effects on the system switchover time.

However, depending on when the error occurred, the system switchover time may be affected.

May: May have effects on the system switchover time.

Specifying the operands listed below makes it possible to minimize the effects these errors have on the system switchover time:

- pd_cancel_dump=noput
- pd_dump_suppress_watch_time

Yes: Does often have effects on the system switchover time.

[1] In the case of this error, HiRDB does not usually terminate abnormally. However, when the pd_down_watch_proc operand is specified, the number of server processes terminating abnormally is monitored and HiRDB is terminated abnormally if this number exceeds a specified value.

[2] HiRDB issues SIGKILL internally and terminates processing. Forced termination resulting when PDCWAITTIME is exceeded or the pdcancel command is issued is not included.

25.20.2 RDAREA I/O error (path error) has occurred

This section explains the procedure for performing system switchover when an RDAREA input/output error (path error) has occurred. For this purpose, an I/O error means an error that occurs when HiRDB manipulates a file, when HiRDB can no longer distinguish a file, when file manipulation fails. The error code returned from the request for access to the HiRDB file system is `-1544`.

(1) System switchover preparations

(a) Specify the pd_db_io_error_action operand

If `unitdown` is specified in the `pd_db_io_error_action` operand, HiRDB (or unit in the case of a HiRDB/Parallel Server) terminates abnormally when an RDAREA I/O error occurs and system switchover is performed. When the cause of the I/O error is a path error, job tasks can continue because I/O processing can be performed after system switchover is performed. For this purpose, a path error means a status in which files cannot be accessed because the path of communication between HiRDB and the files was interrupted.

For details about specifying `unitdown` in the `pd_db_io_error_action` operand, see *18.20 Actions to take when an RDAREA I/O error occurs*.

(b) Check the value specified in the pd_mode_conf operand

If system switchover is to be performed when HiRDB terminates abnormally, specify `pd_mode_conf=MANUAL2`. If system switchover is not to be performed (if HiRDB is to restarted on the abnormally terminated system), specify `pd_mode_conf=MANUAL1`.

(c) Specify the switchtype operand for HA monitor or Hitachi HA Toolkit Extension (applicable to the server mode only)

Specify `switch` in the `switchtype` operand for HA monitor or Hitachi HA Toolkit Extension. When `switch` is specified, system switchover will be performed when HiRDB terminates abnormally.

(d) Monitoring of the system switchover time period (applicable to the monitor mode only)

When in the monitor mode, a system cannot be switched automatically even if HiRDB terminates abnormally. Systems can be switched only by a user operation (such as by executing a system switchover shell script). Example system switchover operations are explained below.

- When HiRDB terminates abnormally, the `KFPS01821-E` message is output to the `syslogfile`. Use JP1 or a monitor shell script to monitor this message and perform system switchover when this message is output. In the case of a HiRDB/Parallel Server, this message is output to `syslogfile` on either the server machine running the system manager or on the server machine with the unit that

terminated abnormally.

- Execute the UAP regularly and see if the database can be accessed. If the UAP cannot access the database, perform a system switchover.

(2) Operation

When an I/O error occurs and HiRDB terminates abnormally, perform a system switchover and continue the processing in progress when the error occurred. Also, read the output messages and resolve the error. Then, either perform another system switchover or terminate and restart HiRDB, as appropriate. If the I/O error recurs after the system switchover, the RDAREA is shut down. In such a case, use the database recovery utility (`pdrstr` command) to recover the RDAREA.

25.21 Actions to take when a stopped unit prevents switching of the system manager unit

When system switchover for the system manager unit occurs while there is a stopped unit, it will not be possible to start the system manager unit at the switching destination. As a result, system switchover for the system manager unit fails.

However, if you take either of the following actions, you can execute system switchover for the system manager unit even when there is a stopped unit:

- Specify reduced activation
- Specify the `pd_ha_mgr_rerun` operand

The required conditions and operation timing restrictions depend on which of these methods you use. The method you choose should be appropriate for your system.

25.21.1 Using reduced activation

When there is a stopped unit, you can execute system switchover for the system manager unit by specifying the following operands:

- `pd_start_level = 1`
- `pd_start_skip_unit =` *unit-identifier-of-stopped-unit*
- `pd_reduced_check_time =` *startup-time-limit*

Table 25-56 shows how to specify these operands and the actions HiRDB takes during system switchover.

Table 25-56: Operands related to reduced activation and actions HiRDB takes during system switchover

Condition		Actions HiRDB takes when system switchover occurs for the system manager unit
pd_start_level specification value	**pd_start_skip_unit specification**	
`0` (default value)	—	If there is a stopped unit, system switchover for the system manager unit fails.
`1`	Not specified	Even if there is a stopped unit, system switchover for the system manager unit executes. However, a wait occurs for receipt of a startup processing completion notice from each unit. The startup time limit is specified in the `pd_reduced_check_time` operand (default value is 20 minutes).
	Specified	Even if there is a stopped unit, system switchover for the system manager unit executes. No checking wait time occurs.

Reference note:

- When system switchover for the system manager unit occurs, a check is made for whether or not front-end servers and back-end servers are active at the system manager unit at the switching destination. This is why a wait occurs for receipt of a startup processing completion notice from each unit. If system switchover occurs while there are units that are stopped, the operating status of such units cannot be determined; as a result, system switchover may fail or take a long time to be completed.
- If any of the following conditions is not satisfied, system switchover for the system manager unit fails:
  - A front-end server is active
  - A back-end server is active
  - A dictionary server is active

When you specify the `pd_start_skip_unit` operand, you must terminate HiRDB. Therefore, you should consider the other method if your system must remain active around the clock.

25.21.2 Specifying the pd_ha_mgr_rerun operand

(1) Processing difference depending on the operand specification

When `notwait` is specified in the `pd_ha_mgr_rerun` operand, HiRDB does not wait to receive a startup processing completion notice from each unit during system switchover for the system manager unit (at the time of startup processing at the switching destination). Table 25-57 shows the processing by HiRDB depending on the value specified in the `pd_ha_mgr_rerun` operand.

Table 25-57: Processing by HiRDB depending on the value specified in the pd_ha_mgr_rerun operand

Item	pd_ha_mgr_rerun value	
	wait (default value)	**notwait**
Whether system switchover for the system manager unit can be executed when there is a stopped unit.	System switchover cannot be executed (system switchover fails).	System switchover can be executed.
Processing that occurs during system switchover for the system manager unit.	• Checks each unit's version.[1] • Checks the system configuration.[2]	Does not perform the processing described at the left.[3]

[1] Checks whether the version of the system manager in the standby system is the same as the version of other units.

[2] Checks the following:

- Is any front-end server active?
- Is any back-end server active?
- Is a dictionary server active?

[3] When system switchover for the system manager unit is complete, the `KFPS05210-I` message (system startup completion message) is output, even if the HiRDB operating environment is not complete (for example, no front-end server is active).

Reference note:

When `notwait` is specified in the `pd_ha_mgr_rerun` operand, HiRDB does not perform version checking of the units or the system configuration check. Therefore, the time required for system switchover of the system manager unit is reduced.

(2) Required system configuration

When `notwait` is specified in the `pd_ha_mgr_rerun` operand, the system must be configured so that all the following conditions are satisfied:

- The rapid system switchover facility is used for the system manager unit.
- Only the rapid system switchover facility is used for units subject to system switchover (the user server hot standby function or a standby-less system switchover facility cannot also be used).
- The system manager and the dictionary server are defined in the same unit.
- A HiRDB/Parallel Server consists of at least two units.
- HiRDB Datareplicator is not be used.

(3) Environment setting

The procedure for setting the environment is as follows:

Procedure

1. Use the `pdadmvr` command to check that all units in the primary and standby system are of the same version.
2. Specify `notwait` in the `pd_ha_mgr_rerun` operand.

Reference note:

When `notwait` is specified in the `pd_ha_mgr_rerun` operand, HiRDB does not perform unit version checking when system switchover for the system manager unit occurs. Because correct HiRDB operation cannot be guaranteed if there is a version mismatch between units, you must implement step 1 to ensure all units are of the same version.

(4) Notes about system switchover for the system manager unit

(a) When jobs cannot be executed after system switchover for the system manager unit

When system switchover for the system manager unit is completed, the `KFPS05210-I` message (system startup completion message) is output, even if the HiRDB operating environment is not complete (for example, no front-end server is active). Consequently, if jobs cannot be executed after completion of system switchover for the system manager unit, a UAP could terminate in an error. For this reason, you should use the `pdls` command to check the operating status of each server.

(b) When system switchover for the system manager unit occurs while HiRDB is starting or terminating

If system switchover for the system manager unit occurs while HiRDB is starting (or terminating), the system manager unit starts (or terminates) regardless of the operating status of other units. Therefore, units other than the system manager unit may not be able to start (or terminate). In this case, you must terminate HiRDB forcibly.

You can avoid this situation by taking the following actions when starting or terminating HiRDB:

- When starting HiRDB

  Make sure HiRDB has started completely, then start the system manager unit of the standby system.

- When stopping the HiRDB system

  First terminate the system manager unit of the standby system; then terminate HiRDB.

(c) About the status of the stopped unit

After system switchover, you can execute the `pdls -d svr` command for a unit that was stopped at the time of system switchover for the system manager unit. `STOP(N)`, indicating normal termination status following execution of the `pdstop` command) will be displayed as long as the unit remains stopped.

If a network error causes the unit to remain in `STOP(N)` status even though it has actually started, first resolve the network error, terminate the unit, and then restart it.

Chapter

26. Using the Facility for Monitoring MIB Performance Information

This chapter explains the use of the facility for monitoring MIB performance information; this facility uses MIB to collect HiRDB operation information.

26.1 Overview of the facility for monitoring MIB performance information

This section provides an overview of the facility for monitoring MIB performance information.

26.1.1 About the facility for monitoring MIB performance information

SNMP is a protocol for managing management-target objects (such as routers, printers, and databases) on a network. By using SNMP, HiRDB operating information (performance information) can be collected at regular intervals using an SNMP agent and MIB. The collected operating information is sent to a Management Framework where it is managed. This is called the facility for monitoring MIB performance information.

The Management Framework provides a facility for converting the collected operating information into graphs, enabling the HiRDB operating information to be evaluated over a period of time. You can also set events, such as issuance of a warning when the value of an operating information item that is being monitored exceeds a specified value. In this way, changes in HiRDB's operating status can be monitored and the network administrator can take appropriate action before an incipient change results in an error.

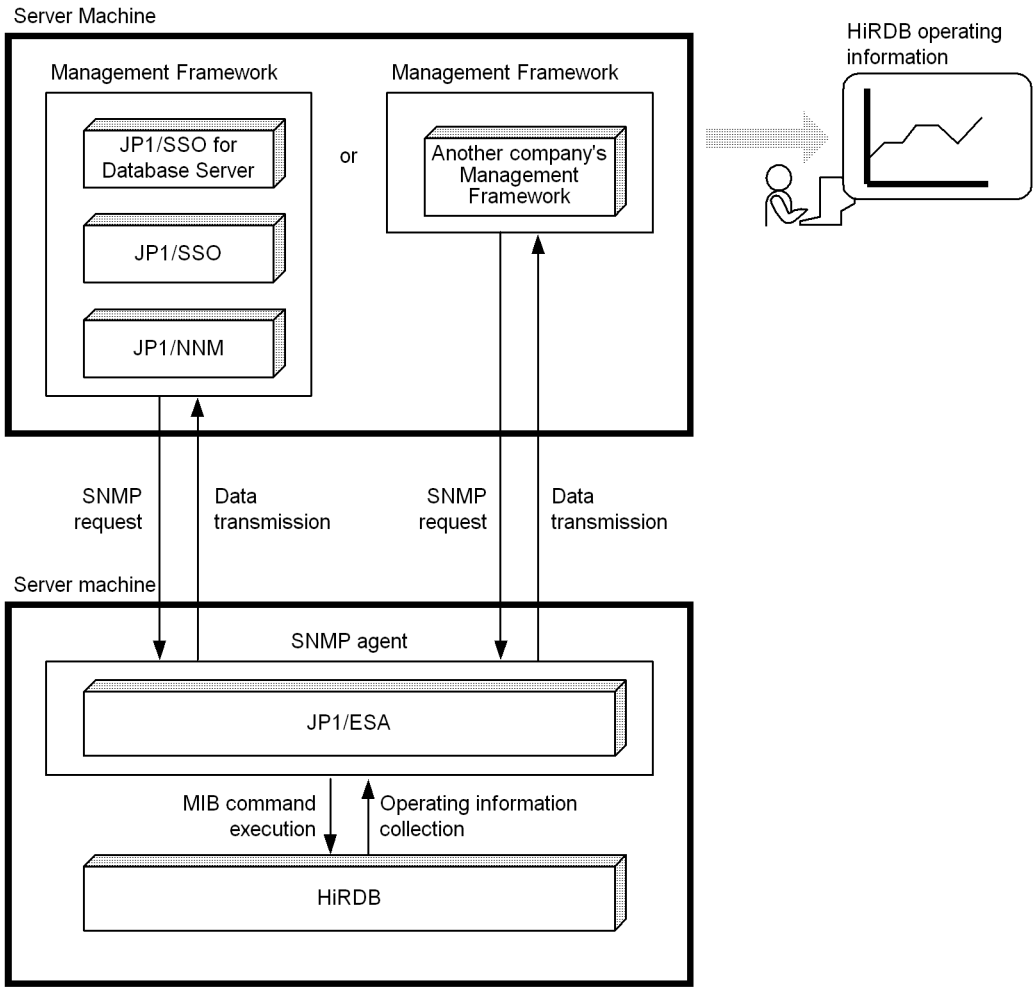
MIB records the HiRDB operating information provided by HiRDB. MIB also prescribes the method of collecting the HiRDB operating information and executes HiRDB commands, such as `pdbufls`, or utilities in response to requests from the Management Framework for the purpose of collecting the HiRDB operating information.

Reference note:

A Management Framework performs management of the collected HiRDB operating information, user notification, and creation of documentation for investigating operating problems.

Figure 26-1 provides an overview of the facility for monitoring MIB performance information.

Figure 26-1: Overview of the facility for monitoring MIB performance information



Explanation

JP1/ESA, which is an SNMP agent, uses MIB commands to collect HiRDB operating information at specified intervals. This operating information is managed by the Management Framework.

MIB:

Variables that represent information that can be accessed using SNMP are described in a text file coded in ASN.1.

MIB commands:

Commands used to collect HiRDB operating information. These commands are executed by JP1/ESA on HiRDB. A HiRDB command or utility is executed as an extension of an MIB command and returns its execution results to JP1/ESA.

SNMP:

A protocol for monitoring and controlling management-target objects (such as routers, printers, and databases) on a network via the network.

SNMP agent:

A program that transfers HiRDB operating information to the Management Framework or that collects HiRDB operating information in response to requests from the Management Framework. The following product is used as the SNMP agent:

- JP1/Cm2/Extensible SNMP Agent

Management Framework:

An SNMP-compatible management program. The following are recommended for use as the Management Framework:

- JP1/Cm2/Network Node Manager, JP1/Performance Management/SNMP System Observer, and JP1/Performance Management/SNMP System Observer for Database Server
- Management Frameworks of other companies

Reference note:

In addition to the method that uses MIB, JP1/PFM-Agent for HiRDB can also be used to monitor performance information. You must select either the facility for monitoring MIB performance information or JP1/PFM-Agent for HiRDB, whichever is appropriate for your system environment.

26.1.2 Objectives of the facility for monitoring MIB performance information

The facility for monitoring MIB performance information can be used to achieve the following objectives:

- Continuous analysis of the HiRDB operating status

  Because you collect and summarize on a regular basis such operating information as RDAREA usage rate and number of server processes, you can quickly and easily analyze the HiRDB operating status from the monitored HiRDB itself and display operating trends and history as graphs.

- Early detection of system problems and creation of documents for investigation

of problems

When performance begins to decline in the monitored HiRDB, such as the global buffer hit rate starts to fall, the HiRDB administrator can be alerted to the incipient problem by means of an e-mail. This enables corrective action to be implemented early. Moreover, because information about the problem can be displayed as a graph, documents for investigating and resolving the problem can be prepared easily and expeditiously.

26.1.3 MIB definition file

The MIB definition file stores the following types of information, which can be accessed via SNMP:

- Variable name
- Variable's object ID
- Variable's data type
- Variable's access restriction

26.1.4 MIB environment definition file

You specify the HiRDB to be monitored in the MIB environment definition file. You also specify the information (such as a user name) that will be required to execute HiRDB commands and utilities as extensions of MIB commands.

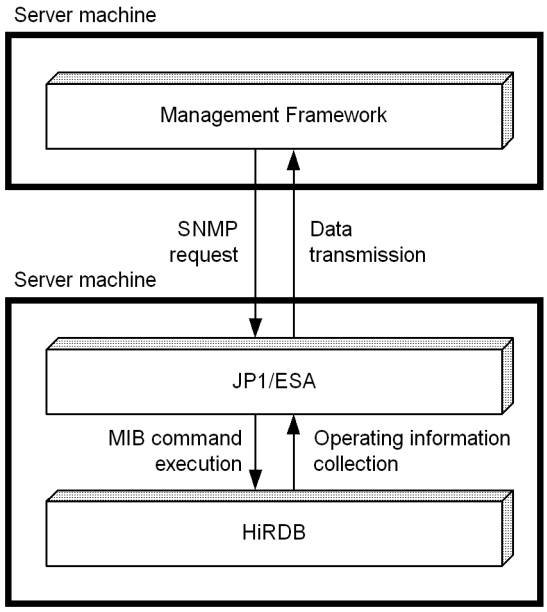
26.2 System configuration

This section provides examples of system configurations for using the facility for monitoring MIB performance information.

(1) HiRDB/Single Server

Figure 26-2 shows a system configuration for applying the facility for monitoring MIB performance information to a HiRDB/Single Server.

Figure 26-2: System configuration for a HiRDB/Single Server



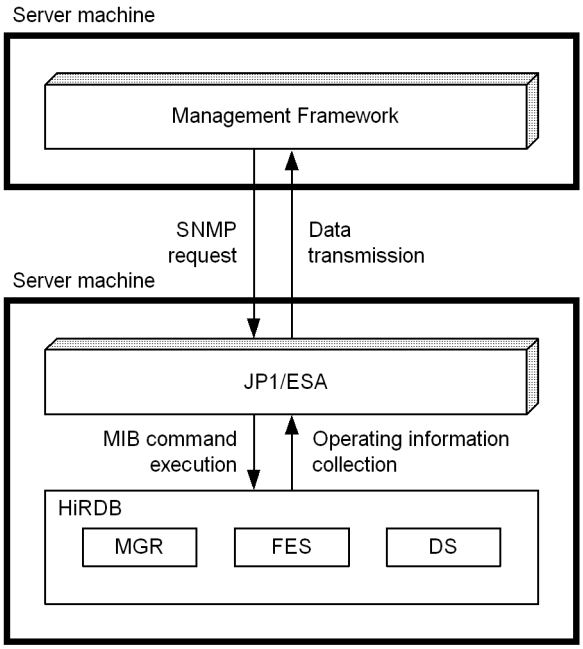
Explanation

JP1/ESA is installed in the server machine where the monitored HiRDB is located.

(2) HiRDB/Parallel Server

Figure 26-3 shows a system configuration for applying the facility for monitoring MIB performance information to a HiRDB/Parallel Server.

Figure 26-3: System configuration for a HiRDB/Parallel Server



Explanation

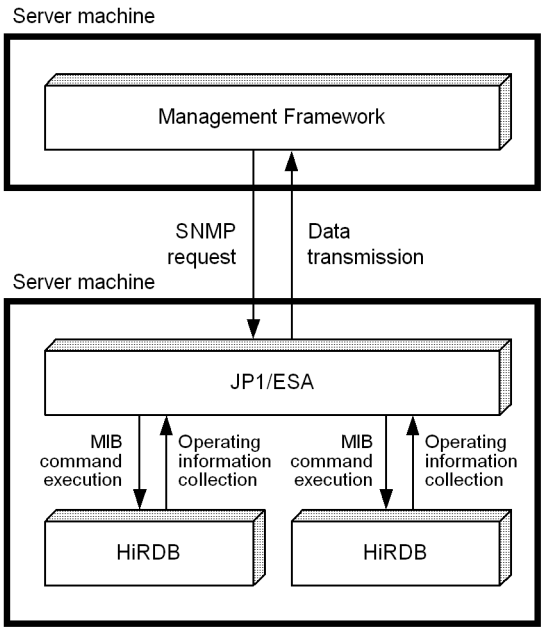
JP1/ESA is installed in the server machine where the system manager is defined, because that is the server machine where performance information is collected; there is no need to install JP1/ESA in other server machines.

MIB commands are executed at the server machine where the system manager is defined.

(3) Multi-HiRDB

Figure 26-4 shows a system configuration for applying the facility for monitoring MIB performance information to a multi-HiRDB configuration of HiRDB/Single Servers.

Figure 26-4: System configuration for a multi-HiRDB configuration of HiRDB/ Single Servers



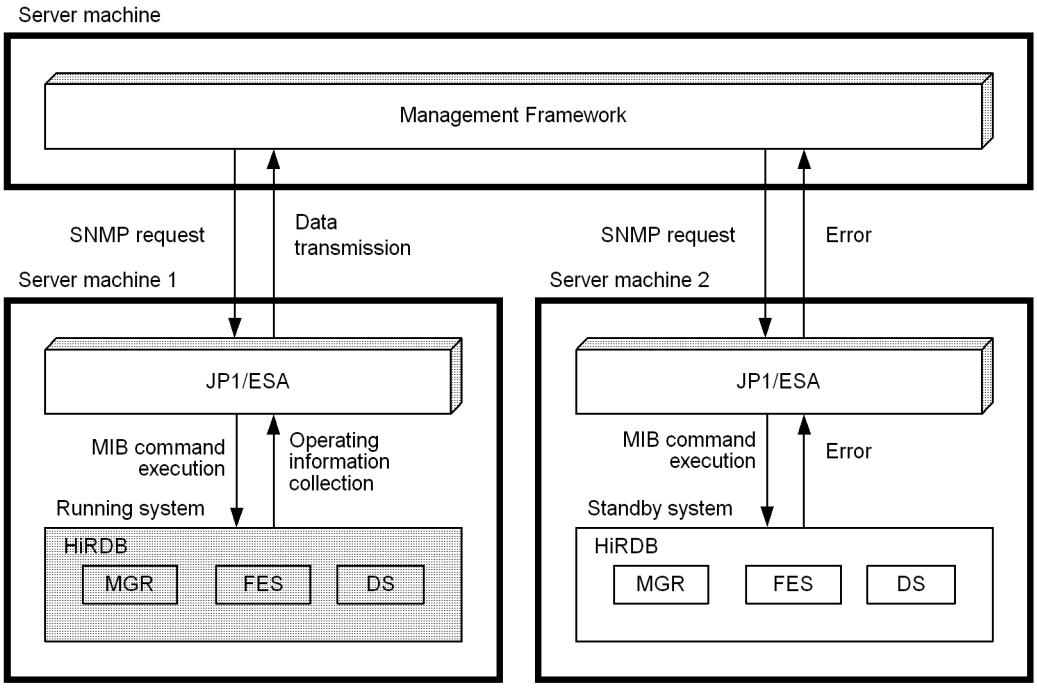
Explanation

In the case of a multi-HiRDB configuration, a single MIB command can be used to monitor multiple HiRDBs. You must specify in the MIB environment definition file all the HiRDBs that are to be monitored. Note that there is no requirement that you monitor all the HiRDBs; it is possible to monitor only selected HiRDBs.

(4) Using a system switchover facility

Figure 26-5 shows a system configuration for applying the facility for monitoring MIB performance information to a 1-to-1 switchover configuration.

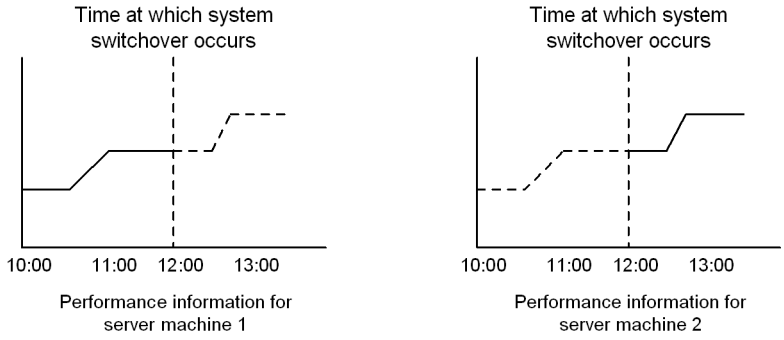
Figure 26-5: System configuration for a 1-to-1 switchover configuration



Explanation

If you are using a system switchover facility, you must also set up the server machine in the standby system for performance monitoring. That way, if system switchover should occur, monitoring will be continued for the server machine at the switching destination. However, as is shown in Figure 26-6, the collected performance information is not continuous. After switchover, it will appear as though a different HiRDB were being monitored.

Figure 26-6: Performance information when the facility for monitoring MIB performance information is applied to a 1-to-1 switchover configuration



26.3 Environment setup

This section explains the HiRDB environment setup when you use the facility for monitoring MIB performance information. You must perform all the tasks described in the subsections below.

(1) Creating the common MIB environment definition file (pdmibtgt)

You must create a common MIB environment definition file (pdmibtgt). Specify in this file the HiRDB identifier of the HiRDB to be monitored and the HiRDB directory name. A specification example follows:

```
pdmib -a HiRDB-system-identifier -d HiRDB-directory-name
```

Explanation

- Create the common MIB environment definition file in the HiRDB installation directory: `/sample/pdmibtgt`.
- In the case of a HiRDB/Parallel Server, create the common MIB environment definition file in the server machine where the system manager is defined.
- To monitor multiple HiRDBs in a multi-HiRDB environment, specify multiple lines. A specification example follows:

```
pdmib -a HRD1 -d /hirdb_x1
pdmib -a HRD2 -d /hirdb_x2
```

(2) Creating an individual MIB environment definition file (pdmibenv)

You must create an individual MIB environment definition file (pdmibenv). Specify in this file the environment for executing HiRDB commands and utilities. A specification example follows:

```
putenv PDHOST host-name-of-HiRDB-server-to-be-connected
putenv PDDIR HiRDB-directory-name-of-HiRDB-server-to-be-connected
putenv PDNAMEPORT port-number-of-HiRDB-server-to-be-connected
putenv PDUSER authorization-identifier-and-password
putenv LANG character-code-classification
```

Explanation

- The specification of each operand is the same as in the client environment

definition. For details about the client environment definition, see the manual *HiRDB Version 8 UAP Development Guide*.

- Create the individual MIB environment definition file in `$PDDIR/conf/pdmibenv` (`$PDDIR` is the HiRDB directory name specified in (1)).

Note:

If you use the `pdchgconf` command to modify `PDNAMEPORT` while HiRDB is running, you must also modify the `pdmibenv` file.

(3) Registering the MIB definition file

Register into JP1/ESA the MIB definition file provided by HiRDB. In the case of a HiRDB/Parallel Server, register it in the server machine where the system manager is defined. For details about registering into JP1/ESA, see the manual *Job Management Partner 1/Consolidated Management 2/Extensible SNMP Agent*.

The path name for the MIB definition file is as follows:

- `/opt/HiRDB_S/sample/hirdbMIB` (HiRDB/Single Server)
- `/opt/HiRDB_P/sample/hirdbMIB` (HiRDB/Parallel Server)

Notes about a multi-HiRDB

- Register the MIB definition file of the HiRDB in whose directory the `pdmibtgt` file was created.
- If the HiRDB whose MIB definition file was registered is uninstalled, you will have to re-register the MIB definition file of a HiRDB that has not been uninstalled.

(4) Setting up the remote shell execution environment (HiRDB/Parallel Server only)

Because MIB commands are executed with the superuser privilege, a superuser must set up the remote shell execution environment. Settings that permit mutual logon between any server comprising the HiRDB/Parallel Server and the server machine where the system manager is defined must be specified in `/etc/hosts.equiv` or `/.rhosts`.

(5) Setting up environment variables

Add the installation directory `/lib` to the superuser's `SHLIB_PATH` environment variable. For the Solaris or Linux version, add the installation directory `/lib` to the superuser's `LD_LIBRARY_PATH` environment variable. For the AIX 5L version, add the installation directory `/lib` to the superuser's `LIBPATH` environment variable.

(6) Selecting the performance information to be collected

(a) Using JP1/SSO for Database

Use JP1/SSO for Database to select the performance information that you wish to collect. For details about the selection method, see the manual *Job Management Partner 1/Performance Management/SNMP System Observer for Extended Resource Management*.

Note:

Depending on the environment applicable to the history to be monitored and the MIB for which information is to be collected, an SNMP timeout error may occur. In such a case, adjust appropriately the SNMP timeout value for JP1/NNM or JP1/ESA.

(b) Using another company's Management Framework

See the relevant manuals for other company's Management Framework.

26.4 MIB definition file

This section explains the conventions for the MIB definition file and the MIB tables that are stored in the MIB definition file.

(1) Conventions for the MIB definition file provided by HiRDB

Table 26-1 explains the conventions for the MIB definition file provided by HiRDB.

Table 26-1: Conventions for the MIB definition file provided by HiRDB

Item	Conventions
File name	File name is `hirdbMIB`.
Object ID	HiRDB's object ID is `.iso.org.dod.internet.private.enterprises.hitachi.systemExMib.hirdbMibs.HiRDB`. In numeric format, it is `1.3.6.1.4.1.116.5.24.2`. Objects containing HiRDB performance information are located under this object ID.
Comment (`DESCRIPTION`)	A comment for the object ID can be specified on the first line, followed on the second and subsequent lines by JP1/ESA commands.
Index (`INDEX`)	Indexes identify an MIB object. For example, to identify information on a particular RDAREA, the following two indexes are required: • HiRDB system index • RDAREA index
Access restriction (`ACCESS`)	Access restriction for the object. In HiRDB, all objects except tables and entries are read-only. `GetRequest` or `GetNextRequest` is allowed, but `SetRequest` is not. Tables and entries are set to `not-access` and cannot be accessed.
Object type (`SYNTAX`)	`DisplayString` (ASCII) is used for characters, and `INTEGER` ($-(2^{31}-1)$ to $2^{31}-1$) or `Gauge` (0 to $2^{32}-1$) is used for numbers. `SEQUENCE` is used for tables.

(2) MIB tables provided by HiRDB

Table 26-2 lists the MIB tables provided by HiRDB.

Table 26-2: List of MIB tables provided by HiRDB

Table name	Explanation
Server status table (`hirServerStatusTable`)	Stores a snapshot of performance information on a server's operating status at a given point in time. For details, see *26.5 Server status table (hirServerStatusTable)*.
Work table HiRDB file system area table (`hirFileSystemTable`)	Stores performance information collected over a given time period on a HiRDB file system area for work table files. For details, see *26.6 Work table HiRDB file system area table (hirFileSystemTable)*.

Table name	Explanation
RDAREA table (hirRdareaStatusTable)	Stores performance information collected over a given time period on RDAREAs. For details, see *26.7 RDAREA table (hirRdareaStatusTable)*.
RDAREA details table (hirRdareaDetStatusTable)	Stores performance information collected over a given time period on RDAREAs. For details, see *26.8 RDAREA details table (hirRdareaDetStatusTable)*.
Global buffer table (hirBufferStatusTable)	Stores performance information collected over a given time period on global buffers. For details, see *26.9 Global buffer table (hirBufferStatusTable*).
HiRDB file system area (RDAREAs) table (hirRdareaFileTable)	Stores performance information collected over a given time period on a HiRDB file system area for RDAREAs. For details, see *26.10 HiRDB file system area (RDAREAs) table (hirRdareaFileTable)*.
SYS statistics table (hirStatisInfSysTable)	Collects statistical information related to system operations for servers. For details, see *26.11 SYS statistics table (hirStatisInfSysTable)*.

26.5 Server status table (hirServerStatusTable)

The server status table stores a snapshot of performance information on a server's operating status at a given point in time. An instance is created for each unit or server.

Notes on collecting performance information

Set the performance information collection interval to no less than 60 seconds so that HiRDB performance is not affected adversely.

Table 26-3 shows the configuration of the server status table.

Table 26-3: Configuration of the server status table

ID	Object	Explanation	Type	Privilege	Data source
1.	`hirServerStatusTable`	Server status table	`SEQUENCE`	not-access	—
1.1	`hirServerStatusEntry`	Server operating status entry	`SEQUENCE`	not-access	—
1.1.1	`hirServerStatusSysIndex`	HiRDB system identifier index	`INTEGER`	read-only	MIB command
1.1.2	`hirServerStatusHostIndex`	HiRDB host identifier index	`INTEGER`	read-only	MIB command
1.1.3	`hirServerStatusUnitIndex`	HiRDB unit identifier index	`INTEGER`	read-only	MIB command
1.1.4	`hirServerStatusServerIndex`	HiRDB server identifier index	`INTEGER`	read-only	MIB command
1.1.5	`hirServerStatusInstance`	Instance name *HiRDB-identifier*:*host-name*:*unit-identifier*:*server-name*	`DisplayString`	read-only	MIB command
1.1.6	`hirServerStatusHirID`	HiRDB identifier	`DisplayString`	read-only	MIB command
1.1.7	`hirServerStatusHost`	Host name	`DisplayString`	read-only	`pdls -d svr`
1.1.8	`hirServerStatusServerName`	Server name `********` is displayed for a unit.	`DisplayString`	read-only	`pdls -d svr`

ID	Object	Explanation	Type	Privilege	Data source
1.1.9	hirServerStatusStartTime	Start time of server or unit 999999 is displayed if the server or unit is stopped.	DisplayString	read-only	pdls -d svr
1.1.10	hirServerStatusStatus	Server or unit status One of the following: 1: ACTIVE 2: STOP 3: STOP (N) 4: STOP (F) 5: STOP (A) 6: START (I) 7: SUSPEND 8: STARTING 9: STOPPING 10:TRNPAUSE	INTEGER	read-only	pdls -d svr
1.1.11	hirServerStatusUnitID	Unit identifier	DisplayString	read-only	pdls -d svr
1.1.12	hirServerStatusDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

26.6 Work table HiRDB file system area table (hirFileSystemTable)

The work table HiRDB file system area table stores performance information collected over a given time period on a HiRDB file system area for work table files.

Notes on collecting performance information

Set the performance information collection interval to no less than 600 seconds so that HiRDB performance is not affected adversely.

Table 26-4 shows the configuration of the work table HiRDB file system area table.

Table 26-4: Configuration of the work table HiRDB file system area table

ID	Object	Explanation	Type	Privilege	Data source
3.	hirFileSystemTable	Work table HiRDB file system area table	SEQUENCE	not-access	—
3.1	hirFileSystemEntry	Work table HiRDB file system area entry	SEQUENCE	not-access	—
3.1.1	hirFileSystemSysIndex	HiRDB system identifier index	INTEGER	read-only	MIB command
3.1.2	hirFileSystemFileIndex	HiRDB file system identifier index	INTEGER	read-only	MIB command
3.1.3	hirFileSystemInstance	Instance name *HiRDB-identifier*:*host-name*:*HiRDB-file-system-area-name*	DisplayString	read-only	MIB command
3.1.4	hirFileSystemHirID	HiRDB identifier	DisplayString	read-only	MIB command
3.1.5	hirFileSystemAvailableExpandCount	Maximum number of HiRDB file system area extensions	INTEGER	read-only	pdfstatfs -d -b
3.1.6	hirFileSystemAvailableFileSize	Maximum size of each file that can be allocated in the HiRDB file system area (KB)	INTEGER	read-only	pdfstatfs -d -b
3.1.7	hirFileSystemCurrentExpandCount	Total number of HiRDB file system area extensions	INTEGER	read-only	pdfstatfs -d -b
3.1.8	hirFileSystemCurrentFileCount	Number of HiRDB files created	INTEGER	read-only	pdfstatfs -d -b

ID	Object	Explanation	Type	Privilege	Data source
3.1.9	hirFileSystemFreeAreaCount	Total number of non-contiguous free areas	INTEGER	read-only	pdfstatfs -d -b
3.1.10	hirFileSystemHiRDBFileSystemAreaName	HiRDB file system area name	DisplayString	read-only	pdfstatfs -d -b
3.1.11	hirFileSystemPeakCapacity	Largest space* used by the user since the maximum value allocated in the HiRDB file system area was reset to 0 (KB)	INTEGER	read-only	pdfstatfs -d -b
3.1.12	hirFileSystemRemainingFileCount	Number of HiRDB files that can still be created (*max-number-of-files-that-can-be-created* − *number-of-created-files*)	INTEGER	read-only	pdfstatfs -d -b
3.1.13	hirFileSystemRemainingUserArea	Size of unused area in the area allocated to the user (area not allocated as HiRDB files)(KB)	INTEGER	read-only	pdfstatfs -d -b
3.1.14	hirFileSystemSectorSize	HiRDB file system area sector size (KB) Sector size specified in -s option of pdfmkfs command (default is 1024)	INTEGER	read-only	pdfstatfs -d -b
3.1.15	hirFileSystemHost	Host name	DisplayString	read-only	pdfstatfs -d -b
3.1.16	hirFileSystemUserAreacapacity	Size of the HiRDB file system area in the user area (KB)	INTEGER	read-only	pdfstatfs -d -b
3.1.17	hirFileSystemDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

* Because the PEAK_CAPACITY value is not cleared until the pdfstatfs -c command is executed on the applicable HiRDB file system area, this field's value does not decrease.

26.7 RDAREA table (hirRdareaStatusTable)

The RDAREA table stores performance information collected over a given time period on RDAREAs. One line is created for each RDAREA.

Notes on collecting performance information

- Set the performance information collection interval to not less than 600 seconds so that HiRDB performance is not affected adversely.
- For a shared RDAREA, performance information of updatable back-end servers only is collected.

Table 26-5 shows the configuration of the RDAREA table.

Table 26-5: Configuration of the RDAREA table

ID	Object	Explanation	Type	Privilege	Data source
4.	`hirRdareaStatusTable`	RDAREA table	`SEQUENCE`	not-access	—
4.1	`hirRdareaStatusEntry`	RDAREA entry	`SEQUENCE`	not-access	—
4.1.1	`hirRdareaStatusSysIndex`	HiRDB system identifier index	`INTEGER`	read-only	MIB command
4.1.2	`hirRdareaStatusRdareaIndex`	RDAREA identifier index	`INTEGER`	read-only	MIB command
4.1.3	`hirRdareaStatusInstance`	Instance name *HiRDB-identifier*:*server-name*:*RDAREA-name*	`DisplayString`	read-only	MIB command
4.1.4	`hirRdareaStatusHirID`	HiRDB identifier	`DisplayString`	read-only	MIB command
4.1.5	`hirRdareaStatusBufferName`	Global buffer name	`DisplayString`	read-only	`pdbufls -k def-s SERVR_NAME`
4.1.6	`hirRdareaStatusExtensionSegmentSize`	Number of segment extensions `0` if RDAREA extension is not specified.	`INTEGER`	read-only	Dictionary table*
4.1.7	`hirRdareaStatusFileCount`	Number of HiRDB files	`INTEGER`	read-only	Dictionary table*

ID	Object	Explanation	Type	Privilege	Data source
4.1.8	hirRdareaStatusIndexCount	Number of stored indexes (number defined)	INTEGER	read-only	Dictionary table*
4.1.9	hirRdareaStatusPageSize	Page size (bytes)	INTEGER	read-only	Dictionary table*
4.1.10	hirRdareaStatusRDAREAName	RDAREA name	DisplayString	read-only	pddbls -r ALL -a
4.1.11	hirRdareaStatusRDAREAStatus	RDAREA status One of the following: CLOSE, CLOSE HOLD, CLOSE HOLD (INQ), CLOSE HOLD (CMD), CLOSE HOLD (BU), CLOSE HOLD (BU I), CLOSE HOLD (BU W), CLOSE HOLD (BU IW), CLOSE HOLD (SYNC), CLOSE HOLD (ORG), CLOSE ACCEPT-HOLD, HOLD, HOLD (INQ), HOLD (CMD), HOLD (BU), HOLD (BU I), HOLD (BU W), HOLD (BU IW), HOLD (SYNC), HOLD (ORG), ACCEPT-HOLD, OPEN	DisplayString	read-only	pddbls -r ALL -a
4.1.12	hirRdareaStatusRDAREAType	RDAREA type One of the following: MAST, DDIR, DDIC, DLOB, USER, ULOB, LIST, RGST, RLOB	DisplayString	read-only	pddbls -r ALL -a
4.1.13	hirRdareaStatusSegmentSize	Segment size (pages)	INTEGER	read-only	Dictionary table*
4.1.14	hirRdareaStatusServerName	Server name	DisplayString	read-only	pddbls -r ALL -a
4.1.15	hirRdareaStatusTableCount	Number of stored tables (number defined)	INTEGER	read-only	Dictionary table*
4.1.16	hirRdareaStatusTotalRDAREASegments	Total number of segments in the RDAREA	INTEGER	read-only	pddbls -r ALL -a

ID	Object	Explanation	Type	Privilege	Data source
4.1.17	hirRdareaStatusUnusedRDAREASegments	Number of unused segments in theRDAREA	INTEGER	read-only	pddbls -r ALL -a
4.1.18	hirRdareaStatusDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

* Result of the following SQL statement:

```
SELECT
PAGE_SIZE,SEGMENT_SIZE,FILE_COUNT,N_TABLE,N_INDEX,EXTENSION
_SEGMENT_SIZE
      FROM "MASTER".SQL_RDAREAS
```

26.8 RDAREA details table (hirRdareaDetStatusTable)

The RDAREA details table stores performance information collected over a given time period on RDAREAs. One line is created for each RDAREA.

Notes on collecting performance information

- Set the performance information collection interval to not less than 3600 seconds so that HiRDB performance is not affected adversely.
- For a shared RDAREA, the performance information of updatable back-end servers only is collected.
- If the `pddbst` command cannot be executed, `0` (for a numeric value field) or a blank (for a character string field) is set in any field that acquires data from the `pddbst` command.

  The `pddbst` command can be executed when both the following conditions are satisfied:

  - The target RDAREA is a data dictionary RDAREA, user RDAREA, LOB RDAREA, registry RDAREA, or registry LOB RDAREA.
  - The target RDAREA is open without shutdown or referencing-possible hold in effect.

Table 26-6 shows the configuration of the RDAREA details table.

Table 26-6: Configuration of the RDAREA details table

ID	Object	Explanation	Type	Privilege	Data source
5.	`hirRdareaDetStatusTable`	RDAREA details table	`SEQUENCE`	not-access	—
5.1	`hirRdareaDetStatusEntry`	RDAREA details entry	`SEQUENCE`	not-access	—
5.1.1	`hirRdareaDetStatusSysIndex`	HiRDB system identifier index	`INTEGER`	read-only	MIB command
5.1.2	`hirRdareaDetStatusRdareaIndex`	RDAREA identifier index	`INTEGER`	read-only	MIB command
5.1.3	`hirRdareaDetStatusInstance`	Instance name *HiRDB-identifier*:*server-name*:*RDAREA-name*	`DisplayString`	read-only	MIB command

ID	Object	Explanation	Type	Privilege	Data source
5.1.4	hirRdareaDetStatusHirID	HiRDB identifier	DisplayString	read-only	MIB command
5.1.5	hirRdareaDetStatusAutoExtendErrorCode	Error code when automatic extension cannot be executed	INTEGER	read-only	pddbst -k phys -f
5.1.6	hirRdareaDetStatusAutoExtendStatus	Automatic extension facility's status: SUP: Suppressed NOSUP: Not suppressed	DisplayString	read-only	pddbst -k phys -f
5.1.7	hirRdareaDetStatusAutoExtendUse	Automatic extension facility's use status: USE: Being used NOUSE: Not being used	DisplayString	read-only	pddbst -k phys -f
5.1.8	hirRdareaDetStatusBufferName	Global buffer name	DisplayString	read-only	pdbufls -k def-s SERVR_NAME
5.1.9	hirRdareaDetStatusExtensionSegmentSize	Number of segment extensions 0 if RDAREA extension is not specified.	INTEGER	read-only	Dictionary table[1]
5.1.10	hirRdareaDetStatusFileCount	Number of HiRDB files	INTEGER	read-only	Dictionary table[1]
5.1.11	hirRdareaDetStatusFreezeSpecified	Whether or not RDAREA updating is frozen	DisplayString	read-only	pddbst -k phys -f[2]
5.1.12	hirRdareaDetStatusFullUsedPage	Ratio of full pages (%)	INTEGER	read-only	pddbst -k phys -f[9]
5.1.13	hirRdareaDetStatusFullUsedPages	Number of full pages	INTEGER	read-only	pddbst -k phys -f[9]
5.1.14	hirRdareaDetStatusGenNumber	Generation number of the RDAREA	INTEGER	read-only	pddbst -k phys -f[4]
5.1.15	hirRdareaDetStatusHoldCode1	Hold cause code for the RDAREA	INTEGER	read-only	pddbst -k phys -f

ID	Object	Explanation	Type	Privilege	Data source
5.1.16	hirRdareaDetStatusHoldCode2	Hold cause code immediately prior to Hold Status 1	INTEGER	read-only	pddbst -k phys -f
5.1.17	hirRdareaDetStatusHoldStatus1	Hold type of the RDAREA: 1: Error shutdown (shut down by an error) 2: Command shutdown (command shutdown based on error detection by HiRDB) 3: Value output by pddbst (value other than FLT or CMD) 4: Blank (other cases)	DisplayString	read-only	pddbst -k phys -f
5.1.18	hirRdareaDetStatusHoldStatus2	Hold status immediately prior to Hold Status 1	DisplayString	read-only	pddbst -k phys -f
5.1.19	hirRdareaDetStatusHoldTime1	Hold time of the RDAREA	DisplayString	read-only	pddbst -k phys -f
5.1.20	hirRdareaDetStatusHoldTime2	Hold time immediately prior to Hold Status 1	DisplayString	read-only	pddbst -k phys -f
5.1.21	hirRdareaDetStatusIndexCount	Number of stored indexes (number defined)	INTEGER	read-only	Dictionary table[1]
5.1.22	hirRdareaDetStatusLastSegment	Position information indicating the last segment being used Always indicates the last segment if Segment Over is Y.	INTEGER	read-only	pddbst -k phys -f[6]
5.1.23	hirRdareaDetStatusLobmapOver	Whether or not all LOB management entries are being used: Y: All are being used N: Some entries are unused	DisplayString	read-only	pddbst -k phys -f[5]
5.1.24	hirRdareaDetStatusOriginal RDAREAName	Original RDAREA name	DisplayString	read-only	pddbst -k phys -f[3]
5.1.25	hirRdareaDetStatusPageSize	Page size (bytes)	INTEGER	read-only	Dictionary table[1]

ID	Object	Explanation	Type	Privilege	Data source
5.1.26	hirRdareaDetStatusRDAREAName	RDAREA name	Display String	read-only	pddbls -r ALL -a
5.1.27	hirRdareaDetStatusRDAREAStatus	RDAREA status One of the following: CLOSE, CLOSE HOLD, CLOSE HOLD (INQ), CLOSE HOLD (CMD), CLOSE HOLD (BU), CLOSE HOLD (BU I), CLOSE HOLD (BU W), CLOSE HOLD (BU IW), CLOSE HOLD (SYNC), CLOSE HOLD (ORG), CLOSE ACCEPT-HOLD, HOLD, HOLD (INQ), HOLD (CMD), HOLD (BU), HOLD (BU I), HOLD (BU W), HOLD (BU IW), HOLD (SYNC), HOLD (ORG), ACCEPT-HOLD, OPEN	Display String	read-only	pddbls -r ALL -a
5.1.28	hirRdareaDetStatusRDAREAType	RDAREA type One of the following: MAST, DDIR, DDIC, DLOB, USER, ULOB, LIST, RGST, RLOB	Display String	read-only	pddbls -r ALL -a
5.1.29	hirRdareaDetStatusReplicaRDAREAs	Number of replica RDAREAs	INTEGER	read-only	pddbst -k phys -f[4]
5.1.30	hirRdareaDetStatusSegmentOver	Whether or not LOB RDAREA storage is disorganized: Y: Disorganized N: Organized	Display String	read-only	pddbst -k phys -f[5]
5.1.31	hirRdareaDetStatusSegmentSize	Segment size	INTEGER	read-only	Dictionary table[1]
5.1.32	hirRdareaDetStatusServerName	Server name	Display String	read-only	pddbls -r ALL -a
5.1.33	hirRdareaDetStatusTableCount	Number of stored tables (number defined)	INTEGER	read-only	Dictionary table[1]
5.1.34	hirRdareaDetStatusTotalPages	Total number of pages in segments in the RDAREA (*number-of-used-pages* + *number-of-unused-pages*)	INTEGER	read-only	pddbst -k phys -f[10]

ID	Object	Explanation	Type	Privilege	Data source
5.1.35	hirRdareaDetStatusTotalRDAREASegments	Total number of segments in the RDAREA	INTEGER	read-only	pddbls -r ALL -a
5.1.36	hirRdareaDetStatusUnusedRDAREASegments	Number of unused segments in the RDAREA	INTEGER	read-only	pddbls -r ALL -a
5.1.37	hirRdareaDetStatusUsed	Rate of used segments	INTEGER	read-only	Fixed to 0
5.1.38	hirRdareaDetStatusUsedPage	Percentage of used pages (%)	INTEGER	read-only	pddbst -k phys -f[8]
5.1.39	hirRdareaDetStatusUsedPages	Number of used pages	INTEGER	read-only	pddbst -k phys -f[7]
5.1.40	hirRdareaDetStatusDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

[1] Result of the following SQL statement:

```
SELECT
PAGE_SIZE,SEGMENT_SIZE,FILE_COUNT,N_TABLE,N_INDEX,EXTENSION
_SEGMENT_SIZE FROM "MASTER".SQL_RDAREAS
```

[2] Left blank except in the case of a LOB RDAREA.

[3] Left blank if HiRDB Staticizer Option is not installed.

[4] 0 if HiRDB Staticizer Option is not installed.

[5] Left blank except in the case of a LOB RDAREA.

[6] 0 except in the case of a LOB RDAREA.

[7] Value of USED_AREA_SEG in the case of a LOB RDAREA.

[8] Value of PERCENT_USED in the case of a LOB RDAREA.

[9] `0` in the case of a LOB RDAREA.

[10] Value of `TOTAL_AREA_SEG` in the case of a LOB RDAREA.

26.9 Global buffer table (hirBufferStatusTable)

The global buffer table stores performance information collected over a given time period on global buffers. One line is created for each global buffer.

Notes on collecting performance information

Set the performance information collection interval to no less than 60 seconds so that HiRDB performance is not affected adversely.

Table 26-7 shows the configuration of the global buffer table.

Table 26-7: Configuration of the global buffer table

ID	Object	Explanation	Type	Privilege	Data source
6.	hirBufferStatusTable	Global buffer table	SEQUENCE	not-access	—
6.1	hirBufferStatusEntry	Global buffer entry	SEQUENCE	not-access	—
6.1.1	hirBufferStatusSysIndex	HiRDB system identifier index	INTEGER	read-only	MIB command
6.1.2	hirBufferStatusBufferIndex	Global buffer identifier index	INTEGER	read-only	MIB command
6.1.3	hirBufferStatusInstance	Instance name *HiRDB-identifier*:*server-name*:*global-buffer-name*	DisplayString	read-only	MIB command
6.1.4	hirBufferStatusHirID	HiRDB identifier	DisplayString	read-only	MIB command
6.1.5	hirBufferStatusBufferName	Global buffer name	DisplayString	read-only	pdbufls -k sts -d
6.1.6	hirBufferStatusBufferPoolHitRate	Global buffer pool hit rate (%)	INTEGER	read-only	pdbufls -k sts -d
6.1.7	hirBufferStatusCurrentReferenceBuffers	Current number of reference buffers	INTEGER	read-only	pdbufls -k sts -d

ID	Object	Explanation	Type	Privilege	Data source
6.1.8	hirBufferStatusCurrentUpdateBuffers	Current number of update buffers	INTEGER	read-only	pdbufls -k sts -d
6.1.9	hirBufferStatusDBSyncs	Number of synchronization points that have occurred	Gauge	read-only	pdbufls -k sts -d
6.1.10	hirBufferStatusLOBBufferInputPages	Number of LOB buffer batch input pages	Gauge	read-only	pdbufls -k sts -d
6.1.11	hirBufferStatusLOBBufferOutputPages	Number of LOB buffer batch output pages	Gauge	read-only	pdbufls -k sts -d
6.1.12	hirBufferStatusLOBBufferReadRequests	Number of LOB buffer read requests	Gauge	read-only	pdbufls -k sts -d
6.1.13	hirBufferStatusLOBBufferWriteRequests	Number of LOB buffer write requests	Gauge	read-only	pdbufls -k sts -d
6.1.14	hirBufferStatusOutOfBuffer	Number of times buffer shortage occurred	Gauge	read-only	pdbufls -k sts -d
6.1.15	hirBufferStatusPrefetchBufferShortages	Number of times prefetch buffer shortage occurred	Gauge	read-only	pdbufls -k sts -d
6.1.16	hirBufferStatusPrefetchHitRate	Prefetch hit rate (%)	INTEGER	read-only	pdbufls -k sts -d
6.1.17	hirBufferStatusPrefetchInputPages	Number of prefetch input pages	Gauge	read-only	pdbufls -k sts -d
6.1.18	hirBufferStatusPrefetchReadRequests	Number of prefetch read requests	Gauge	read-only	pdbufls -k sts -d
6.1.19	hirBufferStatusReads	Number of real read requests from disk	Gauge	read-only	pdbufls -k sts -d

ID	Object	Explanation	Type	Privilege	Data source
6.1.20	hirBufferStatusReferenceBufferFlushes	Number of reference buffer flushes	Gauge	read-only	pdbufls -k sts -d
6.1.21	hirBufferStatusReferenceGets	Number of reference GETs	Gauge	read-only	pdbufls -k sts -d
6.1.22	hirBufferStatusReferenceHitRate	Reference request hit rate	INTEGER	read-only	pdbufls -k sts -d
6.1.23	hirBufferStatusServerName	Server name	DisplayString	read-only	pdbufls -k sts -d
6.1.24	hirBufferStatusUpdateBufferFlushes	Number of update buffer flushes	Gauge	read-only	pdbufls -k sts -d
6.1.25	hirBufferStatusUpdateGets	Number of update GETs	Gauge	read-only	pdbufls -k sts -d
6.1.26	hirBufferStatusUpdateHitRate	Update request hit rate (%)	INTEGER	read-only	pdbufls -k sts -d
6.1.27	hirBufferStatusUpdatedBufferTrigger	Number of update buffers that become output triggers during deferred write triggers	INTEGER	read-only	pdbufls -k sts -d
6.1.28	hirBufferStatusWaits	Number of times buffer lock-release wait occurred	Gauge	read-only	pdbufls -k sts -d
6.1.29	hirBufferStatusWrites	Number of real write operations to disk	Gauge	read-only	pdbufls -k sts -d
6.1.30	hirBufferStatusDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

26.10 HiRDB file system area (RDAREAs) table (hirRdareaFileTable)

The HiRDB file system area (RDAREAs) table stores performance information collected over a given time period on a HiRDB file system area for RDAREAs. One line is created for each combination of a HiRDB file system area and an RDAREA.

Notes on collecting performance information

- Performance information is collected while HiRDB statistical information is being collected.

  Trigger for starting statistical information collection

  • When the `pdstbegin` command is executed with `fil` specified for the statistical information type (`-k` option) while HiRDB is running

  • When HiRDB is started by the `pdstbegin` operand with `fil` specified for the statistical information type (`-k` option)

  Trigger for terminating statistical information collection

  • When the `pdstend` command is executed while HiRDB is running

  • When HiRDB is terminated
- For a shared RDAREA, performance information is collected for each server on a separate line.
- Performance information cannot be collected on a unit that is stopped.

Table 26-8 shows the configuration of the HiRDB file system area (RDAREAs) table.

Table 26-8: Configuration of the HiRDB file system area (RDAREAs) table

ID	Object	Explanation	Type	Privilege	Data source
9.	`hirRdareaFileTable`	HiRDB file system area (RDAREAs) table	`SEQUENCE`	not-access	—
9.1	`hirRdareaFileEntry`	HiRDB file system area entry	`SEQUENCE`	not-access	—
9.1.1	`hirRdareaFileSysIndex`	HiRDB system identifier index	`INTEGER`	read-only	MIB command
9.1.2	`hirRdareaFileServerIndex`	Server identifier index	`INTEGER`	read-only	MIB command
9.1.3	`hirRdareaFileFileIndex`	HiRDB file identifier index	`INTEGER`	read-only	MIB command

ID	Object	Explanation	Type	Privilege	Data source
9.1.4	hirRdareaFileRdareaIndex	RDAREA identifier index	INTEGER	read-only	MIB command
9.1.5	hirRdareaFileInstance	Instance name *HiRDB-identifier*:*server-name*:*HiRDB-file-system-area-name*:*RDAREA*	DisplayString	read-only	MIB command
9.1.6	hirRdareaFileHirID	HiRDB identifier	DisplayString	read-only	MIB command
9.1.7	hirRdareaFileAIORead	System-specific information (AIO READ)	INTEGER	read-only	pdstedit (fil)[1]
9.1.8	hirRdareaFileAIOWrite	System-specific information (AIO WRITE)	INTEGER	read-only	pdstedit (fil)[1]
9.1.9	hirRdareaFileClose	Number of closing operations that occurred (CLOSE)	INTEGER	read-only	pdstedit (fil)[1]
9.1.10	hirRdareaFileFreeMBytes	Size of unused area in the user area (area not allocated as HiRDB files)(KB)	INTEGER	read-only	pdfstatfs FS_NAME
9.1.11	hirRdareaFileHiRDBFileSystemAreaName	HiRDB file system area name (full path) (FS NAME)	DisplayString	read-only	pdstedit (fil)[2]
9.1.12	hirRdareaFileHiRDBFileSystemAreaType	HiRDB file system area type (DB, DB(NOLOB), SVR)	DisplayString	read-only	pdfstatfs FS_NAME
9.1.13	hirRdareaFileHost	Host name (HOST)	DisplayString	read-only	pdstedit (fil)
9.1.14	hirRdareaFileIOError	Number of input/output errors (IO ERROR)	INTEGER	read-only	pdstedit (fil)[1]
9.1.15	hirRdareaFileListIO	System-specific information (LIST IO)	INTEGER	read-only	pdstedit (fil)[1]
9.1.16	hirRdareaFileMbytes	Total size of user areas in the HiRDB file system area (MB)	INTEGER	read-only	pdfstatfs FS_NAME
9.1.17	hirRdareaFileOpen	Number of opening operations that occurred (OPEN)	INTEGER	read-only	pdstedit (fil)[1]

ID	Object	Explanation	Type	Privilege	Data source
9.1.18	hirRdareaFileRDAREAName	RDAREA name (RDAREA NAME)	DisplayString	read-only	pdstedit (fil)
9.1.19	hirRdareaFileServerName	Server name (SERVER NAME)	DisplayString	read-only	pdstedit (fil)
9.1.20	hirRdareaFileSyncRead	Number of synchronous read operations that occurred (SYNC READ)	INTEGER	read-only	pdstedit (fil)[1]
9.1.21	hirRdareaFileSyncWrite	Number of synchronous write operations that occurred (SYNC WRITE)	INTEGER	read-only	pdstedit (fil)[1]
9.1.22	hirRdareaFileDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

Note: Information in parentheses following pdstedit is the edit item in the statistics analysis utility.

[1] Operations are separated into groups based on the combination of RDAREA NAME and HiRDB file system area name obtained by removing the HiRDB file name from HOST, SERVER, and FILE NAME in fil_DAT in order to compute the total of the numeric value data in the fields. Overflow is indicated when ********** is output instead of numeric data, and 0 is set.

[2] Created by removing the HiRDB file name from the full-path HiRDB file name. For example, /users/hirdb_s/area/rdsys04 is obtained from /users/hirdb_s/area/rdsys04/rddata01.

26.11 SYS statistics table (hirStatisInfSysTable)

The SYS statistics table collects statistical information related to system operations for servers. One line is created for each server.

Notes on collecting performance information

- Performance information is collected while HiRDB statistical information is being collected.

  Trigger for starting statistical information collection

  • When the `pdstbegin` command is executed with `sys` specified for the statistical information type (`-k` option) while HiRDB is running

  • When HiRDB is started by the `pdstbegin` operand with `sys` specified for the statistical information type (`-k` option)

  Trigger for terminating statistical information collection

  • When the `pdstend` command is executed while HiRDB is running

  • When HiRDB is terminated
- Performance information cannot be collected on a unit that is stopped.
- Set the collection interval to be the same as the interval specified in the `-m` option of the `pdstbegin` command or the `pdstbegin` operand.

Table 26-9 shows the configuration of the SYS statistics table.

Table 26-9: Configuration of the SYS statistics table

ID	Object	Explanation	Type	Privilege	Data source
11.	`hirStatisInfSysTable`	SYS statistics table	`SEQUENCE`	not-access	—
11.1	`hirStatisInfSysEntry`	SYS statistical information entry	`SEQUENCE`	not-access	—
11.1.1	`hirStatisInfSysSysIndex`	HiRDB system identifier index	`INTEGER`	read-only	MIB command
11.1.2	`hirStatisInfSysHostIndex`	HiRDB host identifier index	`INTEGER`	read-only	MIB command
11.1.3	`hirStatisInfSysUnitIndex`	HiRDB unit identifier index	`INTEGER`	read-only	MIB command

ID	Object	Explanation	Type	Privilege	Data source
11.1.4	hirStatisInfSysServerIndex	HiRDB server identifier index	INTEGER	read-only	MIB command
11.1.5	hirStatisInfSysInstance	Instance name *HiRDB-identifier*:*host-name*:*server-name*	DisplayString	read-only	MIB command
11.1.6	hirStatisInfSysHirID	HiRDB identifier	DisplayString	read-only	MIB command
11.1.7	hirStatisInfSysHost	Host name (HOST)	DisplayString	read-only	pdstedit (sys)
11.1.8	hirStatisInfSysServerName	Server name (SERVER_NAME)	DisplayString	read-only	pdstedit (sys)
11.1.9	hirStatisInfSysScheduleQueueLenFreq	Number of scheduling queues that occurred (SCHEDULE_QUEUE_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.10	hirStatisInfSysScheduleQueueLenMax	Maximum scheduling queue size (SCHEDULE_QUEUE_LEN_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.11	hirStatisInfSysScheduleQueueLenMin	Minimum scheduling queue size (SCHEDULE_QUEUE_LEN_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.12	hirStatisInfSysScheduleQueueLenAvg	Average scheduling queue size (SCHEDULE_QUEUE_LEN_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.13	hirStatisInfSysMessageLenFreq	Number of schedule messages issued (MESSAGE_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.14	hirStatisInfSysMessageLenMax	Maximum schedule message size (MESSAGE_LEN_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.15	hirStatisInfSysMessageLenMin	Minimum schedule message size (MESSAGE_LEN_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.16	hirStatisInfSysMessageLenAvg	Average schedule message size (MESSAGE_LEN_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.17	hirStatisInfSysServerAbort	Number of times servers terminated abnormally (SERVER_ABORT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.18	hirStatisInfSysSystemServerAbort	Number of times internal servers used by HiRDB terminated abnormally (SYSTEM_SERVER_ABORT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.19	hirStatisInfSysProcessCountMax	Maximum number of processes generated (PROCESS_COUNT_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.20	hirStatisInfSysProcessCountMin	Minimum number of processes generated (PROCESS_COUNT_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.21	hirStatisInfSysProcessCountAvg	Average number of processes generated (PROCESS_COUNT_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.22	hirStatisInfSysCommit	Number of commits (COMMIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.23	hirStatisInfSysRollback	Number of rollbacks (ROLLBACK)[3]	INTEGER	read-only	pdstedit (sys)
11.1.24	hirStatisInfSysLockWaitTimeFreq	Number of lock-release waits that occurred (LOCK_WAIT_TIME_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.25	hirStatisInfSysLockWaitTimeMax	Maximum lock-release wait time (LOCK_WAIT_TIME_MAX)[1] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.26	hirStatisInfSysLockWaitTimeMin	Minimum lock-release wait time (LOCK_WAIT_TIME_MIN)[2] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.27	hirStatisInfSysLockWaitTimeAvg	Average lock-release wait time (LOCK_WAIT_TIME_AVG)[3] (milliseconds)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.28	hirStatisInfSysLockQueueLenFreq	Number of users who had to wait for lock release (LOCK_QUEUE_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.29	hirStatisInfSysLockQueueLenMax	Maximum number of users who had to wait for lock release (LOCK_QUEUE_LEN_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.30	hirStatisInfSysLockQueueLenMin	Minimum number of users who had to wait for lock release (LOCK_QUEUE_LEN_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.31	hirStatisInfSysLockQueueLenAvg	Average number of users who had to wait for lock release (LOCK_QUEUE_LEN_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.32	hirStatisInfSysDeadlock	Number of deadlocks (DEADLOCK)[3]	INTEGER	read-only	pdstedit (sys)
11.1.33	hirStatisInfSysUseLockTableFreq	Number of phenomena that increased the usage rate of the table for managing locked resources by at least 5% (USE_LOCK_TABLE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.34	hirStatisInfSysUseLockTableMax	Maximum usage rate of the table for managing locked resources (USE_LOCK_TABLE_MAX)[1] (%)	INTEGER	read-only	pdstedit (sys)
11.1.35	hirStatisInfSysUseLockTableMin	Minimum usage rate of the table for managing locked resources (USE_LOCK_TABLE_MIN)[2] (%)	INTEGER	read-only	pdstedit (sys)
11.1.36	hirStatisInfSysUseLockTableAvg	Average usage rate of the table for managing locked resources (USE_LOCK_TABLE_AVG)[3] (%)	INTEGER	read-only	pdstedit (sys)
11.1.37	hirStatisInfSysSegmentSizeAvg	Average size of shared memory allocated to servers and internal servers used by HiRDB (SEGMENT_SIZE_AVG)[3] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.38	hirStatisInfSysStaticSizeAvg	Average size of the allocated static shared memory (STATIC_SIZE_AVG)[3] (KB)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.39	hirStatisInfSysDyanamicSizeAvg	Average size of the allocated dynamic shared memory (DYNAMIC_SIZE_AVG)[3] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.40	hirStatisInfSysSizeForBufferAvg	Average size of the allocated shared memory for global buffers (SIZE_FOR_BUFFER_AVG)[3] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.41	hirStatisInfSysSyncGetIntervalTimeFreq	Number of synchronization point dumps collected (SYNC_GET_INTERVAL_TIME_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.42	hirStatisInfSysSyncGetIntervalTimeMax	Maximum interval for collecting a synchronization point dump (SYNC_GET_INTERVAL_TIME_MAX)[1] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.43	hirStatisInfSysSyncGetIntervalTimeMin	Minimum interval for collecting a synchronization point dump (SYNC_GET_INTERVAL_TIME_MIN)[2] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.44	hirStatisInfSysSyncGetIntervalTimeAvg	Average interval for collecting a synchronization point dump (SYNC_GET_INTERVAL_TIME_AVG)[3] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.45	hirStatisInfSysSyncGetTimeFreq	Number of synchronization point dumps collected (SYNC_GET_TIME_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.46	hirStatisInfSysSyncGetTimeMax	Maximum time required for collecting a synchronization point dump (SYNC_GET_TIME_MAX)[1] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.47	hirStatisInfSysSyncGetTimeMin	Minimum time required for collecting a synchronization point dump (SYNC_GET_TIME_MIN)[2] (milliseconds)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.48	hirStatisInfSysSyncGetTimeAvg	Average time required for collecting a synchronization point dump (SYNC_GET_TIME_AVG)[3] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.49	hirStatisInfSysLogBufferFull	Number of times the log buffer was full (LOG_BUFFER_FULL)[3]	INTEGER	read-only	pdstedit (sys)
11.1.50	hirStatisInfSysLogWaitThread	Number of times log input/output had to wait because no current buffer was available (LOG_WAIT_THREAD)[3]	INTEGER	read-only	pdstedit (sys)
11.1.51	hirStatisInfSysLogOutputBlockLenFreq	Number of times log block was output (LOG_OUTPUT_BLOCK_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.52	hirStatisInfSysLogOutputBlockLenMax	Maximum log output block size (LOG_OUTPUT_BLOCK_LEN_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.53	hirStatisInfSysLogOutputBlockLenMin	Management log output block size (LOG_OUTPUT_BLOCK_LEN_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.54	hirStatisInfSysLogOutputBlockLenAvg	Average log output block size (LOG_OUTPUT_BLOCK_LEN_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.55	hirStatisInfSysLogNotBusLenFreq	Number of times log was output in non-bus area (LOG_NOT_BUS_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.56	hirStatisInfSysLogNotBusLenMax	Maximum log block size in non-bus area (LOG_NOT_BUS_LEN_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.57	hirStatisInfSysLogNotBusLenMin	Minimum log block size in non-bus area (LOG_NOT_BUS_LEN_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.58	hirStatisInfSysLogNotBusLenAvg	Average log block size in non-bus area (LOG_NOT_BUS_LEN_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.59	hirStatisInfSysLogWaitBufferForIOFreq	Number of cases in which the system had to wait for log input/output completion (LOG_WAIT_BUFFER_FOR_IO_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.60	hirStatisInfSysLogWaitBufferForIOMax	Maximum number of buffer sectors waiting for log input/output (LOG_WAIT_BUFFER_FOR_IO_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.61	hirStatisInfSysLogWaitBufferForIOMin	Minimum number of buffer sectors waiting for log input/output (LOG_WAIT_BUFFER_FOR_IO_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.62	hirStatisInfSysLogWaitBufferForIOAvg	Average number of buffer sectors waiting for log input/output (LOG_WAIT_BUFFER_FOR_IO_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.63	hirStatisInfSysLogWriteToFile	Number of times data was written into system log file (LOG_WRITE_TO_FILE)[3]	INTEGER	read-only	pdstedit (sys)
11.1.64	hirStatisInfSysLogWriteError	Number of log write errors (LOG_WRITE_ERROR)[3]	INTEGER	read-only	pdstedit (sys)
11.1.65	hirStatisInfSysLogFileSwapTimeFreq	Number of times log files were swapped (LOG_FILE_SWAP_TIME_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.66	hirStatisInfSysLogFileSwapTimeMax	Maximum time for log file swapping (LOG_FILE_SWAP_TIME_MAX)[1] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.67	hirStatisInfSysLogFileSwapTimeMin	Minimum time for log file swapping (LOG_FILE_SWAP_TIME_MIN)[2] (milliseconds)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.68	hirStatisInfSysLogFileSwapTimeAvg	Average time for log file swapping (LOG_FILE_SWAP_TIME_AVG)[3] (milliseconds)	INTEGER	read-only	pdstedit (sys)
11.1.69	hirStatisInfSysLogInputDataFreq	Number of logs that were input during rollback (LOG_INPUT_DATA_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.70	hirStatisInfSysLogInputDataMax	Maximum size of log input data during rollback (LOG_INPUT_DATA_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.71	hirStatisInfSysLogInputDataMin	Minimum size of log input data during rollback (LOG_INPUT_DATA_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.72	hirStatisInfSysLogInputDataAvg	Average size of log input data during rollback (LOG_INPUT_DATA_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.73	hirStatisInfSysLogReadFromFile	Number of times logs were read during rollback (LOG_READ_FROM_FILE)[3]	INTEGER	read-only	pdstedit (sys)
11.1.74	hirStatisInfSysLogReadError	Number of log read errors (LOG_READ_ERROR)[3]	INTEGER	read-only	pdstedit (sys)
11.1.75	hirStatisInfSysDicTblDEFGetReq	Number of requests for collecting table definition information (DIC_TBL-DEF_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.76	hirStatisInfSysDicTableCacheHit	Number of hits for the buffer for table definition information (DIC_TABLE_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.77	hirStatisInfSysDicCachedTblDEFFreq	Number of definition information items in the buffer for table definition information (DIC_CACHED_TBL-DEF_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.78	hirStatisInfSysDicCachedTblDEFMax	Maximum number of definition information items in the buffer for table definition (DIC_CACHED_TBL-DEF_MAX)[1]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.79	hirStatisInfSysDicCachedTblDEFMin	Minimum number of definition information items in the buffer for table definition (DIC_CACHED_TBL-DEF_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.80	hirStatisInfSysDicCachedTblDEFAvg	Average number of definition information items in the buffer for table definition (DIC_CACHED_TBL-DEF_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.81	hirStatisInfSysDicUseTblDEFSizeFreq	Number of definition information items collected in the buffer for table definition information (DIC_USE_TBL-DEF_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.82	hirStatisInfSysDicUseTblDEFSizeMax	Maximum size of the area used in the buffer for table definition information per item of table definition information collected in the buffer for table definition information (DIC_USE_TBL-DEF_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.83	hirStatisInfSysDicUseTblDEFSizeMin	Minimum size of the area used in the buffer for table definition information per item of table definition information collected in the buffer for table definition information (DIC_USE_TBL-DEF_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.84	hirStatisInfSysDicUseTblDEFSizeAvg	Average size of the area used in the buffer for table definition information per item of table definition information collected in the buffer for table definition information (DIC_USE_TBL-DEF_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.85	hirStatisInfSysDicCacheTblDEFSizeFreq	Number of table definition information items in the buffer for table definition information (DIC_CACHED_TBL-DEF_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.86	hirStatisInfSysDicCacheTblDEFSizeMax	Maximum size of the area used in the buffer for table definition information (DIC_CACHED_TBL-DEF_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.87	hirStatisInfSysDicCacheTblDEFSizeMin	Minimum size of the area used in the buffer for table definition information (DIC_CACHED_TBL-DEF_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.88	hirStatisInfSysDicCacheTblDEFSizeAvg	Average size of the area used in the buffer for table definition information (DIC_CACHED_TBL-DEF_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.89	hirStatisInfSysDicAccessPrivCheck	Number of times table access privilege information was collected (DIC_ACCESS_PRIV_CHECK)[3]	INTEGER	read-only	pdstedit (sys)
11.1.90	hirStatisInfSysDicAccessCacheHit	Number of hits for the buffer for table access privilege information (DIC_ACCESS_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.91	hirStatisInfSysDicConDbaDefGetReq	Number of requests for collecting user privilege information (DIC_CON/DBA_DEF_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.92	hirStatisInfSysDicConDbaCacheHit	Number of hits for the buffer for user privilege information (DIC_CON/DBA_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.93	hirStatisInfSysDicConDbaCachedUserFreq	Number of users using the buffer for user privilege information (DIC_CON/DBA_CACHED_USER_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.94	hirStatisInfSysDicConDbaCachedUserMax	Maximum number of users using the buffer for user privilege information (DIC_CON/DBA_CACHED_USER_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.95	hirStatisInfSysDicConDbaCachedUserMin	Minimum number of users using the buffer for user privilege information (DIC_CON/DBA_CACHED_USER_MIN)[2]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.96	hirStatisInfSysDicConDbaCachedUserAvg	Average number of users using the buffer for user privilege information (DIC_CON/DBA_CACHED_USER_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.97	hirStatisInfSysDicTransDataLenFreq	Number of requests for collecting table definition information (DIC_TRANS_DATA_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.98	hirStatisInfSysDicTransDataLenMax	Maximum size of communications with the dictionary server (DIC_TRANS_DATA_LEN_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.99	hirStatisInfSysDicTransDataLenMin	Minimum size of communications with the dictionary server (DIC_TRANS_DATA_LEN_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.100	hirStatisInfSysDicTransDataLenAvg	Average size of communications with the dictionary server (DIC_TRANS_DATA_LEN_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.101	hirStatisInfSysDicTrans	Number of times communication occurred with the dictionary server (DIC_TRANS)[3]	INTEGER	read-only	pdstedit (sys)
11.1.102	hirStatisInfSysDicViewDefGetReq	Number of requests for collecting view analysis information (DIC_VIEW_DEF_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.103	hirStatisInfSysDicViewDefCacheHit	Number of hits for the buffer for view analysis information (DIC_VIEW_DEF_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.104	hirStatisInfSysDicViewCachedDef	Number of analysis information items in the buffer for view analysis information (DIC_VIEW_CACHED_DEF)[3]	INTEGER	read-only	pdstedit (sys)
11.1.105	hirStatisInfSysDicUsedViewSizeFreq	Number of analysis information items collected in the buffer for view analysis information (DIC_USED_VIEW_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.106	hirStatisInfSysDicUsedViewSizeMax	Maximum size of the area used in the buffer for view analysis information per view collected in the buffer for view analysis information (DIC_USED_VIEW_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.107	hirStatisInfSysDicUsedViewSizeMin	Minimum size of the area used in the buffer for view analysis information per view collected in the buffer for view analysis information (DIC_USED_VIEW_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.108	hirStatisInfSysDicUsedViewSizeAvg	Average size of the area used in the buffer for view analysis information per view collected in the buffer for view analysis information (DIC_USED_VIEW_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.109	hirStatisInfSysDicViewCacheSizeFreq	Number of view analysis information items in the buffer for view analysis information (DIC_VIEW_CACHE_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.110	hirStatisInfSysDicViewCacheSizeMax	Maximum size of the buffer for view analysis information used (DIC_VIEW_CACHE_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.111	hirStatisInfSysDicViewCacheSizeMin	Minimum size of the buffer for view analysis information used (DIC_VIEW_CACHE_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.112	hirStatisInfSysDicViewCacheSizeAvg	Average size of the buffer for view analysis information used (DIC_VIEW_CACHE_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.113	hirStatisInfSysDicCacheMissViewSizeFreq	Number of times the buffer for view analysis information was missed (DIC_CACHE_MISS_VIEW_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.114	hirStatisInfSysDicCacheMissViewSizeMax	Maximum size of the view analysis information that caused a buffer miss (DIC_CACHE_MISS_VIEW_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.115	hirStatisInfSysDicCacheMissViewSizeMin	Minimum size of the view analysis information that caused a buffer miss (DIC_CACHE_MISS_VIEW_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.116	hirStatisInfSysDicCacheMissViewSizeAvg	Average size of the view analysis information that caused a buffer miss (DIC_CACHE_MISS_VIEW_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.117	hirStatisInfSysSQLOBJGetReq	Number of requests for collecting SQL objects (SQLOBJ_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.118	hirStatisInfSysSQLOBJCacheHit	Number of hits for the SQL object buffer (SQLOBJ_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.119	hirStatisInfSysCachedSQLOBJFreq	Number of SQL objects in the SQL object buffer (CACHED_SQLOBJ_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.120	hirStatisInfSysCachedSQLOBJMax	Maximum number of SQL objects in the SQL object buffer (CACHED_SQLOBJ_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.121	hirStatisInfSysCachedSQLOBJMin	Minimum number of SQL objects in the SQL object buffer (CACHED_SQLOBJ_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.122	hirStatisInfSysCachedSQLOBJAvg	Average number of SQL objects in the SQL object buffer (CACHED_SQLOBJ_AVG)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.123	hirStatisInfSysCachedSQLOBJSizeFreq	Number of SQL objects in the SQL object buffer (CACHED_SQLOBJ_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.124	hirStatisInfSysCachedSQLOBJSizeMax	Maximum value of the combined total size of objects in the SQL object buffer (CACHED_SQLOBJ_SIZE_MAX)[1] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.125	hirStatisInfSysCachedSQLOBJSizeMin	Minimum value of the combined total size of objects in the SQL object buffer (CACHED_SQLOBJ_SIZE_MIN)[2] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.126	hirStatisInfSysCachedSQLOBJSizeAvg	Average value of the combined total size of objects in the SQL object buffer (CACHED_SQLOBJ_SIZE_AVG)[3] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.127	hirStatisInfSysSwapOutSQLOBJ	Number of SQL objects taken out of the SQL object buffer (SWAP_OUT_SQLOBJ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.128	hirStatisInfSysSQLOBJLenFreq	Number of SQL objects in the SQL object buffer (SQLOBJ_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.129	hirStatisInfSysSQLOBJLenMax	Maximum size of SQL objects in the SQL object buffer (SQLOBJ_LEN_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.130	hirStatisInfSysSQLOBJLenMin	Minimum size of SQL objects in the SQL object buffer (SQLOBJ_LEN_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.131	hirStatisInfSysSQLOBJLenAvg	Average size of SQL objects in the SQL object buffer (SQLOBJ_LEN_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.132	hirStatisInfSysSTROBJGetReq	Number of requests to collect stored procedure objects (STROBJ_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.133	hirStatisInfSysSTROBJCacheHit	Number of SQL object buffer hits by stored procedure objects (STROBJ_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.134	hirStatisInfSysCachedSTROBJFreq	Number of stored procedure objects in the SQL object buffer (CACHED_STROBJ_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.135	hirStatisInfSysCachedSTROBJMax	Maximum number of stored procedure objects in the SQL object buffer (CACHED_STROBJ_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.136	hirStatisInfSysCachedSTROBJMin	Minimum number of stored procedure objects in the SQL object buffer (CACHED_STROBJ_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.137	hirStatisInfSysCachedSTROBJAvg	Average number of stored procedure objects in the SQL object buffer (CACHED_STROBJ_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.138	hirStatisInfSysCachedSTROBJSizeFreq	Number of stored procedure objects in the SQL object buffer (CACHED_STROBJ_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.139	hirStatisInfSysCachedSTROBJSizeMax	Maximum value of the combined total size of stored procedure objects in the SQL object buffer (CACHED_STROBJ_SIZE_MAX)[1] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.140	hirStatisInfSysCachedSTROBJSizeMin	Minimum value of the combined total size of stored procedure objects in the SQL object buffer (CACHED_STROBJ_SIZE_MIN)[2] (KB)	INTEGER	read-only	pdstedit (sys)
11.1.141	hirStatisInfSysCachedSTROBJSizeAvg	Average value of the combined total size of stored procedure objects in the SQL object buffer (CACHED_STROBJ_SIZE_AVG)[3] (KB)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.142	hirStatisInfSysSwapOutSTROBJ	Number of stored procedure objects taken out of the SQL object buffer (SWAP_OUT_STROBJ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.143	hirStatisInfSysSTROBJLenFreq	Number of stored procedure objects in the SQL object buffer (STROBJ_LEN_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.144	hirStatisInfSysSTROBJLenMax	Maximum size of stored procedure objects in the SQL object buffer (STROBJ_LEN_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.145	hirStatisInfSysSTROBJLenMin	Minimum size of stored procedure objects in the SQL object buffer (STROBJ_LEN_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.146	hirStatisInfSysSTROBJLenAvg	Average size of stored procedure objects in the SQL object buffer (STROBJ_LEN_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.147	hirStatisInfSysSTROBJRecompile	Number of times stored procedure objects were recompiled (STROBJ_RECOMPILE)[3]	INTEGER	read-only	pdstedit (sys)
11.1.148	hirStatisInfSysProcessServiceCountMax	Maximum number of server processes that are executing service (PROCESS_SERVICE_COUNT_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.149	hirStatisInfSysProcessServiceCountMin	Minimum number of server processes that are executing service (PROCESS_SERVICE_COUNT_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.150	hirStatisInfSysProcessServiceCountAvg	Average number of server processes that are executing service (PROCESS_SERVICE_COUNT_AVG)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.151	hirStatisInfSysProcessRequestOver	Number of service requests exceeding the maximum number of processes that can be activated (PROCESS_REQUEST_OVER)[3]	INTEGER	read-only	pdstedit (sys)
11.1.152	hirStatisInfSysResponseOnOwnUnitFreq	Number of service responses to the local unit server (RESPONSE_ON_OWN_UNIT_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.153	hirStatisInfSysResponseOnOwnUnitMax	Maximum service response time to the local unit server (RESPONSE_ON_OWN_UNIT_MAX)[1] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.154	hirStatisInfSysResponseOnOwnUnitMin	Minimum service response time to the local unit server (RESPONSE_ON_OWN_UNIT_MIN)[2] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.155	hirStatisInfSysResponseOnOwnUnitAvg	Average service response time to the local unit server (RESPONSE_ON_OWN_UNIT_AVG)[3] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.156	hirStatisInfSysResponseToOtherUnitFreq	Number of service responses to other unit servers (RESPONSE_TO_OTHER_UNIT_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.157	hirStatisInfSysResponseToOtherUnitMax	Maximum service response time to other unit servers (RESPONSE_TO_OTHER_UNIT_MAX)[1] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.158	hirStatisInfSysResponseToOtherUnitMin	Minimum service response time to other unit servers (RESPONSE_TO_OTHER_UNIT_MIN)[2] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.159	hirStatisInfSysResponseToOtherUnitAvg	Average service response time to other unit servers (RESPONSE_TO_OTHER_UNIT_AVG)[3] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.160	hirStatisInfSysExecTimeOnOwnUnitFreq	Number of executions per service from the local unit server (EXEC_TIME_ON_OWN_UNIT_FREQ)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.161	hirStatisInfSysExecTimeOnOwnUnitMax	Maximum execution time per service from the local unit server (EXEC_TIME_ON_OWN_UNIT_MAX)[1] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.162	hirStatisInfSysExecTimeOnOwnUnitMin	Minimum execution time per service from the local unit server (EXEC_TIME_ON_OWN_UNIT_MIN)[2] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.163	hirStatisInfSysExecTimeOnOwnUnitAvg	Average execution time per service from the local unit server (EXEC_TIME_ON_OWN_UNIT_AVG)[3] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.164	hirStatisInfSysExecTimeFromOtherUnitFreq	Number of executions per service from other unit servers (EXEC_TIME_FROM_OTHER_UNIT_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.165	hirStatisInfSysExecTimeFromOtherUnitMax	Maximum execution time per service from other unit servers (EXEC_TIME_FROM_OTHER_UNIT_MAX)[1] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.166	hirStatisInfSysExecTimeFromOtherUnitMin	Minimum execution time per service from other unit servers (EXEC_TIME_FROM_OTHER_UNIT_MIN)[2] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.167	hirStatisInfSysExecTimeFromOtherUnitAvg	Average execution time per service from other unit servers (EXEC_TIME_FROM_OTHER_UNIT_AVG)[3] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.168	hirStatisInfSysSendToOwnProcs	Number of send operations to the local process (SEND_TO_OWN_PRCS)[3]	INTEGER	read-only	pdstedit (sys)
11.1.169	hirStatisInfSysSendToOtherPrcs	Number of send operations to other processes within the local unit (SEND_TO_OTHER_PRCS)[3]	INTEGER	read-only	pdstedit (sys)
11.1.170	hirStatisInfSysSendToOtherUnit	Number of send operations to other units (SEND_TO_OTHER_UNIT)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.171	hirStatisInfSysReceiveFromOwnPrcs	Number of receive operations from the local process (RECEIVE_FROM_OWN_PRCS)[3]	INTEGER	read-only	pdstedit (sys)
11.1.172	hirStatisInfSysReceiveFromOtherPrcs	Number of receive operations from other processes within the local unit (RECEIVE_FROM_OTHER_PRCS)[3]	INTEGER	read-only	pdstedit (sys)
11.1.173	hirStatisInfSysReceiveFromOtherUnit	Number of receive operations from other units (RECEIVE_FROM_OTHER_UNIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.174	hirStatisInfSysTypeDEFGetReq	Number of requests for collecting type information (TYPE-DEF_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.175	hirStatisInfSysTypeDEFCacheHit	Number of hits for the user-defined type information buffer (TYPE-DEF_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.176	hirStatisInfSysCachedTypeDEFFreq	Number of type definition information items in the user-defined type information buffer (CACHED_TYPE-DEF_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.177	hirStatisInfSysCachedTypeDEFMax	Maximum number of type definition information items in the user-defined type information buffer (CACHED_TYPE-DEF_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.178	hirStatisInfSysCachedTypeDEFMin	Minimum number of type definition information items in the user-defined type information buffer (CACHED_TYPE-DEF_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.179	hirStatisInfSysCachedTypeDEFAvg	Average number of type definition information items in the user-defined type information buffer (CACHED_TYPE-DEF_AVG)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.180	hirStatisInfSysTypeDEFCacheSizeFreq	Number of type definition information items in the user-defined type information buffer (TYPE-DEF_CACHE_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.181	hirStatisInfSysTypeDEFCacheSizeMax	Maximum size of the buffer area used for one type definition information item collected in the user-defined type information buffer (TYPE-DEF_CACHE_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.182	hirStatisInfSysTypeDEFCacheSizeMin	Minimum size of the buffer area used for one type definition information item collected in the user-defined type information buffer (TYPE-DEF_CACHE_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.183	hirStatisInfSysTypeDEFCacheSizeAvg	Average size of the buffer area used for one type definition information item collected in the user-defined type information buffer (TYPE-DEF_CACHE_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.184	hirStatisInfSysTypeDEFCacheTotalSizeFreq	Number of user-defined type information items collected (TYPE-DEF_CACHE_TOTAL_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.185	hirStatisInfSysTypeDEFCacheTotalSizeMax	Maximum size of the total area used for user-defined type information buffers (TYPE-DEF_CACHE_TOTAL_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.186	hirStatisInfSysTypeDEFCacheTotalSizeMin	Minimum size of the total area used for user-defined type information buffers (TYPE-DEF_CACHE_TOTAL_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.187	hirStatisInfSysTypeDEFCacheTotalSizeAvg	Average size of the total area used for user-defined type information buffers (TYPE-DEF_CACHE_TOTAL_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.188	hirStatisInfSysTypeDEFCacheAllocSizeFreq	Number of user-defined type information buffers allocated (TYPE-DEF_CACHE_ALLOC_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.189	hirStatisInfSysTypeDEFCacheAllocSizeMax	Maximum size of the user-defined type information buffers allocated (TYPE-DEF_CACHE_ALLOC_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.190	hirStatisInfSysTypeDEFCacheAllocSizeMin	Minimum size of the user-defined type information buffers allocated (TYPE-DEF_CACHE_ALLOC_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.191	hirStatisInfSysTypeDEFCacheAllocSizeAvg	Average size of the user-defined type information buffers allocated (TYPE-DEF_CACHE_ALLOC_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.192	hirStatisInfSysRtnDEFGetReq	Number of requests for collecting routine definition information (RTN-DEF_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.193	hirStatisInfSysRtnDEFCacheHit	Number of hits for the routine definition information buffer (RTN-DEF_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.194	hirStatisInfSysCachedRtnDEFFreq	Number of routine definition information items in the routine definition information buffer (CACHED_RTN-DEF_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.195	hirStatisInfSysCachedRtnDEFMax	Maximum number of routine definition information items in the routine definition information buffer (CACHED_RTN-DEF_MAX)[1]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.196	hirStatisInfSysCachedRtnDEFMin	Minimum number of routine definition information items in the routine definition information buffer (CACHED_RTN-DEF_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.197	hirStatisInfSysCachedRtnDEFAvg	Average number of routine definition information items in the routine definition information buffer (CACHED_RTN-DEF_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.198	hirStatisInfSysRtnDEFCacheSizeFreq	Number of routine definition information items in the routine definition information buffer (RTN-DEF_CACHE_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.199	hirStatisInfSysRtnDEFCacheSizeMax	Maximum size of the buffer area used for one routine definition information item collected in the routine definition information buffer (RTN-DEF_CACHE_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.200	hirStatisInfSysRtnDEFCacheSizeMin	Minimum size of the buffer area used for one routine definition information item collected in the routine definition information buffer (RTN-DEF_CACHE_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.201	hirStatisInfSysRtnDEFCacheSizeAvg	Average size of the buffer area used for one routine definition information item collected in the routine definition information buffer (RTN-DEF_CACHE_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.202	hirStatisInfSysRtnDEFCacheTotalSizeFreq	Number of routine definition information items in the routine definition information buffer (RTN-DEF_CACHE_TOTAL_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.203	hirStatisInfSysRtnDEFCacheTotalSizeMax	Maximum size of the total area used for routine definition information buffers (RTN-DEF_CACHE_TOTAL_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.204	hirStatisInfSysRtnDEFCacheTotalSizeMin	Minimum size of the total area used for routine definition information buffers (RTN-DEF_CACHE_TOTAL_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.205	hirStatisInfSysRtnDEFCacheTotalSizeAvg	Average size of the total area used for routine definition information buffers (RTN-DEF_CACHE_TOTAL_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.206	hirStatisInfSysRtnDEFCacheAllocSizeFreq	Number of routine definition information buffers allocated (RTN-DEF_CACHE_ALLOC_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.207	hirStatisInfSysRtnDEFCacheAllocSizeMax	Maximum size of the routine definition information buffers allocated (RTN-DEF_CACHE_ALLOC_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.208	hirStatisInfSysRtnDEFCacheAllocSizeMin	Minimum size of the routine definition information buffers allocated (RTN-DEF_CACHE_ALLOC_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.209	hirStatisInfSysRtnDEFCacheAllocSizeAvg	Average size of the routine definition information buffers allocated (RTN-DEF_CACHE_ALLOC_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.210	hirStatisInfSysPlgRTNGetReq	Number of requests for collecting routine definitions of functions provided by plug-ins (PLG-RTN_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.211	hirStatisInfSysPlgRTNCacheHit	Number of hits for the buffer for routine definition information of functions provided by plug-ins (PLG-RTN_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.212	hirStatisInfSysRegistryGetReq	Number of requests for collecting registry information (REGISTRY_GET_REQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.213	hirStatisInfSysRegistryCacheHit	Number of hits for the registry information buffer (REGISTRY_CACHE_HIT)[3]	INTEGER	read-only	pdstedit (sys)
11.1.214	hirStatisInfSysCachedRegistryDEFFreq	Number of registry information items in the registry information buffer (CACHED_REGISTRY-DEF_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.215	hirStatisInfSysCachedRegistryDEFMax	Maximum number of registry information items in the registry information buffer (CACHED_REGISTRY-DEF_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.216	hirStatisInfSysCachedRegistryDEFMin	Minimum number of registry information items in the registry information buffer (CACHED_REGISTRY-DEF_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.217	hirStatisInfSysCachedRegistryDEFAvg	Average number of registry information items in the registry information buffer (CACHED_REGISTRY-DEF_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.218	hirStatisInfSysRegistryCacheSizeFreq	Number of registry information items in the registry information buffer (REGISTRY_CACHE_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.219	hirStatisInfSysRegistryCacheSizeMax	Maximum size of the registry information buffer per registry information item (REGISTRY_CACHE_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.220	hirStatisInfSysRegistryCacheSizeMin	Minimum size of the registry information buffer per registry information item (REGISTRY_CACHE_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.221	hirStatisInfSysRegistryCacheSizeAvg	Average size of the registry information buffer per registry information item (REGISTRY_CACHE_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.222	hirStatisInfSysRegistryCacheTotalSizeFreq	Number of registry information items in the registry information buffer (REGISTRY_CACHE_TOTAL_SIZE_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.223	hirStatisInfSysRegistryCacheTotalSizeMax	Maximum size of the total area used for the registry information buffer (REGISTRY_CACHE_TOTAL_SIZE_MAX)[1] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.224	hirStatisInfSysRegistryCacheTotalSizeMin	Minimum size of the total area used for the registry information buffer (REGISTRY_CACHE_TOTAL_SIZE_MIN)[2] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.225	hirStatisInfSysRegistryCacheTotalSizeAvg	Average size of the total area used for the registry information buffer (REGISTRY_CACHE_TOTAL_SIZE_AVG)[3] (bytes)	INTEGER	read-only	pdstedit (sys)
11.1.226	hirStatisInfSysRegisterdPortsFreq	Number of HiRDB reserved ports used (REGISTERED_PORTS_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.227	hirStatisInfSysRegisterdPortsMax	Maximum number of HiRDB reserved ports used (REGISTERED_PORTS_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.228	hirStatisInfSysRegisterdPortsMin	Minimum number of HiRDB reserved ports used (REGISTERED_PORTS_MIN)[2]	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.229	hirStatisInfSysRegisterdPortsAvg	Average number of HiRDB reserved ports used (REGISTERED_PORTS_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.230	hirStatisInfSysAssigndPortsFreq	Number of ports allocated automatically by the OS when the number of HiRDB reserved ports was insufficient (ASSIGNED_PORTS_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.231	hirStatisInfSysAssigndPortsMax	Maximum number of ports allocated automatically by the OS when the number of HiRDB reserved ports was insufficient (ASSIGNED_PORTS_MAX)[1]	INTEGER	read-only	pdstedit (sys)
11.1.232	hirStatisInfSysAssigndPortsMin	Minimum number of ports allocated automatically by the OS when the number of HiRDB reserved ports was insufficient (ASSIGNED_PORTS_MIN)[2]	INTEGER	read-only	pdstedit (sys)
11.1.233	hirStatisInfSysAssigndPortsAvg	Average number of ports allocated automatically by the OS when the number of HiRDB reserved ports was insufficient (ASSIGNED_PORTS_AVG)[3]	INTEGER	read-only	pdstedit (sys)
11.1.234	hirStatisInfSysDirectoryUserCheckTimeFreq	Number of requests for authenticating directory-registered users (DIRECTORY_USER_CHECK_TIME_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.235	hirStatisInfSysDirectoryUserCheckTimeMax	Maximum time for authenticating directory-registered users (DIRECTORY_USER_CHECK_TIME_MAX)[1] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.236	hirStatisInfSysDirectoryUserCheckTimeMin	Minimum time for authenticating directory-registered users (DIRECTORY_USER_CHECK_TIME_MIN)[2] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.237	hirStatisInfSysDirectoryUserCheckTimeAvg	Average time for authenticating directory-registered users (DIRECTORY_USER_CHECK_TIME_AVG)[3] (microseconds)	INTEGER	read-only	pdstedit (sys)

ID	Object	Explanation	Type	Privilege	Data source
11.1.238	hirStatisInfSysGroupCheckTimeFreq	Number of requests for checking groups (GROUP_CHECK_TIME_FREQ)[3]	INTEGER	read-only	pdstedit (sys)
11.1.239	hirStatisInfSysGroupCheckTimeMax	Maximum time for checking a group (GROUP_CHECK_TIME_MAX)[1] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.240	hirStatisInfSysGroupCheckTimeMin	Minimum time for checking a group (GROUP_CHECK_TIME_MIN)[2] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.241	hirStatisInfSysGroupCheckTimeAvg	Average time for checking a group (GROUP_CHECK_TIME_AVG)[3] (microseconds)	INTEGER	read-only	pdstedit (sys)
11.1.242	hirStatisInfSysDummy	Cannot be referenced.	INTEGER	read-only	MIB command

Legend:

—: Not applicable

Note

Information in parentheses following pdstedit is the edit item in the statistics analysis utility.

[1] If multiple items of statistical information are collected for a server, the maximum value is used.

[2] If multiple items of statistical information are collected for a server, the minimum value is used.

[3] If multiple items of statistical information are collected for a server, the average value is used.

26.12 Disk usage

This section explains disk usage when the facility for monitoring MIB performance information is used. Table 26-10 shows the disk usage for MIB tables. These areas are used for returning results to JP1/ESA. Table 26-11 shows the disk usage in other areas.

Table 26-10: Disk usage for MIB tables (bytes)

Purpose	Directory/file name	Size
Server status table	`/opt/HiRDB_S/sample/snmptmp` or `/opt/HiRDB_P/sample/snmptmp`	178 × (*unit-count* + *server-count*)
Work table HiRDB file system area table	`/opt/HiRDB_S/sample/snmptmp3` or `/opt/HiRDB_P/sample/snmptmp3`	2212 × *number-of-HiRDB-file-system-areas-for-work-tables*
RDAREA table	`/opt/HiRDB_S/sample/snmptmp4` or `/opt/HiRDB_P/sample/snmptmp4`	215 × *RDAREA-count*
RDAREA details table	`/opt/HiRDB_S/sample/snmptmp5` or `/opt/HiRDB_P/sample/snmptmp5`	502 × *RDAREA-count*
Global buffer table	`/opt/HiRDB_S/sample/snmptmp6` or `/opt/HiRDB_P/sample/snmptmp6`	351 × *global-buffer-count*
HiRDB file system area (RDAREAs) table	`/opt/HiRDB_S/sample/snmptmp9` or `/opt/HiRDB_P/sample/snmptmp9`	345 × *RDAREA-count*
SYS statistics table	`/opt/HiRDB_S/sample/snmptmp11` or `/opt/HiRDB_P/sample/snmptmp11`	2719 × *server-count*

Note

The maximum sizes are indicated.

Table 26-11: Disk usage for other areas (bytes)

Directory/file name		Purpose	Size
`$PDDIR/spool/pdmibwork` directory	`pdmibwork/sys_DAT` `pdmibwork/fil_DAT`	`pdstedit` execution result[1]	See *File size when executing the statistics analysis utility (pdstedit)* in the manual *HiRDB Installation and Design Guide*.
	`pdmibwork/new_parameter`	Final log information[1]	32
	`pdmibwork/log/stlog01` `pdmibwork/log/stlog02`	Statistics log output facility command log	Maximum of 1 MB each for `stlog01` and `stlog02`
	`pdmibwork/stjtmp/pdstj_tmp`	Statistics log temporary file[1]	Combined total of the maximum sizes of `pdstj1` and `pdstj2`
	Manager host's `pdmibwork/sys_DAT_copy` `pdmibwork/fil_DAT_copy`	`pdstedit` execution result for remote copy[2]	See *File size when executing the statistics analysis utility (pdstedit)* in the manual *HiRDB Installation and Design Guide*.
	Manager host's `pdmibwork/new_parameter_copy`	Final log information for remote copy[2]	32
	`pdmibwork/pdmibstm_sys.dat` `pdmibwork/pdmibstm_fil.dat`	Statistical information management data	1 KB + (32 × *unit-count*)
	`pdmibwork/mibstrerr`	Final standard error output of the statistical information output facility	Maximum of 1 KB
	`pdmibwork/pdmibidx`	Index value management data	Maximum: 47 × *host-count* + 19 × *unit-count* + 23 × *server-count* + 45 × *RDAREA-count* + 31 × *global-buffer-count* + 46 × *HiRDB-file-system-area-count*

[1] Deleted after information is collected by executing the `pdstedit` command.

[2] Deleted after remote copy.

Chapter

27. Using a Distributed Database (applicable to HP-UX and AIX 5L only)

This chapter explains the environment setup and operating procedures for a distributed database. The Distributed Database System DF/UX program product is required in order to use the distributed database facility.

This chapter assumes that the reader is familiar with the *Distributed Database System DF/UX*.

This chapter contains the following sections:

27.1 Overview of a distributed database

This section explains a distributed database implemented under HiRDB. The following items are explained here:

- Scope of distributed database
- Remote database access facility
- Character codes environment
- Handling of authorization identifiers
- Handling of passwords
- Notes on establishing connection with another node's HiRDB

27.1.1 Scope of distributed database

Table 27-1 shows the scope of a distributed database implemented under HiRDB.

Table 27-1: Scope of distributed database

Distributed client	Distributed server type				
	HiRDB	XDM/RD	ORACLE (HI-UX/WE2 Version)	RDB1 E2	SQL/K
HiRDB	Y[1]	Y	Y[2]	Y[2]	Y[2]
XDM/RD	Y	—	—	—	—
ORACLE (HI-UX/WE2 Version)	Y[3]	—	—	—	—
RDB1 E2	N	—	—	—	—
SQL/K	N	—	—	—	—

Y: Distributed database can be implemented.

N: Distributed database cannot be implemented.

— : Not applicable.

Notes

- The distributed database facility is limited to HP-UX and AIX 5L (excluding the 64-bit-mode HiRDB and POSIX library version of HiRDB). However,

versions earlier than 07-02 support the HI-UX/WE2 version of HiRDB. The distributed database facility is supported by this HI-UX/WE2 version of HiRDB.

- OSI-RDA is used as the protocol.
- XDM/RD and XDM/RD E2 are referred to collectively as XDM/RD.

[1] The distributed database facility can be used between different platforms. For example, a distributed database can be constructed for HP-UX HiRDB and AIX 5L HiRDB.

[2] The distributed clients of a distributed database configuration are limited to HP-UX HiRDB.

[3] The distributed clients of a distributed database configuration are limited to HP-UX HiRDB.

27.1.2 Remote database access facility

A distributed database can be implemented under HiRDB by using the remote database access facility of DF/UX (Distributing Facility/for UNIX). The remote database access facility consists of two facilities, the distributed client facility and the distributed server facility.

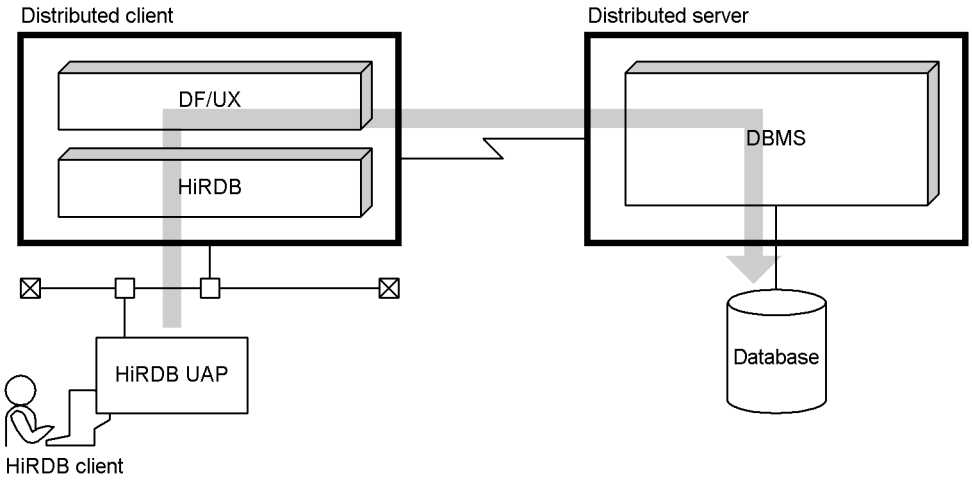
(1) Distributed client facility

When HiRDB at the local node is used as a distributed client, it is possible to access a database at a remote distributed server (a UAP of the HiRDB at the local node can access a remote database at another node's DBMS). This capability is implemented using the distributed client facility of DF/UX.

When the UAP is created, you write the HiRDB SQL statements in the same way as for a UAP used for local access to the HiRDB database at the local node. You specify in the SQL statements the distributed server's table name, a name indicating the location of the database, etc. It must be noted that there are restrictions on UAP creation; for details on these restrictions, see the manual *HiRDB Version 8 UAP Development Guide*.

Figure 27-1 provides an overview of the distributed client facility.

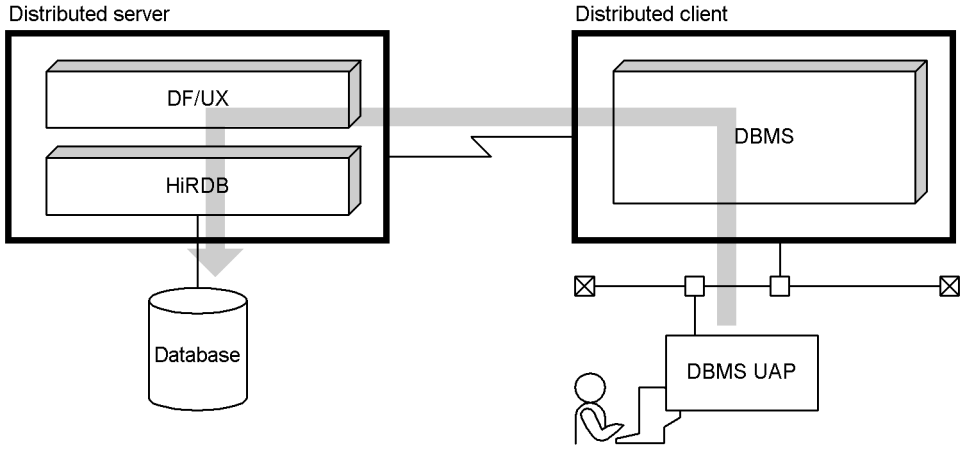
Figure 27-1: Distributed client facility overview



(2) Distributed server facility

When HiRDB at the local node is used as a distributed server, it is possible for a database at a remote distributed client at a remote node to access this distributed server (the DBMS of a remote node can access the HiRDB database at the local node). This capability is implemented using the distributed server facility of DF/UX. Figure 27-2 provides an overview of the distributed server facility.

Figure 27-2: Distributed server facility overview



27.1.3 Character codes environment

An appropriate character codes environment must be set up for the system with which

connection is to be established.

(1) Connection with HiRDB

When the connection is with a HiRDB, the distributed client and the distributed server must both use the same character code set; HiRDB supports the following character code sets:

- Shift JIS Kanji encoding (SJIS)
- EUC Chinese Kanji encoding (EUC/GB)

(2) Connection with Oracle

When the connection is with Oracle, the distributed client and the distributed server must both use the same character code set; HiRDB supports the following character code set:

- Shift JIS Kanji encoding (SJIS)

(3) Connection with XDM/RD

When the connection is with XDM/RD, HiRDB supports the following character code set:

- Shift JIS Kanji encoding (SJIS)

To use Shift JIS Kanji encoding, XDM/RD must be using EBCDIK/KEIS character codes.

(4) Connection with RDB1 E2

When the connection is with RDB1 E2, HiRDB supports the following character code set:

- Shift JIS Kanji encoding (SJIS)

In this case, RDB1 E2 must be using EBCDIK/KEIS character codes.

(5) Connection with SQL/K

When the connection is with SQL/K, HiRDB supports the following character code set:

- Shift JIS Kanji encoding (SJIS)

In this case, SQL/K must be using EBCDIK/KEIS character codes.

27.1.4 Handling of authorization identifiers

The length of the authorization identifier and the permitted characters depend on the system with which connection is established. Table 27-2 shows the permitted authorization identifier length, and Table 27-3 shows the permitted characters that can be used in an authorization identifier.

Table 27-2: Permitted authorization identifier length

System to be connected	Permitted authorization identifier length
HiRDB	Maximum 8 characters
ORACLE	Maximum 8 characters*
XDM/RD	Maximum 7 characters
RDB1 E2	Maximum 8 characters
SQL/K	Maximum 7 characters

* When Oracle is the distributed client, a maximum of 7 characters is permitted for the authorization identifier used to establish connection with HiRDB.

Table 27-3: Characters permitted in an authorization identifier

System to be connected	Characters permitted in authorization identifier					
	A-Z	a-z	0-9	#	@, \	Other
HiRDB	Y	Y	Y	Y	Y	N
ORACLE	Y	Y	Y	Y	N	N
XDM/RD	Y	N	Y	Y	Y	N
RDB1 E2	Y	N	Y	Y	Y	N
SQL/K	Y	N	Y	Y	Y	N

Y: Permitted.

N: Not permitted.

27.1.5 Handling of passwords

The length of the password and the permitted characters depend on the system with which connection is established. Table 27-4 shows the permitted password length, and Table 27-5 shows the permitted characters that can be used in a password.

Table 27-4: Permitted password length

System to be connected	Permitted password length
HiRDB	Maximum 30 characters[1]
ORACLE	Maximum 30 characters[2]
XDM/RD	Maximum 30 characters[3]

System to be connected	Permitted password length
RDB1 E2	Maximum 8 characters
SQL/K	Maximum 30 characters[4]

[1] Can also be omitted.

[2] When Oracle is the distributed client, a maximum of 8 characters is permitted for the password used to establish connection with HiRDB.

[3] When TRUST E2 is used with the XDM/RD system, a maximum of 8 characters is permitted.

[4] SQL/K does not check passwords. The password specified at the HiRDB side is ignored by SQL/K.

Table 27-5: Characters permitted in a password

System to be connected	Characters permitted in password					
	A-Z	a-z	0-9	#	@, \	Other
HiRDB	Y	Y	Y	Y	Y	N
ORACLE	Y	Y	Y	Y	N	N
XDM/RD	Y	N	Y	Y	Y	N
RDB1 E2	Y	N	Y	Y	Y	N
SQL/K	Y	Y	Y	Y	Y	N

Y: Permitted.

N: Not permitted.

27.1.6 Notes on establishing connection with another node's HiRDB

When connection is established with a HiRDB at another node, the HiRDB version should be checked. The following points must be taken into consideration when the two HiRDB versions are different:

1. Only facilities supported by the earlier version of HiRDB are available.
2. Only data types supported by the earlier version of HiRDB are available.
3. If the return code from the distributed server is not listed in the current manual, see the appropriate manual for the distributed server's version.

27.2 Environment setup for a distributed database

This section explains the environment setup for a distributed database. The following items are explained:

- HiRDB environment setup
- DF/UX environment setup
- DF/UX Extension environment setup

27.2.1 HiRDB environment setup

Executor: HiRDB administrator

(1) HiRDB system definitions

In the case of a HiRDB/Single Server, specify the name of the local RD node in the `pd_node_name` operand of the single server definition. In the case of a HiRDB/Parallel Server, specify the name of the local RD node in the `pd_node_name` operand of the front-end server definition.

(2) Client environment definition (for distributed client facility only)

When the distributed client facility is to be used, specify whether or not *batch retrieval* is to be performed. Specify in the `PDRDABLKF` operand the number of rows that are to be subject to batch retrieval. Batch retrieval means transferring from the distributed server to the HiRDB system multiple rows of retrieval results in the batch mode. Specifying the `PDRDABLKF` operand reduces the number of data transfer operations, thereby reducing communications overhead and retrieval time. For details on the `PDRDABLKF` operand, see the manual *HiRDB Version 8 UAP Development Guide*.

When batch retrieval is used, more data is received than is customary during retrieval processing. Therefore, if batch retrieval is not possible due to limitations with respect to the DF/UX transfer buffer, a FETCH statement may result in an error. For details on the error handling procedures and the maximum amount of receive data, see the manual *Distributed Database System DF/UX*.

If the distributed server DBMS does not support batch retrieval, regular retrieval (single-row retrieval) is executed.

27.2.2 DF/UX environment setup

Executor: Superuser and DF/UX system administrator

(1) DF/UX installation

- HiRDB/Single Server

  Install DF/UX on the server machine where the single server is defined.

- HiRDB/Parallel Server

  Install DF/UX on the server machine where the front-end server is defined.

(2) DF/UX environment definition

HiRDB-related operands in the DF/UX environment definition are explained below. For details on DF/UX environment definition, see the manual *Distributed Database System DF/UX*.

(a) Related operands (for distributed client facility only)

The following table shows the operands that are related to use of the distributed client facility:

Operand	Specification
`sql_environment_name`	Specifies the RD node name of the distributed server. Specify the name specified in the `rd_node_name` operand of the SQL environment definition in DF/UX.
`context_name`	Specifies `RDA#SQL#BASIC#OSAS`.
`block_fetch_count`	Becomes the default value of the `PDRDABLKF` operand.

(b) Invalid operands (for distributed client facility only)

When the distributed client facility is being used, the following operands are ignored even if specified:

- `maximum_SQL_length`
- `maximum_parameters`
- `maximum_statements`
- `block_fetch_limit`

(c) Specification of maximum_client_users and maximum_server_users operands

The `maximum_client_users` and `maximum_server_users` operands must be specified so that the following condition is satisfied:

Formula

Value of pd_process_count (in single server definition*)
≧ value of maximum_client_users + value of maximum_server_users

* In the case of a HiRDB/Parallel Server, use the value in the front-end server definition.

(3) DF/UX distribution definition

The operand shown below must be specified in the DF/UX distribution definition. For details on the DF/UX distribution definition, see the manual *Distributed Database System DF/UX*.

Operand	Specification
context_name	Specifies RDA#SQL#BASIC#OSAS.

(4) Creating a server facility executable file (for distributed server facility only)

Use the dfsvplink command of DF/UX to create a server facility executable file. Specify the options listed below in this command. For details on the dfsvplink command, see the manual *Distributed Database System DF/UX*.

dfsvplink command option	Specification
-x	Specifies p_rdb_df_xa_switch as the name of the xa_switch_t structure.
-l	Specifies the following as the libraries of the distributed server's DBMS: • $PDDIR/lib/libdfc.sl* • $PDDIR/client/lib/libzcltx.sl*

* The library suffix depends on the platform; for AIX 5L, it is .a.

(5) Environment variable setup (for distributed server facility only)

As in the case of a local-access UAP, communication between DF/UX and HiRDB uses the HiRDB client's library. Therefore, client environment definitions must be specified in environment variables of the startup command (dfstart command) of DF/UX. To start DF/UX automatically, specify the client environment definitions before the dfstart command in localrc in the /etc/rc file. For details on automatic startup of DF/UX (details of specifying localrc in the /etc/rc file), see the manual *Distributed Database System DF/UX*.

The following are the client environment definitions that must be specified (for details, see the manual *HiRDB Version 8 UAP Development Guide*):

- PDCLTPATH[1]
- PDHOST
- PDNAMEPORT
- PDSWAITTIME[2]

[1] Specifies $PDDIR/spool/dfc as the error log output destination. The HiRDB administrator must create $PDDIR/spool/dfc. Because error logs are output under

the ID of the DF/UX system manager, the privilege to write into the directory must be granted to the DF/UX system manager.

[2] Specifies a value that is greater than the value of the `limit_access_interval` operand of DF/UX. To safeguard against errors in distributed clients, access time is monitored based on the value specified in the `limit_access_interval` operand. If an error occurs in DF/UX, access time is monitored based on the value specified in the `PDSWAITTIME` operand.

27.2.3 DF/UX Extension environment setup

Executor: Superuser and DF/UX system administrator

(1) DF/UX Extension installation

In the case of a HiRDB/Single Server, install DF/UX Extension at the server machine where the single server is defined.

In the case of a HiRDB/Parallel Server, install DF/UX Extension at the server machine where the front-end server is defined.

(2) DF/UX Extension environment definition

The HiRDB-related operand for the DF/UX Extension environment definition is explained below.

`maximum_client_users` must be set so that the following condition is satisfied:

Formula

Value of pd_process_count (single server definition*)
≥ value of maximum_client_users (DF/UX environment definition)
+ value of maximum_server_users (DF/UX environment definition)
+ value of maximum_client_users (DF/UX Extension environment definition)

* In the case of a HiRDB/Parallel Server, use the value in the front-end server definition.

(3) DF/UX Extension distribution definition

The following operand must be specified in the DF/UX Extension distribution definition:

Operand	Specification
`rdb_name`	Must be the same RD node name specified in the SQL statements.

(4) Package creation

DF/UX Extension's package handling utility is used to create a package used by DF/UX Extension. The following points should be noted.

1. When DF/UX Extension's package handling utility creates cursor declarations in the package, it uses C00001-C00100 as the cursor names. However, any cursor names can be used in cursor declarations in a UAP. HiRDB converts the cursor names in the UAP to the cursor names in the package.

27.3 Distributed database security

(1) Security when distributed client facility is used

Required privileges must be registered in the distributed server for the authorization identifier of a UAP that accesses a remote database.

(2) Security when distributed server facility is used

Executor: User with DBA privilege and table owner

The following user privileges must be granted to the authorization identifier of a UAP that accesses a remote database:

- CONNECT privilege
- Access privilege

For details on how to grant user privileges, see *2.2 Setting user privileges*.

27.4 Information output when a communication error occurs (Distributed Server facility only)

If an error occurs during communication between DF/UX and HiRDB, error information (error log) is output to an error log file.

File name

- `$PDCLTPATH/pderrxxxxx-1.trc` (where xxxxx is the DF/UX server process ID)
- `$PDCLTPATH/pderrxxxxx-2.trc`

File size

4096 bytes

For details on the error log files, see the manual *HiRDB Version 8 UAP Development Guide*.

Appendixes

A. Q&A
B. Operations When Using a DVD-RAM Library Device
C. Information Needed for Troubleshooting
D. Notes on Running HiRDB Around the Clock
E. Using Performance Improvement Facilities

A. Q&A

Frequently asked questions about HiRDB operations are answered in this appendix in Q&A format. The following topics are covered:

This appendix contains the following sections:

- A.1 System log files
- A.2 Synchronization point dump files
- A.3 Status files
- A.4 Errors
- A.5 Tables and indexes
- A.6 HiRDB startup
- A.7 HiRDB termination
- A.8 Performance
- A.9 Backup
- A.10 RDAREA recovery
- A.11 Other

A.1 System log files

(1) The KFPS01224-I warning message was displayed indicating that there are no system log files in swappable target status

Question

When HiRDB starts normally, doesn't it continue to use the system log file that was being used during the previous HiRDB session ?

Phenomenon 1

I executed normal startup or forced normal startup of HiRDB, but the `KFPS01224-I` message was displayed indicating that there was no system log file available for swapping. When this problem occurred, the system log files were in the following status:

- There was adequate free space remaining in the system log file that was being used during the previous HiRDB session.
- There was an available system log file that could be swapped in.

Phenomenon 2

I executed normal startup or forced normal startup of HiRDB, but the KFPS01256-E message was displayed indicating that there was no available system log file, and then HiRDB (or unit for a HiRDB/Parallel Server) terminated abnormally with abort code Psjnf07. When this problem occurred, the system log files were in the following status:

- There was adequate free space remaining in the system log file that was being used during the previous HiRDB session.
- There was no available system log file that could be swapped in.

Answer

When HiRDB starts normally, it does not continue to use the system log file that was being used during the previous HiRDB session. Because HiRDB swaps system log files during a normal startup (including a forced normal startup), the current file is swapped. The KFPS01256-E warning message was displayed because the last system log file available to be swapped in was allocated, and now there are no more system log files available for future swapping.

On the other hand, when HiRDB is restarted, the system log files are used as if the previous session was being resumed (system log files are not swapped).

If the same system log files were used after a normal startup, the system log from the previous session and the system log for the current session would be stored in a single system log file. In this case, performing file management tasks (such as unloading into unload log files) would be complicated. Therefore, HiRDB swaps system log files during normal startup so that a system log file containing only the new information following the normal startup is compiled.

When pd_log_rerun_swap=Y is specified, the system log files are also swapped during a restart. If a shortage of space occurs in the system log file, the restart fails; therefore, the number of system log files available to be swapped in (in swappable target status) should be increased.

(2) The KFPS01251-E message was displayed indicating that a system log file was invalid

Question

I executed forced normal startup of HiRDB (or a unit) after forced termination occurred (or the HiRDB unit terminated abnormally due to a software or hardware error), but the KFPS01251-E message was displayed indicating an invalid system log file. As a result, the system log file could no longer be used as a swappable target. What is the cause of this and what action should be taken?

Answer

Cause:

The file is an incomplete file because the primary file was not closed normally when HiRDB (or the unit) terminated abnormally. The same file will continue to be used because such situations occur during a restart. However, this file will be checked when a restart is executed.

Action:

Use the `pdlogunld` command to unload the system log in this system log file. If this system log information is not needed, the `pdlogchg` command can be used to simply change the file's status.

(3) Addition of a system log file and restart of HiRDB were repeated several times

Question

HiRDB terminated abnormally immediately after a restart because there was no system log file available to be swapped in (in swappable target status). So I allocated a new system log file and restarted HiRDB. HiRDB terminated abnormally again due to a shortage of system log files. I repeated the process of adding a system log file and restarting HiRDB, but the problem could not be resolved. Why did HiRDB keep terminating abnormally?

Answer

Possible reasons:

1. `pd_log_rerun_swap=Y` is specified.

   When `Y` is specified in this operand, the current system log files are swapped even during a restart, thus reducing the number of swappable files by one. Therefore, this phenomenon cannot be eliminated by adding only one system log file. Add at least two system log files and restart HiRDB.

2. There is not enough space in the overwrite-enabled system log file.

   During a HiRDB restart, system log information is collected even for rollback. Therefore, repeating the restart increases the number of system log files that are output, thereby increasing the range of the overwrite-disabled system log files. Because a new synchronization point dump is collected upon completion of all rollbacks after a restart, if the unit terminates abnormally after this synchronization point, the range of the overwrite-disabled system log files will not increase.

   At least two system log files should be added at the same time, rather than adding one at a time. The total size of the files to be added should be at least 1.5 times the system log input range at the first restart when this event occurred for the first time (from the location indicated in the `KFPS01262-I`

message to the location indicated in the KFPS01263-I message).

Remarks

This problem can be prevented by providing multiple reserved file groups and specifying pd_log_rerun_reserved_file_open=Y.

A.2 Synchronization point dump files

(1) The KFPS02179-I message is issued repeatedly

Question

Why is the KFPS02179-I message being issued repeatedly (this message says that the time for collecting a synchronization point dump has been reached, but collection of the dump was skipped because the previous synchronization point dump processing had not been completed)?

Answer

This message is issued when HiRDB is waiting for commitment of an updating transaction that was being executed when a synchronization point dump was collected (i.e., HiRDB is waiting for completion of validation). There should be no problem if there is a sufficient number of system log files with sufficient space in them. However, if the unit terminates abnormally in such status, the restart time for HiRDB or the unit may be prolonged.

If there are enough generations and there is enough space in the system log file, HiRDB or the unit may terminate abnormally due to a shortage of overwrite-enabled system log files (the KFPS01256-E message is issued). To avoid such problems, reevaluate the following:

When it takes a long time for the transaction to commit after updating a database or data dictionary

Evaluate the following items:

- Would it be possible to issue the commit statement during the processing?
- If the transaction retrieves a large amount of data after updating, would it be possible to reduce the time from the first update processing in the transaction to the commit point by changing the sequence of the update processing and the retrieval processing?

When there is insufficient space in the system log file or too few generations

Evaluate the following items:

- It is recommended that you provide the amount of file space necessary to store system logs that will be output during operation of HiRDB. If

this is not feasible, try to allocate at least the size recommended in the *HiRDB Version 8 Installation and Design Guide*.

- Do not collect a database update log for a transaction that updates a large amount of data. Whether or not a database update log is to be collected is specified in the `PDDBLOG` operand in the client environment definition. If a database update log is not collected and an error occurs during UAP execution, the database can be restored only to the point where its backup was made.

(2) Number of guaranteed valid generations

Question

What is meant by the number of guaranteed valid generations of the synchronization point dump file?

Answer

Information such as a read operation's starting point in the system log file is acquired in the synchronization point dump file; this information is useful in the event a restart of HiRDB becomes necessary. Thus, the portion of the system log file beginning at the location indicated in the synchronization point dump file is write-protected because it might be needed for a restart.

The number of guaranteed valid generations is the number of generations of synchronization point dump files that are used to write-protect the system log file. In other words, if the number of guaranteed valid generations is 1, only the system log file corresponding to the most recent synchronization point dump file is write-protected. If the number of guaranteed valid generations is 2, the system log file corresponding to the most recent synchronization point dump file and the one immediately preceding it are write-protected. Therefore, if the number of guaranteed valid generations is 2, the number of system log files that are write-protected may increase.

The number of synchronization point dump file guaranteed valid generations is specified in the `pd_spd_assurance_count` operand.

A.3 Status files

(1) When are status files swapped?

Question

Are status files swapped only when a disk error occurs?

Answer

No. Status files are swapped at the following times:

- When an error occurs in a status file

- When a current status file becomes full
- When there is no more usable space in a current status file due to fragmentation
- When the `pdstsswap` command is executed

(2) Does swapping occur even though an error occurs in only one of the status file versions?

Question

Are status files swapped when an error occurs in only one of the file versions?

Does there have to be an error in both file versions before swapping occurs?

Answer

Swapping of status files occurs when an error occurs in either of the file versions. Below is the procedure HiRDB follows from the time an error is detected until swapping is completed:

1. Searches for normal spare files for both versions A and B.
2. The contents of the normal system are copied into versions A and B of other status files (the file space is also defragmented).
3. The status files at the copy destination are used as the current status files.

(3) Handling of status file errors

Question

The following operands are used to specify the HiRDB processing when no more normal status files are available for either file version A or file version B (in the event of an error in file version A or B):

- `pd_syssts_singleoperation=stop|continue` (applicable to unit status files)
- `pd_sts_singleoperation=stop|continue` (applicable to server status files)

Should I specify `stop` or `continue` for these operands?

Answer

Specifying `stop` terminates HiRDB or the unit abnormally. When `continue` is specified, operation with a single set of status files occurs (this is called *single operation*).

Status files are important because they contain information needed for full recovery processing. When `continue` is specified and an error occurs in the status file during single operation, HiRDB or the unit is terminated abnormally

because the error is in both file versions. In this case, full recovery processing cannot be executed because neither file version that was used as the current file can be accessed. Following are guidelines for specifying these operands:

- When providing for full recovery is more important than preventing abnormal termination of HiRDB, `stop` should be specified.
- When you do not want to stop HiRDB (when you give up on full recovery processing, recover the database up to the point of the most recent backup, and reload data), `continue` should be specified.

A.4 Errors

(1) What information do I need in the event of an error?

Question

Which information should I collect for purposes of investigation in the event of an error?

Answer

In general, save the following information onto a medium such as DAT:

1. All files under `$PDDIR/spool`
2. The following shared libraries:
   - `/lib/libM.sl`
   - `/lib/libc.sl`
   - `/usr/lib/libdld.sl`
   - `/usr/lib/librsort.sl`
3. syslog file
4. For an error associated with database conformity, collect the following information:
   - Contents of the user RDAREAs storing the corresponding table
   - Contents of the past log files up to that point
   - Contents of the master directory RDAREA
   - Contents of the data directory RDAREA
   - Contents of data dictionary RDAREAs

Remarks

HiRDB provides the `pdgeter` command for collecting all the above error information except No. 4.

(2) After restarting HiRDB, SPP terminated abnormally due to a DB lock-release wait timeout

Question

After restarting HiRDB, SPP terminated abnormally due to an RDAREA lock-release wait timeout:

1. The database seems to still be locked. What is the cause of this?
2. After SPP's abnormal termination, the status during rollback cannot be restored. What is the cause of this?
3. An attempt to terminate OpenTP1 normally with the `dcstop` command (OpenTP1 command) failed because there is an SPP engaged in rollback processing. What is the cause of this?

Answer

1. The following specifications may be missing in OpenTP1's transaction service definition (`trn`):

   - `trnstring -n HiRDB_DB_SERVER -d`*
   - `set trn_tran_optimum_processing=N`

   These operands suppress OpenTP1's commit optimization and prepare optimization and normally are used when HiRDB is linked with OpenTP1 via the XA interface.

   * Supported by TP1/Server Base 03-03 or later.

2. A HiRDB client environment variable may be missing or an invalid value may have been specified in OpenTP1's transaction service definition (`trn`). Check the following client environment variables:

   - `PDHOST`
   - `PDNAMEPORT`

   At the same time, check OpenTP1's user service default definition (`usrrc`).

3. Same as 2 above.

Note:

When this event occurs, terminate OpenTP1 forcibly, correct the HiRDB system definitions, then restart OpenTP1.

Reference note:

The environment variables common to `trn` and `usrrc` should be specified in env. When they are specified in env, the environment variables are different from those in `trn` or `usrrc`; check the *HiRDB Version 8 UAP Development Guide* for the correct ones.

A.5 Tables and indexes

(1) How can I obtain an existing index's definition?

Question

Is it possible to use SQL to find out if an index has been defined?

Answer

It is possible by searching data dictionary tables; see the examples below.

The definition SQL generation facility of the dictionary import/export utility (`pdexp` command) can also be used for this purpose.

Example 1

Search for tables that have indexes:

```
SELECT TABLE_SCHEMA,TABLE_NAME,TABLE_ID
    FROM MASTER.SQL_TABLES WHERE N_INDEX >= 1
```

Example 2

Display the index names and columns names for a defined table:

```
SELECT
TABLE_SCHEMA,TABLE_NAME,INDEX_NAME,INDEX_ID,COLUMN_NAME
    FROM MASTER.SQL_INDEX_COLINF
        WHERE TABLE_SCHEMA = USER
```

Example 3

Display the table name, table ID, index name, index ID, and column names for a table that has an index:

```
SELECT X.TABLE_SCHEMA,X.TABLE_NAME,X.TABLE_ID,
  INDEX_ID,COLUMN_NAME
    FROM MASTER.SQL_TABLES X,MASTER.SQL_INDEX_COLINF Y
        WHERE X.TABLE_SCHEMA = Y.TABLE_SCHEMA
              AND X.TABLE_SCHEMA = USER
              AND X.TABLE_NAME = Y.TABLE_NAME
              AND N_INDEX >= 1
```

(2) How can I add columns to an existing FIX table?

Question

Is there a way to add columns to a `FIX` table that already contains data?

Answer

Yes, there is. For details about adding columns, see *13.6 Adding a column*.

A.6 HiRDB startup

(1) HiRDB startup is slow

Question

Why does it take so much time (1-2 hours) for all units to start?

When the `pdstart` command was executed, the `KFPS05078-I` message was issued.

Answer

Check the following:

1. If the `KFPS00608-W` message (`-314`) is output more than once, check that the same host name is specified in the `pdunit` and `pdstart` operands for all units and that the specified host name is correct (name of the existing host).
2. Check that all hosts used by HiRDB and the network have started.

(2) The pdstart command resulted in an error (reason code=TIMEOUT)

Question

The `pdstart` command resulted in an error with the `KFPS01861-E` message (`reason code=TIMEOUT`). Why has this happened?

Answer

Cause

Following are possible reasons:

1. It took more time to start up a unit than expected.
2. There is an error in the server common definition or an individual server definition.

Action

Take the following actions:

1. Increase the value specified in the `pd_start_time_out` operand, then re-enter the `pdstart` command.
2. Check the messages HiRDB has output to syslogfile and correct any incorrect definitions that were detected. Then use the `pdsetup -d` command (enter `y` at the `KFPS00036-Q` prompt message) to delete HiRDB from the operating system and re-execute the `pdsetup` command.

(3) The pdstart command resulted in an error (RETURN_CODE=28)

Question

The `pdstart` command resulted in an error, setting `RETURN_CODE=28` for a semaphore-related function such as `semctl` and `semop`. What causes this error?

Answer

The problem was caused by a resource shortage. Change the OS's operating system parameter to the value required by HiRDB. Note that the default value of the operating system parameter is smaller than the value required by HiRDB.

If the `pdstart` command terminates in a timeout error, a semaphore or shared memory shortage may be the cause. For details about the operating system value required by HiRDB, see the manual *HiRDB Version 8 Installation and Design Guide*.

If the operating system parameter value is appropriate, increase the value of the `pd_start_time_out` operand, and then re-execute the `pdstart` command.

(4) The pdstart command resulted in an error (reason code=SETUP)

Question

The `pdstart` command resulted in an error and the `KFPS01801-E` message was issued (`reason code=SETUP`). Why?

Answer

Cause

Following are possible reasons:

1. The HiRDB directory specified in the `PDDIR` environment variable was not registered into the OS with the `pdsetup` command.
2. The process server daemon cannot be started because of a kernel semaphore shortage.

Action

Take the following actions:

1. Use the `pdsetup` command to register the HiRDB directory into the OS.
2. Increase the number of semaphores used, which is defined by the system. Note that you must reboot the OS in order for the new value to take effect.

Reference note:

This event may also occur immediately after OS booting or when another program registered in `inittab` is placed in response wait status.

(5) The pdstart command was invalidated

Question

The `pdstart` command had no effect. Why?

Answer

The HiRDB server daemon may have been unable to start up because of an error.

Investigate the cause of the error by checking the message log (`$PDDIR/spool/pdlog1` or `pdlog2`) and take appropriate action. If the cause of the error cannot be determined from the message log, check the details in the HiRDB message output to syslogfile, and take appropriate action.

(6) A unit cannot be started with the pdstart command

Question

When I executed pdstart, the `KFPS01815-E` message (`errno=11,13,22`) was displayed indicating that semaphore manipulation (semop, semctl) failed. What happened?

Answer

Cause

Following are possible reasons:

1. HiRDB has not been installed on that machine.
2. HiRDB was not registered in the OS with the `pdsetup` command.
3. The installation directory is linked to a shared disk.

Action

Take the following actions:

1. Reboot the server machine, then execute the `pdsetup -d` command to delete HiRDB from the OS. Then re-execute the `pdsetup` command.
2. If HiRDB is installed on the server machine, execute the `pdsetup` command.
3. Create the installation directory on the local server machine's local disk.

(7) There is no response from the pdstart command

Question

Why is there no response from the pdstart command?

Answer

1. Is the front-end server in SUSPEND status?

   If it is not in SUSPEND status, check for another cause. If it is in SUSPEND status, execute ps -ef or grep fesact at the host where the front-end server is started. for multiple front-end servers, execute it at all those hosts.

2. Is there more than one fesact process?

   If not, check for another cause. If there is, an error may have occurred in the network. Terminate HiRDB forcibly with the pdstop -f command.

   Contact the network administrator for error recovery.

(8) HiRDB/Parallel Server will not start

Question

The system manager unit terminated abnormally, and only that unit was restarted. Now, HiRDB startup cannot be completed.

Answer

Abnormal termination of the system manager unit may have caused abnormal termination of another unit. Use the pdls command to determine the units that are not active.

(9) A unit other than the system manager unit will not start

Question

The pdstart command was entered, but a unit other than the system manager unit will not start. Why?

When ps -ef was checked on the unit other than the system manager unit, there was only pdprcd. HiRDB's other daemons have not been started.

Answer

Check the system common definition for the unit that would not start. The value specified for the pdunit or pdstart operand may not match the information in the system common definition for the server machine where the system manager is located.

Correct the system common definition for the unit resulting in the startup error, then start the unit again with pdstart -u.

(10) The front-end server is in SUSPEND status

Question

When the front-end server unit or the front-end server was started independently,

the front-end server was placed in SUSPEND status. How can I place it in ACTIVE status?

Answer

If the front-end server unit or the front-end server is started while the dictionary server is shut down, the status is set to SUSPEND. In such a case, start the dictionary server, then execute pdstart -a -s *front-end-server-identifier*.

(11) Is it possible to start HiRDB automatically?

Question

1. Is it possible for HiRDB to be started and terminated automatically when the OS is started and terminated?
2. pdstart was specified in /etc/localrc in order to start HiRDB when the OS was started, but this did not work. Why?
3. pdstop was specified in /etc/localshutrc in the same manner. Why didn't this work?

Answer

1. To start HiRDB when the OS starts, specify AUTO in the pd_mode_conf operand.
2. Currently, there is no way to terminate HiRDB automatically when the OS terminates. Execute the pdstop command and check that HiRDB has terminated normally, then terminate the OS.
3. Neither the pdstart nor the pdstop command can be specified in the OS's /etc/localrc or /etc/localshutrc environment file. During OS startup, /etc/localrc is activated at the /etc/rc phase, but pdstart cannot be accepted at this point because the HiRDB process server daemon (_prc) has not been started yet. If the pdstart command is specified in /etc/localrc, the command terminates abnormally. To start HiRDB when the OS is started, specify AUTO in the pd_mode_conf operand. On the other hand, /etc/localshutrc is activated as an extension of the shutdown command when the OS execution level is 0, at which point the HiRDB process server daemon has already been terminated. If the pdstop command is specified in /etc/localshutrc, the command results in an error and HiRDB has to be restarted.

(12) The pdstart command was specified in /etc/localrc, but it resulted in an error

Question

- The pdstart command was specified in /etc/localrc, but it resulted in an error.
- The pdstart command was executed using the HA monitor's interface, but

it resulted in an error.

- The pdstart command resulted in an error when it was executed after OS bootup followed by remote login.

Answer

HiRDB operation becomes available near the end of OS bootup processing. Therefore, HiRDB cannot be started by specifying the pdstart command in /etc/localrc, etc. The pdstart command can be executed only after HiRDB's pdprcd daemon has started.

Action 1

Do not execute the pdstart command immediately after OS bootup. For example, use the following procedure to execute the pdstart command:

Procedure:

1. Ensure that pdprcd is active:

   ```
   ps -ef | grep $PDDIR/lib/servers/pdprcd
   ```

2. Wait a few seconds (until startup of the pdprcd daemon is completed).
3. Execute the pdstart command.

Action 2

Automatic startup of HiRDB should be employed. To start HiRDB automatically, specify pd_mode_conf=AUTO. When this is specified, HiRDB starts automatically at the completion of OS bootup (there is no need to enter the pdstart command).

(13) When I closed the window used to enter the pdstart command, HiRDB terminated abnormally

Question

I entered the pdstart command but there was no response from the command. Therefore, I closed the window used to enter the pdstart command. HiRDB terminated abnormally immediately after this. Why?

Answer

If the pdstart command is stopped forcibly, conformity cannot be maintained in the shared resources, resulting in abnormal termination of HiRDB. Similarly, if the window is closed while waiting for a response from the pdstart command, HiRDB terminates abnormally.

Do not close the window used to enter the pdstart command until the command has terminated. This also applies to all other operation commands and utilities; do not close the corresponding window during execution or while waiting for a

response.

A.7 HiRDB termination

(1) When the pdstop -f command was entered, the unit terminated abnormally

Question

When I entered the `pdstop -f` command, why did HiRDB or the unit terminate abnormally with abort code `Polkcrt`?

Answer

A process in critical status was terminated during forced termination processing. Ignore this event because there is no problem in terms of operations.

The `pdstop -f` command terminates forcibly all processes, even processes in critical status. Therefore, although this event can occur, there is no problem because HiRDB is restarted by the next `pdstart` command (the database is restored from the system log).

(2) HiRDB will not terminate normally

Question

Why won't HiRDB terminate normally?

Answer

HiRDB cannot terminate normally while there is a transaction in uncompleted status or a connected user. Use the following commands to check the status of connected users and transactions before attempting to terminate HiRDB normally:

- `pdls -d prc` command to check for connected users
- `pdls -d trn` command to check transaction status

For the actions to be taken, see *18.13 When HiRDB cannot be terminated because a user is still connected* or *18.14 Actions when there is an undetermined transaction*.

(3) When HiRDB terminated normally, the system server terminated abnormally

Question

I executed the `pdstop` and `shutdown` commands consecutively in this order using a shell script, then the system server terminated abnormally. Why?

Answer

The system server does not terminate the moment the `pdstop` command is entered. If the `shutdown` command is executed while the system server is still engaged in termination processing, the system server terminates abnormally.

Do not execute the shutdown command immediately after entering the pdstop command. For example, execute the following commands in the sequence shown:

1. pdstop
2. sleep 60
3. shutdown

(4) When I closed the window used to enter the pdstop command, HiRDB terminated abnormally

Question

I entered the pdstop command, but there was no response from the command. Therefore, I closed the window used to enter the pdstop command. HiRDB terminated abnormally immediately after this. Why?

Answer

If the pdstart command is stopped forcibly, conformity cannot be maintained in the shared resources, resulting in abnormal termination of HiRDB. Similarly, if the window is closed while waiting for a response from the pdstop command, HiRDB terminates abnormally.

Do not close the window used to enter the pdstop command until the command has terminated. This also applies to all other operation commands and utilities; do not close the corresponding window during execution or while waiting for a response.

A.8 Performance

(1) The expected retrieval performance cannot be obtained

Question

I defined an index for a table in order to use the index to achieve high-speed retrieval, but the expected retrieval performance was not obtained. Is it possible that the index was not used?

Answer

An index may not be used for retrieval if the optimizing information collection utility (pdgetcst command) has not been used in the database's most recent status. For details about whether or not execution of the optimizing information collection utility is required, see the manual *HiRDB Version 8 Command Reference*.

If any of the events listed below occurs with respect to a table that has been optimized based on cost using the optimizing information collection utility (pdgetcst command), the pdgetcst command must be executed again to obtain the most recent optimization information:

- A large amount of data has been added, updated, or deleted
- The database load utility (`pdload` command) or the database reorganization utility (`pdrorg` command) has been executed
- Table definition has been modified, or an index has been added or deleted

Remarks

The access path display utility (`pdvwopt` command) can be used to determine whether or not an index is being used for retrieval.

(2) Is it possible to check the buffer utilization status and the number of read/write operations?

Question

How can I check a buffer's utilization status (such as the hits rate) and the number of read/write operations in order to tune the buffer?

Answer

A buffer's hits rate and the real `READ/WRITE` counts can be obtained by editing the global buffer statistical information with the statistics analysis utility (`pdstedit` command). Start collecting statistical information by executing the statistics information output start command (`pdstbegin` command) with `buf` specified as the statistical information type. Note that this is the differential information between synchronization point dumps; therefore, the information will not be edited or output until two synchronization point dumps have been validated. If necessary, use the following commands to forcibly swap the system log files and validate synchronization point dumps:

- `pdststart -k buf`: Starts collection of statistical information
- `pdlogsync`: Validates synchronization point dumps

Range of statistical information collection:

- `pdlogsync`: Validates synchronization point dumps
- `pdstedit`: Edits and outputs the statistical information

Remarks

The same information as obtained by the statistics analysis utility can be displayed easily by entering the `pdbufls` command. In this case, the differential information between two `pdbufls` command entries is displayed; therefore, execute the `pdbufls` command twice, once at the beginning and once at the end of the desired range of statistical information collection.

A.9 Backup

(1) Backup acquisition unit

Question 1

Can you make a backup of just a table?

Answer

No, you cannot. However, you can achieve this with one of the following methods:

- Store one table in one RDAREA and make a backup of the RDAREA.
- Use the database reorganization utility (`pdrorg` command) to unload each table.

Question 2

Can you make a backup that consists of only the changes made since the previous backup?

Answer

Yes, you can. HiRDB supports the differential backup facility. This facility backs up only the information that has changed (differential information) since the previous backup was made. This facility can reduce the amount of time spent making backups. The differential backup facility should be used when the database is large but the volume of updating is small. For details of the differential backup facility, see *6.5 Acquiring a differential backup*.

Question 3

If an error occurs in a particular RDAREA after all RDAREAs have been backed up with the `-a` option specified in the `pdcopy` command, can you recover only that RDAREA from the backup copy of all RDAREAs?

Answer

Yes, you can. A particular RDAREA can be recovered by specifying its name in the `pdrstr` command.

(2) Difference between the pdfbkup and pdcopy commands

Question 1

Can I use a backup (backup of a HiRDB file system area) acquired with the `pdfbkup` command as the input file for the `pdrstr` command?

Answer

No, a backup acquired with the `pdfbkup` command can be used only with the `pdfrstr` command.

Question 2

Can I make a backup with the `pdfbkup` command while HiRDB is running?

Answer

No, you cannot. Use the `pdcopy` command to make a backup while HiRDB is running.

(3) Backing up the master directory RDAREA

Question 1

Can I back up RDAREAs while HiRDB is running?

Answer

Yes, you can. However, if the master directory RDAREA is included, one of the following backup acquisition modes must be used:

- Referencing-permitted mode (`-M r` specified)
- Updatable mode (`-M s` specified)

Question 2

If the `-a` option (back up all RDAREAs) is specified in the `pdcopy` command, will the master directory RDAREA be included? If so, do I need to start HiRDB with the `pdstart -r` command?

Answer

The master directory RDAREA will be included.

HiRDB must be started with the `pdstart -r` command, not just when the master directory RDAREA is backed up, but when both the following conditions are satisfied:

- The master directory RDAREA is included in the backup
- The referencing/updating-impossible mode (`-M x` specified) is specified as the backup acquisition mode

(4) Even though there is available space on the disk, a disk space shortage occurs during execution of the pdcopy command

Question

Why does a disk space shortage occur during execution of the `pdcopy` command, even though there is available space on the disk?

Answer

Following are possible reasons:

- Use of large files is not permitted (`pd_large_file_use = Y` is not

specified).

- The permitted maximum size for the kernel parameter has been exceeded.

In this case, either use large files or change the kernel parameter. You can also solve the problem by specifying multiple backup files. However, if the OS does not support large files, you must set the disk partition size to 2 GB or less to handle multiple files.

A.10 RDAREA recovery

(1) Specification of the pdrstr command

Question 1

Because errors occurred in the master directory RDAREA and in a particular user RDAREA, all RDAREAs were recovered with `-a` (recovery of all RDAREAs) or `-c` (recovery of all RDAREAs in a backup) specified in the `pdrstr` command. In this case, has the information in normal RDAREAs also been replaced?

Does recovery processing first delete the information internally and then import it?

Answer

The information is replaced. The `pdrstr` command first deletes the HiRDB files comprising the RDAREAs to be recovered by the `pdrstr` command and then restores the contents of the backup files. Therefore, the RDAREAs specified as recovery targets during execution of the `pdrstr` command (including normal RDAREAs) are all replaced.

Question 2

When an initialized RDAREA is recovered with the `pdrstr -a` command (recovery of all RDAREAs), is table definition information also recovered?

If not, when should I execute `CREATE TABLE`?

Answer

The table information is recovered. Because the table definition information is stored in a dictionary, it is recovered correctly if the initialized RDAREA is synchronized with the dictionary, data directory, and master directory.

(2) Master directory RDAREA recovery

Question

How should I recover the master directory RDAREA?

Answer

To recover the master directory RDAREA, you must start HiRDB with the

pdstart -r command. For details of the recovery procedure, see *19. Database Recovery Procedures*.

A.11 Other

(1) Increasing the number of executing users

Question

What are the effects of increasing the value of the pd_max_users operand in order to accommodate more users?

Answer

For details about the effects of increasing the value of the pd_max_users operand, see *9.5 Handling an increase in the number of users*.

(2) No UAP is connected, but user identification information is displayed

Question

No UAP is connected, so why is user identification information displayed when the pdls -d act command is executed?

Answer

If transaction rollback was not completed after a server process was aborted or rerun, the user identification information is retained. This user identification information will be deleted once transaction rollback is completed.

(3) Can the system time be changed?

Question

I want to change the system time to do some tests. Is it possible to change the system time?

What would be the effects of forcibly advancing or rolling back the system date or time?

Answer

Although you can advance the system date or time, it is not recommended that you do so.

HiRDB stores date and time information in the system log files, status files, and dictionaries for use during HiRDB restart, etc. Date and time information is also stored in the work files of various utilities for use in several types of checking.

Advancing the date and time

There is no serious problem in advancing the time, but statistical information may not be displayed correctly, or for some period the time may not be updated in messages. If the date and time are changed during HiRDB startup

processing, the results cannot be guaranteed. There may also be OS consequences.

Rolling back the date and time

The results cannot be guaranteed because various problems may result, such as restart errors, database recovery errors, invalid system waits, etc. For these reasons, forcibly rolling back the date or time is not advisable. If the date and time are rolled back for testing purposes, the system and databases will have to be rebuilt from the beginning.

(4) Authorization identifiers and passwords are not being recognized correctly

Question

Why are authorization identifiers and passwords not being recognized correctly?

Answer

It is possible that upper- and lower-case letters are not being recognized correctly. If lower-case letters are included, the character strings must be enclosed in quotation marks ("). Otherwise, the lower-case letters will be regarded as upper-case letters.

Example

```
SELECT * FROM "user05".TABLE05;
```

B. Operations When Using a DVD-RAM Library Device

This appendix explains the procedures for using a DVD-RAM library device as a storage device.

(1) Setting system common definitions

To use a DVD-RAM library device, you must take steps to avoid detection of false timeouts caused by the fact that the physical mounting operation (staging) takes a long time. Therefore, specify a value in the operands listed below that adds the time required for the mounting operation to the previous estimate:

- `pd_watch_time` (maximum response wait time)
- `pd_lck_wait_timeout` (lock-release wait time)

(2) Creating HiRDB file system areas

You create a HiRDB file system area in a normal file or character special file in a DVD-RAM library device.

Creating a HiRDB file system area in a normal file

A HiRDB file system area created in a DVD-RAM library device can be manipulated in the same manner as a HiRDB file system area created on a magnetic disk. However, when considering performance and availability, Hitachi does not recommend using such file system areas for system files (SYS), work files, or list RDAREAs (WORK).

Creating a HiRDB file system area in a character special file

If you create a HiRDB file system area in a character special file in a DVD-RAM library device, you must take into consideration the sector length of the DVD-RAM library device. When the sector length is 512 or 1024 bytes, the HiRDB file system area can be manipulated in the same manner as a HiRDB file system area created in a character special file on a magnetic disk. However, when considering performance and reliability, Hitachi does not recommend using such file system areas for system files (SYS), work files, or list RDAREAs (WORK).

Notes

You should note following when the sector length is 2048 or 4096 bytes:

1. Only the files listed below can be created in a HiRDB file system area created using this procedure:
   - HiRDB files comprising RDAREAs (excluding list RDAREAs)
   - Backup files
   - Unload log files

- Unload data files

1. Execute the pdfmkfs command, then use the -s option to specify the sector length. You cannot use the -k option to specify SYS, WORK, or SVR.
2. When you use the database initialization utility (pdinit command) to define an RDAREA, specify in the page operand (page length) in the utility control statement a value that is a multiple of the sector length.
3. When you use the database structure modification utility (pdmod command) to add, extend, or reinitialize an RDAREA, specify in the page operand (page length) in the utility control statement a value that is a multiple of the sector length.

   Hitachi cautions against omitting the page operand during reinitialization of an RDAREA. If the page length of the RDAREA before reinitialization is not a multiple of the sector length, the RDAREA cannot be reinitialized.
4. When you use the pdfrstr command to restore a HiRDB file, the record length of the HiRDB file to be restored must be a multiple of the sector length of the HiRDB file system area into which the file will be restored. The record length of a HiRDB file can be checked with the pdfls command.
5. When you use the database recovery utility (pdrstr command) to recover an RDAREA, the page length of the RDAREA to be recovered must be a multiple of the sector length of the HiRDB file system area. If the page length is not a multiple of the sector length, the RDAREA cannot be recovered. For example, if the medium that a HiRDB file system area is created on is being replaced because of a physical failure, check that the sector length of the HiRDB file system area on the new medium is the same as on the medium being replaced.

(3) RDAREA opening trigger attribute

When you define an RDAREA on a DVD-RAM library device, use SCHEDULE as the RDAREA opening trigger attribute. If you use INITIAL (the default), all HiRDB files will be opened when HiRDB is restarted because RDAREA information resides in memory. When opening processes accumulate on a DVD-RAM library device, disk swapping may occur often and result in the HiRDB startup process timing out. By using the DEFER attribute, you can avoid accumulation of opening processes when HiRDB starts, but they will accumulate when HiRDB terminates normally.

Also, when you use a normal file with the INITIAL or DEFER attribute, reactivation will not be performed smoothly because fsck will be executed on all media HiRDB opens during the next OS startup if a power outage or restart occurred while HiRDB was still active.

RDAREA opening trigger attributes are not suitable for operations with DVD-RAM library devices, because the RDAREA opening trigger attribute is fixed to INITIAL

for the master directory RDAREA, data directory RDAREA, data dictionary RDAREAs, data dictionary LOB RDAREAs, and registry RDAREA. You should create these RDAREAs on a magnetic disk.

To change the opening trigger attribute of RDAREAs to SCHEDULE, specify pd_rdarea_open_attribute_use=Y and take one of the following actions depending on the number of RDAREAs to be assigned the SCHEDULE attribute:

When changing many RDAREAs to the SCHEDULE attribute

Specify pd_rdarea_open_attribute=SCHEDULE. This specification makes all RDAREAs in the entire system valid except for the master directory RDAREAs, data directory RDAREAs, data dictionary RDAREAs, data dictionary LOB RDAREAs, and the registry RDAREA.

When changing a small number of RDAREAs to the SCHEDULE attribute

Define each such RDAREA by specifying it in a control statement of one of the following utilities:

- Specify SCHEDULE in the open attribute operand of the database initialization utility (pdinit command)
- Specify SCHEDULE in the open attribute operand of the database structure modification utility (the pdmod command)

Such a specification is valid only for the specified RDAREA. Also, when used in conjunction with specification of the pd_rdarea_open_attribute operand, the specification for the utility has the higher priority.

Reference note:

Specification of an opening trigger attribute in this manner does not take effect immediately after the database structure modification utility is used to add an RDAREA because the opening trigger attribute starts out as INITIAL. To activate specification of the opening trigger attribute, you must restart HiRDB. The specified opening trigger becomes effective regardless of the startup mode.

C. Information Needed for Troubleshooting

The information needed for troubleshooting can be divided broadly into operating system information and HiRDB information. Operating system information is collected using operating system commands. HiRDB information is collected using HiRDB commands.

If you use problem-solving support or Q&A support services, the information explained in Table C-1 may be needed. Table C-1 explains the information needed for troubleshooting. The priority for collecting information is divided into three categories: a performance problem, a no-response, and abnormal termination. There are seven priority levels, with 1 being the highest priority and 7 being the lowest.

Table C-1: Information needed for troubleshooting

No.	Category	Collected information	Collection method	Prf.	NR	AT
1	OS	syslogfile	Use an operating system function (command).	1	1	1
2		CPU utilization and device status	Use the OS's `sar` command. For details about this command, see the documentation for the operating system.	3	4	3
3		CPU activity or memory status for the process	Use the OS's `top` command. For details about this command, see the documentation for the operating system.	3	4	3
4		Virtual memory information	Use the OS's `vmstat` command. For details about this command, see the documentation for the operating system.	3	4	3
5		Network status information	Use the OS's `netstat` command. For details about this command, see the documentation for the operating system.	3	4	3
6	HiRDB	HiRDB error information	Use the `pdgeter` command. Save the output information onto a medium such as a DAT.	2	2	2
7		Error log files	Error log files are output to the directory `$PDDIR/spool/errlog`.	2	2	2
8		Command log file	Command log files are output to the directory `$PDDIR/spool/cmdlog`.	2	2	2

No.	Category	Collected information	Collection method	Prf.	NR	AT
9		HiRDB system definition information	Save files in the directory `$PDDIR/conf` on a medium such as a DAT.	4	5	4
10		Table and index definition information	Save files describing definition SQL used when defining tables or indexes on a medium such as a DAT.	4	—	—
11		Statistical information	Use the statistical information output start and end commands (`pdstbegin` and `pdstend`) to collect the information listed below: • Statistical information on system operations (`sys`) Recommended interval: 1 hour • Statistical information on the global buffer pool (`buf`) • Statistical information for HiRDB files on database manipulation (`fil`) This information is output to the directories `$PDDIR/spool/pdstj1` and `pdstj2`. After collecting the statistical information, use the statistics analysis utility (`pdstedit` command) to edit the statistical information.	6	—	—
12		Server communication control information	Use the `pdls -d rpc -a` command. Execute this command at 60-second intervals and output to the target file using an appending redirect (>>). Save this file on a medium such as a DAT. • Recommended collection interval: 60 seconds • Disk space needed to collect information one time: approximately 62 KB.	5	4	4

No.	Category	Collected information	Collection method	Prf.	NR	AT
13		Information on the status of server transactions	Use the `pdls -d tm -a` command. Execute this command at 60-second intervals and output to the target file using an appending redirect (>>). Save this file on a medium such as a DAT. • Recommended collection interval: 60 seconds • Disk space needed to collect information one time: approximately 30 KB	5	6	5
14		Information on the status of server processes	Use the `pdls -d prc -a` command. Execute this command at 60-second intervals and output to the target file using an appending redirect (>>). Save this file on a medium such as a DAT. • Recommended collection interval: 60 seconds • Disk space needed to collect information one time: approximately 28 KB	5	6	5
15		Information on server lock	Use the `pdls -d lck -a` command. Execute this command at 60-second intervals and output to the target file using an appending redirect (>>). Save this file on a medium such as a DAT. • Recommended collection interval: 60 seconds • Disk space needed to collect information one time: approximately 62 KB	5	—	—
16		Information on the global buffer usage status	Use the `pdbufls` command. Execute this command at 60-second intervals and output to the target file using an appending redirect (>>). Save this file on a medium such as a DAT. • Recommended collection interval: 60 seconds • Disk space needed to collect information one time: approximately 3 KB	5	—	—

No.	Category	Collected information	Collection method	Prf.	NR	AT
17		SQL trace files and error log files	SQL trace files and error log files are output to the directory specified by the `PDCLTPATH` operand of the client environment definition. When the `PDCLTPATH` operand is omitted, these files are output to the directory in which the AP is currently running. Save the output files on a medium such as a DAT. File names start with `pderr` or `pdsql`.	—	6	5
18		System log files	Use `pdlogunld` to unload the system log. Save the unload log files on a medium such as a DAT.	6	7	6
19		Core files	If a utility stops responding, use the following procedure to collect the core files: Procedure 1. Use the `pdls -d prc` command to check the server process ID of the utility. 2. Use the `pdcancel -d` command to cancel the process checked in step 1.	—	3	—

Legend:

Prf.: Performance problem

NR: No-response

AT: Abnormal termination

—: Collection of this information is not necessary.

Notes

- The necessary disk space is a reference value. The actual value will depend on the system configuration.
- Available disk space is suppressed because files that can be added to by redirecting output information increase in size incrementally. Therefore, create a general-purpose shell script that swaps files and reuses them in a specific generation.

D. Notes on Running HiRDB Around the Clock

This appendix explains the procedures for and provides notes about running HiRDB continuously around the clock. These procedures and notes are listed below.

1. System reconfiguration command (`pdchgconf` command)
2. Specifying HiRDB system definitions
3. Making backups
4. Reorganizing databases
5. Reusing used free pages and free space within pages
6. Expanding RDAREAs
7. Dynamic updating of global buffers
8. Deleting troubleshooting information
9. System switchover facility
10. Program maintenance facility (upgrade to update version)
11. Recovery-unnecessary front-end server (HiRDB/Parallel Server only)

D.1 System reconfiguration command (pdchgconf command)

It is usually necessary to terminate HiRDB in order to modify a HiRDB system definition (other than a UAP environment definition). However, by using the system reconfiguration command, you can modify HiRDB system definitions while HiRDB is still active. The actions listed below can be executed while HiRDB is active:

- Adding, removing, or moving units
- Adding, removing, or moving servers
- Adding system files
- Adding, removing, or modifying global buffers

The system reconfiguration command is very convenient in the case of a system that runs around the clock without interruption. For details about using this command, see the references below.

- For details about using the system reconfiguration command about modifying HiRDB system definitions, see *9.2 Modifying HiRDB system definitions while HiRDB is running (system reconfiguration command)*.
- For details about adding, removing, or moving units or servers, see *11. Modifying the System Configuration*.

- For details about adding system files, see *3.5.8 Adding a new system log file*, *4.3.5 Adding a new synchronization point dump file*, or *5.2.6 Adding a new status file*.
- For details about adding, removing, or modifying a global buffer, see *15.2 Creating an RDAREA (RDAREA addition)*.

Note that HiRDB Advanced High Availability must be installed in order to use the system reconfiguration command.

D.2 Specifying HiRDB system definitions

To modify an operand in a HiRDB system definition, HiRDB must be terminated. When you wish to modify any of the operands explained in Table D-1, you should determine in advance the new values to be specified. Care is required in specifying values for the operands described in Table D-1 in the case of a HiRDB that operates around the clock.

When HiRDB Advanced High Availability is installed, HiRDB system definitions can be modified while HiRDB remains active because the system reconfiguration command becomes available for use.

Table D-1: Operands requiring caution when specifying new values while HiRDB is running around the clock

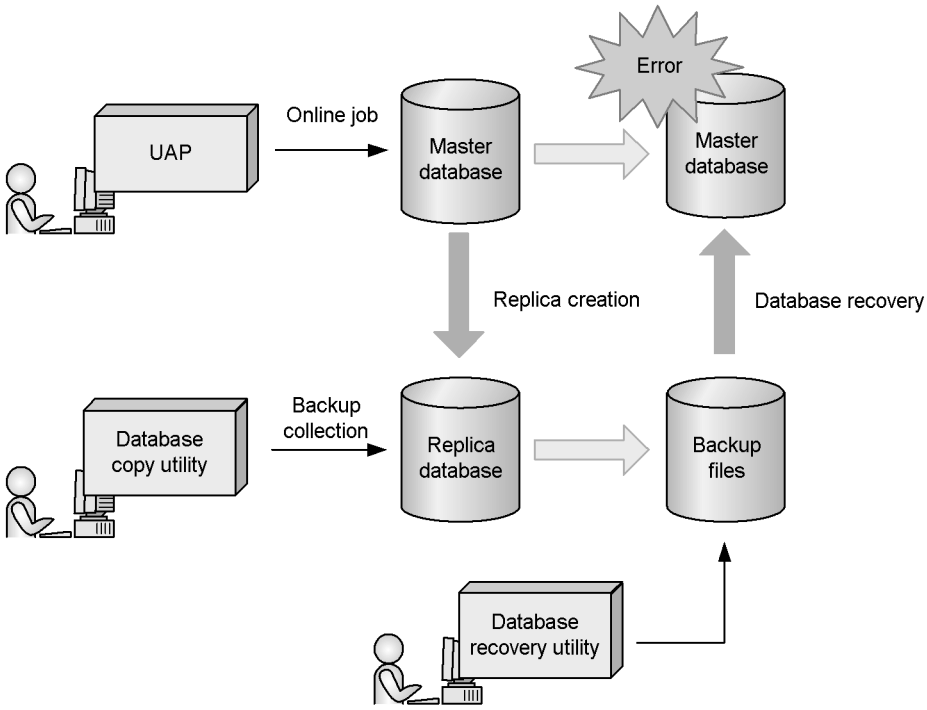
Operand	Caution
pd_max_users pd_max_server_process pd_max_bes_process pd_max_dic_process	When the number of concurrently connected clients is expected to increase, specify values for these operands that provide a sufficient margin. Also, reducing the value of the pd_process_count operand makes it possible to minimize the number of unnecessary (unconnected) processes. When the pd_max_server_process operand is omitted, HiRDB calculates a value for it automatically.
pd_stj_file_size	The file size depends on the statistical information acquired, but Hitachi recommends specifying a value greater than the default (which is 1024 kilobytes).
pd_spd_syncpoint_skip_limit	When specifying this operand, be aware of the amount of data handled by batch jobs to be executed.
pd_lck_pool_size	When the job mode changes and the number of resources to be locked might increase until the resources are committed, take that potential increase into account when determining the value of this operand.
pd_lck_until_disconnect_cnt	When the job mode might change, take that into account when determining the value of this operand.

Operand	Caution
pd_sds_shmpool_size pd_dic_shmpool_size pd_bes_shmpool_size	The values specified in these operands are calculated from the values of other operands in this list. Therefore, when you change the value specified in any of those other operands, you must re-calculate the values to be specified in these operands. For details about which other operands affect these operands, see *Formula for calculating the shared memory used by single servers* or *Formula for calculating the shared memory used by each server* in the manual *HiRDB Version 8 Installation and Design Guide*. When these operands are omitted, HiRDB calculates values for them automatically.
pdlogadfg -d sys pdlogadpf -d sys	Be sure to provide sufficient extra overall space for system log files. Use these operands to define reserved files, and you can add content to system log files as necessary.
pd_svr_castoff_size	Specifying this operand while processes reside in the system increases the amount of memory used. Therefore, confirm that specifying this operand is necessary.
pd_log_sdinterval	Do not specify 0 as the elapsed time.

D.3 Making backups

Hitachi recommends using the inner replica facility because you can reference and update the database while making a backup. Figure D-1 shows how a backup is made using the inner replica facility.

Figure D-1: Making a backup using the inner replica facility



Explanation

A backup is made from a replica database, which means that the master database can continue to be referenced and updated while the backup is being made.

When an error occurs in the master database, the backup of the replica database can be used to recover the master database.

For details about the inner replica facility, see the manual *HiRDB Staticizer Option Version 7 Description and User's Guide*.

When the inner replica facility is not used

When the inner replica facility is not used, Hitachi recommends using the updatable mode for making backups (specifying `-M s`). In such a case, you should make note of the following items:

- The `pdcopy` command may timeout waiting for a lock-release from an update transaction while a backup is being made. To avoid this situation, either increase the lock-release wait time of the `pdcopy` command (`-j` option) or execute the `pdcopy` command when no update transactions have

been issued.

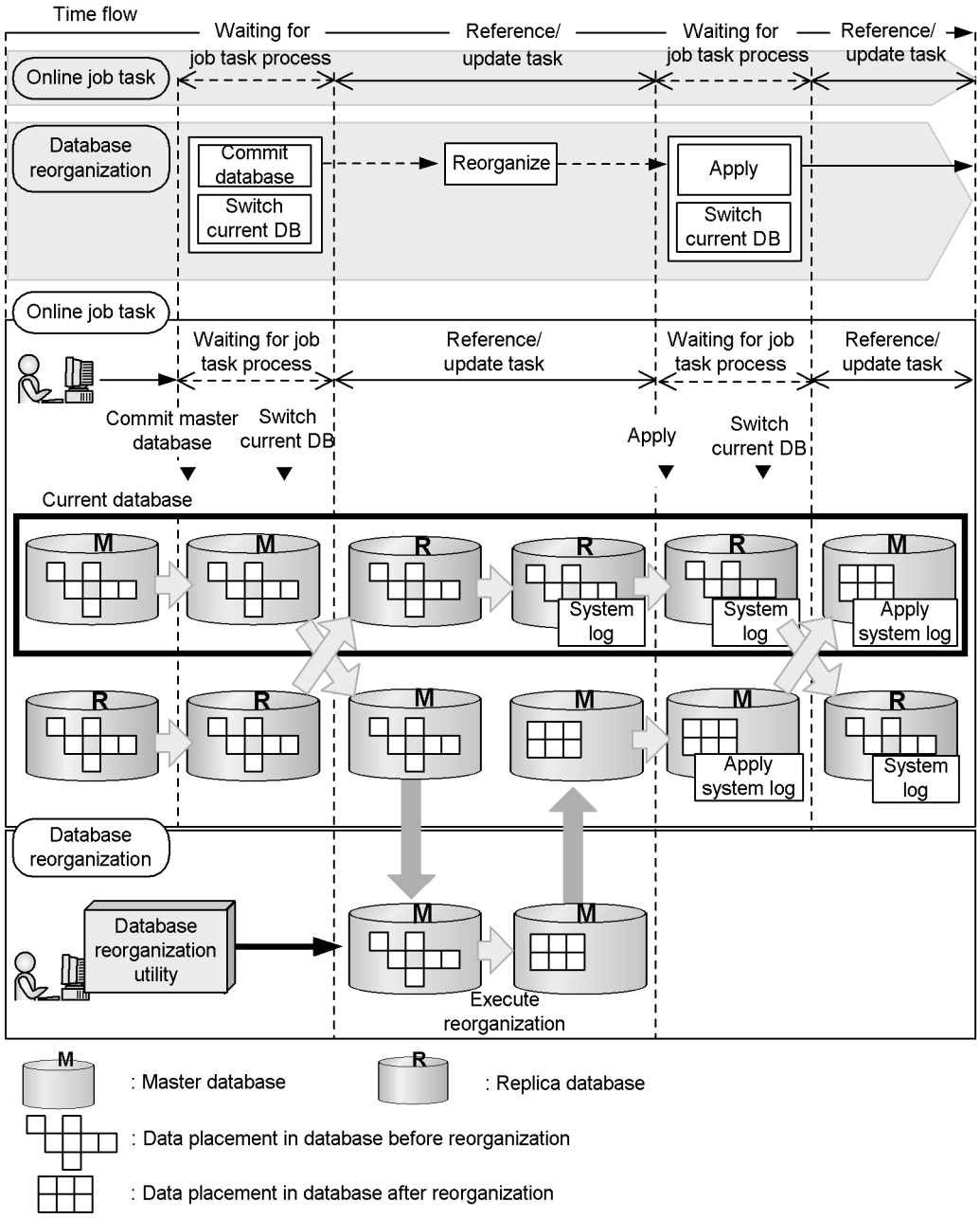
- The database cannot be recovered from the backup alone. The database must be recovered from a synchronization point subsequent to when the backup was made. Therefore, the backup and the system log from the most recent synchronization point subsequent to when the backup was made are required in order to recover the database. When a database is recovered to the most recent synchronization point, the run ID and generation number of the system log file needed for recovering the target RDAREAs is output to the process results output file for the database copy utility. You must manage all subsequent system log files.

  When you recover a database up to the first synchronization point after the backup was made, you must manage the system log files from the last synchronization point before the backup was made to the first synchronization point after the backup was made.

D.4 Reorganizing databases

Hitachi recommends using the inner replica facility because you can reference and update the database while reorganizing it. This is called *updatable online reorganization*. If the inner replica facility is not used, the database cannot be referenced or updated while it is being reorganized. Using the inner replica facility minimizes the impact on online job tasks because job processing uses the replica database while the master database is being reorganized. Figure D-2 shows how a database is reorganized using the inner replica facility.

Figure D-2: Database reorganization using the inner replica facility



Explanation

The database reorganization utility (`prdorg` command) is used to reorganize the master database, and the replica database is used for the online job tasks.

Before reorganizing the current database, another database is committed and the current database is switched to that database. The current database cannot be referenced or updated during this time. After the current database is switched, the master database is reorganized and the replica database is used for performing online job tasks. After reorganization is complete, the system log for the replica database becomes input information and the update processes performed during the reorganization are applied to the master database. This is called *reflection processing*.

For details about the inner replica facility, see the manual *HiRDB Staticizer Option Version 7 Description and User's Guide*.

When the inner replica facility is not used

When the inner replica facility is not used, Hitachi recommends using the `pdhold` command to place RDAREAs storing a reorganization table in hold status so UAPs do not wait for lock-release. This is because EX (the update mode) locks the table during reorganization. UAPs that were accessing in the reorganization table are terminated immediately without waiting for lock-release because RDAREAs in hold status cannot be accessed. Set in the reorganization table the amount of time RDAREAs will not be accessible, and. reorganize the table within that amount of time.

D.5 Reusing used free pages and free space within pages

Job tasks that repeatedly insert or delete rows in a table reduce the storage efficiency of tables or indexes and become a factor in reducing performance. Therefore, you must regularly use the database reorganization utility to reorganize tables or indexes. However, you must stop job tasks that access the table to be reorganized because that table or index cannot be accessed by other UAPs or by utilities while the database reorganization utility is executing. If job tasks cannot be stopped or indexes cannot be reorganized, you can use the following facilities:

- Release used free pages
- Free space reusage facility

Releasing used free pages makes it possible to reuse used free pages. Using the free space reusage facility makes it possible to reuse the free space in a page. Combining these facilities makes it possible to increase the time between reorganizations of tables and indexes because data storage efficiency in increased and performance degradation is minimized.

(1) Release used free pages

When you delete a large amount of table data, such as with a batch job for example, there are cases where some of the pages (data pages) storing that table data become used free pages. Also, if you have defined an index, some of the pages (index pages) storing the index key values become used free pages. Executing the free page release utility (`pdreclaim` command) makes it possible to change these used free pages into unused pages which can then be reused. This is called *releasing used free pages*.

For details about releasing used free pages, see *15.9 Re-using used free pages and used free segments*.

(2) Free space reusage facility

When the `INSERT` process cannot store data in a segment, data is stored in a free area in a used segment (a free area on a page) without allocating a new unused segment. This is called the *free space reusage facility*. Using this facility makes it possible to reuse free areas that became unavailable when data was deleted.

To use the free space reusage facility, specify `SEGMENT REUSE` when you execute `CREATE TABLE`. For details about the free space reusage facility, see the manual *HiRDB Version 8 Installation and Design Guide*.

D.6 Expanding RDAREAs

Adding data may cause a space shortage in an RDAREA. The methods for ensuring a space shortage does not occur are explained below.

(1) RDAREA automatic extension

When a space shortage occurs in an RDAREA, segments can be added automatically to increase the size of the RDAREA, provided that space is available in the HiRDB file system area. This is called *RDAREA automatic extension*. For details about the RDAREA automatic extension, see *15.7 RDAREA automatic extension*.

To automatically extend RDAREAs, there must be free space in the HiRDB file system area. Therefore, include sufficient extra space when estimating the HiRDB file system area size.

Note that stopping job tasks is not necessary for extending RDAREAs automatically.

(2) RDAREA expansion

There is a limit to the number of times RDAREA automatic extension explained in (1) above can be performed. When automatic extension is no longer possible, you can expand the RDAREA manually. For details about RDAREA expansion, see *15.3 Increasing the size of an RDAREA (RDAREA expansion)*.

A maximum of 16 HiRDB files comprise an RDAREA, so no more than 16 HiRDB files can be added to an RDAREA.

Stopping job tasks is not necessary in order to expand an RDAREA.

(3) RDAREA reinitialization

There is a limit to the number of times RDAREA expansion explained in (2) above can be performed. When expansion is no longer possible, you can reinitialize the RDAREA and increase its size. For details about RDAREA reinitialization, see *15.4 Increasing the size of an RDAREA or modifying its attributes (RDAREA reinitialization)*.

Note that an RDAREA that is being reinitialized cannot be accessed. Therefore, you should us method (1) or (2) above whenever possible.

D.7 Dynamic updating of global buffers

Hitachi recommends that you install HiRDB Advanced High Availability and specifying `Y` in the `pd_dbbuff_modify` operand. This makes it possible to add, modify, and delete global buffers while HiRDB is active. This capability is called *dynamic updating of global buffers*. You should use dynamic updating of global buffers in the following situations:

- For allocating global buffers to an added RDAREA
- For modifying a global buffer to be allocated to an RDAREA
- For modifying the tuning results or definition of a global buffer

Once HiRDB has terminated, you must change the value specified in the `pdbuffer` operand, because dynamically modified global buffer information becomes invalid when normal termination or planned termination occurs. However, if you use the system reconfiguration command (`pdchgconf` command), you can change the value specified in the `pdbuffer` operand while HiRDB is running.

For details about dynamic updating of global buffers, see *9.3 Adding, modifying, and deleting global buffers while HiRDB is running (dynamic updating of global buffers)*. For details about using the system reconfiguration command, see *9.2 Modifying HiRDB system definitions while HiRDB is running (system reconfiguration command)*.

D.8 Deleting troubleshooting information

When a server process or a client is terminated forcibly, HiRDB outputs troubleshooting information to the `$PDDIR/spool` directory. Also, when you terminate a command or utility while it is executing by pressing **Ctrl-C**, for example, temporary work files generated by the command or utility remain in the `$PDDIR/tmp` directory. If these troubleshooting information and temporary work files are not deleted, they may eventually cause a space shortage on the disk containing the HiRDB directory. To prevent a space shortage from occurring on the disk containing the HiRDB directory and causing HiRDB abnormal termination, HiRDB regularly deletes the files listed below:

- Troubleshooting information files (files in the $PDDIR/spool directory)
- Temporary work files (work files in the ($PDDIR/tmp directory)

These files are normally deleted every 24 hours. This deletion interval can be changed with the pd_spool_cleanup_interval operand. In addition, files that were output a specific number of days in the past can be deleted by specifying an appropriate value in the pd_spool_cleanup_interval_level operand.

You can also delete troubleshooting information files and temporary work files (files in the $PDDIR/tmp directory) with the pdcspool command.

You select the troubleshooting information to be deleted by specifying it in the pdcspool command option or the pd_spool_cleanup_interval_level operand.

Note:

- There are situations in which troubleshooting information files output by a command or utility that a user other than the HiRDB administrator executed cannot be deleted. In such cases, a user who has troubleshooting information file deletion privileges uses the OS's rm or other command to delete the files.
- If a HiRDB operation command or utility is aborted when the TMPDIR environment variable is specified, there are situations in which HiRDB creates files beginning with the letters pdcmd or plcmd in the directory specified in TMPDIR. If these files are not deleted even after the HiRDB operation command or utility has terminated, the OS's rm or other command can be used to delete them.

D.9 System switchover facility

If you use the system switchover facility, Hitachi recommends that you include the facilities listed below. Using these facilities minimizes job task downtime by reducing the amount of time required to implement system switchover.

- Rapid system switchover facility
- Standby-less system switchover facility
- Transaction queuing facility (HiRDB/Parallel Server)

(1) Rapid system switchover facility

The *rapid system switchover facility* activates server processes and system servers in the HiRDB standby system in advance, instead of having to wait for them to be activated during system switchover. This facility reduces the system switchover time by the amount of time that is saved by not having to activate server processes and

system servers during system switchover. For details about the rapid system switchover facility, see *25.18.2 Rapid system switchover facility*.

(2) Standby-less system switchover facility

In contrast to the standby system switchover facility that prepares an HiRDB standby system, it is not necessary to have a standby system when you use the *standby-less system switchover facility*. This facility transfers processing to another active unit instead of switching to a HiRDB standby system when an error occurs. This facility reduce the system switchover time by using an active unit.

(3) Transaction queuing facility (HiRDB/Parallel Server only)

When system switchover occurs in a unit with a back-end server or dictionary server, the back-end server or dictionary server cannot accept transactions until switchover is completed. This means that transactions processed by a back-end server or dictionary server that is involved in switchover result in an error.

The *transaction queuing facility* queues such transactions on the front-end server until system switchover is completed so that errors are not generated. This facility reduces transaction errors during system switchover. For details about the transaction queuing facility, see *25.19 Transaction queuing facility*.

D.10 Program maintenance facility (upgrade to update version)

When you upgrade to the update level (called *upgrading to the update HiRDB version*), it is not necessary to terminate HiRDB normally. Bu using the program maintenance facility, the version is upgraded to the update level while HiRDB is still active.

The update level version here refers to the underlined part of the revision number, as in 07-00-/A. For example, when upgrading from 07-00-/A to 07-00-/B, it is not necessary to terminate HiRDB normally. This makes it possible to upgrade to the update level without stopping job tasks. For details about the program maintenance facility, see the manual *HiRDB Version 8 Installation and Design Guide*.

D.11 Recovery-unnecessary front-end server (HiRDB/Parallel Server only)

When a front-end server unit terminates abnormally, any transaction that was being executed from this front-end server may end up in uncompleted status. A transaction that is thrown into uncompleted status may still be maintaining a lock on the database, thus preventing other transactions from referencing or updating the database. Ordinarily, a transaction in uncompleted status must be completed from the front-end server, so you must first recover the front-end server unit from the error and then restart the unit.

When a recovery-unnecessary front-end server is used, a transaction in uncompleted status is completed automatically when the front-end server unit terminates

abnormally (the transaction is completed without waiting for the front-end server to restart). You can install a front-end server unit in an application server that does not use system switchover, and use reduced operation for the front-end server unit.

Whereas normally the `pdrplstart` and `pdrplstop` commands can be executed only when all units and servers are running, use of a recovery-unnecessary front-end server means that these commands will execute even though a recovery-unnecessary front-end server or recovery-unnecessary unit is stopped.

For details about recovery-unnecessary front-end servers, see the manual *HiRDB Version 8 Installation and Design Guide*.

To use a recovery-unnecessary front-end server, HiRDB Non Recover FES must be installed.

E. Using Performance Improvement Facilities

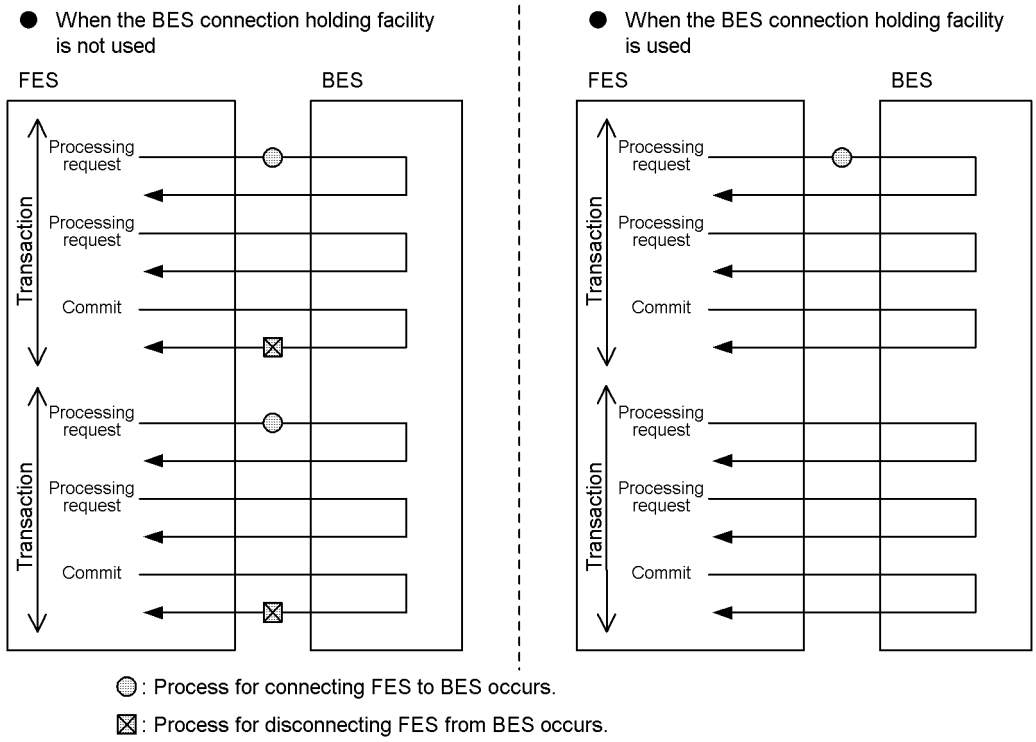
This Appendix describes facilities designed to improved system performance and explains their use.

E.1 BES connection holding facility (HiRDB/Parallel Server only)

(1) About the BES connection holding facility

When a HiRDB/Parallel Server is used, the front-end server must be connected to a back-end server to start a transaction. Consequently, in a system with a high volume of traffic that requires high throughput, this connection process can become a performance bottleneck. When the BES connection holding facility is used, the connection between the front-end server and a back-end server is not terminated when a transaction terminates. This means that if a subsequent transaction will use the same back-end server, no connection processing is needed between the front-end server and the back-end server. In other words, the BES connection holding facility eliminates a performance bottleneck by eliminated the need for connection processing between the front-end server and a back-end server. Figure E-1 shows the transaction flow when the BES connection holding facility is used.

Figure E-1: Transaction flow when the BES connection holding facility is used



Reference note:

- Although the BES connection holding facility also reduces the number of processes for disconnecting the front-end server from a back-end server, this does not have much effect on system performance.
- The BES connection holding facility cannot hold the connection between the front-end server and the dictionary server. When a transaction involving the dictionary server is terminated, the connection between the front-end server and the dictionary server is terminated.

(2) Application criteria

1. Because the time required for connecting the front-end server to a back-end server is normally between several milliseconds and dozens of milliseconds, use of the BES connection holding facility is appropriate in systems in which transaction execution time is relatively short (between a few dozen milliseconds and 2 seconds). Using the BES connection holding facility in systems in which

transactions take a long time to execute cannot be expected to improve performance.

2. If HiRDB is always connected in systems described below, the connection between the front-end server and a back-end server may be held even though the back-end server is not being used. Consequently, a shortage may occur in resources such as memory, sockets, and communication ports (a high load level is maintained for each type of resource). Therefore, care must be exercised when using the BES connection holding facility in such an environment.
   - Systems with a low transaction load
   - Systems in which specific back-end servers are more used heavily than others (do not use the BES connection holding facility with a back-end server that is used infrequently).
3. When the HiRDB XA library is used, one connection may equal one transaction depending on the transaction manager used. In such a case, using the BES connection holding facility results in no benefit. For the list of XA libraries that can be used, see the list of libraries used by each transaction manager in the manual *HiRDB Version 8 UAP Development Guide*.

(3) Environment setting

You must set up the following environment to use the BES connection holding facility:

- Set the BES connection holding facility
- Set a BES connection holding period
- Set a maximum client wait time
- Set the number of back-end server processes

(a) Setting the BES connection holding facility

To use the BES connection holding facility, you must specify either (or both) of the following operands:

1. `pd_bes_connection_hold` operand
2. `PDBESCONHOLD` operand in a client environment definition

When the `pd_bes_connection_hold` operand is specified, the BES connection holding facility is applied to all UAPs. When the `PDBESCONHOLD` operand of a client environment definition is specified, the BES connection holding facility is applied only to UAPs executed from the client for which the `PDBESCONHOLD` operand was specified. The `PDBESCONHOLD` operand has higher priority than the `pd_bes_connection_hold` operand. Operand combinations are explained below.

- When `pd_bes_connection_hold = Y` is specified

  The BES connection holding facility is applied to all UAPs.

- When `PDBESCONHOLD = YES` is specified

  The BES connection holding facility is applied only to UAPs executed from the client for which the operand is specified.

- When `pd_bes_connection_hold = Y` and `PDBESCONHOLD = NO` are both specified

  The BES connection holding facility is applied to all UAPs except for UAPs executed from a client for which `PDBESCONHOLD = NO` is specified.

(b) Setting the BES connection holding period

Specify how long the BES connection is to be held open after a transaction terminates. When the specified BES connection hold time expires, the connection between the front-end server and the back-end server will be terminated when the next transaction that is executed terminates.

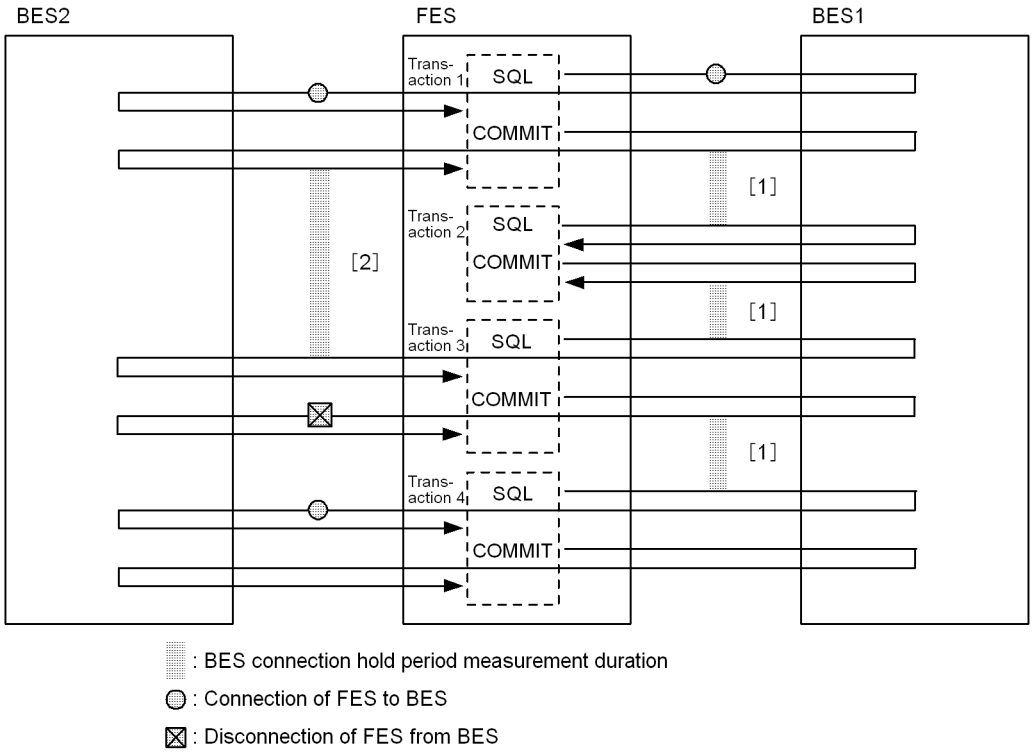
How the BES connection holding period is measured

Measurement of the BES connection holding period starts and ends as explained below:

- Measurement starting point: When transaction processing terminates
- Measurement ending point: When the next transaction's processing starts

Figure E-2 shows how the BES connection holding is measured and the processing performed by HiRDB.

Figure E-2: Measurement of the BES connection holding period and processing by HiRDB



Explanation

1. Because interval [1] is within the BES connection holding period, the connection between the FES and BES1 remains open.
2. Because interval [2] exceeds the BES connection holding period, the connection between the FES and BES2 will be terminated when the next transaction (transaction 3) terminates. When the next transaction to be executed after that (transaction 4) starts, the FES and BES2 must be connected anew.

Notes

Note that the connection between the front-end server and a back-end server does not terminate as soon as the BES connection holding period expires. The following summarizes the processing performed by HiRDB in the event the BES connection holding period expires:

1. At the point when the BES connection holding period expires

HiRDB does not take any action.

2. When a transaction starts

   Because the connection between the front-end server and the back-end server is still being held, no processing is required to connect the front-end server to the back-end server.

3. When this transaction terminates

   Terminates the connection between the front-end server and the back-end server.

4. When the next transaction starts

   Connects the front-end server to the back-end server.

Specification guidelines for the BES connection holding period

- For a system in which the UAP connection period is brief, set the BES connection holding period to `0`.
- For a system that is always connected or in which the UAP connection period is prolonged, do not specify the BES connect hold period; instead, use the default value (1 second).

Operation when 0 is specified as the BES connection holding period

When `0` is specified as the BES connection holding period, time is not monitored and the connection between the front-end server and the back-end server is held open indefinitely. However, such a connection will be terminated in the cases explained in *(4)(b) Cases in which a connection hold is cancelled*.

Specifying the BES connection holding period

To specify the BES connection holding period, you specify either (or both) of the following operands:

- `pd_bes_conn_hold_trn_interval` operand
- `PDBESCONHTI` operand in a client environment definition

The BES connection holding period specified with the `pd_bes_conn_hold_trn_interval` operand is applied to all UAPs. In contrast, the BES connection holding period specified with the `PDBESCONHTI` operand in a client environment definition is applied only to UAPs executed from the client for which the `PDBESCONHTI` operand was specified. The `PDBESCONHTI` operand has higher priority than the `pd_bes_conn_hold_trn_interval` operand. Operand combinations are explained below.

- When pd_bes_conn_hold_trn_interval = 10 is specified

  The BES connection holding period is 10 seconds for all UAPs.

- When PDBESCONHTI = 10 is specified

  The BES connection holding period is 10 seconds for UAPs executed from the client for which this operand is specified.

- When pd_bes_conn_hold_trn_interval = 10 and PDBESCONHTI = 20 are specified

  The BES connection holding period is 20 seconds for UAPs executed from the client for which the PDBESCONHTI operand is specified and 10 seconds for all other UAPs.

(c) Setting the maximum client wait time

If a back-end server unit being held in connected status terminates abnormally, the next transaction to be executed by the back-end server may enter no-response status. Also, when the transaction queuing facility is used, the retry processing specified by the pd_ha_trn_restart_retry_time operand may not occur, resulting in a no-response (a no-response may also occur during planned system switchover). For this reason, you should use the PDCWAITTIME operand in the client environment definition to specify a maximum client wait time so that situations in which transactions stop responding can be handled. If there is no response from a transaction within the time specified by the PDCWAITTIME operand, the transaction is cancelled.

(d) Setting the number of back-end server processes

To use the BES connection holding facility, you should set the number of individual back-end servers to be greater than the total number of front-end servers in the system. To do so, specify values for the system definition operands so that the following condition is satisfied:

value-of-pd_max_bes_process-operand ≥

value-of-pd_max_users-operand × *number-of-front-end-servers*

If this condition is not satisfied, a shortage may occur in the number of available back-end server processes, and, as a result, the message queue monitoring facility may cause HiRDB (or unit for a HiRDB/Parallel Server) to terminate abnormally or an SQL error may occur.

To execute utilities while UAPs are running, the number of back-end server processes you specify must also include the number needed by the utilities.

(4) Notes

(a) Do not use for definition SQLs

Do not use the BES connection holding facility for a UAP that executes a definition

SQL. If the facility is used for such a UAP, connection between the front-end server and back-end server may occur simultaneously with connection between the dictionary server and the back-end server, and, as a result, a shortage may occur in the number of available back-end server processes, causing an SQL error. You can prevent this situation from occurring by specifying `NO` in the `PDBESCONHOLD` operand in the client environment definition of any UAP that executes a definition SQL.

(b) Cases in which a connection hold is cancelled

Even when the BES connection holding facility is used, the connection between the front-end server and a back-end server is terminated unconditionally in the following cases:

- Connection between the front-end server and the client is terminated by a `DISCONNECT` statement (or `xa_close` when HiRDB XA library is used) or because the time specified by the `PDCWAITTIME` operand expires.
- Rollback (including internal rollback) occurs.
- The facility for monitoring the memory size of server processes determines that the memory usage upper limit has been exceeded.
- A large amount of memory is used internally.
- The `pdpfresh` command (without the `-f` option specified) is executed.
- The cursor of a holdable cursor is closed.

(c) Connection may not be terminated even when the BES connection holding period has expired

In the following cases, the connection between the front-end server and back-end server is not terminated even though the BES connection holding period has expired:

- A holdable cursor is used
- A local buffer specified in the UAP environment definition is used

Index

B

C

D

E

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Reader's Comment Form

We would appreciate your comments and suggestions on this manual. We will use these comments to improve our manuals. When you send a comment or suggestion, please include the manual name and manual number. You can send your comments by any of the following methods:

- Send email to your local Hitachi representative.
- Send email to the following address:
  WWW-mk@itg.hitachi.co.jp
- If you do not have access to email, please fill out the following information and submit this form to your Hitachi representative:

Manual name:	
Manual number:	
Your name:	
Company or organization:	
Street address:	
Comment:	

(For Hitachi use)