For Windows Systems
Scalable Database Server

HiRDB Version 8

Command Reference Part I

3020-6-355(E)

HITACHI

■ Relevant program products

List of program products:

For the Windows 2000, Windows XP Professional, Windows XP x64 Edition, Windows Server 2003, Windows Server 2003 x64 Edition, Windows Server 2003 R2, or Windows Server 2003 R2 x64 Edition operating system:

P-2462-7187 HiRDB/Single Server Version 8 08-00

P-2462-7387 HiRDB/Parallel Server Version 8 08-00

P-2462-7H87 HiRDB Non Recover Front End Server Version 8 08-00

P-2462-7J87 HiRDB Advanced High Availability Version 8 08-00

P-2462-7K87 HiRDB Advanced Partitioning Option Version 8 08-00

For the Windows XP x64 Edition or Windows Server 2003 x64 Edition operating system:

P-2962-7187 HiRDB/Single Server Version 8 08-00

P-2962-7387 HiRDB/Parallel Server Version 8 08-00

P-2962-1187 HiRDB/Run Time Version 8 08-00

P-2962-1287 HiRDB/Developer's Kit Version 8 08-00

For the Windows Server 2003 (IPF) operating system:

P-2862-7187 HiRDB/Single Server Version 8 08-00

P-2862-7387 HiRDB/Parallel Server Version 8 08-00

P-2862-1187 HiRDB/Run Time Version 8 08-00

P-2862-1287 HiRDB/Developer's Kit Version 8 08-00

P-2862-7H87 HiRDB Non Recover Front End Server Version 8 08-00

P-2862-7J87 HiRDB Advanced High Availability Version 8 08-00

P-2862-7K87 HiRDB Advanced Partitioning Option Version 8 08-00

For the Windows 2000, Windows XP, Windows XP x64 Edition, Windows Server 2003, or Windows Server 2003 x64 Edition operating system:

P-2662-1187 HiRDB/Run Time Version 8 08-00

P-2662-1287 HiRDB/Developer's Kit 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 *Before Installing* or *Readme file*.

■ Trademarks

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.

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.

Printed in Japan.

■ Edition history

Edition 1 (3020-6-355(E)): March, 2007

■ Copyright

Preface

This manual describes the syntax of the commands used with HiRDB Scalable Database Server HiRDB Version 8.

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 Windows systems
- A basic knowledge of SQL

The following manuals should be read before reading this manual:

- *HiRDB Version* 8 *Installation and Design Guide*
- *HiRDB Version* 8 *System Operation Guide*

Organization of this manual

This manual consists of the following 18 chapters and appendixes:

Chapter *1. Overview of Commands*

Chapter 1 explains the methods of entering commands and the descriptive format for commands.

Chapter *2. Operation Commands*

Chapter 2 explains the operation commands that can be used with HiRDB.

Chapter *3. Database Initialization Utility (pdinit)*

Chapter 3 describes the database initialization utility that defines a physical structure that, in turn, permits files to be used as a HiRDB database.

Chapter *4. Database Definition Utility (pddef)*

Chapter 4 describes the database definition utility that permits modifications of schema definitions and contents.

Chapter *5. Database Load Utility (pdload)*

Chapter 5 describes the database load utility that stores user-provided data in a table.

Chapter *6. Interactive SQL Execution Utility*

Chapter 6 explains the interactive SQL execution utility that can interactively execute SQL statements.

Chapter *7. Database Structure Modification Utility (pdmod)*

Chapter 7 explains the database structure modification utility that can modify the physical structure of a database, such as by adding an RDAREA, expanding the database, or reinitializing the database.

Chapter *8. Database Reorganization Utility (pdrorg)*

Chapter 8 explains the database reorganization utility for maintaining tables and indexes.

Chapter *9. Dictionary Import/Export Utility (pdexp)*

Chapter 9 explains the dictionary import/export utility that migrates table definition information and stored procedure information between HiRDB systems.

Chapter *10. Rebalancing Utility (pdrbal)*

Chapter 10 explains the rebalancing utility that rebalances a table partitioned by the hash partitioning method.

Chapter *11. Free Page Release Utility (pdreclaim)*

Chapter 11 explains the free page release utility that enables you to release free pages (used free pages) during online operation.

Chapter *12. Global Buffer Residence Utility (pdpgbfon)*

Chapter 12 explains the global buffer residence utility that reads table data pages and index pages into a global buffer in advance, such as immediately following HiRDB startup or before starting online jobs.

Chapter *13. Integrity Check Utility (pdconstck)*

Chapter 13 explains the integrity check utility (`pdconstck`), which performs integrity checking and manipulates (sets or releases) the check pending status on tables for which referential constraints or check constraints have been defined.

Chapter *14. Statistics Analysis Utility (pdstedit)*

Chapter 14 explains the statistics analysis utility that edits statistical information, such as information about HiRDB system activities.

Chapter *15. Database Condition Analysis Utility (pddbst)*

Chapter 15 explains the database condition analysis utility, which analyzes the relationship between data dictionary tables or user RDAREAs and the storage status of the index, and summarizes and displays the results of the analysis.

Chapter *16. Optimizing Information Collection Utility (pdgetcst)*

Chapter 16 explains the optimizing information collection utility, which collects information for optimizing based on cost and stores the information in the dictionary table.

Chapter *17. Access Path Display Utility (pdvwopt)*

Chapter 17 explains the access path display utility, which displays access path information that was determined by the SQL optimization processing.

Chapter *18. Database Copy Utility (pdcopy)*

Chapter 18 explains how to use the database copy utility (`pdcopy`), which makes a backup of a database.

Chapter *19. Database Recovery Utility (pdrstr)*

Chapter 19 explains how to use the database recovery utility, which is used to recover databases and re-create log point information files.

Chapter *20. Registry Facility Initialization Utility (pdreginit)*

Chapter 20 explains the registry facility initialization utility, which can be used to register the registry facility for using plug-in utilities.

Appendix *A. List of Commands*

Appendix A provides a list of operation commands and utilities.

Appendix *B. Lock Mode During Command Execution*

Appendix B describes the resources that are locked while an operation command or utility is executing.

Appendix *C. RDAREA Status During Command Execution*

Appendix C describes the RDAREA statuses during execution of commands.

Appendix *D. Maximum Number of Concurrently Executable Utilities*

Appendix D describes the maximum number of utilities that can be executed concurrently.

Appendix *E. Creation of a UOC for Use by pdload and pdrorg*

Appendix E explains the creation of a UOC for use by `pdload` and `pdrorg`.

Appendix *F. Number of Concurrent Command Connections*

Appendix F explains the number of concurrent command connections.

Appendix *G. List of Command Return Codes*

Appendix G explains the return codes from commands.

Related publications

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

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))

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 System Operation Guide* (3000-6-354(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 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.

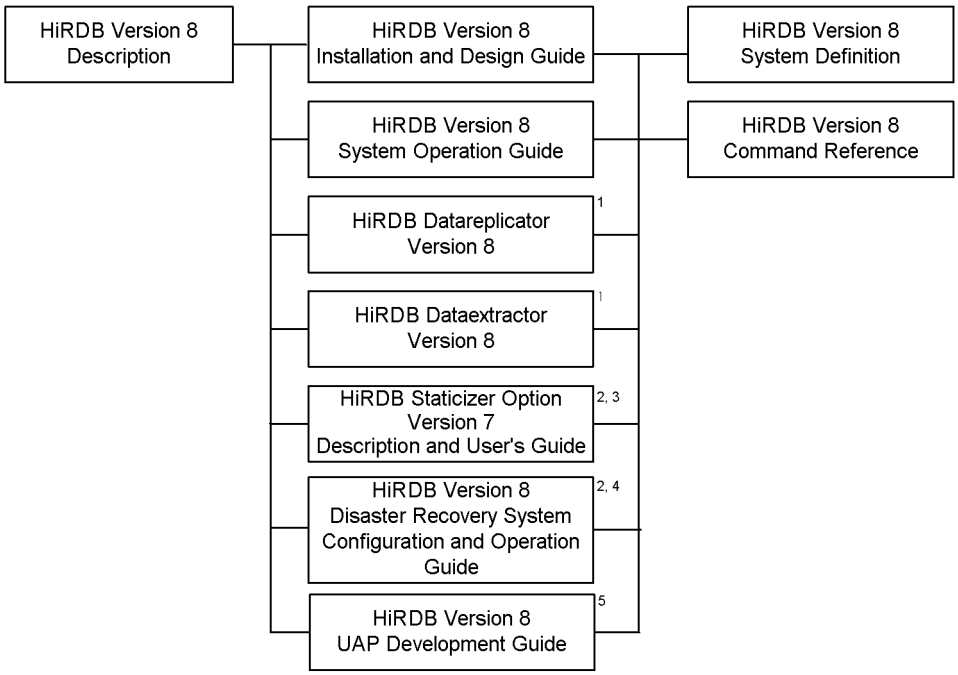
Others

- *HiRDB External Data Access Version 7 Description and User's Guide* (3000-6-284(E))
- *JP1 Version 6 JP1/VERITAS NetBackup v4.5 Agent for HiRDB License Description and User's Guide* (3020-3-D79(E))

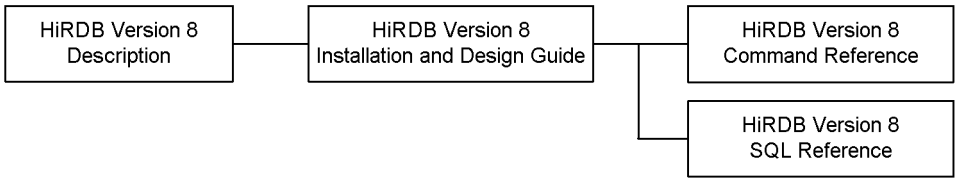
Organization of HiRDB manuals

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 7	HiRDB Datareplicator	
HiRDB Dataextractor Version 7	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 path of the Windows installation directory is represented as `%windir%`.
- The hosts file means the `hosts` file stipulated by TCP/IP. As a rule, a reference to the hosts file means the `%windir%\system32\drivers\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

Abbreviation	Full name or meaning
CMT	Cassette Magnetic Tape
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

Abbreviation	Full name or meaning
FQDN	Fully Qualified Domain Name
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

Abbreviation	Full name or meaning
LU	Logical Unit
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

Abbreviation	Full name or meaning
PP	Program Product
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

Abbreviation	Full name or meaning
VOS K	Virtual-storage Operating System Kindness
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 application log that is displayed by Windows Event Viewer is referred to as the *event log*. The following procedure is used to view the event log.

To view the event log:

1. Choose **Start**, **Programs**, **Administrative Tools (Common)**, and then **Event Viewer**.

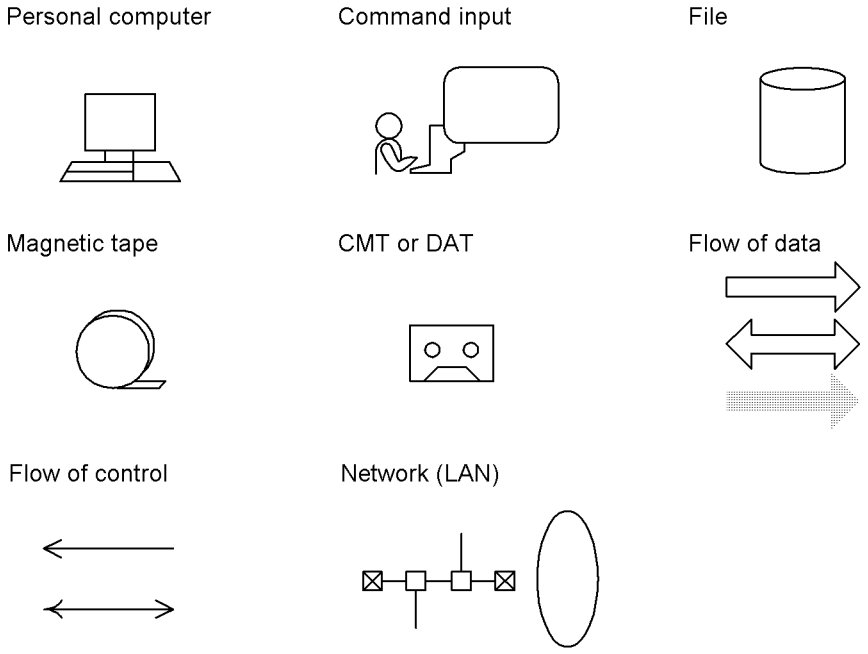
2. Choose **Log**, and then **Application**.

3. The application log is displayed. Messages with **HiRDBSingleServer** or **HiRDBParallelServer** displayed in the **Source** column were issued by HiRDB.

   If you specified a setup identifier when you installed HiRDB, the specified setup identifier follows **HiRDBSingleServer** or **HiRDBParallelServer**.

Conventions: Diagrams

This manual uses the following conventions in diagrams:



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.

Font	Convention
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.
perm	Underlined characters indicate the default value.
[ ]	Square brackets enclose an item or set of items whose specification is optional.
|	Only one of the options separated by a vertical bar can be specified at the same time.

Example font or symbol	Convention
. . .	An ellipsis (...) indicates that the item or items enclosed in ( ) or [ ] immediately preceding the ellipsis may be specified as many times as necessary.
( )	Parentheses indicate the range of items to which the vertical bar (|) or ellipsis (...) is applicable.
～	The user-specified value preceding the swung dash must be specified in accordance with the attributes following the swung dash.
<< >>	Double angle brackets enclose the default value assumed by the system when the specification is omitted.
< >	Angle brackets enclose the syntax element notation for a user-specified value.
(( ))	Double parentheses enclose the permitted range of values that can be specified.

Syntax element notations

Syntax element notations explain the types of user-specified values.

Syntax element notation	Explanation
<*alphabetic*>	Alphabetic characters (A-Z, a-z) and _ (underline)
<*alphabetic symbol*>	Alphabetic characters (A-Z, a-z), #, @, \
<*alphanumeric*>	Alphabetic characters and numeric characters (0-9)
<*alphanumeric symbol*>	Alphabetic symbols and numeric characters
<*unsigned integer*>	Numeric characters
<*hexadecimal*>	Numeric characters and (A-F, a-f)
<*identifier*>[1]	Alphanumeric character string beginning with an alphabetic character
<*symbolic name*>	Alphanumeric symbol string beginning with an alphabetic symbol
<*character string*>	String of any characters
<*path name*>[2, 3]	Includes symbolic names, backslashes (\), periods (.), spaces, and parentheses.
<*HiRDB file name*>	Character string consisting of alphabetic characters (A-Z, a-z), numeric characters (0-9), . (period), _ (underline), and @ (maximum 30 characters)

Note

Alphabetic characters are case sensitive (i.e., lowercase alphabetic characters are distinguished from uppercase alphabetic characters).

[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 operating system being used. The backslash (\) must not be used in a HiRDB file system area name.

[3] If a path name includes any spaces or parentheses, the entire path name must be enclosed in double quotation marks (").

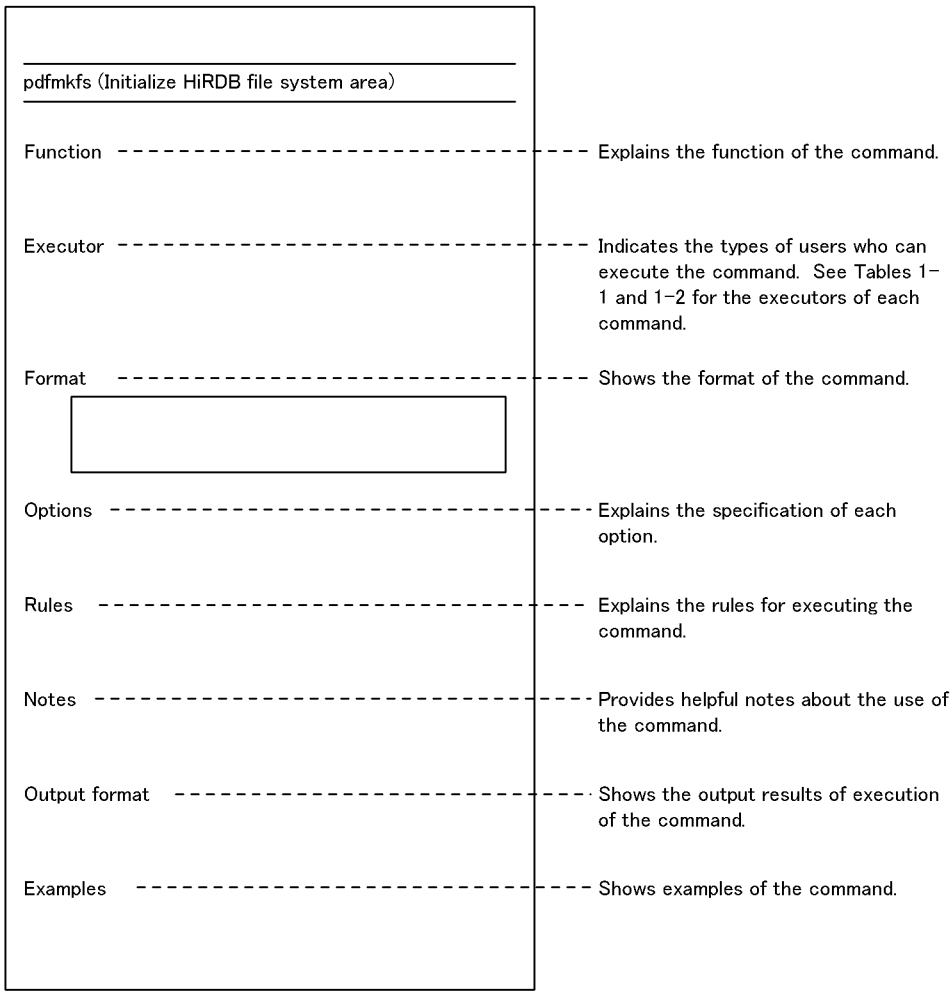
Notations used in formulas

The following notations are used in computational expressions:

Notation	Explanation
↑ ↑	Round up the result to the next integer. Example: The result of ↑ 34 ÷ 3 ↑ is 12.
↓ ↓	Discard digits following the decimal point. Example: The result of ↓ 34 ÷ 3 ↓ is 11.
MAX	Select the largest value as the result. Example: The result of max(10, 2 × 4, 3 + 8) is 11.
MIN	Select the smallest value as the result. Example: The result of min(10, 2 × 4, 3 + 8) is 8.

Conventions in command explanations

This manual explains each command in the format shown as follows; note that this format may differ slightly from one command to another:



Notes on Windows path names

- In this manual, the Windows terms *directory* and *folder* are both referred to as *directory*.
- Include the drive name when you specify an absolute path name.

Example: `C:\win32app\hitachi\hirdb_s\spool\tmp`

- When you specify a path name in a command argument, in a control statement file, or in a HiRDB system definition file, and that path name includes a space or a parenthesis, you must enclose the entire path name in double quotation marks (").

Example: `pdinit -d "C:\Program Files(x86)\hitachi\hirdb_s\conf\mkinit"`

However, double quotation marks are not necessary when you use the `set` command in a batch file or at the command prompt to set an environment variable, or when you specify the installation directory. If you do use double quotation marks in such a case, the double quotation marks become part of the value assigned to the environment variable.

Example: `set PDCLTPATH=C:\Program Files\hitachi\hirdb_s\spool`

- HiRDB cannot use files on a networked drive, so you must install HiRDB and configure the HiRDB environment on a local drive. Files used by utilities, such as utility input and output files, must also be on the local drive.

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

1. Overview of Commands

This chapter explains the methods of entering commands and the descriptive format for commands.

This chapter contains the following sections:

1.1 Overview

The commands used with HiRDB include operation commands and utilities. The same input methods and formats apply to both.

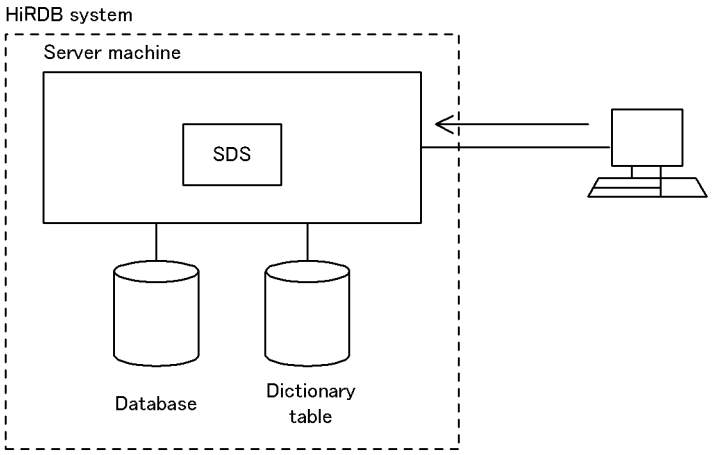
(1) Command input method

A HiRDB command is input at the command prompt.

Before you can execute commands, you must set as the current drive the drive where the HiRDB server is installed. For example, if you have installed the HiRDB server on drive D, set drive D as the current drive in the command prompt and then execute the command.

The command execution mode is shown in Figures 1-1 (for a HiRDB/Single Server) and 1-2 (for a HiRDB/Parallel Server).

Figure 1-1: Command execution mode: HiRDB/Single Server

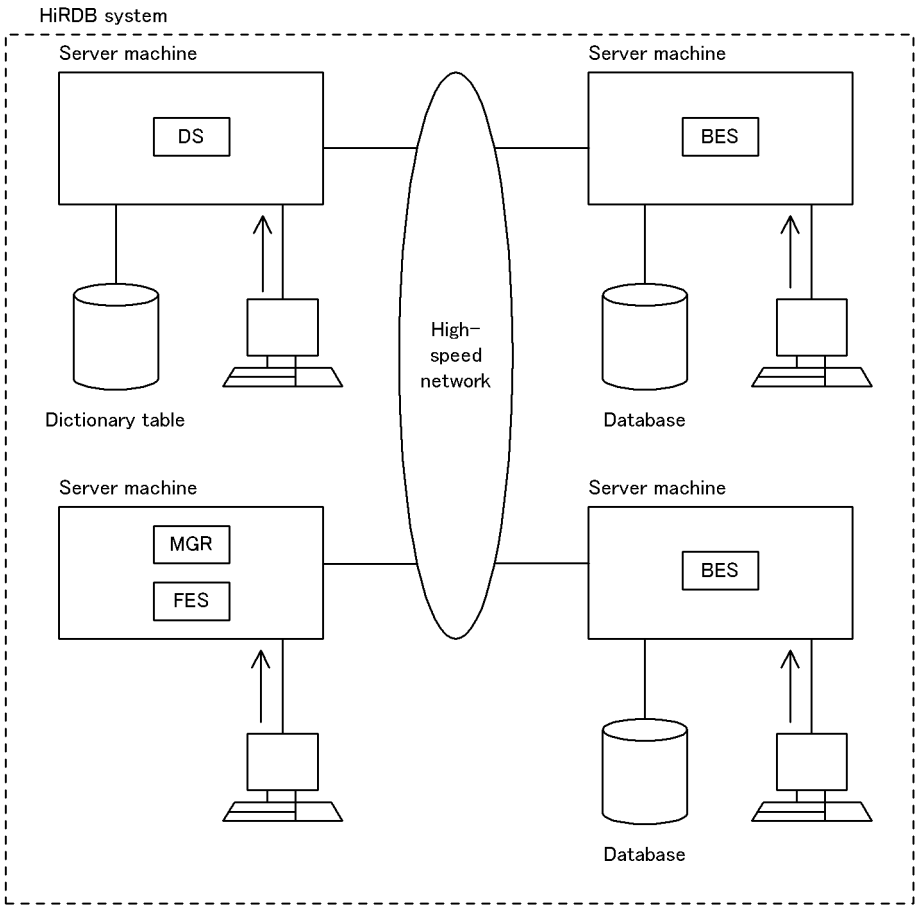


SDS: Single server

Explanation

Enter the command at the server machine where the single server is located.

Figure 1-2: Command execution mode: HiRDB/Parallel Server



MGR: System manager
FES: Front-end server
DS: Dictionary server
BES: Back-end server

Explanation

Enter the command at the server machine where the system manager is located.

Some commands are executed from a server machine other than the system manager. For details about the server machine used to execute each command, see Tables *1-1: List of operation commands* and *1-2: List of utilities*, and the details about each command.

(2) Command descriptive format

The following shows the command descriptive format:
command-name `options` *command-arguments*

(a) command-name

The command name is the file name of the command that is to be executed.

(b) options

An option either qualifies the command's operation or specifies the object of the operation.

In the following explanations, `cmd` indicates a command name.

1. An option is a character string beginning with a minus sign (`-`) that may or may not be followed by one or more flag arguments.

   The following shows the descriptive formats of options:

   -option-flag or -option-flag flag-argument(s)

   option-flag

   One alphanumeric character (case sensitive).

   flag-argument

   Argument of the option flag (multiple flag arguments can be specified by separating them with a comma (`,`)).

2. Multiple option flags without flag arguments can be specified consecutively, preceded by a single minus sign.

   Example

   The following two specifications are treated as being the same:

   ```
   cmd -a -b -c
   ```

   ```
   cmd -abc
   ```

3. If an option requires a flag argument, the flag argument cannot be omitted.

4. When an option flag argument contains a space, the entire flag argument must be enclosed in double quotation marks (`"`).

   Example

   An option with argument "`1 2`" must be specified as follows:

   ```
   cmd -f "1 2"
   ```

   Also, when you specify as a flag argument a path name that contains a space, you must enclose the entire flag argument in double quotation marks.

5. If the same option flag is specified more than once, the last specification is effective.

   Example

   If the following is specified, `-a 2` is effective:

   ```
   cmd -a 1 -a 2
   ```

6. All options must be specified before any command arguments.

   Example

   If option flag `a` does not take a flag argument, `file` and `-b` in the following specification are assumed to be command arguments:

   ```
   cmd -a file -b
   ```

7. Two consecutive minus signs (`--`) indicate the end of the options.

   Example

   In the following specification, `-b` is assumed to be a command argument:

   ```
   cmd -a -- -b
   ```

8. An option consisting of a minus sign only cannot be entered.

   Example

   In the following specification, `-` is assumed to be a command argument:

   ```
   cmd -
   ```

(c) command-arguments

A command argument specifies the direct object of a command's operation. If the object is to be qualified, options are used to specify the object.

(d) Notes

If the control statement file contains a path name that contains a space, the entire path name must be enclosed in double quotation marks.

Example: In the following specification, the `source` statement of `pdload` specifies a path name that contains a space:

```
source host1:"c:\hirdb data\data1","c:\hirdb data\data2"
```

(3) Using a command by specifying an alias

If an operating system command or command in another program has the same name as a HiRDB command, the HiRDB command may fail to execute. If this happens, solve the problem as follows:

- In environmental variable settings, assign a higher priority to the HiRDB

command.

- Execute the command by specifying its absolute path.

In situations when these two methods cannot be used, HiRDB commands can still be executed by assigning arbitrary names to them. For details about how to execute a command in terms of an alias, see the *HiRDB Version 8 Installation and Design Guide*.

(4) Notes about specifying options and control statements

1. When specifying an authorization identifier or a password in a command option, enclose it in `\"`.

   Example

   If you specify `pddbst -u \"hitachi\" -p \"hitachi\"`, HiRDB treats it as `pddbst -u "hitachi" -p "hitachi"`.

2. If you are using a text editor such as Notepad to edit control statements, save the file as a text file using the **Save As** command. If you save it as any other file type, a control statement error may occur during utility execution because the character encoding system has been changed.

3. When you are creating a control statement, be sure to enter a linefeed code at the end of the line. If the system detects EOF without a linefeed code at the end of the line, HiRDB cannot identify the control statement correctly.

1.2 List of operation commands

Table 1-1 lists the operation commands used with HiRDB.

Table 1-1: List of operation commands

Operation command	Description	Executor	HiRDB status	Executing server machine
pdacunlck	Releases the consecutive certification failure account lock state.	HiRDB administrator	ACT	SDS or MGR
pdadmvr	Displays the HiRDB version information.	HiRDB administrator	ANY	Each server machine
pdaudbegin	Starts acquisition of audit trails.	HiRDB administrator	ACT	SDS or MGR
pdaudend	Stops acquisition of audit trails.	HiRDB administrator	ACT[7]	SDS or MGR
pdaudrm	Deletes audit trail files.	HiRDB administrator	ACT	SDS or MGR
pdaudswap	Swaps audit trail files.	Auditor	ACT	SDS or MGR
pdbkupls	Displays information about backup files acquired by pdcopy.	HiRDB administrator	ANY	Server machine where backup files are located
pdbufls	Displays the status of global buffers.	HiRDB administrator	ACT	SDS or MGR
pdbufmod	Adds, deletes, or changes a global buffer during HiRDB operation.	HiRDB administrator	ACT	SDS or MGR
pdcancel	Forcibly disconnects and terminates the process on HiRDB for processing a specified UAP or utility.	HiRDB administrator	ACT	Any server machine
pdcat	Displays status file status.	HiRDB administrator	ANY[2]	Any server machine
	Displays messages in a message log file.	USR	ACT	Any server machine
pdcbl[1]	Preprocesses a UAP written in COBOL.	USR	ANY	Any server machine

Operation command	Description	Executor	HiRDB status	Executing server machine
pdchgconf	Changes system definitions during HiRDB operation.	HiRDB administrator	ACT	SDS or MGR
pdchprc	Changes the number of resident server processes and maximum number of startup server processes, or displays the current number of resident server processes and current maximum number of startup server processes at the time the pdchprc command is executed.	HiRDB administrator	ACT	SDS or MGR
pdclose	Closes RDAREAs.	HiRDB administrator	ACT	SDS or MGR
pdclttrc	Acquires an SQL trace for a client executing SQL statements.	HiRDB administrator	ACT	SDS or MGR
pdcmt	Commits transactions when transactions that are part of a global transaction cannot be concluded.	HiRDB administrator	ACT	Each server machine
pdconfchk	Checks HiRDB system definitions for inconsistencies.	HiRDB administrator	ANY	SDS or MGR
pdcpp[1]	Preprocesses a UAP written in C language.	USR	ANY	Any server machine
pdcspool	Deletes troubleshooting information.	HiRDB administrator	ANY	Each server machine
pddbfrz	Places a full HiRDB file (used through the last page of the file) that constitutes a user LOB RDAREA in frozen update status.	HiRDB administrator	ACT	SDS or MGR
pddbls	Displays the status of RDAREAs.	HiRDB administrator	ACT	SDS or MGR
pddefrev	Generates a definition SQL statement.	User with DBA privilege	ACT	SDS or MGR
pdfbkup	Makes a backup of a HiRDB file system.	HiRDB administrator	ANY[9]	Server machine where the HiRDB file system is located

Operation command	Description	Executor	HiRDB status	Executing server machine
pdffsck	Checks a HiRDB file system area for integrity errors and corrects any errors that are detected.	HiRDB administrator	ANY[11]	Server machine where the HiRDB file system is located
pdfgt	Terminates a transaction forcibly when a transaction branch that is part of a global transaction cannot be terminated for some reason.	HiRDB administrator	ACT	Each server machine
pdfls	Displays information about HiRDB files in a HiRDB file system area.	USR	ANY	Server machine where the HiRDB file system is located
pdfmkfs	Initializes files for a HiRDB file system area.	HiRDB administrator	ANY	Server machine where the HiRDB file system is located
pdfrm	Deletes a HiRDB file from a HiRDB file system area.	HiRDB administrator	ANY[6]	Server machine where the HiRDB file system is located
pdfrstr	Restores a HiRDB file system from a backup.	HiRDB administrator	INA	Server machine where the HiRDB file system is located
pdfstatfs	Displays the status of a HiRDB file system area.	USR	ANY	Server machine where the HiRDB file system is located

Operation command	Description	Executor	HiRDB status	Executing server machine
pdgrprfl	When the Hitachi Directory Server linkage facility is used, updates the user information maintained by HiRDB to the most recent status.	HiRDB administrator	ACT	SDS or MGR
pdhold	Shuts down RDAREAs.	HiRDB administrator	ACT	SDS or MGR
pdjarsync	Registers, re-registers, or deletes a JAR file or displays a list of JAR files.	HiRDB administrator	ANY	SDS or MGR
pdkill	Terminates a specified process forcibly.	HiRDB administrator	ACT	Each server machine
pdlistls	Displays list definition information.	HiRDB administrator	ACT	SDS or MGR
pdlogadpf	During HiRDB operation, assigns a physical file to a file group specified in the HiRDB system definition.	HiRDB administrator	ACT	SDS or MGR
pdlogatul	Restarts, cancels, or displays the status of the automatic log unloading facility.	HiRDB administrator	ANY	SDS or MGR
pdlogchg	Forcibly places a specified file group in unload completed status or HiRDB Datareplicator extraction completed status.	HiRDB administrator	ANY[2]	SDS or MGR
pdlogcls	Closes an open physical file that comprises a file group so that it cannot be used by the active HiRDB.	HiRDB administrator	ACT	SDS or MGR
pdloginit	Creates a log file in the HiRDB file system and initializes it so that it can be used by HiRDB.	HiRDB administrator	ANY[2]	SDS or MGR
pdlogls	Displays information about log files.	HiRDB administrator	ANY[2, 3]	SDS or MGR
pdlogopen	Opens a closed physical file that comprises a file group so that it can be used by the active HiRDB.	HiRDB administrator	ACT	SDS or MGR
pdlogrm	Deletes a log file from the HiRDB file system.	HiRDB administrator	ANY[2]	SDS or MGR
pdlogswap	Swaps log files.	HiRDB administrator	ACT	SDS or MGR

Operation command	Description	Executor	HiRDB status	Executing server machine
pdlogsync	Collects a synchronization point dump.	HiRDB administrator	ACT	SDS or MGR
pdlogucat	Displays information about an unload log file.	HiRDB administrator	ANY	Server machine where the unload log file is located
pdlogunld	Unloads a file group into a destination file and places it in unload completed status.	HiRDB administrator	ANY[2]	SDS or MGR
pdls	Displays the following types of status: • User status • Status of audit trail files • System status when system switchover facility is used • Server lock status • Information about server's shared memory • Server processing status • Communication control information on connected servers • HiRDB Datareplicator linkage status • Server scheduling status • Whether or not an output type for statistical information by unit and server is specified • Unit or server status file status • Unit and server status • Server transaction status • Unit activity status	HiRDB administrator[8]	ANY when -d aud, ha, or rpl is specified; otherwise, ACT	Any server machine when -d ha, lck, svr, trn, or ust is specified; otherwise, SDS or MGR
pdntenv	Displays or sets up an HiRDB operating environment.	HiRDB administrator	INA	Each server machine
pdobils	Displays the utilization status of SQL objects stored in the SQL object buffer.	HiRDB administrator	ACT	Any server machine
pdocb[1]	Preprocesses a UAP coded in OOCOBOL.	USR	ANY	Any server machine
pdocc[1]	Preprocesses a UAP coded in C++.	USR	ANY	Any server machine

Operation command	Description	Executor	HiRDB status	Executing server machine
pdopen	Opens RDAREAs.	HiRDB administrator	ACT	SDS or MGR
pdpfresh	Terminates the current server process and starts a new process.	HiRDB administrator	ACT	SDS or MGR
pdplgrgst	Registers plug-ins and deletes them from a HiRDB.	User with DBA privilege	ACT	Any server machine
pdprgcopy	Copies a HiRDB update version into the HiRDB directory.	HiRDB administrator	ANY	SDS or MGR
pdprgrenew	Updates the current HiRDB to the HiRDB update version.	HiRDB administrator	ACT	SDS or MGR
pdrbk	Rolls back transactions when a transaction branch that is part of a global transaction could not be concluded.	HiRDB administrator	ACT	Each server machine
pdrdrefls	Displays information about related RDAREAs.	HiRDB administrator	ACT	SDS or MGR
pdrels	Releases RDAREAs from shutdown status.	HiRDB administrator	ACT	SDS or MGR
pdrplstart	Starts HiRDB Datareplicator linkage.	HiRDB administrator	ACT[4]	SDS or MGR
pdrplstop	Stops HiRDB Datareplicator linkage.	HiRDB administrator	ACT[5]	SDS or MGR
pdstart	Starts a HiRDB system.	HiRDB administrator	INA	SDS or MGR
	Starts a unit or server.	HiRDB administrator	ACT	SDS or MGR
pdstbegin	Starts output of statistical information to the statistics log file.	HiRDB administrator	ACT	SDS or MGR
pdstend	Stops output of statistical information to the statistics log file.	HiRDB administrator	ACT	SDS or MGR
pdstjswap	Swaps statistics log files.	HiRDB administrator	ACT	SDS or MGR
pdstjsync	Outputs information about the statistics log buffer to the statistics log file.	HiRDB administrator	ACT[10]	Any server machine

Operation command	Description	Executor	HiRDB status	Executing server machine
pdstop	Terminates a HiRDB system, unit, or server.	HiRDB administrator	ACT	SDS or MGR
pdstscls	Closes an open status file.	HiRDB administrator	ACT	SDS or MGR
pdstsinit	Initializes a physical file comprising a status file (logical file).	HiRDB administrator	ANY[2]	SDS or MGR
pdstsopen	Opens an initialized or closed status file.	HiRDB administrator	ACT	SDS or MGR
pdstsrm	Deletes a status file.	HiRDB administrator	ANY[2]	SDS or MGR
pdstsswap	Swaps status files.	HiRDB administrator	ACT	SDS or MGR
pdsvhostname	Displays a specified server's host name in DAT format.	HiRDB administrator	ACT	SDS or MGR
pdtrndec	Completes any uncompleted transactions.	HiRDB administrator	ACT	SDS or MGR
pdtrnqing	Starts or releases transaction queuing for a back-end server.	HiRDB administrator	ACT[12]	Any server machine
pdusrchk	Checks whether or not users registered in HiRDB are registered in the Directory Server.	HiRDB administrator	ACT	SDS or MGR
pdvrup	Maintains the master directory RDAREA and data dictionary RDAREAs when the HiRDB version has been upgraded.	HiRDB administrator	ACT	SDS or MGR

Legend:

ANY: Can be executed whether or not HiRDB is active.

ACT: Can be executed only while HiRDB is active.

INA: Can be executed only while HiRDB is inactive.

USR: Can be executed by a user without privileges.

SDS: For a HiRDB/Single Server, the command is executed on the server machine of the single server.

MGR: For a HiRDB/Parallel Server, the command is executed on the server

machine where the system manager is located.

Note 1

Log files include system log files and synchronization point dump files.

Note 2

A file that is specified in an operation command or utility that is executing must not be in use at the same time by another operation command, utility, or UAP; doing so will result in an error and cause the processing results to be invalidated.

1

For details about `pdcbl`, `pdcpp`, `pdocb`, and `pdocc`, see the *HiRDB Version 8 UAP Development Guide*.

2

This command cannot be executed during the following periods:

- During HiRDB startup processing (from output of `KFPS01800-I` message to output of `KFPS05210-I` message)
- During HiRDB termination processing (from output of `KFPS05220-I` message to output of `KFPS01850-I` message)

3

When the `-d spd` option is specified, this command can be executed only while HiRDB is active.

4

This command can be executed only when all units and servers in the HiRDB system are active.

5

For a HiRDB/Single Server, this command can be executed only while HiRDB is active. For a HiRDB/Parallel Server, this command can be executed if the unit where the system manager is located is active, even if other units are inactive.

6

If the usage purpose of the HiRDB file system area that contains the HiRDB file to be deleted is other than UTL, this command cannot be executed in the HiRDB file system area that contains that HiRDB file while the active HiRDB is in use (if the HiRDB file in the HiRDB file system area has been opened even once). If the usage purpose is UTL, the command can be executed regardless of whether or not the HiRDB file system area is in use. However, it cannot be executed if the specified HiRDB file is in use by the active HiRDB (opened by another process).

7

This command can be executed after acquisition of an audit trail has started.

8

Only the auditor can execute `-d aud` during HiRDB operation. When HiRDB is not running, any HiRDB administrator can execute this command.

9

To execute this command during HiRDB operation, specify the `-f` option.

10

This command can be executed when HiRDB is running and statistical information is being acquired.

11

Depends on whether or not the `-c` option is specified and on the purpose of the HiRDB file system area. For details, see the rules in *pdffsck*.

12

Whether or not the command can be executed depends on the back-end server's operating status. For details, see the rules in *pdtrnqing*.

1.3 List of utilities

Table 1-2 lists the utilities used with HiRDB.

Table 1-2: List of utilities

Utility	Description	Executor	HiRDB status	Executing server machine
Database initialization utility (pdinit)	Initializes a database (creates RDAREAs).	HiRDB administrator	ACT[3]	SDS or MGR
Database definition utility (pddef)	Executes definition SQL statements.	User with CONNECT privilege[4]	ACT	Any server machine
Database load utility (pdload)	Stores user-created data in a table.	Depends on the operating mode[1]	ACT	SDS or MGR
SQL interactive execution utility[5]	Executes SQL commands interactively.	User with CONNECT privilege[4]	ACT	Any server machine
Database structure modification utility (pdmod)	Modifies the physical structure of a database, such as by adding, extending, or reinitializing RDAREAs.	HiRDB administrator	ACT	SDS or MGR
Database reorganization utility (pdrorg)	Maintains tables such as by reorganizing tables and re-creating indexes.	Depends on the function[2]	ACT	SDS or MGR
Dictionary import/export utility (pdexp)	Migrates table definition information and stored procedure information.	User with DBA privilege	ACT	SDS or MGR
Rebalancing utility (pdrbal)	Corrects an imbalance of data that results when RDAREAs are added to rebalancing tables.	User with DBA, SELECT, INSERT, or DELETE privilege	ACT	SDS or MGR
Free page release utility (pdreclaim)	Releases free pages (used free pages) during online operation.	User with DBA, INSERT, or DELETE privilege	ACT	SDS or MGR

Utility	Description	Executor	HiRDB status	Executing server machine
Global buffer residence utility (pdpgbfon)	Reads table page information into a global buffer.	User with DBA privilege or SELECT privilege for the table, or schema owner	ACT	SDS or MGR
Integrity check utility(pdconstck)	Performs integrity checking and manipulates (sets or releases) the check pending status on tables for which referential constraints or check constraints have been defined.	User with DBA privilege or table owner	ACT	SDS or MGR
Statistics analysis utility (pdstedit)	Inputs information from unload data files and system log files and edits the statistical information.	HiRDB administrator	ANY	Any server machine
Database condition analysis utility (pddbst)	Analyzes the status of tables and indexes in user RDAREAs or the storage status of a data dictionary RDAREA, and displays the analysis result.	User with DBA privilege	ACT	SDS or MGR
Optimizing information collection utility (pdgetcst)	Collects optimization information in order to optimize based on cost and stores the information in a data dictionary table.	User with SELECT privilege for the target table	ACT	SDS or MGR
Access path display utility (pdvwopt)	Displays access path information determined by SQL optimization processing.	USR	ANY	SDS or FES
Database copy utility (pdcopy)	• Makes a backup of a database. • Acquires log point information and creates a log point information file.	HiRDB administrator	ACT	SDS or MGR
Database recovery utility (pdrstr)	• In the event of an error, restores a database from a backup copy and information logged after the backup was made. • Re-creates the log point information file.	HiRDB administrator	ACT	SDS or MGR
Registry facility initialization utility (pdreginit)	Creates registry RDAREAs, registry LOB RDAREAs, and registry management tables for controlling and operating the registry facility.	HiRDB administrator	ACT	SDS or MGR

Legend:

ANY: Can be executed whether or not HiRDB is active.

ACT: Can be executed only while HiRDB is active.

SDS: For a HiRDB/Single Server, the command is executed on the server machine of the single server.

MGR: For a HiRDB/Parallel Server, the command is executed on the server machine where the system manager is located.

FES: For HiRDB/Parallel Server, the command is executed on the server machine where the front end server is located.

Note

A file that is specified in an operation command or utility that is executing must not be used at the same time by another operation command, utility, or UAP; doing so will result in an error and cause the processing results to be invalid.

1

In the addition mode (`-d` option omitted), this utility can be executed if the user has the `INSERT` privilege.

In the creation mode (`-d` option specified), the `INSERT` and `DELETE` privileges are required.

For an audit trail table, the audit privilege is required.

2

Reorganization of a dictionary can be executed only by users with the `DBA` privilege.

To unload a table, either the `DBA` privilege or the `SELECT` privilege for the table is required.

To reload a table, either the `DBA` privilege or the `INSERT` and `DELETE` privileges for the table are required.

To create, batch-create, or reorganize an index, no privilege is required.

3

This utility can be executed only when HiRDB is in end-of-initialization wait status (HiRDB is placed in wait status after issuing the `KFPS05201-Q` message).

4

The privilege that is required depends on the SQL to be executed. For details about the privileges required to execute SQL commands, see the manual *HiRDB Version 8 SQL Reference*.

5

HiRDB SQL Executor is required in order to execute the SQL interactive execution utility.

1.4 Utility I/O files

1.4.1 Files and volumes

Files and volumes are defined as follows in a HiRDB system:

- File

  Logical unit used by utilities

- Volume

  Physical storage unit

  In the case of files, volume indicates one file; in the case of a tape device, volume indicates one tape medium.

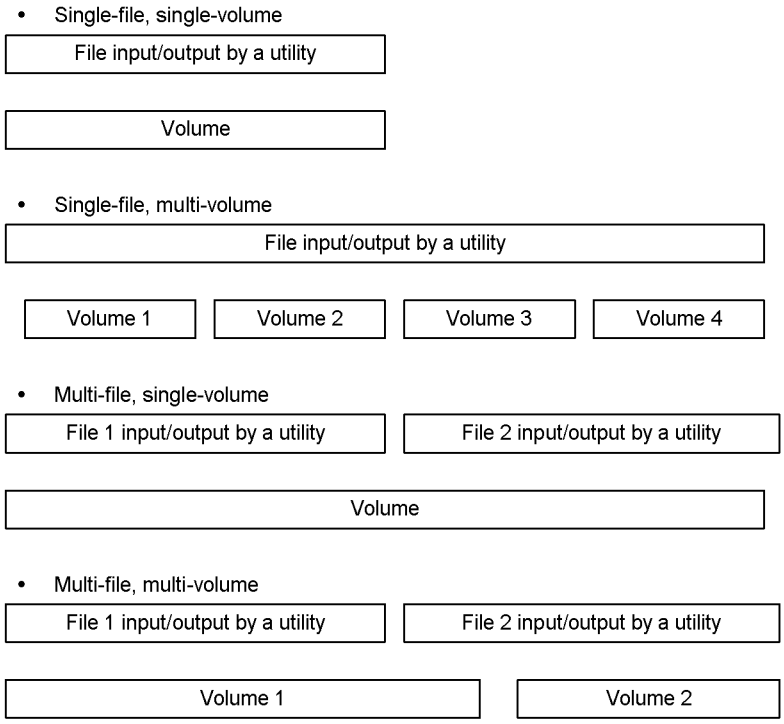
  In the case of the HiRDB file system, volume indicates one HiRDB file.

Typically, one file is stored on one volume (single-file, single-volume). However, a file that is too large for a single volume is divided and stored on multiple volumes (single-file, multi-volume). Conversely, multiple small files can be stored on a single volume (multi-file, single-volume) if formatting permits.

When files are added to a multi-file, single-volume, the storage of the last file may extend onto the next volume (multi-file, multi-volume).

Figure 1-3 shows the concepts of files and volumes.

Figure 1-3: Concepts of files and volumes



1.4.2 File modes usable by utilities

All HiRDB utilities use files as I/O files unless otherwise specified; the file mode is *single-file, single-volume* (utilities cannot handle any file larger than 2^{63} bytes). Files on a network drive cannot be used as I/O files for utilities. Use files on local drives.

Some utilities can handle file modes other than single-file, single-volume. Table 1-3 lists the utilities that can handle I/O files other than single-file, single-volume files.

Table 1-3: Utilities that can handle I/O files other than single-file, single-volume files

Utility	File type	Single-file, multi-volume regular files	Tape device	HiRDB files
Database load utility (pdload)	Input data files, LOB input files, LOB input files by column	C	V*	N
Database reorganization utility (pdrorg)	Unload data files, LOB data unload files		C*	C -f hirdb
Database copy utility (pdcopy)	Backup files			C -k i
Database recovery utility (pdrstr)				
Dictionary import/ export utility (pdexp)	Export files	N		N

Legend:

C:

The files can be used. If a specification on the command line is required, the option is also shown; in such a case, all file-specific information must be specified in addition to the command options. For details about the options, see the explanations of the commands.

V:

Variable-length blocked mode can be used; DAT format can be used in the fixed-length blocked mode. The binary format cannot be used.

N:

Not supported.

*

Not supported in the 64-bit mode.

(1) Tape devices

Table 1-4 lists the tape devices that are supported by the utilities.

Table 1-4: Tape devices supported by the utilities

Device name	Direct input/output by the utilities
CMT	N
DAT	Y
OMT	N
CGMT	N
8mm	N
DLT	Y
LTO	Y

Legend:

Y: Can be used.

N: Cannot be used.

When the utilities access the tape device for direct input/output, they assume a single volume and access the device as a single tape drive. They do not provide Autoloader or Tape Library control.

Except for a CMT that can be accessed only in the fixed-length blocked mode, we recommend that you use the tape device in the variable-length blocked mode. You can reduce the number of input/output operations in the variable-length blocked mode, in which the utilities input/output data in units of 32 kilobytes while they input/output data in units of 512 bytes in the fixed-length blocked mode. When using the fixed-length blocked mode, make sure that the block size is 512 bytes.

For details about each tape device, see the applicable OS manual.

If you use a tape device with `pdrorg`, `pdload`, or `pdrstr`, do not specify the device driver's minor number in such a manner that the tape will not be rewound during close processing. If you execute the utility without rewinding the tape, you must rewind the tape using a command such as `mt` (magnetic tape manipulation program) after the utility has terminated.

To save multiple backups on the same medium, use the utility to output the backups to files and then copy them onto the desired medium.

(2) HiRDB files

To use a HiRDB file as an I/O file, a HiRDB file system must first be created with the `pdfmkfs` command in which `-k UTL` is specified, and the appropriate option shown in Table 1-3 must be specified when the HiRDB utility is executed.

HiRDB utilities use HiRDB files in the following modes:

- Single-file, single-volume
- Single-file, multi-volume

For details about HiRDB files, see the *HiRDB Version 8 Installation and Design Guide*.

1.4.3 Tape device access facility

You can use the tape device access facility to access DAT, DLT, and LTO files.

(1) Files supported by the tape device access facility

The tape device access facility supports the following files:

- Input data files (specified in `source` statement of `pdload`)
- Unload data files (specified in `unload` statement of `pdrorg`)
- LOB data unload files (specified in `lobunld` statement of `pdrorg`)
- Backup files (specified in the `-b` option for `pdcopy` or `pdrstr`)

(2) File specification method

To specify a DAT, DLT, or LTO file enter `\\.\tape[`*n*`]` as the file name, where *n* is the tape drive number. When *n* is omitted, `0` is assumed.

To check the value of *n*, choose **Administrative Tools**, **Computer Management**, **Storage**, **Removable Storage**, **Physical Location**, and then choose Device Information, which is a tape device property.

(3) Hardware compression facility

You can use the hardware compression facility in accessing a DAT, DLT, or LTO. Note that this setting is applicable only if the DAT, DLT, or LTO supports hardware compression.

To reference or update the specified value, use the `pdntenv` command. The compression option is enabled (on) as the default setting immediately after installation.

1.5 Notes on command execution

1.5.1 Forced termination by interruption of command execution

While a command is executing, do not terminate it forcibly by using a signal interrupt (such as **CTL** + **C** keys, **CTL** + \ keys, or the `pdkill` command). HiRDB may terminate itself depending on the timing. Use the `pdcancel` command to terminate utilities.

1.5.2 Batch specification of RDAREA names in operation commands

You can use the methods described below to specify multiple RDAREA names in operation commands. These methods enable you to specify a group of RDAREAs.

Item	Pattern character string	Description	Example of specification
Right truncation	*nnn*`*`	The command searches for RDAREAs whose names begin with *nnn*.	`PDUSER*` (the command searches for RDAREAs whose names begin with `PDUSER`)
Right and left truncation	`*`*nnn*`*`	The command searches for RDAREAs whose names contain *nnn*.	`*PDUSER*` (the command searches for RDAREAs whose names contain `PDUSER`)
Left truncation	`*`*nnn*	The command searches for RDAREAs whose names end with *nnn*.	`*PDUSER` (the command searches for RDAREAs whose names end with `PDUSER`)

(1) Operation commands supporting batch specification of RDAREA names

The following operation commands support batch specification of RDAREA names:

- `pdclose`
- `pddbfrz`
- `pddbls`
- `pdhold`
- `pdopen`
- `pdrels`

(2) Notes

1. Do not specify an asterisk (`*`) at any location other than the beginning or end of a

pattern character string. If an asterisk is specified at a location other than the beginning or end of a pattern character string, the system treats the asterisk as part of the character string.

2. Do not include a percent sign (%) or underscore (_) in the pattern character string.
3. To treat the pattern as being case sensitive, enclose the entire pattern character string in \". If the pattern character string is not enclosed in \", the system treats it as being all uppercase letters. Also, if the pattern character string contains a space, enclose the pattern character string in \".
4. When there an RDAREA name is duplicated in the RDAREAs grouped by batch specification of RDAREA names and in the RDAREAs specified by another RDAREA name specification (including RDAREA names grouped by another batch specification of RDAREA names), whether or not the RDAREA is processed more than once depends on the command. The following table describes each command's processing when an RDAREA name is duplicated:

Command	When multiple RDAREA name specifications contain a duplicate RDAREA	When there are duplicate RDAREAs in the RDAREAs grouped by batch specification of RDAREA names and in the RDAREAs specified by another RDAREA name specification
pdclose	Y	N
pddbfrz	Y	N
pddbls	Y	N
pdhold	Y	N
pdopen	Y	N
pdrels	Y	N

Legend:

Y: Eliminates a duplicated RDAREA name from processing.

N: Does not eliminate any duplicated RDAREA names from processing; performs processing as many times as an RDAREA name is specified.

1.5.3 Specifying the LANG environment variable

Some commands require specification of the LANG environment variable.

If the LANG environment variable is not specified but the LC_ALL or LC_* environment variable is specified, the latter takes effect. If the LC_ALL or LC_* environment variable is not specified, the specified LANG environment variable takes effect.

Take this into account when executing commands that require specification of the

LANG environment variable.

1.6 Functions that become available when optional HiRDB program products are installed

Table 1-5 describes optional HiRDB program products. When you install these programs, the functions described in the table become available.

Table 1-5: HiRDB optional program products

Name of optional program product	Functions that become available when optional program product is installed
HiRDB External Data Access, HiRDB External Data Access Adapter	HiRDB External Data Access facility
HiRDB Advanced High Availability	• Server mode system switchover function • User server hot standby • Rapid system switchover facility • Standby-less system switchover (1:1) facility • Standby-less system switchover (effects distributed) facility • Dynamic updating of global buffers (`pdbufmod` command) • System reconfiguration command (`pdchgconf` command)
HiRDB LDAP Option	Sun Java System Directory Server linkage facility
HiRDB Advanced Partitioning Option	• Matrix partitioning of tables • Changing the partition storage conditions (`ALTER TABLE`)
HiRDB Non Recover FES	Recovery-unnecessary front-end server

Chapter
2. Operation Commands

This chapter explains each of the operation commands.

pdlistls (Display list definition information)
pdlogadpf (Allocate log file)
pdlogatul (Control the automatic log unloading facility)
pdlogchg (Change status of log file)
pdlogcls (Close open log file)
pdloginit (Initialize log file)
pdlogls (Display log-related file information)
pdlogls [-d spd] (Display synchronization point dump file information)
pdlogls [-d sys] (Display system log file information)
pdlogopen (Open log file)
pdlogrm (Delete log file)
pdlogswap (Swap log files)
pdlogsync (Collect a synchronization point dump)
pdlogucat (Display unload log file information)
pdlogunld (Unload log file)
pdls (Display HiRDB system status)
pdls [-d act] (Display user status)
pdls [-d aud] (Display the status of audit trail files)
pdls [-d ha] (Display HiRDB system status)
pdls [-d lck] (Display server lock control status)
pdls [-d mem] (Display server shared memory status)
pdls [-d prc] (Display server process status)
pdls [-d rpc] (Display server communication control information)
pdls [-d rpl] (Display HiRDB Datareplicator linkage status)
pdls [-d scd] (Display server schedule status)
pdls [-d stj] (Display whether or not statistical data output type by unit and server is specified)
pdls [-d sts] (Display status file status)
pdls [-d svr] (Display status by unit and server)
pdls [-d trn] (Display server transaction status)
pdls [-d ust] (Display unit activity status)
pdntenv (Set up HiRDB operating environment)
pdobils (Display SQL object buffer statistics)
pdopen (Open RDAREAs)
pdpfresh (Refresh server process)
pdplgrgst (Register and delete plug-ins)
pdprgcopy (Copy a HiRDB update version)
pdprgrenew (Update a HiRDB update version)
pdrbk (Roll back transactions)
pdrdrefls (Display information about related RDAREAs)
pdrels (Release RDAREAs from shutdown status)
pdrplstart (Start HiRDB Datareplicator linkage)
pdrplstop (Stop HiRDB Datareplicator linkage)
pdstart (Start HiRDB system, unit, or server)

pdstbegin (Start output of statistical information)
pdstend (Stop output of statistical information)
pdstjswap (Swap statistics log files)
pdstjsync (Apply statistics log buffer to statistics log file)
pdstop (Terminate HiRDB system, unit, or server)
pdstscls (Close open status file)
pdstsinit (Initialize status file)
pdstsopen (Open status file)
pdstsrm (Delete status file)
pdstsswap (Swap status files)
pdsvhostname (Display server's host name)
pdtrndec (Settle unsettled transactions forcibly and automatically)
pdtrnqing (Start or release the transaction queuing facility)
pdusrchk (Check the Directory Server for user information)
pdvrup (Upgrade HiRDB version)

pdacunlck (Release consecutive certification failure account lock state)

Function

The pdacunlck command releases identifiers from consecutive certification failure account lock state authorization whose account lock period has not expired, or authorization identifiers that are in permanent consecutive certification failure account lock state.

When executed, the pdacunlck command sets to the null value the consecutive certification failure account lock date and time and the consecutive certification failure count for an authorization identifier if its consecutive certification failure account lock date and time in the SQL_USERS data dictionary table is not the null value.

Executor

HiRDB administrator

Format

```
pdacunlck {authorization-identifier[,authorization-identifier]...|ALL}
```

Command arguments

- *authorization-identifier*[,*authorization-identifier*]...

Specifies the authorization identifiers whose consecutive certification failure account lock state is to be released. You can specify a maximum of 128 authorization identifiers.

- ALL

Specifies that all authorization identifiers are to be released from consecutive certification failure account lock state.

A user attempting to execute the pdacunlck command with ALL specified cannot connect to HiRDB if the user privilege information buffer does not contain information about the CONNECT permissions and the consecutive certification failure account lock date and time are not the null value.

Rules

1. The pdacunlck command can be executed only when HiRDB is running.
2. The pdacunlck command must be executed at the server machine that contains the single server or where the system manager is located.

3. The `pdacunlck` command cannot be executed if the data dictionary RDAREA is shut down.

Notes

1. When an authorization identifier is specified, the command places that row of the `SQL_USERS` data dictionary table in the lock mode. When `ALL` is specified, the command locks the rows of the `SQL_USERS` data dictionary table for all users whose consecutive certification failure account lock date and time are not the null value.
2. For the `pdacunlck` command, return code `0` indicates normal termination, and return code `8` indicates abnormal termination.
3. If an authorization identifier is enclosed in `\"`, the command treats it as being case sensitive. If it is not enclosed in `\"`, the command treats it as all uppercase letters.

pdadmvr (Acquire HiRDB product information)

Function

Displays the HiRDB version information.

Executor

HiRDB administrator

Format

```
pdadmvr [-s]
```

Options

- `-s`

Specifies that the coding following the revision number is to be displayed as the version information. The following shows an example of the version information display:

- When the -s option is omitted: `08-00`
- When the -s option is specified: `08-00-01`

Rules

1. The `pdadmvr` command can be executed at any time, whether or not HiRDB is active.
2. The `pdadmvr` command must be executed at each server machine.

Note

1. When the `pdadmvr` command is executed, return code `0` indicates normal termination, and return code `8` indicates abnormal termination (such as an invalid option).

Output format

```
pdadmvr -s

parallel 08-00-01 Object-Option 32bit No-Staticizer-Option Exd
   ①       ②          ③          ④            ⑤              ⑥

Advanced-High-Availability Advanced-Partitioning-Option Non-Recover-FES
            ⑦                          ⑧                       ⑨
```

A circled number in the figure corresponds to the item number in the following table:

No.	Display item	Explanation
1	HiRDB type	• `single`: HiRDB/Single Server • `parallel`: HiRDB/Parallel Server
2	HiRDB version	HiRDB version
3	Object-Option	`Object-Option` is always displayed.
4	HiRDB addressing mode	• `32bit`: 32-bit addressing mode • `64bit`: 64-bit addressing mode
5	No-Staticizer-Option	`No-Staticizer-Option` is always displayed.
6	Whether or not HiRDB External Data Access has been installed	• `Exd`: Installed • `No-Exd`: Not installed
7	Whether or not HiRDB Advanced High Availability has been installed	• `Advanced-High-Availability`: Installed • `No-Advanced-High-Availability`: Not installed
8	Whether or not the HiRDB Advanced Partitioning Option has been installed	• `Advanced-Partitioning-Option`: Installed • `No-Advanced-Partitioning-Option`: Not installed
9	Whether or not a HiRDB recovery-unnecessary FES has been installed	• `Non-Recover-FES`: Installed • `No-Non-Recover-FES`: Not installed

pdaudbegin (Start acquisition of audit trails)

Function

The pdaudbegin command starts acquisition of audit trails.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

```
pdaudbegin
```

- HiRDB/Parallel Server

```
pdaudbegin [-u unit-identifier]
```

Options

- -u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit that is to start acquisition of an audit trail.

If the -u option is omitted, the command starts acquiring audit trail on all units.

Rules

1. The pdaudbegin command can be executed only while HiRDB is active.
2. The pdaudbegin command must be executed at the server machine that contains the single server or where the system manager is located.
3. Executing the pdaudbegin, pdaudend, pdaudrm, or pdaudswap command while the pdaudbegin command is executing results in a command error.
4. The pdaudbegin command results in a command error if the pd_aud_file_name operand has not been specified in the system definition or the audit trail file specified in the pd_aud_file_name operand is missing.
5. At a HiRDB/Parallel Server, if you execute the pdaudbegin command omitting the -u option and an error occurs during command processing or audit trail acquisition has already begun, the command displays an error message at the corresponding unit. The command starts acquisition of audit trails on units where no error resulted.

Notes

1. The result of `pdaudbegin` command can be checked by using the `pdls -d aud` command, by checking for any error messages, or on the basis of the return code from execution of the command. A return value of `0` for this command indicates normal termination; `4` indicates that only some of the units have terminated normally; and `8` indicates abnormal termination.
2. You can use the `pdls -d aud` command to determine whether or not audit trails are being acquired.

pdaudend (Stop acquisition of audit trails)

Function

The pdaudend command stops acquisition of audit trails.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

```
pdaudend
```

- HiRDB/Parallel Server

```
pdaudend [-u unit-identifier]
```

Options

- -u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit that is to stop acquisition of an audit trail.

If the -u option is omitted, the command stops acquisition of audit trails on all units.

Rules

1. The pdaudend command can be executed only while HiRDB is active and audit trail acquisition has started.
2. The pdaudend command must be executed at the server machine that contains the single server or where the system manager is located.
3. A command error results if you execute the pdaudbegin, pdaudend, pdaudrm, or pdaudswap command while the pdaudend command is executing.
4. The pdaudend command results in a command error if the pd_aud_file_name operand has not been specified in the system definition or the audit trail file specified in the pd_aud_file_name operand is missing.
5. At a HiRDB/Parallel Server, if you execute the pdaudend command omitting the -u option and an error occurs during command processing or audit trail acquisition has already stopped, the command displays an error message at the corresponding unit. The command stops acquisition of audit trails on the units

where no error resulted.

6. Once audit trail acquisition begins, the system keeps acquiring audit trails until the unit is stopped or you execute the `pdaudend` command.

Notes

1. The result of `pdaudend` command can be checked by using the `pdls -d aud` command, by checking for any error messages, or on the basis of the return code from execution of the command. A return value of `0` for this command indicates normal termination; `4` indicates that only some of the units have terminated normally; and `8` indicates abnormal termination.

pdaudrm (Delete audit trail files in shutdown status)

Function

The pdaudrm command deletes audit trail files that are in shutdown status.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

```
pdaudrm -g audit-trail-file-name [-f]
```

- HiRDB/Parallel Server

```
pdaudrm -u unit-identifier -g audit-trail-file-name [-f]
```

Options

- -u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit that contains the audit trail file to be deleted.

- -g *audit-trail-file-name*

Specifies the name of the audit trail file to be deleted.

- -f

Specifies that the audit trail file specified in the -g option is to be deleted even if it is waiting for data loading.

You can use the pdls -d aud command to check for audit trail files.

Rules

1. The pdaudrm command can be executed only while HiRDB is active.
2. The pdaudrm command must be executed at the server machine that contains the single server or where the system manager is located.
3. A command error results if you execute the pdaudbegin, pdaudend, pdaudrm, or pdaudswap command while the pdaudrm command is executing.
4. The pdaudrm command results in a command error if the pd_aud_file_name

operand has not been specified in the system definition or the audit trail file specified in the `pd_aud_file_name` operand is missing.

5. You cannot delete the current audit trail file.

Notes

1. The result of `pdaudrm` command can be checked by using the `pdls -d aud` command, by checking for any error messages, or on the basis of the return code from execution of the command. A return value of `0` for this command indicates normal termination; `8` indicates abnormal termination.

pdaudswap (Swap audit trail files)

Function

The pdaudswap command swaps audit trail files (changes the audit trail file that is the current audit trail file).

Executor

Auditor

To execute the pdaudswap command, you must specify the -U option or set the auditor's authorization identifier and password in the PDUSER environment variable. If the auditor's authorization identifier and password are not correct, a command error results.

Format

- HiRDB/Single Server

```
Pdaudswap [-U authorization-identifier]
```

- HiRDB/Parallel Server

```
pdaudswap -u unit-identifier [-U authorization-identifier]
```

Options

- -u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit that contains the audit trail file to be swapped.

- -U *authorization-identifier*

Specifies the auditor's authorization identifier. When this option is specified, a message requesting entry of a password is displayed.

Rules

1. The command uses the authorization identifier specified in this option and the password that is entered to connect to HiRDB and checks to determine whether this authorization identifier has the audit privilege.
2. You can specify this option when you execute the pdaudswap command in an environment in which you can enter a response to a response-request message.

3. When this option is omitted, the pdaudswap command assumes the value set in the PDUSER environment variable at the time of command execution. If this option is omitted and the PDUSER environment variable has not been set, the pdaudswap command cannot be executed. If this option is omitted, the PDUSER environment variable must be set; the following shows an example:

Example

PDUSER=\"*authorization-identifier*\"/\"*password*\"

Rules

1. The pdaudswap command can be executed only while HiRDB is active. You cannot use this command when HiRDB was started by the pdstart -r command.
2. The pdaudswap command must be executed at the server machine that contains the single server or where the system manager is located.
3. A command error results if you execute the pdaudbegin, pdaudend, pdaudrm, or pdaudswap command while the pdaudswap command is executing.
4. The pdaudswap command results in a command error if the pd_aud_file_name operand has not been specified in the system definition or the audit trail file specified in the pd_aud_file_name operand is missing.
5. If there is no swappable file generation or no audit trail file generation has been created, a command error results.
6. If an open error occurs on the swappable audit trail file generation, the command automatically uses another generation of the file as the target.
7. If pd_aud_no_standby_file_opr=down was specified in the system definition and the number of available generations is 1 or fewer, executing the pdaudswap command results in a command error.

Notes

1. The result of pdaudswap command can be checked by using the pdaudswap command, by checking for any error messages, or on the basis of the return code from execution of the command. A return value of 0 for this command indicates normal termination; 8 indicates abnormal termination.

pdbkupls (Display backup file information)

Function

The pdbkupls command displays information about backup files acquired by pdcopy (such as a list of RDAREAs and the backup acquisition date and time).

Use the pdbkupls command:

- When you want to check the contents of a backup (for example, to determine what is included in the most recent backup)
- When you want to retain the acquired backup information as history

Executor

HiRDB administrator

Format

```
pdbkupls -b {backup-file-name[,backup-file-name]...
            |policy-name}
      [-k {u|i|n}]
      [-o backup-information-output-file-name] [-c]
      [-a] [-s progress-message-output-interval]
      [-U {backup-search-condition-start-date_time,backup-search-condition-end-date_time
             |,backup-search-condition-end-date_time}]
```

Options

■ -b {*backup-file-name*[,*backup-file-name*]...|*policy-name*}

Specifies the names of the backup files.

When *backup-file-name*[,*backup-file-name*]... is specified:

～ <path name> ((up to 1023 bytes))

Specifies file names or the names of HiRDB files.

If you specify a file name, also specify -k u; if you specify a HiRDB file name, also specify -k i.

When *policy-name* is specified:

～ <identifier> ((up to 128 bytes))

Specifies a NetBackup policy name. In this case, also specify `-k n`.

Rules

1. If a host name is specified, an error results.
2. Multiple file names can be specified only if the backup consists of multiple files. When the backup does not consist of multiple files, all file names after the first one specified are ignored. If the backup consists of multiple files, only one file is specified, and the `-a` option is also specified, the command terminates with an error.

- `-k {u|i|n}`

Specifies the type of backup file.

`u`:

Regular file.

`i`:

HiRDB file.

`n`:

Backup file acquired using NetBackup.

- `-o` *backup-information-output-file-name* ～ <path name> ((up to 1023 bytes))

Specifies a file to which the backup file information is to be output. When this option is omitted, the backup file information is output to the standard output.

- `-c`

Specifies that the backup file information is to be output in CSV format. When this option is omitted, the backup file information is output to the standard output.

- `-a`

Specifies that all information is to be output.

When this option is omitted, the command outputs the HiRDB identifier, backup acquisition mode, backup acquisition time, and whether or not a log point information file was specified.

- `-s` *progress-message-output-interval* ～ <unsigned integer> ((1-1024))

Specifies that progress messages are to be output during execution of the `pdbkupls` command, and specifies the message output interval in terms of the amount of backup file data to be processed in each interval (in gigabytes). For example, when `-s 100` is specified, a progress message is displayed after each 100 gigabytes of data has been processed.

Note that this option is applicable only when the -a option is specified.

- -U
  {*backup-search-condition-start-date_time* , *backup-search-condition-end-date_time* | , *backup-search-condition-end-date_time*}

Specifies a time period when only backup information acquired during the specified period under the specified policy is to be included in the output backup information. This option is applicable only when the NetBackup linkage facility is used (-k n is specified). When the NetBackup linkage facility is not used, this option is ignored, if specified.

If *backup-search-condition-start-date_time* and *backup-search-condition-end-date_time* are both specified, the most recent backup acquired during the specified period is used. If only *backup-search-condition-end-date_time* is specified, the backup that was current at the specified time is used.

Rules

- When this option is omitted, the command uses the most recent backup among the backups acquired under the specified policy.
- You can use the bpimagelist command (with the -policy option specified) to check backup acquisition dates/times.
- When you specify a *backup-search-condition-start-date_time* or *backup-search-condition-end-date_time*, specify an underscore (_) between the date and the time, as shown below. If the time is omitted, 000000 is assumed as the start time and 235959 is assumed as the end time.

```
-U YYYYMMDD[_hhmmss] , YYYYMMDD[_hhmmss]
```

YYYY: Year ～ <unsigned integer> ((1990-2037))

Specifies the year.

MM: Month ～ <unsigned integer> ((01-12))

DD: Date ～ <unsigned integer> ((01-31))

hh: Hour ～ <unsigned integer> ((00-23))

mm: Minute ～ <unsigned integer> ((00-59))

ss: Second ～ <unsigned integer> ((00-59))

To specify the end date/time only, specify the comma (,) and then the end date/time.

Rules

1. You can execute the `pdbkupls` command whether or not HiRDB is active.
2. Execute the `pdbkupls` command at the server machine where the host containing the backup files is located. If you use NetBackup, execute the command at the server machine that contains the host to which backup files were output during execution of `pdcopy`.
3. The `pdbkupls` command can process only backup files acquired by `pdcopy`. If any other backup file is specified, the command results in an error.
4. To use NetBackup, you must have JP1/VERITAS NetBackup Agent for HiRDB License; for details, see the *JP1/VERITAS NetBackup v4.5 Agent for HiRDB License Description and User's Guide*.
5. A differential backup file cannot be specified. If a differential backup file is specified, an error results.

Notes

1. For the `pdbkupls` command, return code `0` indicates normal termination, and return code `12` indicates abnormal termination. If the return code is `12`, see the previous message and eliminate the cause of the error.
2. If `utf-8` is selected as the character encoding in the `pdntenv` command, a BOM is not added to a file that is output by `pdbkupls`.

Output format

The following conventions apply to the output format:

1. (*linefeed*) indicates a linefeed code (CR+LF).
2. If there is no output information, the command outputs `*` for the item in the standard format and nothing in the CSV format (2 consecutive commas are displayed).
3. *rr...r* through *uu...u* are output for each RDAREA, in ascending order of the RDAREA IDs (in which case, (*linefeed*) and `<<RDAREA information>>` are also output for each RDAREA).
4. When data is output to the backup information output file, EOF is displayed in the row immediately following the last data.
5. A linefeed is performed after each 1,024 bytes of data (not including linefeed codes).

The following shows the output formats:

- When the -a option is omitted

Standard format:

```
<<System information>>(linefeed)
<HiRDB system id> : aaaa(linefeed)
(linefeed)
<<Backup file information>>(linefeed)
<Backup mode> : dd...d(linefeed)
<Backup start Time> : ee...e(linefeed)
<Logpoint information Y/N> : g(linefeed)
```

CSV format:

```
aaaa,dd...d,ee...e,g(linefeed)
```

Explanation

For details about *aaaa* through *g*, see the description for *When the -a option is specified*.

- When the -a option is specified

Standard format:

```
<<System information>>(linefeed)
<HiRDB system id> : aaaa(linefeed)
(linefeed)
<<Backup file information>>(linefeed)
<Backup file count> : bbbb(linefeed)
<Backup file name> : cc...c(linefeed)
<Backup mode> : dd...d(linefeed)
<Backup start time> : ee...e(linefeed)
<Backup end time> : ff...f(linefeed)
<Logpoint information Y/N> : g(linefeed)
<Get RDAREA's count> : hhhh(linefeed)
<Get RDAREA's list> : ii...i(linefeed)
<Errskip Y/N> : j(linefeed)
<Replica RDAREA generation No> : kk(linefeed)
<Server name> : ll...l(linefeed)
<Server RUNID> : mm...m(linefeed)
<Server group name> : nn...n(linefeed)
<Server block No> : oo...o(linefeed)
<Server generation No> : pp(linefeed)
<Server Log start time> : qq...q(linefeed)
(linefeed)
<<RDAREA information>>(linefeed)
<RDAREA name> : rr...r(linefeed)
<RDAREA id> : ss...s(linefeed)
<RDAREA kind> : tt...t(linefeed)
<RDAREA last LSN> : uu...u(linefeed)
        :
```

CSV format:

```
aaaa,bbbb,cc...c,dd...d,ee...e,ff...f,g,hhhh,ii...i,j,kk,ll...l,mm...m,nn...n,
oo...o,pp,qq...q,rr...r,ss...s,tt...t,uu...u  . . .
```

Explanation

aaaa

Identifier of the HiRDB used to acquire the backup (1 to 4 characters).

bbbb

Number of backup files specified in the `-b` option (1 to 4 decimal digits).

cc...c

Names of the backup files specified in the `-b` option (maximum of 1,023 characters). Backup file names are output in the order specified in the `-b` option of `pdcopy`, separated by the comma.

dd...d

Value of the -M option when the backup was acquired (1 to 10 characters).

For -M x or -M r, the value is EXCLUSIVE; for -M s, the value is SHARED.

ee...e

Time the first record was written during backup acquisition (*YYYY-MM-DD hh:mm:ss*).

ff...f

Time the last record was written during backup acquisition (*YYYY-MM-DD hh:mm:ss*).

g

Whether or not a log point information file was created during backup acquisition (1 character).

If a log point information file was created, the value is Y; if not, the value is N.

hhhh

Number of RDAREAs stored in the backup files (1 to 4 decimal digits).

ii...i

Name of each RDAREA stored in the backup file (1 to 30 characters). Multiple RDAREA names are separated by the comma and output in ascending order of the RDAREA IDs used during backup acquisition.

j

Whether or not the -J option was specified during backup acquisition (1 character).

If the -J option was specified, the value is Y; if not, the value is N.

kk

The value 0 is always displayed.

ll...l

Name of the server that contains the RDAREAs (1 to 8 characters).

mm...m

Log server run ID (information used by the system) (8 hexadecimal characters).

nn...n

Log group name (information used by the system) (1 to 8 characters).

oo...o

Block number (8 hexadecimal characters).

pp

Generation number (1 to 2 decimal digits).

qq...q

Time at which use of the system log file began (*YYYY-MM-DD hh:mm:ss*).

rr...r

Name of an RDAREA (1 to 30 characters).

ss...s

RDAREA ID (1 to 8 decimal digits).

tt...t

RDAREA type (1 to 15 characters):

`MASTERDIRECTORY`: Master directory RDAREA

`DATADIRECTORY`: Data directory RDAREA

`DATADICTIONARY`: Data dictionary RDAREA

`SYSTEM_LOB`: Data dictionary LOB RDAREA

`USER`: User RDAREA

`USER_LOB`: User LOB RDAREA

`REG`: Registry RDAREA

`REG_LOB`: Registry LOB RDAREA

uu...u

RDAREA update sequence number (information used by the system) (17 characters).

pdbufls (Display global buffer information)

Function

The pdbufls command displays the status of global buffers.

Executor

HiRDB administrator

Format

```
pdbufls [-k output-type][-d] [-x [-y]]
        [{-s server-name[,server-name]...
        | -a global-buffer-name[,global-buffer-name]...}]
```

Options

- -k *output-type* ～ <<sts>>

Specifies the type of information to be displayed:

def

Global buffer definition information.

sts

Global buffer statistical information. The statistical information accumulated since the last time the pdbufls command was executed is displayed. The first time you execute the pdbufls command, or if you specify the -d option, the system displays the statistical information accumulated since HiRDB started.

all

Both global buffer definition information and statistical information.

- -s *server-name* ～ <identifier> ((1-8))

Specifies a server name in order to display information about all global buffers at that server. A maximum of 128 server names that can be specified; subsequent server name specifications are ignored.

- -a *global-buffer-name* ～ <identifier> ((1-16))

Specifies the name of a global buffer for which information is be displayed. If duplicate global buffer names are specified, the duplications are eliminated. A maximum of 128 global buffer names can be specified; subsequent global buffer name specifications are ignored.

- -d

Specifies that the system is to display the statistical information about a global buffer since HiRDB started. If `-k def` is specified, the system ignores this option, if specified.

- `-x`

Specifies that the information is to be displayed in DAT format. When `-k all` is specified, this option cannot be specified.

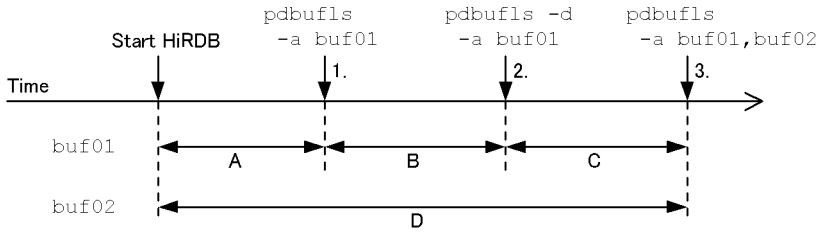
- `-y`

When the `-x` option is specified, specifies that a header is to be displayed.

Rules

1. The `pdbufls` command can be executed only while HiRDB is active.
2. The `pdbufls` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. If the `-s` and `-a` options are both omitted, global buffer information for all servers is displayed
4. Figure 2-1 shows the collection range of global buffer statistical information.

   Figure 2-1: Collection range of global buffer statistical information



Explanation

1. The range of information to be displayed is A, which is from HiRDB startup to 1. In this case, the previous statistical information collection time is not displayed.
2. Because the `-d` option is specified, the system displays the information in the range A + B, which is from HiRDB startup to 2. In this case, the previous statistical information collection time is not displayed. This global buffer information does not take effect on the next `pdbufls` command.
3. The information displayed with `buf01` is in the range B + C, which is from 1 through 3. The information displayed with `buf02` is in the range of D, which is from HiRDB startup to 3. For `buf01`, the statistical information collection time of 1 above is displayed; for `buf02`, the previous statistical information collection time is not displayed.

Notes

1. The following are the `pdbufls` command's return codes:

   `0`: Normal termination

   `4`: Warning termination (some server processing terminated with an error)

   `8`: Abnormal termination

   `12`: Abnormal termination (an event occurred that prevented output of an error message)

   When the return code is `12`, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the `pdbufls` command. If no error message was output to the event log, contact the customer engineer.

Output format

```
-k def specified
DEFINE OF GLOBAL BUFFER
  EDIT TIME aa...a
  BUFNAME SVID   TYPE   SIZE NUM    WRATIO RDAREA/INDEX NAME
                 PRMAX  PRNUM  CSIZE  MAPS
  bb...b     cc...c   d       eeK  ff..f      ggg     hh...h
                 kk...k   lll      mm...m  nn...n
       [NOT FOUND ii...i=jj...j]
```

Explanation

aa...a

Time when `pdbufls` command was executed (*year/month/date hour:minute:second*).

bb...b

Global buffer name (up to 16 characters).

cc...c

Server name (up to 8 characters).

d

Global buffer type:

`R`: RDAREA

`I`: Index

`B`: LOB

O: Other[1]

ee

Buffer length (in KB; 2 decimal digits).

ff...f

Number of buffer sectors (1 to 6 decimal digits in the 32-bit mode; 1 to 10 decimal digits in the 64-bit mode).

ggg

Updated output pages at the time of deferred write trigger (percentage; 3 decimal digits). This is the value specified in the `-w` option in the `pdbuffer` operand in the system common definition.

hh...h

RDAREA name or index name (up to 43 characters).

If multiple names are defined, multiple names are output. However, there is no output in the case of the *Other*[1] global buffer type. There is also no output in the case of RDAREA or index deletion. The output format is as follows:

RDAREA name: `RDAREA-`*name*

Index name: *authorization-identifier.index-identifier*

ii...i

`SERVER` when the `-s` option is specified; `BUFFER POOL` when the `-a` option is specified.

jj...j

Server name when the `-s` option is specified (up to 8 characters); global buffer name when the `-a` option is specified (up to 16 characters).

kk...k

Maximum number of concurrent prefetch operations (5 decimal digits). This is the value specified in the `pdbuffer -m` option in the system common definition.

lll

Maximum number of batch input pages (3 decimal digits). This is the value specified in the `pdbuffer -p` option in the system common definition.

mm...m

Internal information used by the system. `******` is always displayed.

nn...n

Internal information used by the system. ** is always displayed.

[1] Global buffer specified in the -o option of the pdbuffer operand in the system common definition.

```
-k sts specified
STATISTICS OF GLOBAL BUFFER
EDIT TIME aa...a
BUFFNAME SVID  HIT(REF,UPD)  RFGET  READ  RFFLS  REFBUF     WAITL
         LAST-EXEC-TIME      UPGET  WRITE UPFLS  UPBUF(TRG) SYNC
                             PRRED  PRHIT PRINS  PRREQ      INSB
                              LRREQ  LWREQ LRPAG  LWPAG
                              CINSM  CFMAX CFAVG
bb...b       cc...c   eee(fff,ggg)   hh...h   jj...j   ll...l    nn...n        qq...q
             dd...d                  ii...i   kk...k   mm...m   oo...o(pp...p) rr...r
                                     uu...u   vvv      ww...w   xx...x         yy...y
                                      zz...z   AA...A   BB...B   CC...C
                                      DD...D   EE...E   FF...F
        [NOT  FOUND ss...s=tt...t]
```

Explanation

aa...a

Time when pdbufls command was executed (year/month/date hour:minute:second).

bb...b

Global buffer name (up to 16 characters).

cc...c

Server name (up to 8 characters).

dd...d

Last time statistical information was acquired for each global buffer (*year/month/date hour:minute:second*)

If no statistical information has been collected or if the -d option is specified, ****-**-** **:**:** is displayed.

eee

Hits rate for the global buffer (percentage; 3 decimal digits).

fff

Hits rate for referencing requests (percentage; 3 decimal digits).

ggg

Hits rate for updating requests (percentage; 3 decimal digits).

hh...hh

Number of reference GETs.[1]

ii...i

Number of update GETs.[1]

jj...j

Number of actual READs.[1]

kk...k

Number of actual WRITEs.[1]

ll...l

Number of reference buffer flushes.[1]

mm...m

Number of update buffer flushes.[1]

nn...n

Number of current reference buffers (1 to 6 decimal digits in the 32-bit mode; 1 to 10 decimal digits in the 64-bit mode).

oo...o

Number of current update buffers (1 to 6 decimal digits in the 32-bit mode; 1 to 10 decimal digits in the 64-bit mode).

pp...p

Number of update buffers for the next time output of a deferred write trigger occurs (1 to 6 decimal digits in the 32-bit mode; 1 to 10 decimal digits in the 64-bit mode).

qq...q

Number of times a wait for buffer lock release occurred.[1]

rr...r

Number of DB synchronization points.[1]

ss...s

SERVER when the -s option is specified; BUFFER POOL when the -a option is specified.

tt...t

Server name when the `-s` option is specified (up to 8 characters); global buffer name when the `-a` option is specified (up to 16 characters).

uu...u

Number of prefetch input pages.[1]

vvv

Prefetch hit range (3 decimal digits).

ww...w

Number of prefetch buffer shortages.[1]

xx...x

Number of prefetch `READ` requests.[1]

yy...y

Number of out of buffer errors.[1]

zz...z

Number of LOB buffer `read` requests.[1]

AA...A

Number of LOB buffer `write` requests.[1]

BB...B

Number of pages input into LOB buffer in batch.[1]

CC...C

Number of pages output from LOB buffer in batch.[1]

DD...D

Internal information used by the system.

EE...E

Internal information used by the system.

FF...F

Internal information used by the system.

[1] The display format is as follows:

Numeric range		Display format	
0 ～	999	0 ～	999
1,000 ～	9,994	1.00K ～	9.99K[1]
9,995 ～	99,949	10.0K ～	99.9K[1]
99,950 ～	999,499	100K ～	999K[1]
999,500 ～	9,994,999	1.00M ～	9.99M[1]
9,995,000 ～	99,949,999	10.0M ～	99.9M[1]
99,950,000 ～	999,499,999	100M ～	999M[1]
999,500,000 ～	4,294,967,295	1.00G ～	4.29G[1]

K: Kilo

M: Mega

G: Giga

[1] The decimal place beyond the displayed digits is rounded off.

■ DAT output format (when `-x` and `-y` options are specified)

- -k def

```
"DEFINE_OF_GLOBAL_BUFFER"[CR]
"EDIT_TIME"[CR]
"aa...a"[CR]
"BUFFNAME","SVID","TYPE","SIZE","NUM","WRATIO","RDAREA/
INDEX_NAME","PRMAX","PRNUM","CSIZE","MAPS"[CR]
"bb...b","cc...c","d",ee,ff...f,ggg,"hh...h",kk...k,lll,mm...m,nn[CR]
```

- -k sts

```
"STATISTICS_OF_GLOBAL_BUFFER"[CR]
"EDIT_TIME"[CR]
"aa...a"[CR]
"BUFFNAME","SVID","LAST_EXEC_TIME","HIT","REFHIT","UPDHIT","RFGET","UPGET","READ","W
RITE",
"RFFLS","UPFLS","REFBUF","UPBUF","TRG","WAITL","SYNC","PRRED","PRHIT","PRINS","PRREQ
","INSB",
"LRREQ","LWREQ","LRPAG","LWPAG","CINSM","CFMAX","CFAVG"[CR]
"bb...b","cc...c","dd...d",eee,fff,ggg,"hh...h","ii...i","jj...j","kk...k",
"ll...l","mm...m",nn...n,oo...o,pp...p,"qq...q","rr...r","uu...u",vvv,"ww...w","xx...x","yy...y",
"zz...z","AA...A","BB...B","CC...C","DD...D","EE...E","FF...F"[CR]
```

Note

`[CR]` indicates a linefeed.

Rules for output in the DAT format

1. The elements are separated by the comma (,), and information about one global buffer is displayed on each line. If multiple RDAREAs have been allocated to the same global buffer, one line of global buffer definition information is displayed for each allocated RDAREA, and all items other than the RDAREA name are repeated.
2. When a character element is included, it is enclosed in double quotation marks (`"`). Global buffer statistical information including the unit is treated as a character element.
3. Because the `pdbufls` command's execution time is included on the title line, this information is displayed only when title line output (`-y` option) is specified.
4. If the following information items cannot be acquired, the command displays the null value for a numeric element and only `""` for a character element:

   - Buffer length and number of buffer sectors in the global buffer definition information

   - Previous time the global buffer statistical information was acquired

   The internal information used by the system will be the null value or `""`.
5. When information is displayed in DAT format, the following error message is not output:

   `[NOT FOUND` *ss...s=tt...t*`]`

pdbufmod (Change global buffer dynamically)

Function

The pdbufmod command adds, deletes, or changes a global buffer during HiRDB operation.

The change made by pdbufmod is not inherited after the normal termination of unit or server. Therefore, you need to apply this change using the pdbuffer operand in the system definition before the next normal start of HiRDB. The pdchgconf command lets you change the system definitions without having to terminate HiRDB.

You can use the pdbufmod command if HiRDB Advanced High Availability has been installed.

Executor

HiRDB administrator

Format

```
pdbufmod -k change-type -a global-buffer-name
        [{-r RDAREA-name[,RDAREA-name]...|-b RDAREA-name[,RDAREA-name]...
            |-o|-i authorization-identifier.index-identifier}]
        [-n buffer-sectors-count [-l buffer-size]]
        [-m maximum-concurrent-prefetch-operations-count] [-p maximum-batch-input-pages-count]
        [-w updated-output-page-rate-during-deferred-write-trigger]
        [-y update-buffer-sectors-count-for-deferred-write-trigger-event]
```

Options

- -k *change-type*

Specifies the type of change to be made.

add:

Specifies that a new global buffer is to be added or another global buffer is to be added to the existing global buffers.

To add a new global buffer, make sure that the -n option is specified. If you are adding a new global buffer specifying the -r, -b, or -i option, the RDAREA or index to be specified must be defined beforehand.

If the existing shared memory segment does not have enough space, the new global buffer is created in a new shared memory segment.

`del`:

Specifies that a global buffer is to be deleted or specific RDAREAs are to be deleted from the global buffer.

To delete specific RDAREAs, specify the RDAREAs in the `-r` or `-b` option.

You must place the RDAREAs allocated to a global buffer to be deleted or the RDAREAs to be deleted from a global buffer in shutdown and closed status beforehand.

`upd`:

Specifies that the global buffer definition is to be changed. The global buffer definition items include the maximum number of concurrent prefetch operations, maximum number of batch input pages, and updated output page rate during deferred write trigger.

Once the global buffer definition is changed, the contents of the global buffer cache are discarded.

If the change made to the global buffer definition results in an increase in the memory size for the global buffer and there is not enough space in the existing shared memory segment, the command creates the global buffer with the definitions in a new shared memory segment.

The following table shows the relationship between the change type and other options:

Other option	-k change-type		
	add	del	upd
`-a`	M	M	M
`-r`	O	O	N
`-b`	O	O	N
`-o`	O[1]	N	N
`-i`	O[1]	N	N
`-n`	O[1, 2]	N	O
`-l`	O[1]	N	O
`-m`	O[1]	N	O
`-p`	O[1]	N	O

Other option	-k change-type		
	add	del	upd
-w	O[1]	N	O
-y	O[1]	N	O

Legend:

M: Mandatory

O: Optional

N: Not specifiable

[1] You can specify this operand when a new global buffer is to be added.

[2] To add a new global buffer, make sure that this operand is specified.

- `-a` *global-buffer-name* ～ <identifier> ((1-16))

Specifies the name of the global buffer to be added, deleted, or changed.

If you are specifying `-k add` (to add an RDAREA to an existing global buffer), `-k del`, or `-k upd`, specify the name of a global buffer that has already been defined. If you are specifying `-k add` (to add a new global buffer), specify a global buffer name that is unique in the server.

- `-r` *RDAREA-name*`[,`*RDAREA-name*`]...` ～ <identifier> ((1-30))

Specifies the names of the RDAREAs that are to be allocated to the global buffer or that are to be deleted from the global buffer.

Rules

1. A batch specification of RDAREA names is not permitted.
2. If you specify the number of buffer sectors with the `-n` option and also specify multiple RDAREAs for a single global buffer, the largest page size in the RDAREAs becomes the buffer size. Therefore, we recommend that you allocate RDAREAs with the same page size to a global buffer. If any of the RDAREAs with identical page size contains a table that serves a different purpose or to which frequent random or sequential access is made, that RDAREA should be allocate to a separate global buffer in order to improve buffering efficiency.
3. You cannot specify a duplicate RDAREA name. If you specify a duplicate RDAREA name, the command eliminates the RDAREA from processing.
4. You can specify a maximum of 128 RDAREA names. If you specify more

than 128 RDAREA names, the command ignores the excess RDAREA names.

5. If an RDAREA name is enclosed in \", the command treats it as being case sensitive. If it is not enclosed in \", the command treats it as in all uppercase letters. If an RDAREA name contains a space, enclose the RDAREA name in \".
6. To specify a user RDAREA, the data dictionary RDAREA must be in one of the following statuses:

   - Open and shutdown release status

   - Open and command shutdown status

- -b *RDAREA-name*[,*RDAREA-name*]... ～ <identifier> ((1-30))

Specifies the names of LOB RDAREAs that are to be allocated to the global buffer or that are to be deleted from the global buffer.

Effective specification method

Specify the following LOB RDAREAs:

- LOB RDAREA that stores a plug-in index
- LOB RDAREA with a small amount of data that is accessed frequently

We recommend that you allocate a single LOB RDAREA to a single LOB global buffer.

Rules

1. You can specify data dictionary LOB RDAREAs, user LOB RDAREAs, and registry LOB RDAREAs. An error results if any other type of RDAREA is specified.
2. A batch specification of RDAREA names is not permitted.
3. If you specify the -b option, you cannot specify the -m, -p, or -w option.
4. If you have not allocated the global buffer (when the -o option is specified), you must specify the LOB RDAREA specified in the -r option also in the -b option.
5. You cannot specify a duplicate RDAREA name. If you specify a duplicate RDAREA name, the command eliminates the RDAREA from processing.
6. You can specify a maximum of 128 RDAREA names. If you specify more than 128 RDAREA names, the command ignores the excess RDAREA names.
7. If an RDAREA name is enclosed in \", the command treats it as being case sensitive. If it is not enclosed in \", the command treats it as in all uppercase

letters. If an RDAREA name contains a space, enclose the RDAREA name in \".

8. If you specify a user LOB RDAREA, the data dictionary RDAREA must be in one of the following statuses:

   - Open and shutdown release status

   - Open and command shutdown status

- -o

Specifies that a global buffer is to be allocated to all RDAREAs that are not specified with the -r option. You can specify the -o option only once. If this option is specified more than once, the first option specified takes effect.

- -i *authorization-identifier*.*index-identifier*

  authorization-identifier: ～ <identifier> ((1-8))

  index-identifier: ～ <identifier> ((1-30))

Specifies the name of the index (*authorization-identifier*.*index-identifier*) to which an index global buffer is to be allocated.

Effective specification method

Specify an index that is used frequently. By allocating a global buffer to a frequently-used index, you can reduce the number of input/output operations because the percentage of index pages made resident in memory increases.

This effect is especially large if you allocate an index defined for a cluster or unique key to a global buffer. Note that the index identifier of a cluster key is determined by HiRDB. After defining a table, search the dictionary table (INDEX_NAME column of SQL_INDEXES table) to obtain the index identifier.

Allocating all indexes to index global buffers degrades the overall utilization factor of global buffers. Therefore, carefully select the appropriate indexes for allocation of global buffers, taking memory size into consideration.

Rules

1. If a pair of authorization identifier and index identifier is enclosed in \", the command treats it as being case sensitive. If it is not enclosed in \", the command treats it as in all uppercase letters. If the index identifier contains a space, enclose it in \".
2. If you specify *authorization-identifier*.*index-identifier*, the data dictionary RDAREA must be in one of the following statuses:

   - Open and shutdown release status

   - Open and command shutdown status

■ `-n` *buffer-sectors-count*

In 32-bit mode: ～ <unsigned integer> ((4-460000))

In 64-bit mode: ～ <unsigned integer> ((4-1073741824))

Specifies the number of global buffer sectors.

Guidelines for specification value

- Specifying a value that is greater than necessary reduces the number of input/output operations, but it increases overhead for searching the buffer.
- A global buffer is allocated in shared memory. If its size is larger than necessary, frequent paging occurs when other memory is used, thereby adversely affecting the performance.
- Depending on the timing of synchronization point acquisition and the percentage of update pages in the global buffer, output operations may concentrate on the database. Therefore, you also need to take into account the balance of input/output operations.
- If there are too may buffer sectors, allocation of shared memory may fail.
- Determine the number of global buffer sectors based on the following table:

Condition		Determining the number of global buffer sectors
Global buffer with the `-r` or `-o` option specified	HiRDB/Single Server	Number of concurrent SQL processing requests × number of pages used per SQL statement (about 3 to 6)
	HiRDB/Parallel Server	Number of concurrently executable users × average number of tables accessed concurrently per transaction × 3 × n[1]
Global buffer with the `-b` option specified		Total number of LOB RDAREAs × percentage of residency[2](%)
Global buffer with the `-i` option specified		Number of pages containing index × percentage of residency[3](%)

[1] The buffer hit rate cannot be improved by the number of buffer sectors obtained based on n=1. Therefore, determine the appropriate buffer space as a coefficient (buffer hit rate).

[2] Although the desired percentage of residency is 1 (100%), determine this value taking into account some factors, such as the memory capacity and the frequency of data access. For details about the total number of pages in a user LOB

RDAREA and a register RDAREA, see the *HiRDB Version 8 Installation and Design Guide*.

[3] Although the desired percentage of residency is 1 (100%), determine this value taking into account such factors as the memory size and the importance of the index. For details about the total number of pages in an index, see the *HiRDB Version 8 Installation and Design Guide*.

- Set this value in such a manner that the buffer hit rate becomes 80% or higher for applications using HiRDB. You can check the buffer hit rate with the following methods:

  - *Update buffer hit rate* and *reference buffer hit rate* provided by the statistics analysis utility (statistical information on global buffers).

  - `pdbufls` command's header name `HIT`

■ `-l` *buffer-size* ～ <unsigned integer> ((even number in the range 4-30))

Specifies the size of the global buffer in kilobytes. Specify this option together with the `-n` option.

Guidelines for specification value

Normally, you will omit this option, in which case the command assumes the maximum page size of the RDAREAs allocated to this global buffer as the buffer size. In the following cases, however, consider specifying a value:

- If you intend to add an RDAREA whose page length will be greater than this value or you will increase the page length of an RDAREA by initializing it in the future, we recommend that you specify a sufficiently large value in this option. However, if you can terminate HiRDB, there is no need to specify such a large value because the maximum page length will be assumed for the buffer the next time HiRDB starts.

Rules

1. If this option's value is less than the maximum RDAREA page size, the command assumes the maximum RDAREA page size as the buffer size.
2. If you specify an odd number in this option, the command adds 1 to the specified value in determining the buffer size.

■ `-m` *maximum-concurrent-prefetch-operations-count* ～ <unsigned integer> ((0-95000)) <<0>>

Specifies the maximum number of prefetch operations that can be executed concurrently. When a large amount of data is searched using the raw I/O facility, the prefetch facility can reduce the input/output time.

If you specify 0 in this option or omit this option, the prefetch facility will not be used. To use the prefetch facility, you must specify 1 or a greater value.

Guidelines for specification value

This option specifies the number of concurrently executable SQL statements to which the prefetch facility is applied in tables that are contained in the RDAREAs allocated to the global buffer. The prefetch facility is applied to the following SQL statements, where a value of 2 is used as the execution count for 3:

1. SELECT, UPDATE, and DELETE statements that do not use an index (excluding the = and IN conditions)
2. SELECT, UPDATE, and DELETE statements that perform searches in ascending order* using an index (excluding the = and IN conditions)
3. SELECT, UPDATE, and DELETE statements that perform searches in ascending order* using a cluster key (excluding the = and IN conditions)

* For a multicolumn index, the searches are conducted in the order specified in the index definition.

Rules

If you use the prefetch facility, you need to re-estimate the shared memory size because a buffer dedicated to batch input is allocated from the shared memory separately from the global buffers.

For details about the formulas for determining the size of shared memory required for the prefetch facility and global buffers, see the *HiRDB Version 8 Installation and Design Guide*.

- `-p` *maximum-batch-input-pages-count* ～ <unsigned integer> ((2-256)) <<32>>

Specifies the maximum number of batch input pages for the prefetch facility. This specification is applicable only when 1 or a greater value is specified in the `-m` option.

Guidelines for specification value

Based on the shared memory size and cost performance, specify the value that satisfies the following condition:

a × *b* = 64 to 128 (kilobytes)

a: Page length of the RDAREA containing the data or index of the table that is subject to prefetch processing
b: Maximum number of batch input pages

- `-w` *updated-output-page-rate-during-deferred-write-trigger*

  ～ <unsigned integer> ((0-100)) <<20>>

Specifies the percentage of updated output pages during deferred write trigger. For details about deferred write trigger, see the *HiRDB Version 8 Installation and Design Guide*.

If this operand's value is 0, the system does not output pages updated by deferred write trigger.

Guidelines for specification value

- Determine the appropriate value by checking the relevant information, such as the number of input/output operations per global buffer and the hit rate for updated pages, using the statistics analysis utility. If the hit rate for updated pages is high for a global buffer, specify a low output rate for this global buffer; if the hit rage is low, specify a high output rate.
- If the specified value is too large, update processing occurs frequently, resulting in frequent input/output operations. The number of pages to be written in the database also increases during delayed write processing, thereby adversely affecting throughput. On the other hand, if the specified value is too small, the number of pages to be written in the database increases during a synchronization point dump. Therefore, reduce the number of times the same pages are written in the database between synchronization point dumps.
- Specify a value of `0` if all pages are in the global buffer and the same pages are frequently updated by multiple transactions. This can reduce the total number of pages that are written in the database between the synchronization points.

■ `-y` *update-buffer-sectors-count-for-deferred-write-trigger-event*

～ <unsigned integer> ((2-2147483647))

Specifies the number of update buffer sectors for when deferred write processing is to be executed. When the number of update buffer sectors reaches the specified value, the deferred write trigger writes updated pages to the disk.

If the `-y` option is omitted, the number of update buffer sectors for the deferred write trigger event depends on the combination of the `pd_dbbuff_rate_updpage` and `pdbuffer -y` operands, as shown below:

pd_dbbuff_rate_updpage operand specification	pdbuffer -y operand specification	Number of update buffer sectors for deferred write trigger event
Specified	Specified	`pdbuffer -y` operand value
	Omitted	Number of global buffer sectors x `pd_dbbuff_rate_updpage` operand value
Omitted	Specified	`pdbuffer -y` operand value

pd_dbbuff_rate_upd page operand specification	pdbuffer -y operand specification	Number of update buffer sectors for deferred write trigger event
	Omitted	Determined by HiRDB

Specification guidelines

Normally, you omit this operand. There may be times when deferred write processing is not completed within the synchronization point interval. In such a case, you may want to reduce the writing time by reducing the number of updated buffers and accepting a slight reduction in the updated buffer hit rate; you do this by specifying this operand. Assume 50% (the initial value set by HiRDB) as the specification guideline, and determine the operand value by referring to the section *Tuning the synchronization point processing time when deferred write processing is used* in the manual *HiRDB Version 8 System Operation Guide*.

Rules

If the value specified in this option is greater than the number of global buffer sectors, the number of global buffer sectors is used.

Notes

If the value of the `-y` option is too small, there is an increase in the frequency of deferred write processing, resulting in an increase in workload. For this reason, it is important to specify an appropriate value based on the specification guidelines.

Rules

1. The `pdbufmod` command can be executed only while HiRDB is active.
2. The `pdbufmod` command must be executed at the server machine that contains the single server or where the system manager is located.
3. When the `pdbufmod` command is executed, the system locks the related RDAREAs and their RDAREA status in the EX mode. Any other transaction or utility accessing these RDAREAs is placed in the wait status.
4. You cannot execute the `pdbufmod` command in the following cases:
   - The rapid system switchover facility or standby-less system switchover facility is being used.
   - `pd_dbbuff_modify=Y` is not specified in the system definition.
5. You cannot execute more than one `pdbufmod` command at the same time.
6. The maximum number of global buffer sectors that can be defined and the maximum number of new shared memory segments that can be added are determined by the values of the `pd_max_add_dbbuff_no` and

pd_max_add_dbbuff_shm_no operands in the system definition.

Notes

1. Deletion of shared memory segments that is accompanied by deletion of a global buffer may be delayed depending on the command's termination timing. To immediately delete them, execute the pdpfresh command.
2. If the unit is terminated or HiRDB is terminated forcibly while the pdbufmod command is executing, use the pdbufls -k def command after the restart to make sure that the previous operation has been completed. If it has not been completed, re-execute the pdbufmod command.
3. The following shows the pdbufmod command's return codes:

   0: Normal termination

   4: Warning termination (some server processing terminated abnormally)

   8: Abnormal termination

   12: Abnormal termination (an event occurred that prevented output of an error message)

   If the return code is 12, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the pdbufmod command. If no error message has been output to the event log, contact the customer engineer.

Examples

Example 1:

This example adds the global buffer gbuf01 and then allocates the added RDAREA RDAREA1 to this global buffer.

```
pdbufmod -k add   .................1
         -a gbuf01   ..............2
         -r RDAREA1   .............3
         -n 1000   ................4
```

Explanation

1. Specifies that a global buffer is to be added.
2. Name of the global buffer to be added
3. RDAREA to be allocated
4. Number of sectors for the global buffer

Example 2:

This example deletes the global buffer `gbuf01` to which no RDAREA has been allocated.

```
pdbufmod -k del   .................1
         -a gbuf01   ..............2
```

Explanation

1. Specifies that a global buffer is to be deleted.
2. Name of the global buffer to be deleted

Example 3:

This example changes the number of sectors for the global buffer `gbuf01` from 1000 to 2000.

```
pdbufmod -k upd   .................1
         -a gbuf01   ..............2
         -n 2000   ................3
```

Explanation

1. Specifies that a global buffer definition is to be changed.
2. Name of the global buffer whose definition is to be changed
3. Number of buffer sectors after change

pdcancel (Forcibly terminate UAP or utility processing)

Function

The pdcancel command forcibly disconnects and terminates the process on HiRDB for processing a specified UAP or utility.

When a process is still active on HiRDB although the corresponding UAP has been terminated due to an error, etc., this command is used to terminate the process. This command forcibly terminates a process in a single server on the HiRDB that has been connected with a UAP or terminates processes in front-end and back-end servers.

The pdcancel command also cancels utility processing that is executing.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdcancel {-U| -u UAP-identifier [-i process-ID]| -i process-ID}
```

HiRDB/Parallel Server

```
pdcancel {-x host-name | -X unit-identifier}
         {-U| -u UAP-identifier [-i process-ID]| -i process-ID}
```

Options

- **UAP**

-x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host containing the front-end server to which the corresponding UAP is connected.

The host name can be obtained with the pdls command.

-X *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit containing the front-end server to which the corresponding UAP is connected.

-U

Specifies that all UAPs running on the unit with the specified host name are to be forcibly terminated.

-u *UAP-identifier* ～ <character string> ((1-30))

Specifies the UAP identifier (PDCLTAPNAME) of the client UAP corresponding to

the process on HiRDB that is to be forcibly terminated. If more than one UAP with the same identifier is running on the corresponding unit, the -i option must be specified.

-i *process-ID* ～ <unsigned integer> ((1-10))

Specifies the process ID of the process that is to be forcibly terminated. When this option is specified together with the -u option, the process ID of a single server or front-end server must be specified. The pdls command (with prc specified in the -d option) can be used to display process IDs. The process ID of the process to be forcibly terminated can be determined from the service request acceptance time for the corresponding UAP.

■ **Utility**

-x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host containing the system manager. In the case of the database initialization utility (pdinit) or the database structure modification utility (pdmod), the host name of the dictionary server must be specified.

The host name can be obtained by the pdls command.

-X *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit where the system manager is located. However, if the database initialization utility or the database structure modification utility is being terminated forcibly, the identifier of the unit which where the dictionary server is located must be specified.

-u *UAP-identifier* ～ <identifier> ((1-8))

Specifies the name of the server processing the utility. A server name begins with 0m in the list of names displayed by the pdls command (prc specified in the -d option). For example, the server name for the database load utility (pdload) is displayed as 0mload0. To forcibly terminate the database structure modification utility, pdmod must be specified as the UAP identifier.

The following table lists the server name for each utility:

Utility	Server name*
Database initialization utility (pdinit)	0minit*x*
Database load utility (pdload)	0mload*x*
Database structure modification utility (pdmod)	pdmod
Database reorganization utility (pdrorg)	0mrorg*x*
Dictionary import/export utility (pdexp)	0mexp*x*

Utility	Server name*
Rebalancing utility (pdrbal)	0mrbal*x*
Free page release utility (pdreclaim)	0mrorg*x*
Global buffer residence utility (pdpgbfon)	0mrorgx
Database condition analysis utility (pddbst)	0mdbst
Optimizing information collection utility (pdgetcst)	0mgcst
Database copy utility (pdcopy)	0bcpy*x*
Database recovery utility (pdrstr)	0brstr

* *x* at the end of a name is a numeric character (0, 1, 2, ...).

`-i` *process-ID* ～ <unsigned integer> ((1-10))

Specifies the process ID of the utility that is to be terminated forcibly. This option must be specified in order to terminate a utility.

Rules

1. The `pdcancel` command can be executed only while HiRDB is active.
2. The `pdcancel` command can be executed from any server machine.
3. In the case of UAP-connected processes, the `pdcancel` command can be used only to forcibly terminate processes engaged in transaction processing. If transaction processing has not started or has already terminated, but the single server or front-end server does not send a response to the UAP, terminate the single server or front-end server process by executing the `pdkill` command. Whether or not the process corresponding to the specified UAP is engaged in transaction processing can be determined from the information displayed by the `pdls` command (`trn -a` specified in the `-d` option). If information is displayed for the UAP, the process is engaged in transaction processing; otherwise, the process is not engaged in transaction processing.
4. If the `pdcancel` command is not executed for a process that is still engaged in processing at the HiRDB side although the corresponding UAP had been terminated due to an error, the process will be forcibly terminated when the time specified in the corresponding client environment definition (`PDSWAITTIME`) is reached.

Notes

1. The result of the `pdcancel` command can be checked by the `pdls` command (`-d prc` specified).

2. The `pdcancel` command may not execute if a non-alphanumeric character was specified for `PDCLTAPNAME` in the client environment definition.
3. A utility should not be cancelled unnecessarily.
4. Return code `0` for the `pdcancel` command indicates normal termination, and return code `8` indicates abnormal termination (such as an invalid option or `rsh` error).

pdcat (Display messages in file)

Common function

The pdcat command displays the contents of HiRDB system files for a specified type of display object.

Common format

pdcat [-d *display-object-type*] [*display-object-type-options*]

Note

The specifiable options depend on the display object type.

Common options

- -d *display-object-type*

Specifies the type of file to be displayed:

sts

Status file

Default

Message log file

Common note

1. Return code 0 for the pdcat command indicates normal termination, and return code 8 indicates abnormal termination.

pdcat [-d sts specified]

Function

Outputs status file contents to the standard output.

Executor

HiRDB administrator

Format

Unit status files

```
pdcat -d sts [{-x host-name| -u unit-identifier}]
      -f physical-filename [-v][-e]
```

Server status files

```
pdcat -d sts -s server-name -f physical-filename [-v][-e]
```

Options

- {`-x` *host-name* | `-u` *unit-identifier*}

Specifies the identifier of the unit or the name of the host corresponding to the unit status file. If this option is omitted, the information for the entire HiRDB system is displayed.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the name of the host corresponding to the unit status file.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit corresponding to the unit status file.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the server name corresponding to the server status file. Even though there is only one HiRDB server in a HiRDB/Single Server, it must be specified in order to identify the object of server status file operations.

- `-f` *physical-filename* ～ <pathname> ((up to 167 characters))

Specifies the absolute pathname of the physical file. A physical file name must be specified as *HiRDB-file-system-area-name**HiRDB-filename*.

- `-v`

Specifies that the following items concerning the status file are to be checked:

- Record integrity by first and last record numbers.

- Record length and records count, based on the file management information displayed by the `pdstsinit` command
- Completion of file updating, based on the status file management information
- Records count and record type in the status file management information

If checking detects an error, an error message is output after the contents of the specified status file have been displayed.

If this option is omitted, status file checking is not performed.

■ `-e`

Specifies that the status file is to be opened in the lock mode. If the `pdcat` command is executed with this option specified at the time the unit or server is being started, HiRDB may terminate abnormally due to a lock error.

When this option is omitted, the status file is not opened in the lock mode, and the correct status may not be displayed if the specified status file is being used by the HiRDB.

Rules

1. The `pdcat` command can be executed at any time, whether or not HiRDB is active. However, it cannot be executed during HiRDB startup or termination processing.
2. The `pdcat` command can be executed from any server machine.

Output format

```
Path name:aa...aa
 Initial_time Current_time  R_le   R_co  factor  Available manage
 bb...b       cc...c        ddddd  ee...e  fff%      gg...g      hh...h
```

Explanation

aa...a

Name of the physical file (up to 63 digits)

bb...b

Initialization date and time (in the format *year/month/date hour:minute:second*)

cc...c

Current date and time (in the format *year/month/date hour:minute:second*)

If not being used as the current file `--/--/-- --:--:--` is displayed.

ddddd

Record length (decimal)

ee...e

Records count (decimal)

fff

Record utilization in the file (%)

gg...g

Number of contiguous free records in the file (decimal)

hh...h

Number of management records in the file (decimal)

The information displayed for a file that has been corrupted or in which an error has occurred may not be correct.

pdcat [-d option omitted]

Function

The pdcat command displays messages contained in a message log file.

Executor

Any user (with or without privileges)

Format

HiRDB/Single Server

```
pdcat [-a| b | ab  ] [-c significance-code[,significance-code]...]
      [-y yyyymmdd] [-t hhmmss] [-T hhmmss] [-n records-count]
      [filename]
```

HiRDB/Parallel Server

```
pdcat [-a| b | ab  ] [-c significance-code[,significance-code]...]
      [-x host-name| -u unit-identifier]] [-y yyyymmdd]
      [-t hhmmss] [-T hhmmss] [-n records-count]
      [filename]
```

Options

- -{a | b | ab}

-a

Specifies that all message information except the message IDs and message texts is to be displayed.

-b

Specifies that the message IDs and message texts only are to be displayed.

-ab

Specifies that all message information is to be displayed.

- -c *significance-code*

Specifies a significance code when specific messages are to be selected on the basis of their significance codes (for details about significance codes, see the manual *HiRDB Version 8 Messages*). When this option is omitted, messages are not selected on the basis of significance codes.

- {-x *host name*| -u *unit-identifier*}

To display messages by host or unit, specifies the unit's unit identifier or the host's host name. When this option is omitted, messages for all hosts or units are displayed.

`-x` *host-name* ～ <identifier> ((1-32))

To display messages by host, specifies the host's name.

To display server-related messages when the standby-less system switchover (effects distributed) facility is used, you must specify the name of a host whose server is running. This is because messages on the server in the switched system are being output to the message log for the host whose server is running.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

To display messages by unit, specifies the unit identifier of that unit.

To display server-related messages when the standby-less system switchover (effects distributed) facility is used, you must specify the identifier of a unit whose server is running. This is because messages on the server in the switched system are being output to the message log for the unit whose server is running.

- `-y` *yyyymmdd*

Specifies a date, when only messages for and subsequent to the specified message log date are to be displayed (in the format *year-month-date*, where the year consists of the four-digit calendar year, and the month and date each consist of two digits). When this option is omitted, messages are not selected on the basis of the message log date.

- `-t` *hhmmss*

Specifies a time, when only messages for and subsequent to the specified message log time are to be displayed (in the format *hour-minute-second*). When this option is omitted, messages are not selected on the basis of a message log begin time.

- `-T` *hhmmss*

Specifies a time, when only messages for and prior to the specified message log time are to be displayed (in the format *hour-minute-second*). When this option is omitted, messages are not selected on the basis of a message log end time.

- `-n` *records-count* ～ <unsigned integer> ((1-32767))

Specifies the maximum number of message log records to be displayed. When this option is omitted, all selected message log records are displayed.

Command arguments

- *filename* ～ <path-name>

Specifies the name of the message log file to be displayed. A file created by copying a message log file can be specified. When this option is omitted, the messages in the following message log files are merged into chronological order and then displayed:

- `%PDDIR%\spool\pdlog1`
- `%PDDIR%\spool\pdlog2`

Rules

1. The `pdcat` command can be executed only while HiRDB is active.
2. The `pdcat` command can be executed from any server machine.

Notes

1. For a HiRDB/Parallel Server, the storage location of the message log file depends on the `pd_mlg_msg_log_unit` operand value in the system definition, as shown below:
   - When `pd_mlg_msg_log_unit=manager` is specified

     Unit where the system manager is located
   - When `pd_mlg_msg_log_unit=local` is specified

     All units
2. If `pd_mlg_msg_log_unit=local` was specified, the `pdcat` command displays the contents of the message log file for the corresponding unit. If `pd_mlg_msg_log_unit=manager` was specified or the `pd_mlg_msg_log_unit` operand was omitted, and if the system manager is located on the unit, the command displays the contents of the message log file for that unit; if the system manager is not located on the unit, the command displays nothing.

Output format

aa...a bb...b cc...c dddd ee...e ff...f gg...g hhhh ii...i
jjj kk...k ll...l

Explanation

aa...a

Message serial number (7 decimal digits).

bb...b

Process ID (10 decimal digits).

cc...c

Message serial number within the process (server) (7 decimal digits).

dddd

HiRDB identifier (4 characters).

ee...e

Date (in the format *year/month/date*) (10 decimal digits).

ff...f

Time (in the format *hour:minute:second*) (8 decimal digits).

gg...g

Request source host name (first 8 characters).

When the standby-less system switchover (effects distributed) facility is used, this is the name of the host whose server is running.

hhhh

Unit identifier (4 characters).

When the standby-less system switchover (effects distributed) facility is used, this is the identifier of the unit whose server is running.

ii...i

Message output request source server name (8 characters).

jjj

Internal information used by the system (3 characters).

kk...k

Message ID (11 decimal digits).

ll...l

Message text (up to 223 characters).

pdchgconf (Reconfigure the system)

Function

The `pdchgconf` command changes system definitions during HiRDB operation.

Before executing the `pdchgconf` command, check that the definitions after the change will be valid (execute the `pdconfchk -d chgconf` command).

You can use the `pdchgconf` command only if HiRDB Advanced High Availability has been installed.

Executor

HiRDB administrator

Format

```
pdchgconf
```

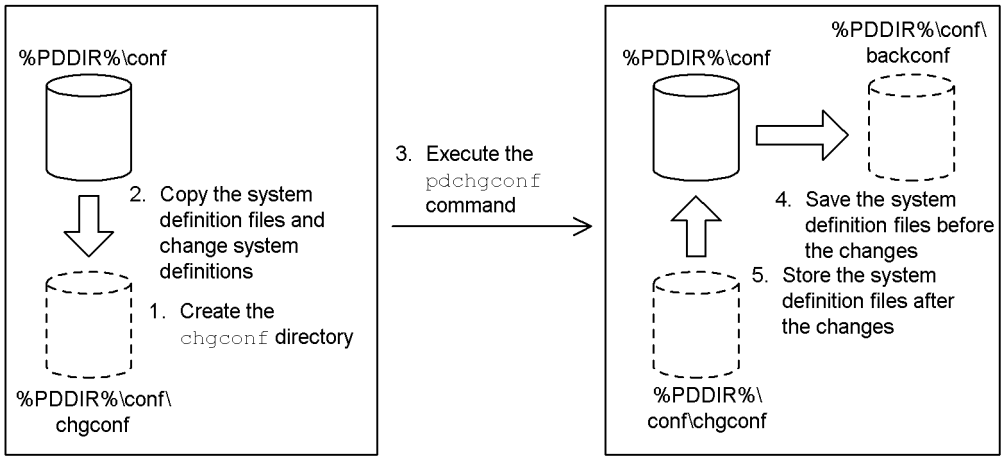
Options

None

Changing procedure

Figure 2-2 provides an overview of changing system definitions.

Figure 2-2: Overview of changing system definitions (pdchgconf command)



Explanation

1. Before executing the pdchgconf command, create the chgconf directory in %PDDIR%\conf in order to store the system definition file after the change. In the case of a HiRDB/Parallel Server, create this directory in all units.
2. After copying all system definition files from %PDDIR%\conf to the chgconf directory, make desired changes to the system definitions.

   For details about the system definitions that can be changed, see the *HiRDB Version 8 System Operation Guide*.
3. Execute the pdchgconf command.
4. When the pdchgconf command terminates, the system definition files before the changes are saved in the backconf directory in %PDDIR%\conf.
5. The system definition files after the changes are stored in %PDDIR%\conf.

Rules

1. The pdchgconf command can be executed only while HiRDB is active.
2. The pdchgconf command must be executed at the server machine that contains the single server or where the system manager is located.

Notes

1. You cannot use the pdchgconf command to change the pd_system_id or pd_master_file_name operand. Nor can you delete this information from the current system file or an overwrite-disabled system file. An attempt to change or delete this information will result in an error.
2. When you execute the pdchgconf command, the system log files are swapped. Before executing the pdchgconf command, make sure that there is enough space for operation with the new configuration. If you execute the pdchgconf command while there is no swappable system log file, HiRDB displays the KFPS01256-E message and stops (displays abort code Psjnf07 or Psjn381). If HiRDB has stopped, provide a swappable system log file and then start HiRDB. If there is no swappable system log file, you need to add one. For details about how to handle when there is no swappable file, see the *HiRDB Version 8 System Operation Guide*.
3. Although the system log files are swapped when you execute the pdchgconf command, no message to that effect is displayed. Therefore, if you need the message log in the message log file, back up the message log file before executing the pdchgconf command.
4. If the following conditions apply, you may not be able to execute the pdchgconf command:
   - For a HiRDB/Parallel Server, some units or servers have terminated

(including reduced activation). If only recovery-unnecessary front-end servers are inactive, you may be able to execute the pdchgconf command; for details, see *Notes when a recovery-unnecessary front-end server is used*.

- For a HiRDB/Parallel Server, a communication error has occurred in the network between units.
- The pdrplstop command is executing.
- A transaction or utility was still active 15 minutes after the pdchgconf command was executed.
- HiRDB was started with the pdstart -r command.
- While HiRDB Datareplicator is linked, the value of the pd_rpl_init_start operand does not match the actual data extraction mode.*

* The details are as follows:

Value of the pd_rpl_init_start operand before change	HiRDB Datareplicator linkage status	Value of the pd_rpl_init_start operand after change	Whether or not the pdchgconf command can be executed
Y	Active	Y	Y
		N or omitted	N
	Has been stopped by the pdrplstop command	Y	N
		N or omitted	Y
N or omitted	Inactive	Y	N
		N or omitted	Y
	Has been activated by the pdrplstart command	Y	Y
		N or omitted	N

Legend:

Y: Executable

N: Not executable

5. The following limitations apply while the pdchgconf command is executing:
   - UAP response time may be delayed. The guideline for the pdchgconf command's execution time is: *time required for normal termination of HiRDB + time required for normal startup of HiRDB*.

- Do not execute any other operation command or utility.
- The system switchover facility is not available. If the system switchover facility is using Hitachi HA Toolkit Extension, stop the standby system before executing the pdchgconf command, update the system definitions after the pdchgconf command terminates normally, and then restart the standby system.

6. To add or delete a unit or server while HiRDB Datareplicator is linked, you must re-create HiRDB Datareplicator's extraction environment. If a HiRDB transaction is processed and data is added, updated, or deleted in the extracted database while HiRDB Datareplicator's extraction environment is being re-created, inconsistency occurs in the target database. In this case, do not use the pdchgconf command to change the configuration.
7. If the pdchgconf command's execution fails, the system automatically restores the previous system definitions, using the system definition files stored in the backconf directory under %PDDIR%\conf. If this recovery processing fails, HiRDB may terminate itself. In such a case, check the error messages and take appropriate action.
8. During execution of the pdchgconf command, error messages and abort codes that accompany HiRDB startup and termination may be displayed. In such a case, there may be a problem with the new system definitions or environment. Take appropriate action according to the displayed error messages and abort codes.
9. Return code 0 for the pdchgconf command indicates normal termination, and return code 8 indicates abnormal termination.
10. While only recovery-unnecessary front-end servers are inactive, you still may not be able to execute the pdchgconf command in the following cases:
    - The pd_mode_conf operand value in the system definition is AUTO.
    - After the system definition is changed, a currently inactive unit's front-end server will no longer be a recovery-unnecessary front-end server.

Notes when a recovery-unnecessary front-end server is used

1. If a recovery-unnecessary front-end server has stopped normally and the corresponding unit is running, the pdchgconf command cannot be executed.
2. If all units are running except ones that contain a recovery-unnecessary front-end server and no individual server has stopped, the pdchgconf command can be executed even when a unit containing a recovery-unnecessary front-end server is inactive. However, if *Notes 5* through *9* are true, the pdchgconf command cannot be executed.

   The pdchgconf command replaces the system definition file in %PDDIR%\conf in the inactive unit with the system definition file in %PDDIR%\conf\chgconf.

The command then restarts the inactive unit in order to apply the new system definition. If a problem occurs, such as a machine shutdown or a network error, the pdchgconf command may not replace the system definition file in the inactive unit or may not restart the unit. In such a case, the command displays the KFPS04665-W message. If this message is displayed, you must take actions (i) and (ii) below before restarting the unit. If these actions are not taken, HiRDB cannot accept the correct changes to the system definition, resulting in invalid operation. If such invalid operation occurs, immediately terminate the corresponding unit and take actions (i) and (ii):

(i) Replace the system definition file in %PDDIR%\conf or %PDCONFPATH% with the new one.

(ii) Use the pdconfchk command to make sure that there are no differences in the system definitions between the units.

pdchprc (Change number of startup server processes)

Function

The pdchprc command changes the number of resident server processes and the maximum number of startup server processes. It also displays the current number of resident server processes and the current maximum number of startup server processes.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdchprc [-p resident-processes-count[,maximum-startup-processes-count]]
```

HiRDB/Parallel Server

```
pdchprc {-a| -s server-name}
        [-p resident-processes-count[,maximum-startup-processes-count]]
```

Options

- -p *resident-processes-count*
  [,*maximum-startup-processes-count*] ～ <unsigned integer> ((0-2048))

Specifies new values for the number of resident server processes and maximum number of startup server processes. If the maximum number of startup processes is omitted, the number of resident processes is assumed. If the -p option is omitted, the command displays the current number of resident server processes and current maximum number of startup server processes at the time the pdchprc command is executed.

The number of processes can be any value in the range 0 to the maximum number of startup server processes specified in the HiRDB system definition. The maximum number of startup server processes is shown below.

- Single server or front-end server

  Value of pd_max_users *operand in the system common definition*

- Back-end server

  Value of pd_max_bes_process *operand in the server common definition or back-end server definition*

- Dictionary server

  Value of pd_max_dic_process *operand in the server common definition or*

dictionary server definition

- `-a`

Specifies that the number of resident server processes and maximum number of startup server processes for every server that is active when the `pdchprc` command is executed are to be displayed. If this option is specified together with the `-p` option, the command changes the number of resident server processes and the maximum number of startup server processes for every active front-end server and back-end servers to the values specified in the `-p` option.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of a server whose number of resident server processes and maximum number of startup server processes are to be changed or displayed.

Rules

1. The `pdchprc` command can be executed only while HiRDB is active.
2. The `pdchprc` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. An error results if the number of resident processes specified in the `-p` option is greater than the maximum number of startup processes.
4. If either value specified in the `-p` option is greater than the maximum number of startup server processes specified in the HiRDB system definition, HiRDB displays a warning message and assumes the maximum number of startup server processes specified in the HiRDB system definition.
5. Whenever the number of resident processes or the maximum number of startup processes is changed by the `-p` option, HiRDB displays a message to that effect at each applicable server.
6. The new number of resident server processes and maximum number of startup server processes set by the `pdchprc` command remain in effect at each applicable server until the server is terminated. Once HiRDB is terminated normally or forcible or planned termination is executed, this information is not inherited at the next restart.

Notes

1. If the server specified in the `-s` option is not active or is terminating, the `pdchprc` command terminates with an error. When the `-a` option is specified, neither the number of resident server processes nor the maximum number of startup server processes is changed at or displayed for any inactive server or any server that is terminating.
2. To change the number of resident processes or the maximum number of startup processes for a dictionary server, the name of the dictionary server must be

specified in the -s option.

3. If the maximum number of startup server processes is changed to 0, all transaction processing underway at the time the pdchprc command executes terminates with an error.
4. If the maximum number of startup processes for a dictionary server or back-end server is changed to a value that is less than the value specified in the HiRDB system definition and if many transactions are concentrated at that server, the transactions the server can no longer handle may terminate with an error.
5. On the HiRDB/Parallel Server, if the system manager and the front-end server are the same unit, changing the maximum number of processes that can be launched on the front-end server to 0 disables the following utilities:
   - Database load utility (pdload)
   - Database structure modification utilities (pdmod)
   - Database reorganization utility (pdrorg)
   - Database condition analysis utility (pddbst)
   - Optimizing Information collection utility (pdgetcst)
6. When the standby-less system switchover (effects distributed) facility is used, if the pdchprc command is used to display the number of resident server processes and the maximum number of active processes, the number of resident back-end server processes and the maximum number of active processes may not match depending on the number of allocated server processes. Additionally, even when you specify the -p option to change the number of resident processes and the maximum number of active processes, the number of resident back-end server processes and the maximum number of active processes may not be changed depending on the number of allocated server processes. For details about the number of allocated server processes when the standby-less system switchover (effects distributed) facility is used, see the *HiRDB Version 8 System Operation Guide*.
7. Return code 0 for the pdchprc command indicates normal termination, and return code 8 indicates abnormal termination (such as an invalid option or rsh error).

Output format

The following information is displayed only when the -p option is omitted:

aa...a *bbbb*,*cccc* *dddd*,*eeee*
aa...a *bbbb*,*cccc* *dddd*,*eeee*

Explanation

aa...a

Server name (1-8 characters)

bbbb

Number of resident server processes specified in the HiRDB system definition

cccc

Maximum number of startup server processes specified in the HiRDB system definition

dddd

New number of resident server processes set by the `pdchprc` command. If it is the same as *bbbb*, `****` is displayed.

eeee

New maximum number of startup server processes set by the `pdchprc` command. If it is the same as *cccc*, `****` is displayed.

pdclose (Close RDAREAs)

Function

The `pdclose` command closes specified RDAREAs. RDAREAs that have been shut down by the `pdhold` command and RDAREAs in error shutdown status can be specified.

Executor

HiRDB administrator

Format

`pdclose -r {`*RDAREA-name*`[,`*RDAREA-name*`]...|ALL}`

Options

- `-r {`*RDAREA-name*`[,`*RDAREA-name*`]...|ALL}`

`-r` *RDAREA-name* ～ <identifier> ((1-30))

Specifies the name of an RDAREA to be closed. You can also use batch specification of RDAREA names. For details about batch specification of RDAREA names, see *1.5.2 Batch specification of RDAREA names in operation commands*.

`ALL`

Specifies that all RDAREAs except the master directory RDAREA are to be closed.

Rules

1. You cannot close an RDAREA that is in reference-possible backup hold (update WAIT mode) or updatable backup-hold status, nor can you close a master directory RDAREA.
2. You cannot specify a duplicated RDAREA name. If the name is specified, the system will eliminate all duplicated RDAREA names.
3. You can specify a maximum of 128 RDAREAs. If more than 128 RDAREA names are specified, the system ignores the excess names.
4. If an RDAREA name is enclosed in `\"`, the system treats it as being case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a space, enclose the entire name in `\"`.
5. If you specify a user RDAREA or user LOB RDAREA, make sure that the corresponding data dictionary RDAREA is in one of the following statuses:

- Open and shutdown release status
- Open and shutdown status placed by the `pdhold` command

6. When you specify both data dictionary and user RDAREAs or both data dictionary and user LOB RDAREAs, be sure to specify the data dictionary RDAREAs last.

Rules

1. The `pdclose` command can be executed only while HiRDB is active.
2. The `pdclose` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. The `pdclose` command locks the specified RDAREAs in the EX mode. If another transaction is accessing a specified RDAREA, the `pdclose` command will go onto wait status until the transaction terminates.
4. If the `pdclose` command is executed on a shared RDAREA, all back-end servers are locked. If there can be multiple concurrent accesses to the corresponding RDAREA, global deadlock may occur, resulting in a timeout. If global deadlock has occurred, re-execute the `pdclose` command.

Note

1. The result of the `pdclose` command can be checked by the `pddbls` command.
2. The following shows the `pdclose` command's return codes:

   `0`: Normal termination

   `4`: Warning termination (some RDAREA processing terminated with an error)

   `8`: Abnormal termination

   `12`: Abnormal termination (an event occurred that prevented output of an error message)

   If the error code is `12`, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the command. If no error message has been output to the event log, contact the customer engineer.

pdclttrc (Acquire SQL trace dynamically)

Function

The pdclttrc command acquires an SQL trace for a client executing SQL statements. The SQL trace information is output to the client.

You can determine the server name and process ID subject to acquisition of the SQL trace from the client's IP address and UAP identification information in the pdls -d prc command's execution results.

Executor

HiRDB administrator

Format

```
pdclttrc [-s server-name] [-p process-ID] [-e] [-l PDUAPREPLVL-value]
        [-m maximum-length-of-?-parameter-and-search-data]
        [-n output-operation-codes-count] [-o SQL-trace-file-size]
```

Options

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server to which the client subject to acquisition of the SQL trace is connected. Specify the name of the single server or a front-end server.

When this option is omitted, the command acquires an SQL trace for all processes connected to all HiRDB servers (single or front-end servers).

For a single server, specifying an invalid server name does not result in an error.

- -p *process-ID* ～ <unsigned integer> ((1-10))

Specifies the process ID of the server to which the client subject to acquisition of the SQL trace is connected.

For a HiRDB/Parallel Server, this option must be specified together with the -s option. For a HiRDB/Single Server, the -s option is optional.

When this option is omitted, the command acquires an SQL trace for all processes connected to the single server or front-end server.

- -e

Stops acquisition of SQL traces.

Rules

1. When the connection has been broken or SQL trace acquisition has stopped because the -n option value was reached, specifying this option does not result in an error.
2. If only this option is specified, the command stops acquisition of SQL traces for all processes that are connected to all HiRDB servers (single or front-end servers).
3. When this option is specified together with the -s option, the command stops acquisition of the SQL traces for all processes connected to the specified server.
4. When this option is specified together with the -p option, the command stops acquisition of the SQL trace for the specified process.
5. A stop request to the client is executed by the next SQL statement. If an internal operation is executed in the meantime, SQL trace information may be output even after its acquisition is stopped by the -e option. For details about internal operations, see the operation codes in *14.4.3 Record formats of DAT-format files*.

- -l *PDUAPREPLVL-value*

To output UAP statistical report information in addition to SQL trace information, specifies the UAP statistical report information that is to be output (value of the PDUAPREPLVL client environment definition). For details about the specification values and information output destination, see the *HiRDB Version 8 UAP Development Guide*.

When the inter-process memory communication facility is used, this option is ignored, if specified. The command displays the messages during process-to-process memory communication.

When this option is specified together with the -e option, this option is ignored.

- -m *maximum-length-of-?-parameter-and-search-data* ～ <unsigned integer> ((4-32008))

Specifies in bytes the maximum length of the search data and ? parameter information that is to be output to the SQL trace.

This option provides the same function as when INOUT is specified in the PDPRMTRC client environment definition.

When this option is specified together with the -e option, this option is ignored.

- -n *output-operation-codes-count* ～ <unsigned integer> ((0-10000)) <<0>>

Specifies the number of operation codes to be output to the SQL trace file. SQL trace acquisition stops when information has been output for the specified number of

operation codes.

When this option is omitted, the command outputs SQL trace information until a pdclttrc command with the -e option specified is executed or until DISCONNECT is executed.

When this option is specified together with the -e option, this option is ignored.

- -o *SQL-trace-file-size* ～ <unsigned integer> ((0,4096-2000000000)) <<32768>>

Specifies in bytes the size of the SQL trace file.

When 0 is specified, the maximum file size is assumed.

When this option is specified together with the -e option, this option is ignored.

Rules

1. The pdclttrc command can be executed only when HiRDB is running.
2. The pdclttrc command must be executed at the server machine that contains the single server or where the system manager is located.
3. The pdclttrc command acquires a SQL trace only if the connected client version is 07-01 or later.
4. When the pdclttrc command is executed, it starts acquisition of the SQL trace for the SQL statement that is currently being processed; however, the command does not display the following information for this SQL statement:
   - SQL start time
   - SQL runtime that is displayed when PDSQLEXECTIME=YES is specified in the client environment definition
   - SQL runtime and differential information, which are output to the UAP statistical report for each SQL statement
5. The SQL trace file and UAP statistical report are output to the directory specified in the PDCLTPATH client environment definition. When PDCLTPATH is omitted, the SQL trace file is output to the directory that is the current directory for UAP execution.

   If both the -l option and the PDREPPATH client environment definition are specified, the SQL trace file is output to the directory specified in PDREPPATH.

   The following table shows the output destination of an SQL trace file when the -l option is specified:

PDCLTPATH	PDREPPATH	Output destination of SQL trace file
Omitted	Omitted	Current directory

PDCLTPATH	PDREPPATH	Output destination of SQL trace file
Specified	Omitted	Directory specified in PDCLTPATH
Omitted	Specified	Directory specified in PDREPPATH
Specified	Specified	

6. When the -l option is specified, the command does not output information from prior to the command's execution because the information for each UAP statement begins with the beginning of the command.
7. The names of the SQL trace files that are output are pd*xxxxxxxxyyyyyyyyyy*-1.trc and pd*xxxxxxxxyyyyyyyyyy*-2.trc.

   xxxxxxxx: Server name (1 to 8 characters)

   yyyyyyyyyy: Server process ID (1 to 10 decimal digits)
8. If an SQL trace has already been acquired at the client, the pdclttrc command is ignored, if executed.
9. If the pdclttrc command is executed while another pdclttrc command is executing, a message is displayed indicating that an SQL trace is being acquired for the current process.

Notes

1. If connection is broken for a reason such as a communication error or DISCONNECT, or if SQL trace information has been output for the number of operation codes specified in the -n option, SQL trace acquisition terminates automatically.
2. The following are the pdclttrc command's return codes:

   0: Normal termination

   8: Error termination

pdcmt (Commit transactions)

Function

The pdcmt command forcibly commits transactions when none of the transactions that are part of a global transaction could be concluded (for example, because of a communications error). All the transactions in the global transaction must be committed, otherwise they will not be synchronized.

The pdcmt command is used to forcibly commit transactions that are shown in the status information displayed by the pdls -d trn command as being in READY status (Status 1) and p status (Status 2).

Executor

HiRDB administrator

Format

```
pdcmt {-x host-name | -u unit-identifier} [-s server-name]
      {-A | -t transaction-identifier}
```

Options

- {-x *host-name* | -u *unit-identifier*}

Specifies the identifier of the unit or the name of the host in which the transactions to be committed are located.

-x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host in which the transactions to be committed are located.

-u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit in which the transactions to be committed are located.

- -s *server-name* ～ <identifier> ((1-8))

Specifies the server name when the unit has multiple servers. If this option is omitted, all servers in the object host are subject to the pdcmt command.

- -A

Specifies that all transactions in the server specified by the -s option that are in READY status and recovery waiting status are to be committed.

- -t *transaction-identifier* ～ <alphanumeric characters> ((16 characters))

Specifies that the transaction identifier of a transaction in READY status and recovery

processing waiting status is to be committed.

Rules

1. The `pdcmt` command can be executed only while HiRDB is active.
2. Execute the `pdcmt` command at each server machine.

Notes

1. The results of the `pdcmt` command can be checked on the basis of the return code from execution of the `pdls` command (with `-d trn` specified) or by checking for any error messages. Return code `0` for the `pdcmt` command indicates normal termination, and return code `1` indicates abnormal termination (such as an invalid option or `rsh` error).
2. Forced commit may not be possible for a transaction that is connected to a recovery-unnecessary front-end server and executed, even if transaction status 1 is `READY` and transaction status 2 is `p`. In such a case, you must complete the transaction automatically; for details about how to complete a transaction that is in uncompleted status for transaction determination, see the manual *HiRDB Version 8 System Operation Guide*.

pdconfchk (Check system definitions)

Function

Checks the contents of system definitions that are required in order to start HiRDB.

If there is an error in the system definitions, the command displays the erroneous definition along with the details.

Executor

HiRDB administrator

Format

```
pdconfchk [-d subdirectory-name] [-n]
```

Option

- `-d` *subdirectory-name* ～ <path name>

Specifies the subdirectory that contains the HiRDB system definition file to be checked, expressed as a path name relative to `%PDDIR%\conf` or `%PDCONFPATH%`. `%PDDIR%\conf` is the directory that contains unit control information definition files. `%PDCONFPATH%` is the directory that contains other HiRDB system definition files.

The following table shows the locations of the HiRDB system definition files that can be checked:

Type of HiRDB system definition file	-d option specified	-d option omitted
Unit control information definition file	`%PDDIR%\conf\`	`%PDDIR%\conf\`*subdirectory*`\`
Other file	`%PDCONFPATH%\`	`%PDCONFPATH%\`*subdirectory*`\`

Criteria

Before changing the HiRDB system definitions for the current environment, you should save the new HiRDB system definitions in a subdirectory and check them. This enables you to check the modified HiRDB system definitions before proceeding.

Rules

1. Create in advance the subdirectory that will be specified.
2. Store in the subdirectory not only the modified HiRDB system definition file but all HiRDB system definition files.
3. For a HiRDB/Single Server, the same path name is set in `%PDDIR%\conf` and `%PDCONFPATH%`.

4. For a HiRDB/Parallel Server, the path name pointed to by `%PDDIR%\conf` and `%PDCONFPATH%` may be different (for example, when the HiRDB system definition file is shared). In this case, create a subdirectory with the same name on both paths.

   If the `%PDDIR%\conf` and `%PDCONFPATH%` settings are different for each unit, create the subdirectory under `%PDDIR%\conf` and `%PDCONFPATH%` and then store the HiRDB system definition files in the subdirectories.

5. The maximum lengths of `%PDDIR%\conf\`*subdirectory* and `%PDCONFPATH%\`*subdirectory* are 220 bytes. Make sure that the name does not exceed 220 bytes.

6. To check the system definitions that are to be changed by the `pdchgconf` command, store the new system definition files in `%PDDIR%\conf\chgconf` and then specify `-d chgconf` when you execute the `pdconfchk` command.

- `-n`

Specifies that checking of the system files and access permissions is to be skipped. Use this option when the system files have not been created by the `pdfmkfs` or `pdloginit` command.

Information subject to checking

- Syntax checking

  The command outputs the result of syntax checking to the standard output and standard error output.

- Checking for system files

The command checks for any system log file, synchronization point dump file, or status file.

- Checking for access privilege

The command checks to see if the HiRDB administrator can access HiRDB files.

- Checking for duplicated system files

The command checks the system log files, synchronization point dump files, and status files for any duplication.

- Checking for hosts

The command checks to see if the host names are specified in the `HOSTS` file.

- Checking between server machines (applicable to HiRDB/Parallel Server)

The command checks between server machines based on the system manager's server machine.

For details about the operands in the system definition that can be checked by the pdconfchk command, see *Operands Checked by the pdconfchk Command* in the manual *HiRDB Version 8 System Definition*.

Rules

1. You can execute the pdconfchk command whether or not HiRDB is active.
2. You can execute the pdconfchk command on the server machine containing the single server or the server machine where the system manager is located.

Notes

1. If the command does not detect any error in the system definitions, it displays KFPS05007-I return code=0 and terminates itself. If the command detects an error, it displays KFPS05007-I return code=8 and terminates itself.
2. During checking between server machines, the command ignores any comment section (any text following *) or any comment line.
3. If the command detects an error during checking between server machines, check the contents of the system definitions in the server machine where the system manager is located. There may be an error in the system definitions in this server machine.
4. During checking between server machines, the command ignores any tab or space that follows a definition statement. Tabs and spaces preceding a definition statement are subject to checking.
5. During checking between server machines, if the order of definitions does not match between system manager definitions and other definitions, the command treats some of the definitions with a wrong sequence as errors.

   Example:

System manager definitions	Other definitions
A	B
B	A

   The command treats system manager definition B and other definition B as different definitions.
6. Return code 0 for the pdconfchk command indicates normal termination, and return code 8 indicates abnormal termination (such as an invalid argument or a pdconfchk command execution error).
7. If the pdconfchk command displays the KFPY01012-E message, the command may terminate with KFPS05007-I return code=0. In this case, eliminate the cause of the error.

8. When execution of the pdconfchk command results in the KFPS05062-W message, the pdconfchk command terminates with KFPS05007-I (return code=0).
9. Even when the pdconfchk command's execution was successful, the KFPO00107-E message may be displayed. To determine the result of definition checking by the pdconfchk command, check the KFPS05007-I message.

Output format

```
<Checking for system files or access privilege>
 [aa...a]
  bbbb     cc...c     dd...d     ee...e
<Checking for duplicate system files>
 [aa...a]
  bbbb     cc...c     dd...d     ee...e
  bbbb     cc...c     dd...d     ee...e
<Checking for hosts>
 [aa...a]
  bbbb     cc...c     dd...d     ff...f
<Checking between server machines (applicable to HiRDB/Parallel
Server)>
 [aa...a]
 <gggg:hhhh>
  gggg;  cc...c;  line = iii <- hhhh;  cc...c;  nothing
  gggg;  cc...c;  nothing  -> hhhh;  cc...c;  line = iii
```

Explanation

aa...a

Type of error detected in the checked item:

`File not found`

File was not found. Possible reasons:

- The specified path name contains invalid characters.
- The HiRDB file name is too long (more than 30 characters).

`File invalid`

File is invalid.

`No Permission`

There is no permission.

`Duplicate filename`

File name is duplicated.

`Hosts name not matched`

Host name does not match.

`Different definition`

Definitions do not match between the system manager unit and non-system manager unit.

bbbb

Unit identifier

cc...c

Name of the system definition file

dd...d

Operand specification

ee...e

File name specified in the operand shown as *dd...d* (absolute pathname)

ff...f

Defined host name if *dd...d* is `pdstart`, `pdunit`, or `pd_hostname`

gggg

Identifier of the unit that contains the system manager

hhhh

Identifier of the unit that contains other than the system manager

iii

Line number in the system definition file

■ **Output example**

```
[File not found]
 unt1   pdutsys   pd_syssts_file_name_1
        C:\hirdb\rdsys011\ut1sts1a                ...1
[File invalid]
 unt1   pdutsys   pd_syssts_file_name_3
        C:\hirdb\yyy\pddir                        ...2
[No permission]
 unt1   bes1   pd_sts_file_name_1
        C:\hirdb\rdsys011\bes1sts1a               ...3
[Duplicate filename]
 unt1   fes1   pdlogadpf -d sys -g fes1log1 -a
        C:\hirdb\rrdsys011\fes1log1               ...4
 unt1   fes1   pdlogadpf -d sys -g fes1log1 -b
```

```
        C:\hirdb\rrdsys011\fes1log1              ...4
 unt1   bes1   pdlogadpf -d spd -g bes1spd1 -a
        C:\hirdb\rrdsys011\fes1log1              ...4
[Hosts name not matched]
 unt1   pdsys     pdstart -x
        dcm3500                                  ...5
[Different definition]
<unt1:unt2>
 unt1   pdsys    line = 2   -> unt2   pdsys
      nothing                                    ...6
 unt1   pdsys    nothing    <- unt2   pdsys
      line = 5                                   ...7
```

Explanation

1. The file that was specified in the `pd_syssts_file_name_1` operand for `pdutsys` on `unt1` was not found.
2. The file specified in the `pd_syssts_file_name_3` operand for `pdutsys` on `unt1` is invalid.
3. There is no access privilege for the file specified in the `pd_sts_file_name_1` operand for `bes1` on `unt1`.
4. The file name specified in the `pdlogadpf -d sys -g fes1log1 -a` operand for `fes1` on `unt1` is the same as the following file names:
   - File name specified in the `pdlogadpf -d sys -g fes1log1 -b` operand for `fes1` on `unt1`
   - File name specified in the `pdlogadpf -d spd -g bes1spd1 -a` operand for `bes1` on `unt1`
5. Host name for `unt1` does not match the host name specified in the `pdstart -x dcm3500` operand for `pdsys` on `unt1`.
6. Definition in line 2 for `pdsys` on `unt1` is not found in `pdsys` on `unt2`.
7. Definition in line 5 for `pdsys` on `unt1` is not found in `pdsys` on `unt1`.

Examples

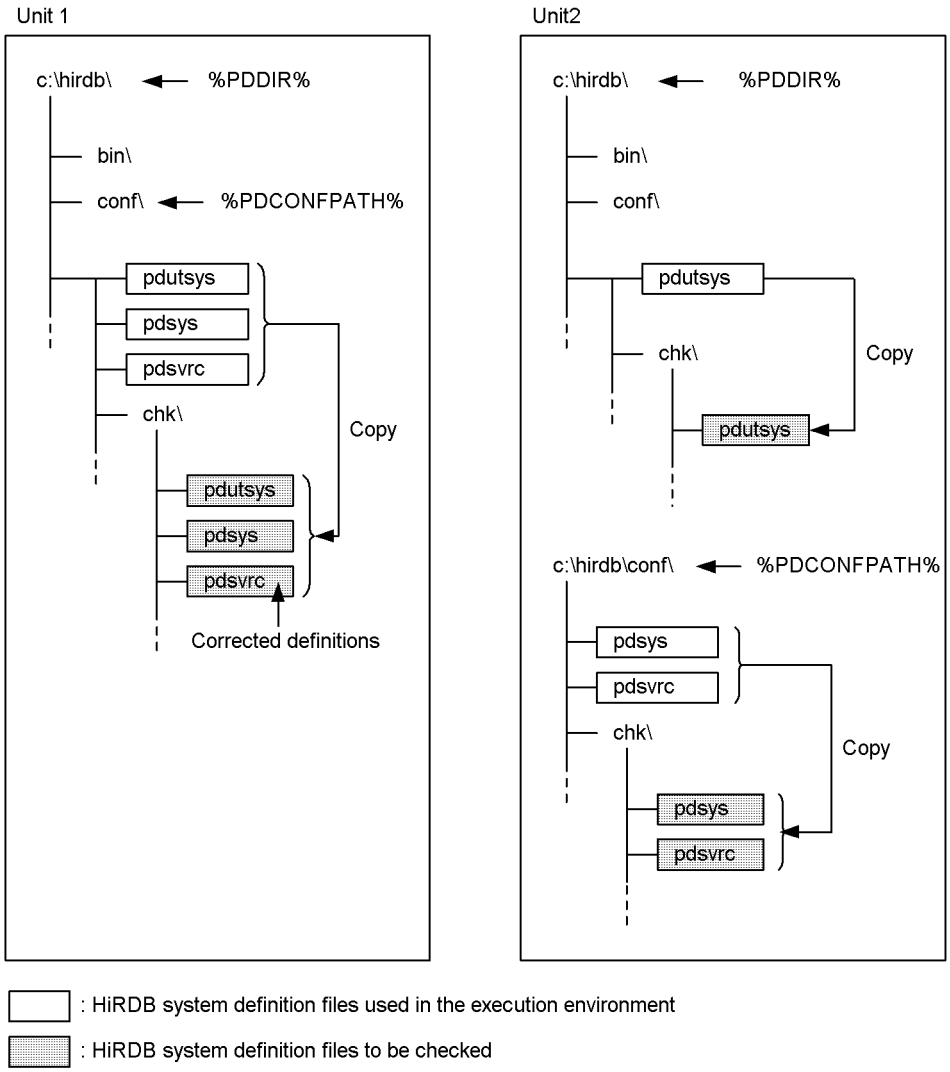
This example changes system definitions of the current HiRDB/Parallel Server (consisting of units 1 and 2). The HiRDB system definition file to be changed is `pdsvrc` on unit 1.

1. Creates the subdirectory `chk` under `%PDDIR%\conf` and `%PDCONFPATH%` on units 1 and 2.
2. Copies `pdutsys` from `%PDDIR%\conf` to `%PDDIR%\conf\chk` on each unit.
3. Copies the HiRDB system definition files from `%PDCONFPATH%` to

%PDCONFPATH%\chk\ on each unit.

4. Changes the definitions in %PDCONFPATH%\chk\pdsvrc on unit 1.
5. Executes pdconfchk -d chk.*
6. Assuming there are no problems with the definitions, this example uses the pdstop command to execute normal termination or planned termination of HiRDB. Note that in the case of planned termination, some definitions cannot be changed.
7. Uses the pdlogunld command to unload the system log file in unload wait status.
8. Copies the HiRDB system definition file changed in step 4 to %PDCONFPATH%.
9. Uses the pdstart command to start HiRDB normally.

* During execution of pdconfchk -d chk, the HiRDB system definition files are checked at the following locations:


Unit 1
c:\hirdb\
%PDDIR%
bin\
conf\
%PDCONFPATH%
pdutsys
pdsys
pdsvrc
chk\
Copy
pdutsys
pdsys
pdsvrc
Corrected definitions
Unit2
c:\hirdb\
%PDDIR%
bin\
conf\
pdutsys
chk\
Copy
pdutsys
c:\hirdb\conf\
%PDCONFPATH%
pdsys
pdsvrc
chk\
Copy
pdsys
pdsvrc
: HiRDB system definition files used in the execution environment
: HiRDB system definition files to be checked

pdcspool (Delete troubleshooting information)

Function

The pdcspool command deletes the troubleshooting information created by HiRDB in the spool (%PDDIR%\spool directory) and the temporary work files created in the temporary work file storage directory (%PDDIR%\tmp directory).

Executor

HiRDB administrator

Format

```
pdcspool [-i] [-d days] [-k type]
```

Options

- -i

Specifies that when a file storing troubleshooting information is to be deleted, a prompt is to be returned to the standard output device requesting confirmation that the file is to be deleted (in the case of temporary work files, the option does not return a prompt). Responding with a y to the prompt causes the file to be deleted. The default is to delete the file without generating a prompt, in which case the name of the deleted file is output to the standard output device.

- -d *days* ～ <unsigned integer> ((0-24855)) <<1>>

This option makes files that were created 24 hours × specified number of days from the time the pdcspool command was executed subject to deletion. If 10 is specified, all files created 240 or more hours before are subject to deletion.

If 0 is specified, all troubleshooting information in the spool and all temporary work files in the temporary work file storage directory are subject to deletion.

If all parameters are omitted, the troubleshooting information in the spool and the temporary work files in the temporary work file storage directory that were created 24 or more hours before are subject to deletion.

If you execute periodically commands that take a long time to execute, such as pdcopy and pdrstr, specify a value that provides sufficient time for the commands to be completed.

- -k *type*

This option specifies the range of troubleshooting information to be deleted.

If this option is omitted, the command deletes all troubleshooting information except for the access path information. Note that you cannot specify this option for temporary

work files.

`all`

Deletes all troubleshooting information including the access path information.

`dump`

Deletes the files internally collected by HiRDB (files other than the deadlock timeout information and access path information files).

Rules

1. The `pdcspool` command can be executed at any time, whether or not HiRDB is active.
2. The `pdcspool` command must be executed at each server machine.

Notes

1. Return code `0` or `4` for the `pdcspool` command indicates normal termination, and return code `12` indicates abnormal termination.
2. The `pdcspool` command may not delete a troubleshooting information file that was output by a command or utility executed by a user other than the HiRDB administrator. In such a case, the user with the deletion privilege for the troubleshooting information files must delete the file. However, do not delete the following directories and files in the temporary work file storage directory (`%PDDIR%\tmp` directory):
   - `home`
   - `pdommenv`
   - Files whose names begin with `CMb`
3. If the `TMP` environment variable has been specified, the temporary work files will be output to the directory specified in this environment variable, not to the `%PDDIR%\tmp` directory. The temporary work files in the directory specified in the `TMP` environment variable are not subject to deletion by the `pdcspool` command. Therefore, you need to delete these files directly using Explorer.
4. The `pdcspool` command makes all files that were created by HiRDB in the spool (`%PDDIR%\spool` directory) and the temporary work file storage directory (`%PDDIR%\tmp` directory) subject to deletion. Therefore, any files created in those directories will be deleted together. The following table describes the directories that are subject to deletion:
   - Spool (`%PDDIR%\spool` directory)

Directory	Description of output file	-k option			Remarks
		all	dump	Omitted	
pdsqldump	Access path information	Y	N	N	This is output when 1 or a greater value is specified in the PDVWOPTMODE client environment definition and then an SQL statement is executed.
pdlckinf	Deadlock/ timeout information	Y	N	Y	This is output when an error occurs during lock control.
save	Save core file, etc.	Y	Y	Y	This is output upon abnormal termination of a process.
pdshmdump	Shared memory dump file	Y	Y	Y	This is output upon abnormal termination of a process or a unit.
pdsysdump(1,2)	Quick dump file	Y	Y	Y	None.
pdsdsdump(1,2)		Y	Y	Y	This is output in a HiRDB/Single Server.
pdfesdump(1,2) pddicdump(1,2) pdbesdump(1,2)		Y	Y	Y	This is output in a HiRDB/ Parallel Server.

Legend:

Y: Deleted.

N: Not deleted.

- Temporary work file storage directory (%PDDIR%\tmp directory)

  The command deletes everything except the following directories and files:

Directory name or file name	Description of output file
home	Current working directory of a process where HiRDB is run
pdommenv	Shared memory information file
File name beginning with CMb	Differential information file for the pdbufls command

pddbfrz (Place a full HiRDB file in a user LOB RDAREA in frozen update status)

Function

The pddbfrz command places a full HiRDB file (used through the last page of the file) that constitutes a user LOB RDAREA in frozen update status. You cannot update or delete LOB data in a HiRDB file that is in frozen update status.

Use this command to back up data without using pdcopy.

Executor

HiRDB administrator

Format

```
pddbfrz [-d] -r {RDAREA-name[,RDAREA-name]...|ALL}
```

Options

- -d

Specifies that HiRDB files are to be released from frozen update status.

When this option is omitted, the command places full HiRDB files in frozen update status. Note that you cannot place the first HiRDB file in an RDAREA in frozen update status.

- -r {*RDAREA-name*[,*RDAREA-name*]...|ALL}

RDAREA-name ～ <identifier> ((1-30))

Specifies the names of the RDAREAs that are to be placed in the frozen update status or released from frozen update status. You can specify only the names of user LOB RDAREAs.

You can also use batch specification of RDAREA names. For details about batch specification of RDAREA names, see the *1.5.2 Batch specification of RDAREA names in operation commands*.

ALL

Specifies that all user LOB RDAREAs are to be placed in the frozen update status. When the -d option is specified, the command releases all user LOB RDAREAs from frozen update status.

Rules

1. An error results if a specified RDAREA is not a user LOB RDAREA, a specified user LOB RDAREA contains a plug-in index, or a specified user LOB RDAREA is unused.
2. If the same RDAREA name is specified more than once, the duplicate RDAREA name is ignored.
3. You can specify a maximum of 128 RDAREA names. If more than 128 RDAREA names are specified, the excess names are ignored.
4. If an RDAREA name is enclosed in `\"`, the system treats it as being case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a space, enclose the entire name in `\"`.

Rules

1. The `pddbfrz` command can be executed only while HiRDB is active.
2. The `pddbfrz` command must be executed at the server machine that contains the single server or where the system manager is located.
3. When you execute the pddbfrz command, the user LOB RDAREA must be in one of the following statuses:
   - Open and shutdown release status
   - Open, and shutdown status placed by the `pdhold` command
4. The `pddbfrz` command locks a specified RDAREA. If you execute the `pddbfrz` command to process an RDAREA that is being accessed by another transaction, the command is placed in wait status until that transaction terminates.
5. Executing the `pddbfrz` command does not place the first HiRDB file in an RDAREA in frozen update status because this file contains HiRDB management records.
6. You can place only HiRDB files that are filled with data in frozen update status. If a HiRDB file is not full during command execution, it will not be placed in frozen update status later when it becomes full of data.

Notes

1. The result of the `pddbfrz` command can be checked by the `pddbst` command.
2. If you want to update data in a HiRDB that is in frozen update status or if you need to access such an RDAREA for reasons such as reorganization with `pdrorg`, use the `pddbfrz -d` command to release the HiRDB file from frozen update status. After updating the RDAREA, back up all HiRDB files again.
3. The following shows the `pddbfrz` command's return codes:

   `0`: Normal termination

4: Warning termination (some RDAREA processing terminated with an error)

8: Abnormal termination

12: Abnormal termination (an event occurred that prevented output of an error message)

If the error code is 12, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the command. If no error message has been output to the event log, contact the customer engineer.

pddbls (Display status of RDAREAs)

Function

The pddbls command displays the status of RDAREAs.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pddbls {-r RDAREA-name [, RDAREA-name]... | -r ALL}
          [-{l | a}] [-b][-x[-y]]
```

HiRDB/Parallel Server

```
pddbls {-r RDAREA-name [, RDAREA-name]... | -r ALL
          | -s server-name [, server-name]...}
          [-{l | a}] [-b][-m][-x[-y]]
```

Options

- -r *RDAREA-name* ～ <identifier> ((1-30))

Specifies the name of an RDAREA whose status is to be displayed. You can also use batch specification of RDAREA names. For details about batch specification of RDAREA names, see *1.5.2 Batch specification of RDAREA names in operation commands*.

The same RDAREA name should not be specified more than once; if specified, a duplicate RDAREA name is ignored. A maximum of 128 RDAREA names can be specified. If more than 128 RDAREA names are specified, the excess names are ignored.

If an RDAREA name is enclosed in \", it is treated as being case sensitive; otherwise, it is treated as all uppercase letters. If an RDAREA name contains a blank, the entire name must be enclosed in \".

- -r ALL

Specifies that information about all RDAREAs is to be displayed.

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of a server when the status of all of that server's RDAREAs is to be displayed. The command displays information about all RDAREAs for the specified server.

A maximum of 128 server names can be specified. If more than 128 server names are

specified, the excess names are ignored.

- -{l |a}

-l

Specifies that only the RDAREA status is to be displayed.

-a

Specifies that all information about RDAREAs other than shared RDAREAs is to be displayed.

When the -m option is specified, the command displays shared RDAREA information for all servers. Note that the segment usage information in a shared RDAREA is displayed only for an updatable back-end server.

- -b

Specifies that information is to be displayed only for RDAREAs in shutdown status. When a server name is specified, the command displays information about the RDAREAs in shutdown status for the specified server only. When this option is omitted, the command displays information about all specified RDAREAs.

- -x

Specifies that the information is to be displayed in DAT format.

- -y

When the -x option is specified, specifies that a header is to be displayed.

- -m

Specifies that shared RDAREA information is to be displayed.

When a shared RDAREA is specified in the -r option and the -m option is also specified, the command displays the applicable RDAREA and shared RDAREA information for all back-end servers.

Rules

1. The pddbls command can be executed only while HiRDB is active.
2. The pddbls command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. When a user RDAREA or a user LOB RDAREA is specified, the corresponding data dictionary RDAREA must be in one of the following statuses:
   - Open and shutdown release status
   - Open and shutdown status placed by the pdhold command

Notes

1. The following shows the `pddbls` command's return codes:

   `0`: Normal termination

   `4`: Warning termination

   `8`: Abnormal termination

   `12`: Abnormal termination (an event occurred that prevented output of an error message)

   If the error code is `12`, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the command. If no error message has been output to the event log, contact the customer engineer.

Output format

```
STATE OF RDAREA
RDAREA ID      STATUS   [SERVER] TYPE
               OPNMODE  [SHARED]
aa...a  bb...b  cc...c   dd...d   eeee
                jj...j   kkk
 [SEGMENT   ff..f/gg...g]
 [UNABLE TO DISPLAY CODE=hh]
    :     :    :
 [NOT FOUND ii...i=aa...a]
 [RDAREA NOT FOUND]
```

Explanation

aa...a

RDAREA name (up to 30 characters).

bb...b

RDAREA number (11 decimal digits).

cc...c

Status of RDAREA (up to 17 characters). The following statuses are displayed for RDAREAs:

RDAREA status	Explanation	Segment information
CLOSE	Closed	DS
CLOSE HOLD	Closed, error shutdown	—

RDAREA status	Explanation	Segment information
CLOSE HOLD (INQ)	Closed, reference-possible command shutdown	—
CLOSE HOLD (CMD)	Closed, command shutdown	—
CLOSE HOLD (BU)	Closed, updatable backup-hold	D
CLOSE HOLD (BU I)	Closed, reference-possible backup-hold	D
CLOSE HOLD (BU W)	Closed, updatable backup-hold (WAIT mode)	D
CLOSE HOLD (BU IW)	Closed, reference-possible backup-hold (update WAIT mode)	D
CLOSE ACCEPT-HOLD	Closed, pdhold command-accepted	—
HOLD	Error shutdown	—
HOLD(INQ)	Reference-possible command shutdown	D[1]
HOLD(CMD)	Command shutdown	D[1]
HOLD (BU)	Updatable backup-hold	D
HOLD (BU I)	Reference-possible backup-hold	D
HOLD (BU W)	Updatable backup-hold (WAIT mode)	D
HOLD (BU IW)	Reference-possible backup-hold (update WAIT mode)	D
ACCEPT-HOLD	pdhold command-accepted	D[1]
OPEN	Open	D

D: Displayed

DS: Displayed when RDAREA open timing is set to DEFER or SCHEDULE.

—: Not displayed.

[1] Segment information is not displayed during close processing.

dd...d

Server name for HiRDB/Parallel Server (up to 8 characters).

eeee

Type of RDAREA:

MAST

Master directory RDAREA

`DDIR`

Data directory RDAREA

`DDIC`

Data dictionary RDAREA

`DLOB`

Data dictionary LOB RDAREA

`USER`

User RDAREA

`ULOB`

User LOB RDAREA

`LIST`

List RDAREA

`RGST`

Registry RDAREA

`RLOB`

Registry LOB RDAREA

ff...f

Number of unused segments in RDAREA (11 decimal digits).

For a LOB RDAREA, the next segment is included in the number of unused segments, but it cannot be reused until the synchronization point is reached. Once the synchronization point is reached, it can be reused.

- Segments that stored data of the `BLOB` type that has been deleted by a `DELETE` statement.
- For `BLOB`-type data updated by the `UPDATE` statement, the segments that stored the data before the updating was performed.
- Segments that stored `BLOB`-type dictionary data deleted by a change procedure, such as `DROP PROCEDURE` or `DROP SCHEMA`.

gg...g

Total number of segments in RDAREA (11 decimal digits).

hh

Return code. The following return codes are displayed:

Return code	Explanation	Required action
01	RDAREA is in a status such that no information can be displayed (see *cc...c* for the RDAREA status).	Not applicable.
02	Information on an RDAREA cannot be displayed because the RDAREA is being accessed in the lock mode by another transaction.	Re-execute the command after the transaction that is accessing the RDAREA in the lock mode has terminated.
03	No global buffer was allocated to the RDAREA.	Allocate a global buffer, then re-execute the command.
04	HiRDB file access error occurred.	Take action appropriate to the corresponding error message.
05	Lock error occurred.	
101	Invalid RDAREA type was specified.	Check for a specification error; if there was no error, contact maintenance personnel.

ii...i

For an RDAREA, RDAREA is displayed; for a server, SERVER is displayed (six characters).

jj...j

User-specified open timing for the HiRDB file system area. This is the HiRDB file system area's open timing specification (INITIAL, DEFER, or SCHEDULE) that was specified in the system common definition or with the database initialization utility or database structure modification utility.

kkk

Shared RDAREA information (applicable when the -m option is specified):

SUP: Shared RDAREA (updatable back-end server)

SRD: Shared RDAREA (reference-only back-end server)

NON: Non-shared RDAREA

When the -m option is omitted or for a HiRDB/Single Server, neither a header nor shared RDAREA information is displayed.

Notes

- If a specified RDAREA is not found, the message NOT FOUND RDAREA= is displayed.
- If a specified server is not found, the message NOT FOUND SERVER= is displayed.

- When the -b option is specified but there is no RDAREA in shutdown status, or when there are no RDAREAs for a server specified in the -s option, the message RDAREA NOT FOUND is displayed.
- When the -a option is specified but the information cannot be displayed, the message UNABLE TO DISPLAY CODE=*hh* is displayed.

■ DAT output format (when -x and -y options are specified)

```
"RDAREA","ID","STATUS","UNUSED_SEGMENT","TOTAL_SEGMENT","SERVER","TYPE","OPNMODE","O
RIGINAL-RDAREA",
"ORIGINAL-ID","GENERATION-NUMBER","REPLICA-STATUS","REPLICA-COUNTER","SHARED"[CR]
"aa...a",bb...b,"cc...c",ff...f,gg...g,"dd...d","eeee","jj...j","",
,,"",,"kkk"[CR]
```

Note

[CR] indicates a linefeed.

Rules for output in DAT format

1. The elements are separated by the comma (,), and information about one RDAREA is displayed on each line.
2. A character element is enclosed in double quotation marks (").
3. If information cannot be acquired for a reason regarding, for example, the options that are specified or the server type, the command displays only a blank or "".
4. If the -a option was specified but segment information cannot be acquired due to its status, the command displays -1 as the number of unused segments in the RDAREA and the return code for the total number of segments in the RDAREA.
5. When information is displayed in DAT format, the following error messages are not output:

   [NOT FOUND *ii...i=aa...a*]

   [RDAREA NOT FOUND]

pddefrev (Generate definition SQL)

Function

The pddefrev command generates a definition SQL for tables, indexes, view tables, stored procedures, and triggers defined in the HiRDB system.

The generated definition SQL can be used as an input to the database definition utility. This command generates the definition SQL by starting the dictionary import/export utility (pdexp) internally.

Executor

User with DBA privilege

Format

```
pddefrev -f control-statements-filename
         -o definition-SQL-output-filename
        [-w work-directory-name]
```

Options

- -f *control-statements-filename* ～ <pathname> ((up to 1023 bytes))

Specifies the name of the control statements file that contains the tables and stored procedures for which the definition SQL is to be generated. For details about the contents of the control statements file, see Chapter *9. Dictionary Import/Export Utility (pdexp)*.

- -o *definition-SQL-output-filename* ～ <pathname> ((up to 1023 bytes))

Specifies the absolute pathname of the file to which the generated definition SQL is to be output. If the specified file already exists, HiRDB writes the generated definition SQL by overwriting the existing information; if the specified file does not already exist, HiRDB creates it.

- -w *work-directory-name* ～ <pathname>

Specifies the absolute pathname of the directory to be used for temporary storage of the export file that will be created internally by the pddefrev command in order to export the dictionary. If this option is omitted, \tmp is assumed.

The following is the name of the export file that is created:

```
pddefrev.exp.HHMMSS
```

The local time is set in *HHMMSS*.

Rules

1. The `pddefrev` command can be executed only while HiRDB is active.
2. The `pddefrev` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. To execute the `pddefrev` command, you must grant access permissions for the definition SQL output file and work directory to the HiRDB administrator beforehand.

Notes

1. The results of the `pddefrev` command can be checked on the basis of the return value and by whether or not there are any error messages. Return code `0` for the `pddefrev` command indicates normal termination, and return code `8` or `12` indicates abnormal termination.
2. Only one `pddefrev` command can be executed at a time. The command cannot be executed while the dictionary import/export utility is executing.
3. The following table lists the definition SQL statements that are generated by this command:

Type	Index definition	Comment definition	Definition SQL to be generated

Base table	Defined	Defined	`CREATE TABLE`
			`CREATE INDEX`
			`COMMENT`

		Not defined	`CREATE TABLE` `CREATE INDEX`
	Not defined	Defined	`CREATE TABLE` `COMMENT`
		Not defined	`CREATE TABLE`
View table	—	Defined	`CREATE VIEW` `COMMENT`
		Not defined	`CREATE VIEW`
Stored procedure	—	—	`CREATE PROCEDURE`
Trigger	—	—	`CREATE TRIGGER`

—: Not applicable

4. The definition SQL statements are generated in the order specified in the control statements file.

5. A semicolon (`;`) is added at the end of the line of a generated definition SQL statement so that the definition SQL can be used as an input to the database definition utility. `end_proc;` is added in the case of `CREATE PROCEDURE` and `CREATE TRIGGER`.
6. Definition SQL statements involving any of the following tables, procedures, or functions are not generated:
   - Table definitions containing abstract data types (`CREATE TABLE`)
   - Index definitions specifying index types (`CREATE INDEX`)
   - Function definitions (`CREATE FUNCTION` and system-defined scalar functions)
   - Procedures specifying an abstract data type as a data type for a parameter in a procedure (`CREATE PROCEDURE`)
   - Procedures calling another procedure from within the procedure definition (`CREATE PROCEDURE`)
   - A procedure declared in an abstract data type (`CREATE TYPE`)
   - Procedures calling a function definition (`CREATE FUNCTION` or system-defined scalar function) from within the procedure definition
7. When `CREATE PROCEDURE` is created, the values of the SQL or extended SQL optimization options are generated in decimal format.
8. Linefeed is performed on the created definition SQL statement in accordance with a condition. For details about the linefeed condition, see *9.3(3) Rules for definition SQL generation*.
9. If you select `utf-8` as the character encoding in the `pdntenv` command, you can use a file with a BOM as the control statement file for `pddefrev`. Note that even when you do this, the BOM is skipped. No BOM is included in the file that is output by `pddefrev`.

pdfbkup (Back up HiRDB file system)

Function

The pdfbkup command makes a backup in a file of the HiRDB file system. Backups can also be made in units of HiRDB files.

During the backup processing, this command also backs up the attributes of the HiRDB file system (specified by the pdfmkfs command). Although the attributes are inherited, once a HiRDB file system is expanded, the pdfrstr command cannot restore the original size from a backup that was made before the expansion.

The number of added areas and the number of noncontiguous free areas can be determined by displaying the status of a HiRDB file system area (by entering the pdfstatfs command). The added areas and noncontiguous free areas can then be merged and the HiRDB file system area used more effectively by taking the following measures:

- Back up the HiRDB file system (pdfbkup command)
- Initialize the HiRDB file system area (pdfmkfs command)
- Restore the backed up HiRDB file system to the initialized HiRDB file system (pdfrstr command)

Executor

HiRDB administrator

Format

```
pdfbkup [-y] [-r] [-i] [-{c|l|f}]
        HiRDB-file-system-area-name[\HiRDB-filename]
        backup-filename
```

Options

- -y

Specifies that the HiRDB file system is to be backed up without obtaining user confirmation. When this option is omitted, the backup is made only after the user confirms the processing.

- -r

Specifies that if a file with the specified backup file name already exists, it is to be deleted and re-created. When this option is omitted, an error occurs if a file with the specified backup file name already exists.

- -i

Specifies that if an error occurs during backup of the HiRDB file system, the backup of the file where the error occurred is to be abandoned and processing is to continue with the next file. When this option is omitted, the command terminates with an error when an error occurs.

- `-{c|l|f}`

Specifies the processing to be performed if another process or user is accessing the specified HiRDB file system area.

`-c`

Cancel backup processing if another process or user is accessing the specified HiRDB file system area

`-l`

Do not make a backup of any HiRDB file currently being accessed by another process or user

`-f`

Execute backup processing even if another process or user is accessing the specified HiRDB file system area

Command arguments

- *HiRDB-file-system-area-name[\HiRDB-filename]*

Specifies the names of the HiRDB file system area and the HiRDB file to be backed up.

HiRDB-file-system-area-name ～ <pathname> ((up to 165 characters))

Specifies the name of the HiRDB file system area.

HiRDB-filename ～ <HiRDB filename>

Specifies the name of an HiRDB file.

- *backup-filename*

Specifies the name of the file in which the backup is to be made.

Rules

1. The `pdfbkup` command can be executed at any time, whether or not HiRDB is active. If you execute the command during operation, specify the `-f` option.
2. The `pdfbkup` command must be executed at each server machine where the HiRDB file system area is stored.
3. Only one HiRDB file system can be backed up into one file. Therefore, a commensurate number of files are needed in order to back up multiple HiRDB file

systems.

4. The backup file is not created if an error occurs during backup processing.
5. A confirmation message is displayed when the `-y` option is omitted. The HiRDB file system is not backed up if any character other than G is entered in response to the confirmation message.
6. Before backing up a HiRDB file, use the `pdfls` command to check its record length. You cannot restore a HiRDB file if the record length of the backup HiRDB file is not a multiple of the sector length of the target HiRDB file system area.
7. If you are executing the command on an HiRDB file system area that is used for shared RDAREAs, execute it from the server machine where the updatable back-end server is located.

Notes

1. The execution results of the `pdfbkup` command can be confirmed by the return code after the command has executed and by whether or not there are any error messages. Return code `0` for this command indicates normal termination, `1` indicates warning termination, and `-1` indicates abnormal termination.
2. If the `pdfbkup` command terminates abnormally, the backup file may have been created but its contents may not be valid.

pdffsck (Check and repair the integrity of a HiRDB file system area)

Function

The pdffsck command checks a HiRDB file system area for integrity errors and repairs any errors that are detected. If an inconsistency is detected, the command displays the corresponding file or area.

You should execute the pdffsck command when either of the following occurs:

- The KFPI21586-W message is displayed during execution of the pdfls command
- The available space in the HiRDB file system area becomes less than the original capacity (for example, 100 gigabytes were allocated, but only 80 gigabytes are now available)

The following table describes the integrity check items and repair methods:

Target	Check item	Cause	Repair method
HiRDB file	Whether or not there are unusable HiRDB files	Cancellation of file creation or deletion processing	Manages the files as being in unused status and the allocated area as free area.
Area in a HiRDB file	Whether or not there is an area that is referenced neither from the free area list nor from the file management table	Cancellation of file extension (automatic extension) processing	Manages the isolated area as a free area.
Management information in the HiRDB file system area	Inconsistent statistical information stored in the management area of the HiRDB file system area	Cancellation of file creation or deletion processing	Correct the information, such as file count statistical information, when each check is completed.

Executor

HiRDB administrator

Format

```
pdffsck [-c] HiRDB-file-system-area-name
```

Options

■ -c

Specifies that the HiRDB file system area is to be checked only. When the -c option is specified, the HiRDB file system area is not repaired.

Command argument

- *HiRDB-file-system-area-name* ～ <path name> ((up to 165 characters))

Specifies the name of the HiRDB file system area.

Rules

1. When you specify the -c option, you can execute the pdffsck command whether or not HiRDB is active. When you omit the -c option, you must execute the pdffsck command as explained below, depending on the purpose of the HiRDB file system area:
   - When -k *HiRDB-file-system-area-for-DB* is specified

     Execute the command after placing all RDAREAs in the HiRDB file system area in command shutdown and closed status with the pdhold -c command, and then terminating the active processes with the pdpfresh command. After executing the pdffsck command, release all RDAREAs from shutdown status with the pdrels command.
   - When -k *HiRDB-file-system-area-for-SYS* is specified

     Execute the command after swapping system log files with the pdlogswap command and then closing the system log file with the pdlogcls command. After executing the pdffsck command, use the pdlogopen command to make the system log file available for use in the active HiRDB.
   - When -k *HiRDB-file-system-area-for-UTL* is specified

     If the HiRDB file system area is being used by a utility, execute the command only after the utility's processing has been completed.
   - When -k *HiRDB-file-system-area-for-WORK* is specified

     Since the HiRDB file system area cannot be restored during operation of HiRDB, execute it after HiRDB has terminated.
2. You can execute the pdffsck command with the -c option specified on a HiRDB file system area that is being accessed by another process or user. The pdffsck command with the -c option omitted cannot be executed on a HiRDB file system area that is being accessed by another process or user.
3. The pdffsck command must be executed at each server machine where the HiRDB file system area is stored.
4. If you are executing the command on a HiRDB file system area for which -k SDB has been specified, execute it from the server machine where the updatable back-end server is located.

Notes

1. If an I/O error occurs, the command cancels the processing.
2. You can use the `pdfls` command to check the HiRDB files that have been created in the target HiRDB file system area.
3. If a HiRDB file system area is checked while the RDAREA automatic extension facility is being used, automatic extension occurs after the check processing has terminated. To determine whether or not the RDAREA automatic extension facility is in effect for the target HiRDB file system area, execute the `pdfstatfs` command.
4. While the `pdffsck` command is executing, do not execute `pdmod` or `pdrstr`. If they are executed, the database may become corrupted.
5. If file repair processing by the `pdffsck` command is cancelled while it is underway, re-execute the command to complete the repair processing.
6. In HiRDB version 07-02 or earlier, the `pdffsck` command may not be able to repair information in the management area that has become corrupted. In such a case, the `KFPI21585-E` message is displayed. As directed in this message, restore the target HiRDB file system area from a backup. For details, see the manual *HiRDB Version 8 System Operation Guide*.

Output format

When the `KFPI21559-W` message is displayed due to an inconsistency detected in the area, the applicable information item listed below is displayed, as appropriate for the detected inconsistency. If the `KFPI21558-I` message is displayed but no inconsistency was detected, the information below is not displayed.

- When there are unusable HiRDB files (cancellation of file creation or deletion processing)

  An unusable HiRDB file is distinguished by the status of the area allocated to it, in terms of whether or not the area itself is usable. If the area is not usable, the command displays the name, start location, and size of every file in the area. Because the area may not be unusable for all the unusable files, the displayed unusable file count may not match the number of files that are listed.

```
unusable file count                 aa...a
file list
     FILE NAME                  OFFSET[kB]     SIZE[kB]
     bb...b                        cc...c       dd...d
       :                              :            :
total unusable area size[kB]             ee...e
```

Explanation

aa...a

Number of unusable HiRDB files (1 to 4 decimal digits)

bb...b

Name of an HiRDB file whose area is not usable (up to 30 characters)

cc...c

Offset to the start of the HiRDB file (up to 10 decimal digits)

dd...d

Size of the HiRDB file (kilobytes) (up to 10 decimal digits)

ee...e

Total size (kilobytes) of all the HiRDB files whose area is not usable (the size that can be reused as free area by repairing) (up to 10 decimal digits)

- When there are unusable areas (cancellation of file extension processing)

  The start location and size of each unusable area is displayed. This information is displayed for all applicable areas. The displayed unusable area count always matches the number of areas that are listed.

```
unusable area count                aa...a
area list
     OFFSET[kB]                      SIZE[kB]
     bb...b                            cc...c
       :                                  :
total unusable area size[kB]            dd...d
```

Explanation

aa...a

Number of unusable areas (up to 4 decimal digits)

bb...b

Offset to the start of an unusable area (up to 10 decimal digits)

cc...c

Size of the unusable area (kilobytes) (up to 10 decimal digits)

dd...d

Total size (kilobytes) of all the areas that are unusable (the size that can be reused as free area by repairing) (up to 10 decimal digits)

- When there are both unusable HiRDB files and unusable areas

```
unusable file count                 aa...a
file list
     FILE NANE                   OFFSET[kB]    SIZE[kB]
     bb...b                         cc...c      dd...d
       :                               :           :
unusable area count                 ff...f
area list
     OFFSET[kB]                     SIZE[kB]
     gg...g                          hh...h
       :                               :
total unusable area size[kB]         ee...e
```

Explanation

aa...a

Number of unusable HiRDB files (1 to 4 decimal digits)

bb...b

Name of an HiRDB file whose area is not usable (up to 30 characters)

cc...c

Offset to the start of the HiRDB file (up to 10 decimal digits)

dd...d

Size of the HiRDB file (kilobytes) (up to 10 decimal digits)

ee...e

Total size (kilobytes) of all HiRDB files whose area is not usable and of the unusable areas (the size that can be reused as free area by repairing) (up to 10 decimal digits)

ff...f

Number of unusable areas (up to 4 decimal digits)

gg...g

Offset to the start of an unusable area (up to 10 decimal digits)

hh...h

Size of an unusable area (kilobytes) (up to 10 decimal digits)

pdfgt (Terminate transaction forcibly)

Function

The pdfgt command terminates a transaction forcibly when a transaction branch that is part of a global transaction could not be terminated (such as because of a communication error, etc.).

If a communication error occurs before an OLTP system and a global transaction are synchronized, or if the OLTP system is down, a transaction will not terminate until synchronization notification is completed. The pdfgt command can be used in such a case to terminate the transaction forcibly. The pdfgt command can also be used to terminate a transaction in which a non-transient error has occurred.

The pdfgt command terminates forcibly a transaction for which transaction status 1 is FORGETTING and transaction status 3 is w, as displayed by the pdls command (-d trn specified).

Executor

HiRDB administrator

Format

```
pdfgt {-x host-name| -u unit-identifier} [-s server-name]
  {-A | -t transaction-identifier}
```

Options

- {-x *host-name*| -u *unit-identifier*}

Specifies the name of the host or the unit identifier of the unit where the transaction to be terminated forcibly is located.

-x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host where the transaction to be terminated forcibly is located.

-u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the unit identifier of the unit where the transaction to be terminated forcibly is located.

- -s *server-name* ～ <identifier> ((1-8))

For a unit with multiple servers, specifies a server name in order to specify a particular server. When this option is omitted, all servers at the object host are subject to the pdfgt command.

- -A

Specifies that all transactions at the server specified by the -s option that are in synchronization wait status and FORGETTING status are to be terminated forcibly.

- -t *transaction-identifier* ～ <identifier> ((16 characters))

Specifies the transaction identifier of a transaction in synchronization wait status and FORGETTING status that is to be terminated forcibly.

Rules

1. The pdfgt command can be executed only while HiRDB is active.
2. The pdfgt command must be executed at each server machine.

Note

1. The results of the pdfgt command can be checked by the pdls command (with -d trn), by the return code of the pdfgt command, or by any error messages. Return code 0 for the pdfgt command indicates normal termination, and return code 1 indicates abnormal termination (such as an invalid option or rsh error).
2. Forced termination may not be possible for a transaction that is connected to a recovery-unnecessary front-end server and executed, even if transaction status 1 is FORGETTING and transaction status 3 is w. In such a case, complete the transaction automatically. For details about how to complete a transaction that is in uncompleted status for transaction determination, see the manual *HiRDB Version 8 System Operation Guide*.

pdfls (Display HiRDB file system area information)

Function

The pdfls command displays information about HiRDB files in a HiRDB file system area. If only the name of a HiRDB file system area is specified in the command argument, information about all HiRDB files in the specified HiRDB file system area is displayed. If a HiRDB file name is also specified, information about only the specified HiRDB file is displayed.

Executor

Any user (with or without privileges)

Format

```
pdfls [-[H][-L| -S][-{t|u}]] HiRDB-file-system-area-name
                                    [\HiRDB-filename]
      | [-x] HiRDB-file-system-area-name[\HiRDB-filename]}
```

Options

- -H

Specifies that the display is to be with headers in alphabetical order of the HiRDB file names.

- -L

Specifies that the HiRDB files are to be displayed in alphabetical order of the HiRDB file names with each file's lock status.

- -S

Specifies that the HiRDB file partition storage status (extent status) is to be displayed in alphabetical order of the HiRDB file names.

- -{t|u}

-t

Specifies that the contents of the HiRDB file system area are to be displayed in the order of the update close date and time, beginning with the most recent one. The update close date and time refer to the most recent date and time the file was written since it was opened. If no write operation has occurred on the HiRDB file, the system treats the file creation date and time as the update close date and time. For a HiRDB file constituting an RDAREA, the actual update close date and time may not be used.

-u

Specifies that the contents of the HiRDB file system area are to be displayed in the order of the close date and time, beginning with the most recent one. The close date and time refer to the most recent date and time the file was read since it was opened. If no read operation has occurred on the file, the system treats the file creation date and time as the close date and time. For a HiRDB file constituting an RDAREA, the actual close date and time may not be used.

If the -H or -L option is specified together with the -t or-u option, the -t or-u option is effective.

- -x

Specifies that only the HiRDB file names are to be displayed (listed horizontally in alphabetical order). When this option is omitted, the HiRDB file names are listed vertically.

Command arguments

HiRDB-file-system-area-name ～ <pathname> ((up to 165 characters))

Specifies the name of the HiRDB file system area whose files are to be displayed.

HiRDB-filename ～ <HiRDB filename>

Specifies the name of the HiRDB file whose information is to be displayed.

Rules

1. The pdfls command can be executed at any time, whether or not HiRDB is active.
2. The pdfls command must be executed separately for each server machine containing the HiRDB file system.
3. If you are executing the command on a HiRDB file system area that is used for shared RDAREAs, execute it from the server machine where the updatable back-end server is located.

Notes

1. Return code 0 for the pdfls command indicates normal termination, and return code -1 indicates abnormal termination.

Output format

-H option specified

```
MODE      UID   GID   RSIZE   RNUM   TIME   FILE
aabbcc    dd...d  ee...e  ffff      gg...g   hh...h   ii...i
```

Explanation

aa

Owner's access privileges:

`rw`

Read and write

`r-`

Read

`-w`

Write

`--`

None

bb

Group's access privileges:

`rw`

Read and write

`r-`

Read

`-w`

Write

`--`

None

cc

Other users' access privileges:

`rw`

Read and write

`r-`

Read

`-w`

Write

`--`

None

dd...d

Owner's name (up to 9 characters).

ee...e

Owner's group name (up to 9 characters).

Zero (0) always is displayed.

ffff

Record length (4 decimal digits).

gg...g

Number of records (8 decimal digits).

hh...h

Update close date and time.

This information is displayed in the format *hour:minute month date year*.

The following output example is for 2005-01-07 (Friday) at 09:05:

09:05 Jan 07 2005

ii...i

HiRDB file name (up to 30 characters).

`-H` and `-L` options specified

```
MODE   UID     GID     PID   L TIME    FILE
aabbcc  dd...d   ee...e   pppp  q  hh...h    ii...i
```

Explanation

aa

Owner's access privileges:

`rw`

Read and write

`r-`

Read

`-w`

Write

`--`

None

bb

Group's access privileges:

`rw`

Read and write

`r-`

Read

`-w`

Write

`--`

None

cc

Other users' access privileges:

`rw`

Read and write

`r-`

Read

`-w`

Write

`--`

None

dd...d

Owner's name (up to 9 characters).

ee...e

Owner's group name (up to 9 characters).

Zero (`0`) always is displayed.

pppp

If there is a process locking the file, the command displays `*`; if there is no process locking the file, the command displays `-`.

q

Lock status identification flag:

E

Locked.

-

Not locked

hh...h

Update close date and time.

This information is displayed in the format *hour:minute month date year*.

The following output example is for 2005-01-07 (Friday) at 09:05:

09:05 Jan 07 2005

ii...i

HiRDB file name (up to 30 characters).

`-H` and `-S` options specified

```
MODE     UID      GID      RSIZE  RNUM     TIME     FILE
aabbcc   dd...d   ee...e   ffff   gg...g   hh...h   ii...i
                  extent size     jj...j
```

Explanation

aa

Owner's access privileges:

r

Read

w

Write

-

Neither read nor write

bb

Group's access privileges:

r

Read

w

Write

-

Neither read nor write

cc

Other users' access privileges:

r

Read

w

Write

-

Neither read nor write

dd...d

Owner's name (up to 9 characters)

ee...e

Owner's group name (up to 9 characters)

Zero (0) always is displayed.

ffff

Record length (4 decimal digits)

gg...g

Number of records (8 decimal digits)

hh...h

Update close date and time

This information is displayed in the format *hour:minute month date year*.

The following output example is for 2005-01-07 (Friday) at 09:05:

09:05 Jan 07 2005

ii...i

HiRDB file name

jj...j

Number of records per HiRDB file extension. If there is more than one HiRDB file extent, this information is displayed for each extension.

pdfmkfs (Initialize HiRDB file system area)

Function

The pdfmkfs command initializes a specified file as a HiRDB file system area.

Executor

HiRDB administrator

Format

```
pdfmkfs [-r] -n HiRDB-file-system-area-size -l max-files-count
        [-k usage-purpose] [-e max-secondary-allocations-count]
        [-s sector-length] [-i] [-r]
         filename
```

Options

- -n *HiRDB-file-system-area-size* ～ <unsigned integer>

Specifies, in megabytes, the space to be allocated as the HiRDB file system area.

pd_large_file_use operand in system definition	File type	Value range
N	File (NTFS)	1-2,047
	Direct disk access (raw I/O)	1-2,047
Y	File (NTFS)	1-1,048,575
	Direct disk access (raw I/O)	1-1,048,575

If you are using a file (NTFS) with a size greater than 100 gigabytes, specify Y in the pd_ntfs_cache_disable operand in the system definition to avoid a shortage of OS resources. Also make sure that HiRDB files for LOB RDAREAs are not intermixed with HiRDB files for RDAREAs.

The management area used for management of a HiRDB file system area is also included in the value of the -n option. Therefore, the actual size allocated to the HiRDB file is *(value of the -n option) - (size of management area)*. For details about determining the size of the management area, see *Notes*. You can use the pdfstatfs command to check the actual size that is allocated to the HiRDB file.

- -l *max-files-count* ～ <unsigned integer> ((1-4096))

Specifies the maximum number of HiRDB files that can be created in the allocated HiRDB file system area.

Once you execute the pdfmkfs command, you can no longer change this value. Therefore, carefully determine this value taking into account the number of HiRDB files to be used and future expansion plans.

- -k *usage-purpose* ～<<SVR>>

Specifies the purpose for which the HiRDB file system area will be used:

DB

HiRDB file system area for RDAREAs

SDB

HiRDB file system area for shared RDAREAs. SDB can be specified for a HiRDB/Parallel Server when direct disk access is used as the HiRDB file system area. For regular RDAREAs, use a HiRDB file system area whose purpose is DB.

When the purpose of initialization is SDB, execute the command at the server machine where the updatable back-end server is located. Do not execute the command from a referencing-possible back-end server.

Once created, a DB-purpose HiRDB file system area cannot be distinguished from a SDB-purpose HiRDB file system area. For this reason, you should assign area names that distinguish between the two purpose types.

SYS

HiRDB file system area for system log files, synchronization point dump files, status files, or audit trail files.

If you specify SYS, do not specify the -e option.

WORK

HiRDB file system area for list RDAREA or work table files.

UTL

HiRDB file system area for utilities.

You can create the following files in a HiRDB file system area for utilities:

- Unload data files
- LOB data unload files
- Index information files (applicable to delayed batch creation of plug-in index)
- Backup files
- Differential backup files
- Differential backup management files

- Unload log files

NUTL

HiRDB file system area for utilities (without using Windows cache). The system accesses this area without using cache regardless of the pd_ntfs_cache_disable operand value specified in the system definition.

Specify NUTL when a shortage of OS resources is expected because the total file size exceeds 100 gigabytes or when direct disk access is used. Normally, however, specify UTL.

SVR

HiRDB file system area that can be used for any purpose other than utilities

SVR can be specified if there is no problem with mixing HiRDB files for different usage purposes in a single HiRDB file system area, such as in a test system. Normally, however, each HiRDB file system area is dedicated to a particular type of usage.

Do not specify SVR when you are creating a HiRDB file system area that will be used for the system switchover facility. For a HiRDB file system area created with SVR specified, HiRDB resumes processing when data is written in the Windows file cache. Therefore, if an error occurs before the data is written on the disk, file conformity will be lost.

Limitations (when using direct disk access (raw I/O))

When using direct disk access, the following limitations apply:

Usage purpose	Availability	Remarks
DB	Y	
SDB	Y	
SYS	Y	
WORK	N	You cannot specify direct disk access in the pdwork operand in the system definition.
UTL	N	
NUTL	Y	
SVR	N	

Legend:

Y: Available.

N: Not available.

- `-e` *max-extensions-count* ～ <unsigned integer> ((0-60000)) <<0>>

Specifies the maximum number of extensions for each HiRDB file in the HiRDB file system area. HiRDB increases the size of HiRDB files according to the specified number of extension. The maximum number of extensions permitted per HiRDB file is 23. If the specified value is *maximum number of files* × 23 or greater, the command ignores the specified value and assumes the *maximum number of files* × 23.

Criteria

1. In the case of a HiRDB file system area for work table files, if the estimated size of a work table file exceeds the value listed below, make sure that the number of extensions is specified. For details about estimating the size of work table files, see the *HiRDB Version 8 Installation and Design Guide*.

   - For HiRDB/Single Server

     128 kilobytes

   - HiRDB/Parallel Server

     512 kilobytes

2. Even when the estimated size does not exceed the above value, or for an HiRDB file system area for other than work table files, we recommend that you specify the number of extensions.
3. If you are using RDAREA automatic extension, make sure that the number of extensions is specified.
4. Operations such as deletion, re-initialization (for reduction of allocation size or with `with reconstruction` specified), and integration of RDAREAs result in deletion of allocated HiRDB files and reduction in size. As a result, the HiRDB file system area becomes fragmented. If you use these operations, you should specify the number of extensions. If the system cannot allocate contiguous free space when an RDAREA is added, extended, or re-initialized, multiple fragmented spaces are used to create the new HiRDB file. In this case, the number of extensions that is required is *number of fragmented free spaces - 1*.

Determining the specification value

Use one of the following formulas to determine the number of extensions:

-k option	Number of extensions to be specified
DB	min (60000, 23 × cumulative total number of final files for the RDAREA to be extended*) If you frequently perform deletion, re-initialization (for reduction of allocation size or with `with reconstruction` specified), or integration of RDAREAs, you should use the following value: min (60000, 23 × cumulative total number of RDAREA structure files)
SDB	
SYS	0
WORK	min(60000, 23 × value of the `-l` option)
UTL	
NUTL	
SVR	min(60000, 23 × cumulative number of files subject to extension) Extension is applied to work table files and final files in the RDAREA subject to extension.*

Note 1

Once you execute `pdfmkfs`, you can no longer change the number of extensions. Therefore, specify an appropriate value taking into account the future extensions.

Note 2

As the number of extensions increases, the memory size required by the processes that use the HiRDB file system area increases.

Note 3

If the specified value is less than the value obtained from the above formula, the limit of the HiRDB file system area is reached before the maximum number of extensions is reached for the HiRDB files.

* Note that the number of last files in an RDAREA subject to extension is not one per RDAREA. This is because when `pdmod` is used to expand an RDAREA, the RDAREA is extended at the end, so that the last HiRDB file specified in the `expand rdarea` statement becomes the last file. Both the last file before and the last file after the expansion are included in the number of last files in the RDAREA subject to extension.

- `-s` *sector-length* ～ <unsigned integer> ((1024,2048,4096)) <<1024>>

If you are using the HiRDB file system area with a device that handles a medium with a physical sector length of 2,048 and 4,096 bytes, such as DVD-RAM device, this option specifies the minimum I/O unit expressed as the sector length. For details about the physical sector length of a medium, see the manual for the medium.

Rules

1. If you specify 2,048 or 4,096 as the sector length, you can specify either DB or NUTL in the -k option. When the -k option is omitted, SVR is assumed. Therefore, when you specify a sector length, make sure that -k DB or -k NUTL is specified.
2. You can use a HiRDB file system area for which a sector length is specified for the following purposes:

   - HiRDB file system area for RDAREAs other than list RDAREAs

   - Output destination of the pdlogunld command (unload log file)

   - Backup files for pdcopy and pdrstr

   - Unload data files for pdrorg

   If you are using the HiRDB file system area for RDAREAs, specify a multiple of the sector length specified in the -s option as the page length in the pdinit or pdmod control statement for creating an RDAREA.
3. You can specify the sector length if you use the direct disk access.

- -i

Specifies that the HiRDB file system area specified in the -n option is to be initialized from the beginning. When this option is omitted, the command creates only the management information for the HiRDB file system area. If the HiRDB file system area is large, it may take time to initialize it.

If you omit the -i option, the command creates only the management information for the HiRDB file system area (an area with a maximum size of about 3.5 megabytes is allocated).

The HiRDB file system area will be extended up to the size specified with the -n option when it is actually used. The OS file system containing this HiRDB file system area may become full when this HiRDB file system area is put into use. To avoid this, specify the -i option.

If you create HiRDB file system area to be used with system switchover facility, specify -i option. If you do not specify -i option, the files might be destroyed when the extension of HiRDB file system area and the power failure are generated at the same time.

- -r

Specifies that no confirmation message is to be displayed when the file is initialized. If this option is omitted and the specified regular file already exists, the command displays a confirmation message asking whether or not to initialize the file. If a letter other than g is entered in response to the confirmation message, the command will not initialize the file.

If you are using direct disk access (raw I/O), you cannot specify the `-r` option (in which case no confirmation message is displayed).

Command arguments

- *filename* ～ <pathname>((up to 165 characters))

Specifies a name for the file to be initialized. This name is used as the name of the HiRDB file system area.

You can specify a maximum of 30 characters for the name of a HiRDB file that is created in the HiRDB file system area. Note that because *HiRDB-file-system-area-name\HiRDB-filename* cannot exceed 167 characters, the maximum length of an HiRDB file name is reduced accordingly if the HiRDB file system area name is greater than 136 characters.

For HiRDB files that are created automatically by HiRDB, make sure that the following maximum lengths for HiRDB file system area names are not exceeded:

HiRDB file created automatically by HiRDB	Maximum length of HiRDB file system area name (in characters)
Work table file (specified with `pdwork` operand in system definition)	141
Audit trail file (specified with `pd_aud_file_name` operand in system definition)	150
Index information file for delayed batch creation of plug-in index (specified with `pd_plugin_ixmk_dir` operand in system definition)	136
Unload log file for system log for the automatic log unloading facility (specified with `pd_log_auto_unload_path` operand in system definition)	136

Rules

1. The `pdfmkfs` command can be executed at any time, whether or not HiRDB is active.
2. The `pdfmkfs` command must be executed separately for each server machine for which a HiRDB file system area is to be initialized.

Notes

1. Return code `0` for the `pdfmkfs` command indicates normal termination, and return code `-1` indicates abnormal termination.
2. If the value specified as the area size is greater than the size of the disk volume or partition during initialization, the partition physically following that partition may be damaged. Therefore, a size that is greater than the size of the disk volume or

partition must not be specified.

3. The following shows the relationships among a HiRDB file system area name, HiRDB file name, and physical file name:

4. The results of executing the `pdfmkfs` command can be checked by the `pdfstatfs` command.
5. For details about the values specifiable in the `-n`, `-l`, and `-e` options, see the following manuals, depending on the usage purpose:
   - `SYS` usage purpose

     Chapter *23. Sample Files* in the *HiRDB Version 8 Installation and Design Guide*.
   - `DB` usage purpose

     Chapter *21. Simple Installation of a HiRDB/Single Server* in the *HiRDB Version 8 Installation and Design Guide*.
   - `WORK` usage purpose

     Chapter *22. Simple Installation of a HiRDB/Parallel Server* in the *HiRDB Version 8 Installation and Design Guide*.

     Sections 3.2.1, 3.3.2, and 3.3.3 in the manual *HiRDB Version 8 System Definition*.
   - `UTL` usage purpose

     Chapter *8. Database Reorganization Utility (pdrorg)*

     Chapter *18. Database Copy Utility (pdcopy)*
   - `SVR` usage purpose

     See the appropriate manuals previously, depending on the usage purpose of the HiRDB file to be created in the HiRDB file system area.
6. If the purpose of use is `WORK`, generates a utilization rate warning message on an initialized HiRDB file system area, based upon the specifications `pd_watch_resource` and `pdwork_wrn_pnt` that are provided in the system definitions.
7. As the maximum value of system resources for the HiRDB administrator, specify

an unlimited value or a value that is greater than the -n option value.

8. A management area is allocated to the HiRDB file system area. The size of the management area to be allocated depends on the type of OS, the -n, -l, -e, and -s options, and the type of files to be used. The formulas for determining the size of a management area are presented below. Note that the sizes of the fixed and variable management sections may increase depending on the process because they are read into memory while the corresponding HiRDB file system area is used.

   Management area (bytes)
   = *Fixed management section* + *variable management section* + *HiRDB file management section* × *a*

   - -n 2047 or less

     Fixed management section = 4096

     Variable management section = ↑ (20 + 4 × *a*)/*c* ↑ × *c* + ↑ (48 + 16 × (*a* + *b*))/*c* ↑ × *c*

     HiRDB file management section = ↑ 320/*c* ↑ × *c*

   - -n 2048 or greater

     Fixed management section = 4096

     Variable management section = ↑ (20 + 4 × *a*)/*c* ↑ × *c* + ↑ (56 + 24 × (*a* + *b*))/*c* ↑ × *c*

     HiRDB file management section = ↑ 480/*c* ↑ × *c*

   Description of variable symbols:

   a: Value of the -l option

   b: Value of the -e option (0 if the option is omitted)

   c: Value of the -s option (512 if the option is omitted)

Examples

Initialize the following HiRDB file system areas:

- c:\sysarea\rdsys011

Command execution

```
pdfmkfs -n 25 .....................1
        -l 10 .....................2
        -e 5 ......................3
        c:\sysarea\rdsys011 .......4
```

Explanation

1. Size of HiRDB file system area: 25 MB
2. Maximum number of files: 10
3. Maximum number of secondary allocations: 5
4. Filename: `c:\sysarea\rdsys011`

pdfrm (Delete HiRDB file)

Function

The pdfrm command deletes a specified HiRDB file from a HiRDB file system area.

Executor

HiRDB administrator

Format

pdfrm [-i] *HiRDB-file-system-area-name**HiRDB-filename*

Options

- -i

  Specifies that a confirmation message is to be displayed before the HiRDB file is deleted. The file can be deleted by entering y as the response to the confirmation message.

Command arguments

- *HiRDB-file-system-area-name**HiRDB-filename*

Specifies the name of the HiRDB file system area containing the HiRDB file to be deleted and the name of the HiRDB file to be deleted.

HiRDB-file-system-area-name ～ <pathname> ((up to 165 characters))

Specifies the name of the HiRDB file system area.

HiRDB-filename ～ <HiRDB filename>

Specifies the name of the HiRDB file.

Rules

1. If the usage purpose of the HiRDB file system area that contains the HiRDB file to be deleted is other than UTL, the pdfrm command cannot be executed in the HiRDB file system area that contains that HiRDB file while the active HiRDB is in use (if the HiRDB file in the HiRDB file system area has been opened even once). If the usage purpose is UTL, the pdfrm command can be executed regardless of whether or not the HiRDB file system area is in use. However, it cannot be executed if the specified HiRDB file is in use by the active HiRDB (opened by another process).
2. The pdfrm command must be executed separately for each server machine containing the HiRDB file system area.
3. An HiRDB file that is in use (opened by another process) cannot be deleted.

4. When the -i option was specified and any character other than y is entered in response to the confirmation message, the HiRDB file will not be deleted.
5. If you are executing the command on an HiRDB file system area that is used for shared RDAREAs, execute it from the server machine where the updatable back-end server is located.

Notes

1. Return code 0 for the pdfrm command indicates normal termination, and return code -1 indicates abnormal termination.
2. The result of the pdfrm command can be checked by the pdfls command.
3. Except when the usage purpose is UTL, the pdfrm command is used to delete a HiRDB file forcibly under emergency conditions. Normally, a HiRDB file should be deleted with a host command, such as the database structure modification utility, the pdlogrm command, or the pdstsrm command, depending on the usage purpose.
4. Because the pdfrm command can delete forcibly a HiRDB file that comprises an RDAREA, care should be taken in executing this command.

pdfrstr (Restore HiRDB file system)

Function

The pdfrstr command restores a HiRDB file system backed up by the pdfbkup command. Restoration can also be performed in units of HiRDB files.

The number of added areas and the number of noncontiguous free areas can be determined by displaying the status of a HiRDB file system area (by entering the pdfstatfs command). The added areas and noncontiguous free areas can then be merged and the HiRDB file system area used more effectively by taking the following measures:

- Back up the HiRDB file system (pdfbkup command)
- Initialize the HiRDB file system area (pdfmkfs command)
- Restore the backed up HiRDB file system to the initialized HiRDB file system (pdfrstr command)

Executor

HiRDB administrator

Format

```
pdfrstr  [-y] [-{t|o|r}]
         backup-filename[\HiRDB-filename]
         HiRDB-file-system-area-name
```

Options

- -y

Specifies that the HiRDB file system is to be restored without obtaining user confirmation. When this option is omitted, the system is restored only after the user confirms the processing.

- -{t|o|r}

Specifies the processing to be performed if a file with the same name as a HiRDB file in the backup file to be restored already exists in the restoration destination HiRDB file system area:

-t

Do not restore if the existing HiRDB file in the restoration destination HiRDB file system area has a more recent time stamp

-o

Do not use the HiRDB file in the backup file for restoration

`-r`

Restore from the HiRDB file in the backup file regardless of the time stamp

When this option is omitted, `-t` is assumed.

Command arguments

- *backup-filename*[*HiRDB-filename*]

Specifies the backup file backed up by the `pdfbkup` command. If a particular HiRDB file is to be restored, also specifies that file.

backup-filename

Specifies the name of the backup file.

HiRDB-filename ～ <HiRDB filename>

Specifies the name of the backed up HiRDB file that is to be restored.

- *HiRDB-file-system-area-name* ～ <pathname> ((up to 165 characters))

Specifies the name of the restoration destination HiRDB file system area.

Rules

1. The `pdfrstr` command can be executed only while HiRDB is inactive.
2. The `pdfrstr` command must be executed at each server machine where the HiRDB file system area is stored.
3. Any HiRDB files in the restoration destination HiRDB file system area that are not in the backup file will remain in the HiRDB file system area after execution of the `pdfrstr` command.
4. A confirmation message is displayed when the `-y` option is omitted. The HiRDB file system is not restored if any character other than G is entered in response to the confirmation message.
5. If the `pdfrstr` command terminates abnormally during execution, the HiRDB file system area at the storage destination may become corrupted. Thus, if there are any HiRDB files in the storage destination HiRDB file system area, you should back them up in advance with the `pdfbkup` command.
6. You cannot restore a HiRDB file if the record length of the backup HiRDB file is not a multiple of the sector length of the target HiRDB file system area. Before backing up a HiRDB file, use the `pdfls` command to check its record length. To check the sector length of the target HiRDB file system area, use the `pdfstatfs` command.

Notes

1. The execution results of the `pdfrstr` command can be confirmed by the return code after the command has executed and by whether or not there are any error messages. Return code `0` for this command indicates normal termination, `1` indicates warning termination, and `-1` indicates abnormal termination.
2. Restoration to a utility HiRDB file system area cannot be made while the utility is executing; in such a case, restoration should be performed after termination of the utility.

pdfstatfs (Display status of HiRDB file system area)

Function

The pdfstatfs command displays the status of a specified HiRDB file system area.

Executor

Any user (with or without privileges)

Format

```
pdfstatfs [-{c|d}] [{-S|-x[-y]}] [-b] HiRDB-file-system-area-name
```

Options

- -{c|d}

c

Specifies that the maximum available area for allocation in the HiRDB file system area is to be set to 0. This is effective when the usage purpose of the HiRDB file system area is WORK, UTL, and NUTL.

d

Specifies that the maximum available area for allocation in the HiRDB file system area is to be displayed.

- -S

Indicates the status of available free areas in the HiRDB file system area.

- -x

Specifies that the information is to be displayed in DAT format.

- -y

When the -x option is specified, specifies that a header is to be displayed.

- -b

Specifies that the sector length of the HiRDB file system area is to be displayed.

Command arguments

- *HiRDB-file-system-area-name* ～ <pathname>((up to 165 characters))

Specifies the name of the HiRDB file system area whose status is to be displayed.

Rules

1. The pdfstatfs can be executed at any time, whether or not HiRDB is active.

2. The `pdfstatfs` command must be executed separately for each server machine containing the HiRDB file system area.
3. If you are executing the command on an HiRDB file system area that is used for shared RDAREAs, execute it from the server machine where the updatable back-end server is located.

Notes

1. Return code `0` for the `pdfstatfs` command indicates normal termination, and return code `-1` indicates abnormal termination.
2. When the maximum available area for allocation is to be displayed by specifying the `-d` option after this value has been set to 0 with the `-c` option, the maximum available area of the HiRDB file system area used since the `-c` option was set can be acquired.
3. Displaying the maximum available area of the HiRDB file system area between the time the `-c` option is specified and the `-d` option is specified is possible only when the usage purpose of the HiRDB file system area is `WORK`, `UTL`, and `NUTL`. When the usage purpose is `SYS`, `DB`, or `SVR`, the maximum available area cannot be set to 0 with the `-c` option; thus, the maximum available area that is displayed is that accumulated from the time the `pdfmkfs` command was executed.
4. To use the OS's no-cache access method with the HiRDB file system area, you need to specify the `pd_ntfs_cache_disable` operand in the system definitions.

Output format

```
 user area capacity          aa...a[kB]
 remain user area capacity   bb...b[kB]
[peak capacity               mm...m[kB]]
 available file size         cc...c[kB]
 available file count        dd...d
 current file count          ee...e
 remain file count           ff...f
 free area count             gg...g
 available expand count      hh...h
 current expand count        ii...i
[sector size                 qq...q[Byte]]
 initialize area kind        jj...j
 initialize user id          kk...k
 initialize time             ll...l

[***** HiRDB file system area space information *****]
[offset[kB]      size[kB]  used/unused]
[    nn...n       oo...o     pp...p]
```

Explanation

aa...a

Size (in KB) in the HiRDB file system area of the user area allocated to the user (up to 10 decimal digits).

bb...b

Size (in KB) in the user area of the unused space (space not allocated to HiRDB files) (up to 10 decimal digits).

cc...c

Size (in KB) of the largest space that can be allocated to a single file (up to 10 decimal digits).

dd...d

Maximum number of files that can be created, as specified with the `-l` option of the `pdfmkfs` command (up to 4 decimal digits).

ee...e

Number of HiRDB files that have already been created (up to 4 decimal digits).

ff...f

Number of HiRDB files that can still be created (maximum number of creatable files - number of already created files) (up to 4 decimal digits).

gg...g

Total number of noncontiguous free areas (up to 5 decimal digits).

hh...h

Maximum number of secondary allocations available to the HiRDB file system area (up to 5 decimal digits).

ii...i

Total number of secondary allocations that have been executed for the HiRDB file system area (up to 5 decimal digits).

jj...j

HiRDB file system area's usage purpose, as specified in the `-k` option of the `pdfmkfs` command (up to 9 characters).

If the usage purpose is `SDB`, `DB` is displayed.

If the usage purpose is `DB`, availability of OS no-cache access is also displayed as follows:

`DB`: Usage purpose is `DB` and no-cache access is not available.

`DB(NOLOB)`: Usage purpose is `DB` and no-cache access is available.

kk...k

Login name of the user who initialized the HiRDB file system area (up to 32 characters).

ll...l

Date and time the HiRDB file system area was initialized (24 characters).

This information is displayed in the format *day month date hour:minute:second year*.

The following output example is for 2005-01-07 (Friday) at 09:05:08:

```
initialize time            Fri Jan 07 09:05:08 2005
```

mm...m

User maximum available space at the current time (up to 10 decimal digits); displayed only when the `-d` option is specified.

nn...n

The leading offset for an area being used or an area not being used in the HiRDB file system area. This is displayed only when the `-S` option is specified.

oo...o

The size of an area being used or an area not being used in the HiRDB file system area. This is displayed only when the -S option is specified.

pp...p

Indicates whether an area is being used.

used

Being used.

unused

Not being used.

This is displayed only when the -S option is specified.

qq...q

Sector length of the HiRDB file system area (1 to 4 decimal digits).

This is the sector length specified in the -s option of the pdfmkfs command (1,024 when the option is omitted).

- DAT output format (when -x -y is specified)

```
"USER_AREA_CAPACITY","REMAIN_USER_AREA_CAPACITY","PEAK_CAPACITY","AVAILABLE_FILE_SIZ
E",
"AVAILABLE_FILE_COUNT","CURRENT_FILE_COUNT","REMAIN_FILE_COUNT","FREE_AREA_COUNT","A
VAILABLE_EXPAND_COUNT",
"CURRENT_EXPAND_COUNT","SECTOR_SIZE","INITIALIZE_AREA_KIND","INITIALIZE_USER_ID","IN
ITILIZE_TIME"[CR]
aa...a,bb...b,mm...m,cc...c,
dd...d,ee...e,ff...f,gg...g,hh...h,
ii...i,qq...q,"jj...j","kk...kk","ll...ll"[CR]
```

Note

[CR] indicates a linefeed.

Rules for output in DAT format

1. The elements are separated by the comma (,), and information is output as a single line of text ending with a carriage return.
2. A character element is enclosed in double quotation marks (").
3. The sector length and the current maximum amount of usage by user are displayed only when the appropriate combination of information acquisition options is specified. If an information acquisition option is not specified, the command displays a blank or only "" without displaying the corresponding

item.

pdgrprfl (Refresh user and role information)

Function

When the Directory Server linkage facility is used, the `pdgrprfl` command updates to the most recent status the user and role information maintained in memory by HiRDB.

You should execute the `pdgrprfl` command in the following situations:

- When you have added, changed, or deleted a user or role with the Directory Server
- When you have added, changed, or deleted a member in a role

This command enables you to establish connection with the HiRDB using the most recent user information provided by the Directory Server. It also enables you to use table access privileges on the basis of the most recent role information in the Directory Server.

For details about the Directory Server linkage facility, see the *HiRDB Version 8 System Operation Guide*.

Executor

HiRDB administrator

Format

```
pdgrprfl
```

Option

None

Rules

1. You can execute the `pdgrprfl` command while HiRDB is active.
2. You can execute the `pdgrprfl` command on the server machine containing the single server or the server machine where the system manager is located.
3. The `pdgrprfl` command is available only when the Directory Server linkage facility is used.
4. Do not execute more than one `pdgrprfl` command at any one time.
5. If you execute the `pdgrprfl` command, the buffer hit rate for the table definition information and user privilege information reduces temporarily.

Notes

1. Return code `0` for the `pdgrprfl` command indicates normal termination, and return code `8` indicates abnormal termination.

Relationship between front-end server activity status and pdgrprfl command

1. You can execute the `pdgrprfl` command whether or not a front-end server is running. However, if there is a front-end server that is preparing to start or stop, the `pdgrprfl` command terminates with an error.
2. If you execute the `pdgrprfl` command when it cannot be determined whether or not a front-end server other than a recovery-unnecessary front-end server is running, an error results because the user or role information in the front-end servers cannot be updated. If you execute the `pdgrprfl` command when it cannot be determined whether or not a recovery-unnecessary front-end server is running, the command skips that front-end server. Its user or role information is updated to the most recent status later when it is restarted.
3. The following table provides details of the relationship between the front-end server activity status and the `pdgrprfl` command:

pdls -d svr command execution result		Front-end server status	pdgrprfl command execution result	
Unit	Server		Recovery-unnecessary front-end server	Other front-end server
STARTING	STOP	Initial status	Command terminated with error.	Command terminated with error.
STARTING	STARTING	Preparing to start	Command terminated with error.	Command terminated with error.
ACTIVE	STARTING			
STARTING	STARTING	Preparing to start (server process has already started)	Command terminated with error.	Command terminated with error.
ACTIVE	STARTING			
ACTIVE	STARTING	Ready to start	Command processing executed.	Command processing executed.
ACTIVE	SUSPEND			
ACTIVE	ACTIVE	Running	Command processing executed.	Command processing executed.
ACTIVE	TRNPAUSE			

pdls -d svr command execution result		Front-end server status	pdgrprfl command execution result	
Unit	Server		Recovery-unnecessary front-end server	Other front-end server
STOP	—	Stopped due to reduced mode operation	Command processing skipped.	Command processing skipped.
ACTIVE	STOP	Stopped by individual server stop command		
STOP(N)	—	Stopped by normal termination	Command processing skipped.	Command processing skipped.
STOP(F)	—	Stopped by forced termination	Command processing skipped.	Command processing skipped.
STOP(A)	—	Server activity status unknown because unit shutdown was detected	Command processing skipped.	Command terminated with error.
STOPPING	—	Preparing to stop	Command terminated with error.	Command terminated with error.
ACTIVE	STOP	Ready to execute the database recovery utility (after executing the `pdstart -r` command)	Command terminated with error.	Command terminated with error.

Legend:

—: Not applicable

pdhold (Shut down RDAREAs)

Function

The pdhold command shuts down specified RDAREAs. If a specified RDAREA is in use, the command shuts it down when its utilization is completed.

Shutdown without specifying any options other than -r is called *command shutdown*.

Executor

HiRDB administrator

Format

```
pdhold -r {RDAREA-name[,RDAREA-name]...|ALL}
       [-c|-i|-b [-w] [-u]]
```

Options

- -r {*RDAREA-name*[,*RDAREA-name*]...|ALL}

RDAREA-name ～ <identifier> ((1-30))

Specifies the names of RDAREAs to be shut down. A shutdown achieved by specifying the -r option is called *command shutdown*.

You can also use batch specification of RDAREA names. For details about batch specification of RDAREA names, see *1.5.2 Batch specification of RDAREA names in operation commands*.

ALL

Specifies that all RDAREAs are to be shut down. The following table describes the RDAREAs that are shut down depending on the combination of other options:

Other option	RDAREAs that are shut down
-b	All RDAREAs except list RDAREAs
Other	All RDAREAs except the master directory RDAREA

Rules:

1. If you specify the -b option, you can also specify a master directory RDAREA; otherwise, you cannot specify a master directory RDAREA. Additionally, if you specify the -b option, you cannot specify a list RDAREA.
2. You cannot specify a duplicated RDAREA name. If specified, the system will eliminate all duplicated RDAREA names.

3. You can specify a maximum of 128 RDAREAs. If more than 128 RDAREA names are specified, the system ignores the excess names.
4. If an RDAREA name is enclosed in \", the system treats it as being case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a blank, enclose the entire name in \".
5. If specifying a user RDAREA or user LOB RDAREA, make sure that the corresponding data dictionary RDAREA is in one of the following statuses:
   - Open and shutdown release status
   - Open and shutdown status placed by the pdhold command
6. When specifying both data dictionary and user RDAREAs or both data dictionary and user LOB RDAREAs, be sure to specify the data dictionary RDAREAs last.

■ -c

Specifies that the RDAREA is to be shut down and then closed. This is called *closed command shutdown*.

If this option is omitted, HiRDB shuts down only the specified RDAREAs.

■ -i

Specifies that the RDAREAs that have been shut down are to be made available to users for referencing. This is called *reference-possible command shutdown*.

If you omit this option, you cannot reference the RDAREAs shut down.

■ -b [-w] [-u]

Specifies that the RDAREAs are to be placed in backup hold status.

The backup-hold status enables a backup copy to be made even in online mode. RDAREAs are placed in backup hold at the following times:

- When a backup is acquired in pdcopy's updatable mode
- When a backup is acquired using a method other than pdcopy, such as a general-purpose backup tool or OS command.
- When dual logical volumes are separated

There are four different backup-hold statuses:

1. Reference-possible backup-hold (-b specified)
2. Reference-possible backup-hold (update WAIT mode) (-b,-w specified)
3. Updatable backup-hold (-b, -u specified)
4. Updatable backup-hold (WAIT mode) (-b, -w, -u specified)

Statuses 1 and 2 above are called *committing a database*.

The following explains each backup-hold status:

Reference-possible backup-hold

If an updating transaction can result in an error until the shutdown status is released, place the RDAREA on reference-possible backup-hold.

Notes

1. For an RDAREA in the reference-possible backup-hold status, no deadlock occurs with an updating transaction.
2. You can use the backup copy obtained in this status to restore the database to the point where the backup was made without having to use the system log. Using the system log existing immediately before the RDAREA was placed in the backup-hold status, you can restore the database to the point where the error occurred.
3. You can reference an RDAREA on reference-possible backup-hold, but an attempt to update the RDAREA results in an SQL error (`-920`) unless the shutdown status is released. If an updating transaction in no-log mode results in an error, all updated RDAREAs, if any, are placed in shutdown status.

Reference-possible backup-hold (update WAIT mode)

If an updating transaction can be placed on hold until the shutdown status is released, place the RDAREA on reference-possible backup-hold (update `WAIT` mode).

Notes

1. You can use the backup copy obtained in this status to restore the database to the point where the backup was made without having to use the system log. Using the system log existing immediately before the RDAREA was placed on backup-hold, you can restore the database to the point of error occurrence.
2. You can reference an RDAREA on reference-possible backup-hold (update `WAIT` mode), but an attempt to update the RDAREA results in the hold status unless the RDAREA is released from the backup-hold status. Therefore, the value of the `pd_lck_wait_timeout` operand in the system definitions and the value of `PDCWAITTIME` in the client environment definitions must be at least the period of time the backup-hold status is in effect. If timeout occurs, the updating transaction results in an SQL error (`-770`).
3. If you place an RDAREA on reference-possible backup-hold (update `WAIT` mode), deadlock may occur with an updating transaction. By specifying `pd_deadlock_priority_use=Y` in the system definition

and a deadlock priority value in the `pd_command_deadlock_priority` operand, you can specify whether the updating transaction or the operation command is to take control in the event of deadlock.

4. If deadlock occurs, the backup-hold status placed by the `pdhold` command is released and the same backup-hold processing is repeated. You can use the backup copy obtained in this status to restore the database to the point where the backup was made. This retry processing is repeated up to five times. If deadlock occurs on the fifth retry, the system releases all the backup-hold statuses placed by the `pdhold` command and terminates the processing with an error.
5. If an updating transaction in no-log mode terminates with an error due to timeout or deadlock, all updated RDAREAs, if any, are placed in shutdown status.

Updatable backup-hold

If you want to immediately place an RDAREA in backup-hold status and access it in this status, place the RDAREA in the updatable backup-hold status. If another command is executing, you can execute the `pdhold` command to place an RDAREA in updatable backup hold status after that command has terminated.

Notes

1. You can reference and update an RDAREA in the updatable backup-hold status.
2. For an RDAREA on updatable backup-hold, an applicable page's physical log is output when the update buffer takes effect on the database, unless the shutdown status is released. Therefore, note the following:

   (a) Check to see if there is enough space in the system log file.

   (b) If updating a large amount of data, do not place the RDAREA on updatable backup-hold (a large amount of plug-in index data is also updated).

   (c) Upon completion of backup processing, immediately release the RDAREA from the updatable backup-hold status.
3. Even during the execution of updating transaction, the `pdhold` command can place an RDAREA on updatable backup-hold.
4. To restore the database using the backup copy obtained in this status, you need the system log acquired after the backup processing and the previous synchronization point. However, if an updating transaction was executed in the pre-update log acquisition or no-log mode after the

previous synchronization point, you cannot restore the database with this backup copy.

5. Shared RDAREAs cannot be placed in updatable backup-hold status.

Updatable backup-hold (WAIT mode)

If an RDAREA can be placed in backup-hold status after completion of an updating transaction and you want to access the RDAREA in this status, place the RDAREA in the updatable backup-hold status (`WAIT` mode).

Notes

1. To restore the database using a backup copy obtained in this status, you do not need the system log if no warning message (`KFPH00157-W`) was output when the RDAREA was released from the backup-hold status. If a warning message was output, you need the system log existing after the previous synchronization point. Using the system log existing immediately before the RDAREA was placed in backup-hold status, you can restore the database to the point of error occurrence, whether or not a warning message was output. If the backup copy was made with `pdcopy` specifying `-M s`, you need the system log existing after the previous synchronization point, whether or not a warning message was output.
2. If you place an RDAREA on updatable backup-hold (`WAIT` mode), you can reference and update the RDAREA.
3. For an RDAREA on updatable backup-hold (`WAIT` mode), an applicable page's physical log is output when the update buffer takes effect on the database unless the shutdown status is released. Therefore, note the following:

   (a) Check to see if there is enough space in the system log file.

   (b) If updating a large amount of data, do not place the RDAREA on updatable backup-hold (a large amount of plug-in index data is also updated).

   (c) Upon completion of backup processing, immediately release the RDAREA from the updatable backup-hold status.
4. If updating transaction processing is underway, the `pdhold` command to place an RDAREA on updatable backup-hold (`WAIT` mode) is placed on hold until the transaction processing is completed.
5. If an RDAREA is on updatable backup-hold, an updating transaction in pre-update log acquisition or no-log mode for this RDAREA is placed on hold until the RDAREA is released from the backup-hold status. Therefore, the value of the `pd_lck_wait_timeout` operand in the

system definitions and the value of PDCWAITTIME in the client environment definitions must be at least the period of time the backup-hold status is in effect. If timeout occurs, the updating transaction in pre-update log acquisition or no-log mode results in an error.

6. If you place an RDAREA on updatable backup-hold (WAIT mode), deadlock may occur with an updating transaction in the pre-update log acquisition or no-log mode. By specifying pd_deadlock_priority_use=Y in the system definition and a deadlock priority value in the pd_command_deadlock_priority operand, you can specify whether the updating transaction or the operation command is to take control in the event of deadlock.
7. If deadlock occurs, the backup-hold status placed by the pdhold command is released and the same backup-hold processing is repeated. You can use the backup copy obtained in this status to restore the database to the point where the backup was made. This retry processing is repeated up to five times. If deadlock occurs on the fifth retry, the system releases all the backup-hold status placed by the pdhold command and terminates the processing with an error.
8. If an updating transaction in pre-update log acquisition or no-log mode terminates with an error due to timeout or deadlock, all updated RDAREAs, if any, are placed in shutdown status.
9. Shared RDAREAs cannot be placed in updatable backup-hold (WAIT mode) status.

- -w

Specifies that the RDAREAs are to be placed on reference-possible backup-hold (update WAIT mode) or updatable backup-hold (WAIT mode).

If you place the RDAREAs on reference-possible backup-hold (update WAIT mode), an updating transaction is placed on hold. If you place the RDAREAs on updatable backup-hold (WAIT mode), an updating transaction in pre-update log acquisition or no-log mode is placed on hold.

- -u

Specifies that the RDAREAs are to be placed on updatable backup-hold.

You can reference and update an RDAREA on updatable backup-hold. However, an attempt to update a UAP or utility in no-log mode is placed on hold until the shutdown status is released. If you omit this option, the system places the specified RDAREAs on reference-possible backup-hold.

Rules of backup-hold

1. While an RDAREA is in the backup-hold status, you cannot execute pdmod (to extend, reinitialize, or change the attributes of an RDAREA), pdload, or pdrorg (to reload) on the RDAREA.
2. The status of an RDAREA on reference-possible backup-hold (update WAIT mode) or updatable backup-hold is not inherited during a rerun.
3. If you place an RDAREA on updatable backup-hold during operation in the no-log mode, be sure to use the pdlogswap -d sys -w command to swap the system log files and validate the synchronization point dump beforehand. Otherwise, you need to use pdrstr to specify a range of recovery in the event of an error.
4. If you terminate or restart HiRDB while making a backup copy in the updatable backup-hold status, the backup copy obtained cannot be guaranteed. In this case, obtain a backup copy again.
5. For an RDAREA on reference-possible backup-hold (update WAIT mode), deadlock may occur between the pdhold command and an updating transaction. For an RDAREA on updatable backup-hold, deadlock may occur between the pdhold command and an updating transaction in the no-log mode. In the event of deadlock, if the RDAREA is already placed on reference-possible backup-hold (update WAIT mode) or updatable backup-hold, execute the pdrels command to release the RDAREA from the shutdown status and then re-execute the pdhold command. If the RDAREA has not been placed on reference-possible backup-hold (update WAIT mode) or updatable backup-hold, re-execute the pdhold command after a while.
6. If update buffer contents become effective for an RDAREA while it is on updatable backup-hold (WAIT mode) status, a warning message (KFPH00157-W) is issued when the backup-hold status is released. If this warning message has not been issued, you can restore data to the point where the backup was made without using the system log. However, if this message has been issued, you need the system log obtained since the previous synchronization point. Whether or not the warning message has been issued, if you use the system log obtained immediately before the backup shutdown, you can restore data to the point where the error occurred. If you have acquired the backup using pdcopy with -M s specified, you need the system log obtained since the previous synchronization point whether or not the warning message has been issued.
7. If the pdhold command that has been issued to place an RDAREA in backup hold status is placed in wait status by another command, the timeout specified in the pd_lck_wait_timeout operand in the system definition does not apply to this pdhold command. The command is kept in wait status until the RDAREA is released from the other command. To terminate the pdhold command in wait status, use the pdcancel command to end the process.

Rules

1. You can execute the pdhold command only while HiRDB is active.
2. Execute the pdhold command at the server machine containing the single server or the server machine where the system manager is located.
3. The pdhold command locks RDAREAs in the EX mode or in the PR mode if the -i or -b option is specified. Therefore, if another transaction is accessing the specified RDAREA, the pdhold command is placed on hold until the transaction terminates. If the -i or -b option is specified, the pdhold command is placed on hold only if an updating transaction is accessing a specified RDAREA. Table 2-1 shows the relationship between an RDAREA's shutdown status and transaction.

Table 2-1: Relationship between RDAREA's shutdown status and transaction

RDAREA shutdown status	Transaction		
	Reference	Update	Update (no log)[1]
Command shutdown	Wait	Wait	Wait
	Error	Error	Error
Reference-possible shutdown	No wait[2]	Wait	Wait
	No wait	Error	Error
Reference-possible backup hold	No wait[2]	Wait	Wait
	No wait	Error	Error
Reference-possible backup hold (update WAIT mode)	No wait[2]	Wait	Wait
	No wait	Wait	Wait
Updatable backup-hold	No wait	No wait	No wait
	No wait	No wait	Wait
Updatable backup-hold (WAIT mode)	No wait[2]	Wait	Wait
	No wait	No wait	Wait

Note

The upper row indicates the case in which a transaction locks the RDAREA first. The lower row indicates the case in which the pdhold command locks the RDAREA first. For example, if a referencing UAP locks the RDAREA first, an attempt to place the RDAREA in the command shutdown status using the pdhold command places the command on hold. If the pdhold

command (command shutdown) locks the RDAREA first, an attempt to execute a UAP to update this RDAREA results in an error.

[1] This indicates a transaction in the no-log mode for which `NO` is specified for `PDDBLOG` in the client environment definition.

[2] If an updating access occurs between a referencing access and the time the command is entered, any subsequent accesses will be placed in wait status.

4. Table 2-2 shows the handling of a transaction in shutdown processing, where *in shutdown processing* means `ACCEPT-HOLD` (the RDAREA status obtained as the execution result by the `pddbls` command):

Table 2-2: Handling of a transaction in shutdown processing

Status in shutdown processing	Referencing transaction	Updating transaction
Command shutdown	Error	Error
Reference-possible shutdown	Wait	Error
Reference-possible backup hold	Wait	Error
Reference-possible backup hold (update WAIT mode)	Wait	Wait
Updatable backup-hold	No wait*	No wait*
Updatable backup-hold (WAIT mode)	Wait	Wait

* The transaction is not placed in wait status because updatable backup-hold status does not lock RDAREAs.

5. You cannot place an RDAREA containing a falsification prevented table that is in reload-not-completed data status into reference-possible command shutdown status (`-i` specified) or backup hold status (`-b` specified). Reloading data to a falsification prevented table with `pdrorg` releases the table from reload-not-completed data status. After the reload-not-completed data status is released, place the table in reference-possible command shutdown or backup hold status.

6. If the `pdhold` command is executed on a shared RDAREA, all back-end servers are locked. If there can be multiple concurrent accesses to the corresponding RDAREA, global deadlock may occur, resulting in a timeout. If global deadlock has occurred, re-execute the `pdhold` command.

7. When RDAREA automatic extension has been applied to an RDAREA, placing the RDAREA in updatable backup-hold status or updatable backup-hold (`WAIT` mode) status suppresses the automatic extension of the RDAREA. To release RDAREA automatic extension from the suppression, execute the `pdrels`

command to release the updatable backup hold or the updatable backup hold (`WAIT` mode).

8. When RDAREA automatic extension has been applied to an RDAREA, before placing the RDAREA in updatable backup-hold status or updatable backup-hold (`WAIT` mode) status, use the database condition analysis utility to check that there is enough unused area. If there is not enough unused area, execute the free page release utility.

Note

1. You can use the `pddbls` command to check the result of the `pdhold` command.
2. The following shows the `pdhold` command's return codes:

   `0`: Normal termination

   `4`: Warning termination (some RDAREA processing terminated with an error)

   `8`: Abnormal termination

   `12`: Abnormal termination (an event occurred that prevented output of an error message)

   If the error code is `12`, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the command. If no error message has been output to the event log, contact the customer engineer.

pdjarsync (Manipulate JAR files)

Function

The pdjarsync command manipulates JAR files.

There are the following types of JAR file manipulation:

- Re-registration of a JAR file between hosts (applicable to HiRDB/Parallel Server only)
- Registration of a JAR file
- Re-registration of a JAR file
- Deletion of a JAR file
- Display of a list of JAR files

This command is used when Java stored procedures or Java stored functions are used.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdjarsync [{-I|-R|-D| -L } ]
           [-u authorization-identifier [,authorization-identifier]...]
           [-f JAR-filename]
```

HiRDB/Parallel Server

```
pdjarsync [{-S|-I|-R|-D| -L}]
           [-x host-name[,host-name]...]
           [-u authorization-identifier [,authorization-identifier]...]
           [-f JAR-filename]
```

Options

- {-S|-I|-R|-D|-L}

Specifies the type of JAR file manipulation.

-S

For the HiRDB/Parallel Server, this option re-registers the specified JAR file between hosts.

The command re-registers (replaces) a JAR file in the host specified with the -x option into the server machine in the HiRDB environment. If the -x option is

omitted, the command re-registers a JAR file in the server machine containing the dictionary server into the server machine in the HiRDB environment.

Specify the `-S` option if you have modified the HiRDB server configuration (such as when increasing the number of server machines) or conducted system switchover.

`-I`

This option registers the specified JAR file.

If you specify this option, be sure to specify the `-u` and `-f` options.

If the specified JAR file has already been registered, the command results in an error.

`-R`

This option re-registers the specified JAR file.

If the specified JAR file has already been registered, the command overwrites the registered JAR file.

If you specify this option, be sure to specify the `-u` and `-f` options.

`-D`

This option deletes the specified JAR file.

If you specify this option, be sure to specify the `-u` and `-f` options.

If the specified JAR file has not been registered, the command results in an error.

If the `pdjarsync` command terminates abnormally, a JAR file under write operation may remain in the system. In this case, specify the `-d` option to delete such remaining JAR files.

`-L`

This option displays a list of registered JAR files.

Table 2-3 shows the relationship between JAR file manipulation type and other options.

Table 2-3: Relationship between JAR file manipulation type and other options

JAR file manipulation type	Other option		
	-x	**-u**	**-f**
`-S`	O*	O	O
`-I`	O	M	M
`-R`	O	M	M

JAR file manipulation type	Other option		
	-x	-u	-f
-D	O	M	M
-L	O	O	O

M: Mandatory

O: optional

* If specifying the -x option, specify only one host name.

- -x *host-name*[,*host-name*]

If specifying the -s option, specify only one host name that is to be the origin of re-registration. If this option is omitted, the system assumes the host where the dictionary server is located.

If you specify an option other than -s, specify the name of the host being manipulated. If omitted, the system assumes all hosts in the HiRDB environment.

- -u *authorization-identifier*[,*authorization-identifier*]...

Specify the authorization identifier of the user who is to manipulate the JAR file. Note that uppercase and lower-case letters are treated as being the same.

If you have specified the -I, -R, and -D options, be sure to specify this option also.

For the authorization identifier, you cannot specify MASTER or HiRDB.

- -f *JAR-filename*

Specify the name of the JAR file to be manipulated. Note that uppercase and lower-case letters are treated as being the same.

If you have specified the -I, -R, and -D options, be sure to specify this option also.

If you specify the -S and -L options but omit this option, the system assumes all JAR files that have been registered.

The JAR files are stored in the directory specified with the pd_java_archive_directory operand in the system definitions.

Rules

1. You can execute the pdjarsync command whether or not HiRDB is active.
2. Execute the pdjarsync command at the server machine containing the single server or the server machine where the system manager is located.
3. Do not execute more than one pdjarsync command at any one time.

Notes

1. If you execute the `INSTALL JAR`, `REPLACE JAR`, or `REMOVE JAR` SQL statement, the `pdjarsync` command is executed internally. In these cases, the `pdjarsync` command is executed from the single server or the front-end server.
2. The following shows the `pdjarsync` command's return code:

   `0`: Normal termination

   `4`: Abnormal termination (for registration or re-registration, the specified JAR has already been registered; for deletion, the specified JAR file has not been registered)

   `8`: Abnormal termination

Output format

```
HOSTNAME(aabbcc)  DATE          TIME        SIZE     OWNER    FILE
dd...d             eeee/ff/gg    hh:ii:jj    kk...k     ll...l     mm...m
  :                :             :           :        :        :
```

Explanation

aabbcc

Time the `pdjarsync` command was executed

dd...d

Host name

eeee/ff/gg

Date of the JAR file*

hh:ii:jj

Time of the JAR file*

kk...k

Size of the JAR file (in bytes)

ll...l

Owner of the JAR file

mm...m

Name of the JAR file

* If an SQL statement (`INSTALL JAR`, `REPLACE JAR`) or JDBC method was used to register or re-register the JAR file, this is the date and time the SQL statement or JDBC method was executed. If the `pdjarsync` command was used to register or re-register

the JAR file, this is the date and time of the registered or re-registered JAR file.

pdkill (Forcibly terminate a process)

Function

The `pdkill` command forcibly terminates a specified process.

Executor

HiRDB administrator

Format

`pdkill` *process-ID*

Option

- *process-ID* ～ <unsigned integer> ((1-5))

Specifies the ID of the process to be terminated forcibly. To check the process ID, use the `pdls -d prc` command.

Rules

1. You can execute the `pdkill` command only when HiRDB is active.
2. Execute the `pdkill` command at each server machine.

Notes

1. You can check the result of the `pdkill` command using the `pdls` command (specifying `-d prc`).
2. Make sure that the specified process ID is correct. If an incorrect process is terminated forcibly, HiRDB may terminate abnormally.
3. Return code `0` for the `pdkill` command indicates normal termination, and return code `-1` indicates abnormal termination.

pdlistls (Display list definition information)

Function

The `pdlistls` command displays the list definition information.

Executor

HiRDB administrator

Format

```
pdlistls [-d {1|2}]
        [-u list-owner's-name]
        [-t list-base-table-owner's-name,list-base-table-name]
```

Options

- `-d {1|2}`

Specifies the list definition information to be displayed.

`-d` option	Information to be displayed					
	Values of system definitions*	List owner's name	Name of list	Number of lists	Base table's owner's name	Name of base table
1	D	D	—	D	—	—
2	D	D	D	—	D	D

D: Displayed

—: Not displayed

* Indicates the values of the `pd_max_list_users` and `pd_max_list_count` operands.

- `-u` *list-owner's-name* ～ <identifier> ((1-30))

To display only the specified list owner's list definition information, specify the name of the list owner with this option. If this option is omitted, the system assumes all list owners.

The specified list owner's name is treated as all uppercase letters unless it is enclosed in `\"`.

- `-t` *list-base-table-owner's-name*, *list-base-table-name*

To display only the specified list base table's list definition information, specify the list

base table's owner's name and list base table's name with this option.

A list base table's owner's name can be expressed as 1-8 characters and a list base table's name can be expressed as an identifier consisting of 1-30 characters. The specified list base table's owner's name and list base table's name are treated as all uppercase letters unless they are enclosed in `\"`. If the name contains a space, enclose it in `\"`.

Rules

1. You can execute the `pdlistls` command only when HiRDB is active.
2. Execute the `pdlistls` command on the server machine containing the single server or the server machine where the system manager is located.

Notes

1. Return code `0` for the `pdlistls` command indicates normal termination, and return code `8` indicates abnormal termination.

Output format

- `-d 1` specified

```
 STATE OD LIST  DEFINITION
 #        OWNER NAME    LIST COUNT
[aa...a      bb...b          cc...c]
           :
[NOT FOUND dd...d]
pd_max_list_users : ee...e
pd_max_list_count : ff...f
```

Explanation

aa...a

Sequence number displayed

bb...b

List owner's name

cc...c

Number of segments in the list

dd...d

Information that was not found

`-t` option specified: `TABLE`

`-u` option specified: `USER`

ee...e

Value of the pd_max_list_users operand in the system definitions (for the single server or dictionary server)

ff...f

Value of the pd_max_list_count operand in the system definitions (for the single server or dictionary server)

- -d 2 specified

```
STATE OD LIST  DEFINITION
#       OWNER NAME  LIST NAME  BASE TABLE OWNER  BASE TABLE
NAME
[aa...a      bb...b          cc...c          dd...d                    ee...e]
                :
[NOT FOUND ff...f]
pd_max_list_users : gg...g
pd_max_list_count : hh...h
```

Explanation

aa...a

Sequence number displayed

bb...b

List owner's name

cc...c

Name of the list

dd...d

List base table's owner's name

ee...e

List base table's name

ff...f

information that was not found

-t option specified: TABLE

-u option specified: USER

gg...g

Value of the pd_max_list_users operand in the system definitions (for the single server or dictionary server)

hh...h

Value of the `pd_max_list_count` operand in the system definitions (for the single server or dictionary server)

pdlogadpf (Allocate log file)

Function

The pdlogadpf command allocates during HiRDB operation a physical file to a file group specified in the HiRDB system definition. The physical file must have been created with the pdloginit command before the pdlogadpf command is executed.

Once the pdlogadpf command is executed, the file group becomes available to the active HiRDB system when it is opened by the pdlogopen command.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogadpf -d spd [-s server-name] -g file-group-name
          -a primary-physical-file-name [-b secondary-physical-file-name]
```

HiRDB/Parallel Server

```
pdlogadpf -d spd -s server-name -g file-group-name
          -a primary-physical-file-name [-b secondary-physical-file-name]
```

Options

- -d spd

Specifies that a physical file is to be allocated as a synchronization point dump file.

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group.

- -g *file-group-name* ～ <identifier> ((1-8))

Specifies the name of the file group to which the physical file is to be allocated. This file group name must have been specified in a single server definition, front-end server definition, back-end server definition, or dictionary server definition.

- -a *primary-physical-file-name* ～ <pathname> ((up to 167 characters))

Specifies the absolute path name of the primary physical file that is to be allocated to the file group. Specify the physical file name in the format *HiRDB-file-system-area-name**HiRDB-file-name*.

- -b *secondary-physical-file-name* ～ <pathname> ((up to 167 characters))

Specifies the absolute path name of the secondary physical file that is to be allocated to the file group. Specify the physical file name in the format

*HiRDB-file-system-area-name**HiRDB-file-name*.

This option is required when dual synchronization point dump files are used. If dual synchronization point dump files are not used, specifying this option will result in an error.

If you use dual synchronization point dump files and specify both the `-a` and the `-b` options, and if an allocation error occurs on either the primary or secondary file, a command error results.

Rules

1. The `pdlogadpf` command can be executed only while HiRDB is active.
2. The `pdlogadpf` command must be executed at the server machine containing the single server or the server machine where the system manager is located.

Notes

1. Executing the `pdlogadpf` command for a file group to which a physical file has already been allocated results in an error.
2. The result of the `pdlogadpf` command can be determined from the command's return code, an error message (if any), or the message log. Return code `0` for this command indicates normal termination, `4` indicates abnormal termination, and `8` indicates abnormal termination (such as invalid option or `rsh` error).

Examples

Allocate a physical file to a file group specified in the system definition.

- Name of file group

  `spdfgf15`

- Name of physical file
  `c:\sysarea\rdsys011\spdf15`

HiRDB/Single Server

```
pdlogadpf -d spd                                  1
          -g spdfgf15                             2
          -a c:\sysarea\rdsys011\spdf15           3
```

Explanation

1. Type of log file: `spd` (synchronization point dump file)
2. Name of file group: `spdfgf15`
3. Name of physical file: `c:\sysarea\rdsys011\spdf15`

HiRDB/Parallel Server

```
pdlogadpf -d spd                                  1
```

```
-s bes1                                  2
-g spdfgf15                              3
-a c:\sysarea\rdsys011\spdf15            4
```

Explanation

1. Type of log file: `spd` (synchronization point dump file)
2. Name of server corresponding to the file group: `bes1`
3. Name of file group: `spdfgf15`
4. Name of physical file: `c:\sysarea\rdsys011\spdf15`

pdlogatul (Control the automatic log unloading facility)

Function

The `pdlogatul` command controls the automatic log unloading facility as follows:

- Restarting the automatic log unloading facility that was terminated during HiRDB operation
- Cancelling the automatic log unloading facility during execution
- Displaying the status of the automatic log unloading facility

For details about the automatic log unloading facility, see the *HiRDB Version 8 System Operation Guide*.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogatul -d sys [-s server-name] [{-i | -b | -t[-w]}]
```

HiRDB/Parallel Server

```
pdlogatul -d sys -s server-name [{-i | -b | -t[-w]}]
```

Options

- `-d sys`

Specifies that the system log files are to be manipulated.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server to be manipulated.

- `-i`

Specifies that the status of the automatic log unloading facility is to be displayed.

- `-b`

Specifies that the automatic log unloading facility is to be restarted.

- `-t`

Specifies that the automatic log unloading facility is to be cancelled.

- `-w`

If the `-t` option is specified to cancel the automatic log unloading facility, this option

specifies that the command termination is to be delayed until the current log unloading process is completed. The `-w` option takes effect only when the `-t` option is specified. If you specify the `-t` option, but omit the `-w` option, the system cancels the automatic log unloading facility immediately after the `pdlogatul` command is executed, forcibly terminating the ongoing unloading process.

When this option is specified, completion of unloading process is delayed by about nine hours (32768 seconds). If the unloading process is not completed within this period of time, the system cancels the automatic log unloading facility, in which case the unloading process will be terminated forcibly.

By specifying this option, you can eliminate the incomplete unload log files (which usually result from cancellation of the automatic log unloading facility) and input all the current unload log files with `pdrstr`.

Rules

1. You can execute the `pdlogatul` command while HiRDB is active. If you specify the `-i` or `-t` option, you can execute the command even during the HiRDB termination processing (during the execution of the `pdstop` command).
2. Execute the `pdlogatul` command at the server machine containing the single server or the server machine where the system manager is located.
3. If there is a current log generation, its log file group information is output.
4. If the `pdlogatul -t -w` command terminates with return code `4`, there may be an incomplete unload log file. In these cases, check the nature of the error on the basis of the output error message, take an appropriate action, and re-execute the `pdlogatul -b` command. Then, execute the `pdlogatul -t -w` command.

Notes

1. The following shows the `pdlogatul` command's return code:

   `0`: Normal termination

   `4`: One of the following:

   - If the `-t` and `-w` options were specified, the system forcibly terminated unloading process and cancelled the automatic log unloading facility because the unloading process was not completed within the waiting period (32768 seconds).
   - If the `-t` and `-w` options were specified, the system cancelled the automatic log unloading facility while monitoring the completion of unloading process for one of the following reasons:
   - None of the directories for unload log files was available.
   - The name of the unload log file was duplicated.

- There were not enough resources (such as a memory shortage or too many active processes)

8: Command error

Output format

```
HOSTNAME : aa...a(bb...b)
SERVER_NAME : cc...c
AUTO_LOG_UNLOAD  NOW_UNLOAD_LOG_GROUP  CREATE_DIR
                 d                     ee...e       ff..f
CURRENT LOG GENERATION INFO.
LOG_GROUP  GEN_NO. SERVER_RUN_ID  RUN_ID  UNLOAD_FILE_NAME
gg...g     hh...h  ii...i         jj...j  kk...k
```

Explanation

aa...a

Host name

bb...b

The time pdlogatul was executed (in the format *hour:minute:second*)

cc...c

Server name

d

Status of the automatic log unloading facility:

ACTIVE: Active

STOP: Inactive

STOPPING: Waiting for unloading process during termination

-: Automatic log unloading facility unavailable

ee...e

File group name for the system log file that is currently unloaded

If the automatic log unloading facility is not active, **** is displayed.

ff..f

Directory name for the unload log file in use

gg...g

File group name for the current generation of system log files

hh...h

Generation number of the current generation of system log

ii...i

Run ID of the current generation of server

jj...j

Run ID of the current generation of system log

kk...k

Name of the file created when the current generation of log was automatically unloaded

pdlogchg (Change status of log file)

Function

The pdlogchg command forcibly places a specified file group in unload completed status or extraction by HiRDB Datareplicator completed status. The status of the current file group cannot be changed.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogchg {-d sys [-s server-name] -g file-group-name [-R]
         |-z log-point-information-filename}
```

HiRDB/Parallel Server

```
pdlogchg {-d sys -s server-name [-u unit-identifier] -g file-group-name
[-R]
         |-z log-point-information-filename [-x host-name]}
```

Options

- -d sys

Specifies that the status of a system log file is to be changed.

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group.

- -u *unit-identifier* ～ <identifier> ((4 characters))

When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk on which the system log file whose status is to be changed was created.

When the applicable server and system manager are running, the -u option is ignored, if specified.

Specifying the -u option results in an error if any of the following is true:

- The specified unit does not belong to the target HA group for the server specified in the -s option.
- The server specified in the -s option does not belong to the HA group.
- The standby-less system switchover (effects distributed) facility is not in effect.

The following table describes whether or not the -u option is required:

Server status	Host containing the disk where file is created	Specification of -u option
Active	Host containing the running back-end server	Optional
Inactive	Host containing the primary back-end server	
	Other host	Mandatory

- -g *file-group-name* ～ <identifier> ((1-8))

Specifies the name of the file group whose status is to be changed.

- -z *log-point-information-filename* ～ <pathname>

Specifies that the status change is to be applied only to file groups older than the log point recorded in the specified log point information file.

One of the following files must be specified as the log point information file:

- File created by specifying the -z option in the pdcopy command
- File created by specifying the -z option in the pdrstr command

- -x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host of the server to which the log file group whose status is to be changed belongs.

When this option is omitted, the host where the command is entered is assumed.

- -R

Specifies that a file group is to be forcibly changed from extracting status to extraction completed status when HiRDB Datareplicator linkage is being executed. When this option is specified, only the extraction status is changed from extracting to extraction completed.

Rules

1. The pdlogchg command can be executed under the following conditions:
   - If pd_log_unload_check=N is specified in the system definitions, HiRDB is running, and the file group to be operated upon is open, the command cannot be executed without the -R option.
   - In all other cases, the command can be executed regardless of whether or not HiRDB is running (except when HiRDB is booting or terminating).
2. The pdlogchg command must be executed at the server machine containing the single server or the server machine where the system manager is located.

Notes

1. The following are the pdlogchg command's return codes:

   0: Normal termination

   4: Abnormal termination

   8: Abnormal termination (such as an invalid option or rsh error)

   12: Abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited)

2. The pdlogchg command must not be executed during HiRDB startup processing. HiRDB startup processing begins when the KFPS01800-I message is output and ends when the KFPS05210-I message is output. If the command is executed during that time, the system log file may not become effective in HiRDB. To make the system log file effective in such a case, its file group must be closed with the pdlogcls command after HiRDB startup processing is completed, then the file group must be opened with the pdlogopen command.

3. Because the pdlogchg command references the HiRDB system definition file, it may be impossible to change the file status if the information in the HiRDB system definition file for the active HiRDB that collected the system log does not match the information in the HiRDB system definition file referenced by the pdlogchg command.

4. Processing may terminate normally even if the pdlogchg command results in an error.

   Whether or not the processing has terminated normally can be determined by checking the termination code (whether or not it is 0) or by checking the status (whether or not the status has changed) with the pdlogls command.

5. The results of the pdlogchg command can be checked by the following methods:

   - HiRDB is active

     By executing the pdlogls command.

   - HiRDB is inactive

     By checking the return code set by the command, by checking any error messages, or by executing the pdlogls command.

6. A command execution error will result if the pdlogchg command is executed when HiRDB is not running and HiRDB is started before the execution of the command terminates. If this problem occurs, re-execute the pdlogchg command by checking the HiRDB operation status.

7. If pd_log_unload_check=N is specified on a server, the command cannot be executed without the -R option if the server has been individually terminated by

issuing the `pdstop -s` command.

8. If `pd_log_rpl_no_standby_file_opr=stop` is specified in the system common definition for HiRDB Datareplicator linkage, the `-R` option is executed in order to forcibly change the status of the file group when data linkage is stopped due to occurrence of an error that satisfies the following three conditions:

   - HiRDB was terminated forcibly because all system log files have been placed in extracting status.
   - The definition cannot be changed for HiRDB operational reasons.
   - An error was generated by the HiRDB Datareplicator and the HiRDB.

   Datareplicator cannot be activated. In such a case, the `pdlogchg` command with the `-R` option specified should be executed for all system log files to change the status from extracting to extraction completed. In this case, data may not be applicable to the target database, resulting in inconsistency between the extracted and target databases because some of the update information for the extracted database is lost. To avoid this, you need to re-create the target database before restarting the data linkage. For details about re-creating a database, see the *HiRDB Datareplicator Version 8 Description, User's Guide and Operator's Guide*.

pdlogcls (Close open log file)

Function

The `pdlogcls` command closes an open physical file that comprises a specified file group, so that it cannot be used by the active HiRDB. However, the following physical files cannot be closed:

- Current system log file or a system log file in overwrite disabled status
- Synchronization point dump file in writing status or overwrite disabled status

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogcls {-d {sys|spd} [-s server-name] -g file-group-name [-a][-b]
```

HiRDB/Parallel Server

```
pdlogcls {-d {sys|spd} -s server-name -g file-group-name [-a][-b]
```

Options

- `-d sys|spd`

Specifies the type of log file to be closed:

`sys`

System log file

`spd`

Synchronization point dump file

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group.

- `-g` *file-group-name* ～ <identifier> ((1-8))

Specifies the name of the file group containing the file to be closed.

- `-a`

When a system log file or synchronization point dump file is to be closed, specifies that the primary physical file is to be closed.

- `-b`

When a system log file or synchronization point dump file is to be closed, specifies that the secondary physical file is to be closed. If the system log file is not dualized, specifying this option results in a command error.

When the `-a` and `-b` options are both omitted, the following processing is performed:

- If the system log file is not dualized, the command closes physical file A in the same manner as when the `-a` option is specified
- If the system log file is dualized, the command closes physical files A and B in the same manner as when the `-a` and `-b` options are both specified

Rules

1. The `pdlogcls` command can be executed only while HiRDB is active.
2. The `pdlogcls` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. If the specified physical file is already closed, a warning message (`KFPS01283-W` or `KFPS02168-W`) is output.

Notes

1. The following are the `pdlogcls` command's return codes:

   `0`: Normal termination

   `4`: Abnormal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)
2. The `pdlogcls` command may result in an error if it is executed while log files are being swapped or being processed by another operation command.
3. The result of the `pdlogcls` command can be checked by the `pdlogls` command.
4. If `pd_log_remain_space_check=safe` is specified in the system definition and closing a system log file with the `pdlogcls` command results in a space shortage, the command issues the `KFPS01280-E` message (reason code: `712`) and results in an error. In such a case, increase the available space for the system log file and then re-execute the `pdlogcls` command. For details about using the facility for monitoring free area for system log files, see the *HiRDB Version 8 System Operation Guide*.

pdloginit (Initialize log file)

Function

The `pdloginit` command creates a log physical file in the HiRDB file system area and initializes it so that it can be used by the active HiRDB.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdloginit {-d sys [-s server-name]
           -f physical-filename -n HiRDB-records-count [-l record-length]
         | -d spd [-s server-name]
           -f physical-filename -n HiRDB-records-count}
```

HiRDB/Parallel Server

```
pdloginit {-d sys -s server-name[-u unit-identifier]
              -f physical-filename -n HiRDB-records-count
              [-l record-length] [-D]
            | -d spd -s server-name[-u unit-identifier]
              -f physical-filename -n HiRDB-records-count [-D]}
```

Options

- `-d sys|spd`

Specifies the type of log file to be initialized:

`sys`

System log file

`spd`

Synchronization point dump file

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group.

- `-f` *physical-filename* ～ <pathname> ((up to 167 characters))

Specifies the absolute pathname of the physical file to be initialized. The physical file name must be specified in the format *HiRDB-file-system-area-name**HiRDB-filename*. A HiRDB file name cannot be duplicated in the HiRDB file system area. If the specified HiRDB file has already been initialized, the command results in an error. A

name beginning with pl cannot be specified as a HiRDB file name.

■ -n *HiRDB-records-count* ～ <unsigned integer> ((12-2097152))

Specifies the number of records in the HiRDB file to be initialized.

For the purposes of this section, a *record* is an area with the following size in the HiRDB file system:

- For -d sys, the record length specified in the -l option
- For -d spd, 4096 bytes

Specification range

For the number of HiRDB records, specify a value in the following range:

- For -d sys

  Maximum value: 2147483648 ÷ (-l option value)

  Minimum value: ↑ ((pd_log_max_data_size value in the system definition) + 368 + 3 × 4096) ÷ (-l option value) ↑ + 1

- For -d spd

  12 to 262144

Guidelines for the specification value

For details about how to estimate the number of HiRDB records, see the *HiRDB Version 8 Installation and Design Guide*.

- For -d sys

  The estimated number of records should be the total number. The standard value for this option is the total number/number of system log files to be created × 1.2.

- For -d spd

  The standard value for this option is the estimated number of records × 1.2.

■ -l *record-length* ～ <unsigned integer> ((1024-4096))

Specify the record length for the HiRDB file to be initialized as a system log file. This option should be specified when it is necessary to create a system log file before modifying the system definition, or when it is desirable to explicitly specify a record length.

Notes

1. Any record length specified here should be the same as one that was specified in the pd_log_rec_leng operand in the system definition. The following values can be assigned:

- If the specified value is 1024, a record length of 1024 is assigned.
- If the specified value is 1025-2048, a record length of 2048 is assigned.
- If the specified value is 2049-4096, a record length of 4096 is assigned.

2. Any other values cause an error. The default is initialization using the value specified in the `pd_log_rec_leng` operand in the system definition. Files that are initialized using a record length different from that specified in the `pd_log_rec_leng` operand cannot be opened as system log files.
3. When specifying a record length, the following factors should be taken into consideration:
   - The shorter a record length, the greater the number of I/O operations required on the data file for handling large-size data. However, this will decrease the amount of free space due to the rounding up of space requiring to the HiRDB record size unit, thus improving the file utilization efficiency.
   - The longer a record length, the smaller the number of I/O operations required on the data file for handling large-size data. However, this will increase the amount of free space due to the rounding up of space requiring to the HiRDB record size unit, thus reducing the file utilization efficiency.
4. As a guide for the record length to be specified, use the average block length that is output to the system log file according to statistical information. It is recommended that the following values be specified in this option:
   - If the average block length is less than or equal to 1024 bytes, specify 1024.
   - If the average block length is 1025-2048 bytes, specify 2048.
   - If the average block length is greater than or equal to 2049 bytes, either specify 4096 bytes or use the default.
5. Note that specifying this option in initializing a file as a synchronization point dump file will cause an error. Similarly, specifying a record length less than the sector length for the HiRDB file system causes an error.

■ `-u` *unit-identifier* ～ <identifier> ((4 characters))

When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk on which the file is to be created.

When the applicable server is running, the `-u` option is ignored, if specified.

Specifying the `-u` option results in an error if any of the following is true:

- The specified unit does not belong to the target HA group for the server specified in the `-s` option.
- The server specified in the `-s` option does not belong to the HA group.

- The standby-less system switchover (effects distributed) facility is not in effect.

The following table describes whether or not the `-u` option is required:

Server status	Host containing the disk where file is created	Specification of -u option
Active	Host containing the running back-end server	Optional
Inactive	Host containing the primary back-end server	
	Other host	Mandatory

■ `-D`

Specifies that a unit or server is to be added by the `pdchgconf` command.

This option initializes the system log file or synchronization point dump file that is needed in order to add a unit or server. When this option is specified, the command will not check the server name.

If you execute the `pdloginit` command with this option specified, make sure that the command is executed at the unit where the system log file or synchronization point dump file is to be created.

Rules

1. The `pdloginit` command can be executed at any time, whether or not HiRDB is active (except during HiRDB startup or termination processing).
2. The `pdloginit` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. If the same environment variable as in the HiRDB directory is not set in the `PDDIR` environment variable during execution of the `pdloginit` command and a command error occurs, no message will be output.
4. In all files in each server, the record length for the system log file should be equal to the value specified in the `pd_log_rec_leng` operand of the HiRDB system definition.

Notes

1. When creating duplicate system log files for redundancy, the physical files for systems A and B comprising a log file group should be initialized so that they are equal in record length and the number of records.
2. If the system log file is to be dualized, physical files A and B should be created on separate hard disks, if possible, for purposes of error handling.
3. The result of the `pdloginit` command can be determined from the command's return code or error message, if any. The return codes are as follows:

`0`: Normal termination

`4`: Abnormal termination

`8`: Abnormal termination (such as invalid option or `rsh` error)

`12`: Abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited)

4. Files cannot be initialized in a HiRDB file system that has a sector length greater than 4096 bytes.

Examples

Example 1

Create system log files.

HiRDB/Single Server

```
pdloginit -d sys                                    1
          -f c:\sysarea\rdsys011\logf11             2
          -n 64                                     3
pdloginit -d sys                                    1
          -f c:\sysarea\rdsys012\logf12             2
          -n 64                                     3
```

Explanation

1. Type of log file to be initialized: `sys` (system log files)
2. Names of physical files: `c:\sysarea\rdsys011\logf11`, `c:\sysarea\rdsys012\logf12`
3. Number of records in each physical file: 64

HiRDB/Parallel Server

```
pdloginit -d sys                                    1
          -s bes1                                   2
          -f c:\sysarea\rdsys011\logf11             3
          -n 64                                     4
pdloginit -d sys                                    1
          -s bes1                                   2
          -f c:\sysarea\rdsys012\logf12             3
          -n 64                                     4
```

Explanation

1. Type of log file to be initialized: `sys` (system log files)
2. Name of server corresponding to the file group: `bes1`
3. Names of physical files: `c:\sysarea\rdsys011\logf11`, `c:\sysarea\rdsys012\logf12`

4. Number of records in each physical file: 64

Example 2

Create synchronization point dump files.

HiRDB/Single Server

```
pdloginit -d spd                                    1
          -f c:\sysarea\rdsys011\spdf11             2
          -n 64                                     3
pdloginit -d spd                                    1
          -f c:\sysarea\rdsys012\spdf12             2
          -n 64                                     3
```

Explanation

1. Type of log file to be initialized: `spd` (synchronization point dump file)
2. Names of physical files: `c:\sysarea\rdsys011\spdf11`, `c:\sysarea\rdsys012\spdf12`
3. Number of records in each physical file: 64

HiRDB/Parallel Server

```
pdloginit -d spd                                    1
          -s bes1                                   2
          -f c:\sysarea\rdsys011\spdf11             3
          -n 64                                     4
pdloginit -d spd                                    1
          -s bes1                                   2
          -f c:\sysarea\rdsys012\spdf12             3
          -n 64                                     4
```

Explanation

1. Type of log file to be initialized: `spd` (synchronization point dump files)
2. Name of server corresponding to the file group: `bes1`
3. Names of physical files: `c:\sysarea\rdsys011\spdf11`, `c:\sysarea\rdsys012\spdf12`
4. Number of records in each physical file: 64

pdlogls (Display log-related file information)

Common function

Displays log-related file information.

Common format

`pdlogls -d` *target-display-type* [*target-display-type-specific-option*]

Common options

- `-d` *target-display-type*

Specifies the log-related file to be displayed.

`spd`

Synchronization point dump files

`sys`

System log files

Common note

1. The following are the `pdlogls` command's return codes:

   `0`: Normal termination

   `4`: Abnormal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)

   `12`: Abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited)

pdlogls [-d spd] (Display synchronization point dump file information)

Function

The `pdlogls` command displays information about synchronization-point dump files.

Executor

HiRDB administrator

Format

HiRDB/HiRDB/Single Server

```
pdlogls -d spd [-s server-name] [-g file-group-name] [-e] [-l]
```

HiRDB/ Parallel Server

```
pdlogls -d spd -s server-name [-g file-group-name] [-e] [-l]
```

Option

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server whose information is to be displayed.

- `-g` *file-group-name* ～ <identifier> ((1-8))

Specifies the name of a file group whose information is to be displayed. If you omit this option, the command displays information about all file groups for the specified server.

- `-e`

Specifies that information about element files is to be displayed. If you omit this option, the command does not display information about element files.

- `-l`

Specifies that the maximum skip count for effective synchronization point dumps is to be displayed. If you omit this option, the command does not display the maximum skip count for effective synchronization point dumps. 0 is displayed if the skipped effective synchronization point dumps monitoring facility is not used.

Rules

1. The `pdlogls` command with the `-d spd` option can be executed only when HiRDB is running.
2. The `pdlogls` command should be executed on a server machine with a single server or the system manager.

Notes

1. Executing the pdlogls command during a file group status change (during swapping or when an operation command for manipulating a file group is being processed) may result in incorrect information being displayed.
2. Each record represents a 4096-byte area in the HiRDB file system area.

Output format

```
HOSTNAME: aa...a(bbccdd)
pd_spd_syncpoint_skip_limit: ee...e
Server  Group   Generation Status Log     Log
name            number           group   block
ff..f    gg...g   hh...h       [i]    jj...j     kk...k
       Status Path
 Sys-A [l]    mm...m
 Sys-B [n]    oo...o
```

Explanation

aa...a

Host name (up to 32 characters)

bbccdd

Hour, minute, second (time the pdlogls command was executed)

ee...e

Maximum skip count for effective synchronization point dumps (decimal). If the skipped effective synchronization point dumps monitoring facility is not used, 0 is displayed.

ff..f

Server name (up to 8 characters)

gg...g

File group name (up to 8 characters)

hh...h

Generation number (8 hexadecimal digits)

i

Generation status

a

Not overwritable

`u`

Being written or overwritable

`r`

Reserved

jj...j

File group name of the system log read by HiRDB when HiRDB restarts (up to 8 characters)

kk...k

Block number of the log read by HiRDB when HiRDB restarts (8 hexadecimal digits)

l

Status of the primary physical file:

`o`: Open

`c`: Closed

`h`: Shutdown

mm...m

Path name of the primary physical file

n

Status of the secondary physical file:

`o`: Open

`c`: Closed

`h`: Shutdown

oo...o

Path name of the secondary physical file

pdlogls [-d sys] (Display system log file information)

Function

The pdlogls command displays information about system log files.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogls -d sys [-s server-name] [-g file-group-name] [-e]
```

HiRDB/Parallel Server

```
pdlogls -d sys -s server-name [-u unit-identifier] [-g file-group-name]
[-e]
```

Options

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server whose information is to be displayed.

- -g *file-group-name* ～ <identifier> ((1-8))

Specifies the name of a file group whose information is to be displayed. If you omit this option, the command displays information about all file groups for the specified server.

- -e

Specifies this option when displaying information about an element file. The default is not to display information about element files.

- -u *unit-identifier* ～ <identifier> ((4 characters))

When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk on which the system log file for which information is to be displayed was created.

When the applicable server and system manager are running, the -u option is ignored, if specified.

Specifying the -u option results in an error if any of the following is true:

- The specified unit does not belong to the target HA group for the server specified in the -s option.
- The server specified in the -s option does not belong to the HA group.

- The standby-less system switchover (effects distributed) facility is not in effect.

The following table describes whether or not the `-u` option is required:

Server status	Host containing the disk where file is created	Specification of -u option
Active	Host containing the running back-end server	Optional
Inactive	Host containing the primary back-end server	
	Other host	Mandatory

Rules

1. The `pdlogls` command with the `-d sys` option can be executed regardless of whether HiRDB is running. However, this command cannot be executed when HiRDB is starting or terminating.
2. The `pdlogls` command should be executed on a server machine with either a single server or the system manager.

Notes

1. Executing the `pdlogls` command during a file group status change (during swapping or when an operation command for manipulating a file group is being processed) may result in incorrect information being displayed.
2. If a `waiting for unload` condition is displayed even after unloading (status change), it is possible that a previously entered `pdlogunld` or `pdlogchg` command generated an error that prevents the updating of the memory status. Even if such a problem arises, the unloading processing itself should have terminated normally. If this happens, use the `pdlogcls` command to close the affected file group and then open it by using the `pdlogopen` command.
3. Any value exceeding the `pdlogadpf -d sys` operand specified in the system definition will be ignored when HiRDB is started, but it will be included in the result of the `pdlogls` command that is executing when HiRDB is terminated.

Output format

```
HOSTNAME : aa...a(bbccdd)
***** Off-line Information *****                                    1
Group  Type Server  Gen No. Status  Run ID  Block No. Ex-Status
ee...e  fff  gg...g  hh...h  ijklmno  pp...p  qq...q rr...r  JKLMNOPQ  2
     Element  Status  Record-count  Sys-A  Sys-B1  Record-size
      ss...s  tuvwxy  AA...A  BB...B  CDEFGH  CDEFGH  IIII  3
```

[1] This information is displayed only when the system log file is dualized.

Explanation

Item 1 is displayed when log file information about the offline status (the corresponding server is shut down) is to be output.[1]

Item 2 indicates the status of the file group.

Item 3 indicates the status of the element file.

aa...a

Host name (up to 32 characters)

bbccdd

Time when `pdlogls` command executed (*hour:minute:second*)

ee...e

File group name (up to 8 characters)

fff

File type (3 characters)

`sys`

System log file

gg...g

Server name (up to 8 characters)

hh...h

Generation number (up to 8 hexadecimal digits)

ijklmno

File group status (7 characters)

i

File group open state

- `o`: Open
- `c`: Close

If the file group being used is closed, use the `pdlogopen` command to open it.

j

File group status

- `c`: Current

- `s`: Standby
- `n`: Reserved

If the file group being used is reserved, use the `pdlogopen` command to set it in the standby status.

k

File group unloading status

- `a`: Waiting for unloading but extraction not complete

  (This includes logs that were used as "current" in the past but are no longer current because of swapping, and therefore need to be unloaded. This file group can also be a system log file on which the extraction of data linkage information by the HiRDB Datareplicator has not completed. Upon completion of the extraction process, the file group shifts into the extraction complete status.)

  Upon unloading using the `pdlogunld` command, the file group shifts into the unload complete status.

- `u`: Waiting for unloading but extraction not complete

  (This includes logs that were used as "current" in the past but are no longer current because of swapping, and therefore need to be unloaded.)

  Upon unloading using the `pdlogunld` command, the file group shifts into the unload complete status.

- `e`: Unloaded but extraction not complete

  (This file group may be a system log file on which the extraction of data linkage information by the HiRDB Datareplicator has not completed. Upon completion of the extraction process, the file group shifts into the extraction complete status.)

- `-`: Unloaded and extraction complete

l

Overwrite enabled/disabled status of the file group

- `d`: Overwrite disabled (contains a log required for HiRDB system recovery)

  If all system log files in the server are in this status, a new system log file should be added.

- `-`: Overwrite enabled (does not contain a log required for HiRDB system recovery)

m

Status of the file group in HiRDB

- `b`: Being used by `pdlogchg` or `pdlogunld` command, or being used in recovery processing
- `-`: Not being used by `pdlogchg` or `pdlogunld` command, nor being used in recovery processing

n

File group utilization history

`c`: The file group is in one of the following statuses; this is displayed only when HiRDB is inactive:

- File group currently being used as *current*
- File group in which control information in the log file still remains as is due to an error that occurred when the file group was previously used as *current*

`-`: The file group previously used as "current" and has correctly been processed.

o

Internal information used by the system

pp...p

Run ID (run ID of the log service when the file was used) (8 hexadecimal digits)

qq...q

Beginning block number (8 hexadecimal digits)

rr...r

Ending block number (8 hexadecimal digits)

ss...s

Element file name (up to 8 characters)

tuvwxy

Element file status (6 characters)

t

Element file open status

- `o`: Open (physical file comprising the element file is open)
- `c`: Closed (physical file comprising the element file is closed)

u

Whether or not the element file is usable

- `n`: Element file not composed of physical files that can be used while HiRDB is running
- `u`: Element file composed of physical files that can be used while HiRDB is running

v

Element file unloading status [2]

- `a`: Waiting for unloading but extraction not complete

  (Contains a log file that must be unloaded; or, is a system log file for which the extraction of data linkage information by the HiRDB Datareplicator is not completed. Upon completion of the extraction process, the element file shifts into the extraction complete status.)

  Upon unloading using the `pdlogunld` command, the element file shifts into the unload complete status.

- `u`: Waiting for unloading but extraction not complete

  (Contains a log that must be unloaded.)

  Upon unloading using the `pdlogunld` command, the element file shifts into the unload complete status.

- `e`: Unloaded but extraction not complete

  (This element file is a system log file on which the extraction of data linkage information by the HiRDB Datareplicator has not been completed. Upon completion of the extraction process, the element file shifts into the extraction complete status.)

- `-`: Unloaded and extraction complete

w

Internal information used by the system

x

Element file shutdown status

- `h`: Shut down (an element file that contains a physical file that has not output a log since an error occurred, either during output of a log to the physical file comprising the element file, or during an I/O operation involving a log file header)
- `-`: Not shut down (normal element file)

y

Whether or not element file can be read:

- `r`: The physical file comprising the element file cannot be read.
- `-`: The physical file comprising the element file can be read.

AA...A

Number of records used in the file (up to 8 hexadecimal digits) [3, 5]

Number of records in the HiRDB file system area to which a log was output

BB...B

Total number of records in the file (up to 8 hexadecimal digits) [4, 5]

Number of records in the HiRDB file system area to which a log was output

CDEFGH

Physical file status (6 characters)

C

Physical file open status

- `o`: Open (physical file being opened)
- `c`: Closed (physical file being closed)

D

Physical file status

- `c`: Current (If the status of the file group is "current," the file group is actually eligible for log output. If the status of the file group is not "current," an error occurred when the file group was "current" in the past. The file group has not been recovered from the error, and therefore status information still remains in the log file header. To use the file group, either perform an error recovery with the `pdlogunld` or `pdlogchg` command, or use the `pdloginit` command to initialize the file group.)
- `s`: Standby

E

Physical file unloading status [2]

- `a`: Waiting for unloading but extraction not complete

  (Contains a log that must be unloaded. This physical file can also be a system log file on which the extraction of data linkage information by the HiRDB Datareplicator has not been completed. Upon completion of

the extraction process, the physical file shifts into the extraction complete status.)

Upon unloading using the `pdlogunld` command, the physical file shifts into the unload complete status.

- `u`: Waiting for unloading but extraction not complete

  (Contains a log that must be unloaded.)

  Upon unloading using the `pdlogunld` command, the physical file shifts into the unload complete status.

- `e`: Unloaded but extraction not complete

  (This physical file is a system log file on which the extraction of data linkage information by the HiRDB Datareplicator has not completed. Upon completion of the extraction process, the physical file shifts into the extraction complete status.)

- `-`: Unloaded and extraction complete

F

Internal information used by the system

G

Physical file shutdown status

- `h`: Shut down (the physical file has not output a log since an error occurred, either during output of a log or during an I/O operation involving a log file header)
- `-`: Not shut down (can output a log)

H

Whether or not the physical file can be read

- `r`: Read disabled status (full or partial recovery fails because the log has been erased (initialized) and cannot be read. This condition can arise if an error occurs in the system log file, the error is resolved, and the physical file is re-created and opened.)
- `-`: Read enabled status

IIII

Record length of the file (decimal display)[6]

Record length in the HiRDB file system to which a log can be output

JKLMNOPQ:

File group extension status (8 characters)

J:

-: Always displayed.

K:

Internal information used by the system

L:

Internal information used by the system

M:

Internal information used by the system

N:

Internal information used by the system

O:

Internal information used by the system

P:

Internal information used by the system

Q:

Internal information used by the system

Notes

[1] The offline (with the corresponding server shutdown) status display (file groups, element files, physical files in groups A and B) is valid only in the unloaded condition.

[2] If the file group is unloaded, it is not necessary to execute the `pdlogunld` or `pdlogchg` command even if the element or physical file is waiting to be unloaded. This is because an error occurred during the execution of the `pdlogunld` or `pdlogchg` command, which prevented status information from being overwritten.

[3] A `0` display means that no log was output. If the element file is in a status other than "current" after full recovery, the number of used records indicated may be incorrect.

[4] The total number of records in a file is equal to the number of HiRDB file records which can be output as log blocks. This number can be determined as follows, based upon the record length specified in the `-l` option of the `pdloginit` command:

- `-l 1024`: number of records specified in `-n` option of `pdloginit` command `-13`
- `-l 2048`: number of records specified in `-n` option of `pdloginit` command `-7`
- `-l 2048`: number of records specified in `-n` option of `pdloginit` command `-4`

[5] A record represents an area of the following size on the HiRDB file system controlled by the Log Service:

- If `sys` is specified in the `-d` option of the `pdloginit` command, the record length specified in the `-l` option
- If `spd` is specified in the `-d` option of the `pdloginit` command, 4096 bytes

pdlogopen (Open log file)

Function

The `pdlogopen` command opens a closed physical file that comprises a specified file group, so that it can be used by the active HiRDB.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogopen {-d {sys|spd} [-s server-name] -g file-group-name [-a][-b]
```

HiRDB/Parallel Server

```
pdlogopen {-d {sys|spd} -s server-name -g file-group-name [-a][-b]
```

Options

- `-d sys|spd`

Specifies the type of log file to be opened:

`sys`

System log file

`spd`

Synchronization point dump file

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group.

- `-g` *file-group-name* ～ <identifier> ((1-8))

Specifies the name of the file group containing the file to be opened.

- `-a`

When a system log file or synchronization point dump file is being opened, this option specifies that the primary physical file is to be opened.

- `-b`

When a system log file or synchronization point dump file is being opened, this option specifies that the secondary physical file is to be opened. If the system log file or synchronization point dump file is not dualized, specifying this option results in a command error.

When the -a and -b options are both omitted, the following processing is performed:

- If the system log file is not dualized, the command opens physical file A in the same manner as when the -a option is specified
- If the system log file is dualized, the command opens physical files A and B in the same manner as when the -a and -b options are both specified

Rules

1. The pdlogopen can be executed only while HiRDB is active.
2. The pdlogopen command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. If the specified physical file is already open, a warning message (KFPS01283-W or KFPS02167-W) is output.

Notes

1. The following are the pdlogopen command's return codes:

   0: Normal termination

   4: Abnormal termination

   8: Abnormal termination (such as an invalid option or rsh error)
2. The pdlogopen command may result in an error if it is executed while the file group status is being changed (such as during file swapping or processing by another operation command).
3. The result of the pdlogopen command can be checked by the pdlogls command.

pdlogrm (Delete log file)

Function

The `pdlogrm` command deletes a log physical file from the HiRDB file system. An open physical file cannot be deleted.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogrm -d sys|spd [-s server-name] -f physical-filename [-u]
```

HiRDB/Parallel Server

```
pdlogrm -d sys | spd -s server-name [-X unit-identifier] -f
physical-filename [-u] [-D]
```

Options

- `-d sys | spd`

Specifies the type of log file to be deleted:

`sys`

System log file

`spd`

Synchronization point dump file

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group that consists of the physical file.

- `-f` *physical-filename* ～ <pathname> ((up to 167 characters))

Specifies the absolute pathname of the physical file to be deleted. The physical file name must be specified in the format *HiRDB-file-system-area-name**HiRDB-filename*.

- `-u`

Specifies that if the physical file specified with the `-f` option is a system log file, it is to be deleted forcibly even if it is in any of the following statuses:

- Unload wait status
- Extracting status with HiRDB Datareplicator linked

When this option is omitted, the system checks the unload status of the physical file; if the file is in unload wait status, the command results in an error.

This option is ignored if specified for a synchronization point dump file.

This option is used when normal deletion processing (pdlogrm command with -u not specified) cannot be executed for some reason. If the file still cannot be deleted by the pdlogrm command with -u specified, the pdfrm command can be used.

- -X *unit-identifier* ～ <identifier> ((4 characters))

When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk from which the file is to be deleted.

When the applicable server is running, the -X option is ignored, if specified.

Specifying the -X option results in an error if any of the following is true:

- The specified unit does not belong to the target HA group for the server specified in the -s option.
- The server specified in the -s option does not belong to the HA group.
- The standby-less system switchover (effects distributed) facility is not in effect.

The following table describes whether or not the -X option is required:

Server status	Host containing the disk where file is created	Specification of -X option
Active	Host containing the running back-end server	Optional
Inactive	Host containing the primary back-end server	
	Other host	Mandatory

- -D

Specifies that a unit or server is to be deleted by the pdchgconf command.

To delete a unit or server, the system deletes that unit's or server's system log files and synchronization point dump files. When this option is specified, the command will not check the server name.

If you are executing the pdlogrm command with this option specified, make sure that the command is executed at the unit that contains the system log files or synchronization point dump files to be deleted.

Rules

1. The pdlogrm command can be executed at any time, whether or not HiRDB is active (except during HiRDB startup or termination processing).

2. The `pdlogrm` command must be executed at the server machine containing the single server or the server machine where the system manager is located.

Note

1. The result of the `pdlogrm` command can be determined from the command's return code or error message, if any. The return codes are as follows:

   `0`: Normal termination

   `4`: Abnormal termination

   `8`: Abnormal termination (such as invalid option or `rsh` error)

   `12`: Abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited)

pdlogswap (Swap log files)

Function

The `pdlogswap` command swaps log files. If there is no file to be swapped in, the command results in an error.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogswap -d sys [-s server-name] [-w [-t
validation-wait-monitoring-time]]
```

HiRDB/Parallel Server

```
pdlogswap -d sys -s server-name [-w [-t
validation-wait-monitoring-time]]
```

Options

- `-d sys`

Specifies that system log files are to be swapped.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group whose files are to be swapped.

- `-w`

Specifies that termination of the command is to be delayed until the synchronization point dump that is obtained after the system log files are swapped has been validated.

If HiRDB is operated without unloading the system log, specifying this option to swap system log files prior to database backup provides the following advantages:

- If the system log files are to be placed in unload completed status after backing up the database, all the system log files used before backup processing can be placed in unload completed status.
- During database recovery, the number of system log files to be read by the database recovery utility can be minimized.

- `-t` *validation-wait-monitoring-time* ～ <unsigned integer> ((10-32768)) <<180>>

Specifies in seconds the amount of time for monitoring for validation of the

synchronization point dump when the -w option is specified (the -w option delays termination of the command until the synchronization point dump has been validated). This option is applicable only when the -w option is specified.

Rules

1. The pdlogswap command can be executed only while HiRDB is active.
2. The pdlogswap command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. Only one pdlogswap command can be executed at a time. However, while a pdlogswap command with the -w option specified is waiting for validation of a synchronization point dump, another pdlogswap command without the -w option specified can be executed.

Notes

1. The results of the pdlogswap command can be checked by the pdlogls command, the KFPS01280-E error message (reason code 105 or 116), or the return code after command execution. The return code can be obtained using echo %errorlevel%. Following are the values returned after command execution:

   0

   Terminated normally.

   4

   System log files were swapped, but command processing was terminated forcibly because the synchronization point dump was not validated within the validation wait monitoring time specified in the -t option.

   8

   Terminated abnormally.

2. When the -w option is specified and the synchronization point dump is not validated within the validation wait monitoring time specified in the -t option, the pdlogswap command is terminated forcibly.
3. If safe or warn is specified in the pd_log_remain_space_check operand and one of the following is applicable, the facility for monitoring free space for system log files may detect a space shortage in the system log file:
   - The pdlogswap command was executed when the size of the file group in the swappable target status and in the standby status was less than the size of the current file group.
   - More than one pdlogswap command was executed consecutively.

For details about the facility for monitoring free space for system log files, see the *HiRDB Version 8 System Operation Guide*.

pdlogsync (Collect a synchronization point dump)

Function

The pdlogsync command collects a synchronization point dump.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogsync -d sys [-s server-name] [-w [-t timeout]]
```

HiRDB/Parallel Server

```
pdlogsync -d sys -s server-name [-w [-t timeout]]
```

Options

- -d sys

Specifies that a synchronization point dump is to be acquired.

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server at which a synchronization point dump is to be collected.

- -w

Specifies this option if the termination of the pdlogsync command is to wait until the system point dump is enabled. This option allows the user to determine whether the synchronization point dump has been enabled, based upon a return code returned from the execution of the pdlogsync command. A return code of 0 indicates that the synchronization point dump has been enabled.

Specify this option when creating a batch file for making backups.

- -t *timeout* ～ <unsigned-integer> ((10-32768)) <<180>>

When the -w option is specified, specifies a timeout value (in seconds) for the pdlogsync command. The pdlogsync command terminates if a synchronization point dump is not enabled by the time specified in this option. However, the processing for enabling the synchronization point dump will continue. A message KFPS02183-I is output when a synchronization dump is enabled. However, this message will not be output if N is specified in the pd_spd_assurance_msg operand in the system definition.

Rules

1. The `pdlogsync` command can be executed only when HiRDB is running.
2. The `pdlogsync` command should be executed on a server machine with either a single server or the system manager.

Notes

1. The following are the `pdlogsync` command's return codes:

   `0`: Normal termination

   `4`: Abnormal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)

pdlogucat (Display unload log file information)

Function

The pdlogucat command displays information about an unload log file.

You use the pdlogucat command to check the contents of a created unload log file.

Executor

HiRDB administrator

Format

```
pdlogucat [-k {std|csv}] unload-log-file-name
```

Options

- -k {std|csv}

Specifies the output format for the information.

std

Output the information in the format of fixed fields with headers.

csv

Output the information in CSV format.

Command arguments

- *unload-log-file-name* ～ <path name>

Specifies the absolute path name of the unload log file.

If you are specifying an unload log file created in a HiRDB file system area, the maximum path name length is 167 characters.

Rules

1. You can execute the pdlogucat command whether or not HiRDB is active.
2. Execute the pdlogucat command at the server machine where the host containing the unload log file is located.

Notes

1. Do not execute the pdlogucat command for an unload log file that is being created by the pdlogunld command or by the automatic log unloading facility. If the pdlogucat command is executed under such circumstances, invalid

information may be displayed.

2. For the pdlogucat command, return code 0 indicates normal termination, and return code 8 indicates abnormal termination.

Output format

- When -k std is specified (output information is fixed-length)

```
Hostname: aa....aa(bbccdd)
File_type  Format_ver  System_id  Unit_id  Serv_name  Run_id   User_run_id  Server_name
Group     Svs_start_time      Gen_num   Unload_time         First_block_num  Buff_leng
Status
eee         fffff        gggg         hhhh       ii....ii    jjjjjjj   kkkkkkkk       ll....ll
mm....mm   nnnn/oo/pp  qq:rr:ss   tt....tt   uuuu/vv/ww  xx:yy:zz   AA....AA           BB....BB
CC....CC
```

Note

One line of header text, delimited by a carriage return, is followed by a similarly delimited line of file information.

If there is no information for an item, * is displayed.

If the version of the specified unload log file is earlier than 07-02, * is displayed for server run ID and file group name.

aa....aa

Host name (up to 32 characters)

bbccdd

Time the pdlogucat command was executed (*hour minute second*)

eee

File type

sys: System log file

fffff

Internal information used by the system

gggg

HiRDB identifier

hhhh

Unit identifier

ii....ii

Internal information used by the system

jjjjjjjj

System log run ID (hexadecimal)

kkkkkkkk

Server run ID (hexadecimal)

ll....ll

Server name (up to 8 characters)

mm....mm

File group name (up to 8 characters)

nnnn/*oo*/*pp qq*:*rr*:*ss*

Current system log file allocation time (*year*/*month*/*date hour*:*minute*:*second*)

tt....tt

Generation number (up to 8 hexadecimal characters)

uuuu/*vv*/*ww xx*:*yy*:*zz*

Unload log file creation start time (*year*/*month*/*date hour*:*minute*:*second*)

AA....AA

Start log block number (up to 8 hexadecimal characters)

BB....BB

Internal information used by the system

CC....CC

System log unload completion status

`completed`: Unload log file that has been created correctly

`failed`: Unload log file whose creation was cancelled

- When -k csv is specified (output information is variable-length)

aaa,*bbbbb*,*cccc*,*dddd*,*ee....ee*,*ffffffff*,*gggggggg*,*hh....hh*,*ii....ii*,*jjjjjjjj*,*kk....kk*,
lllllll,*mm....mm*,*nnnnnn*,*o*

Note

If there is no information for an item, that item is simply omitted. For example, if the system log run ID (*ffffffff*) is not available, the display is as follows

aaa,*bbbbb*,*cccc*,*dddd*,*ee....ee*, ,*gggggggg*,...

aaa

File type

`sys`: System log file

bbbbb

Internal information used by the system

cccc

HiRDB identifier

dddd

Unit identifier

ee....ee

Internal information used by the system

ffffffff

System log run ID (hexadecimal)

gggggggg

Server run ID (hexadecimal)

hh....hh

Server name (up to 8 characters)

ii....ii

File group name (up to 8 characters)

jjjjjjjj

Current system log file allocation time (Julian seconds, in hexadecimal)

kk....kk

Generation number (up to 8 hexadecimal characters)

lllllll

Unload log file creation start time (Julian seconds, in hexadecimal)

mm....mm

Start log block number (up to 8 hexadecimal characters)

nnnnnn

Internal information used by the system

o

System log unload completion status:

`0`: Unload log file that has been created correctly

`1`: Unload log file whose creation was cancelled

pdlogunld (Unload log file)

Function

The `pdlogunld` command unloads a specified file group to a specified output destination file and places the file group in unload completed status.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdlogunld -d sys [-s server-name] -g file-group-name
                 [-o output-destination-filename] [-n] [-f]
                 [-k output-destination-file-type]
                 [-i output-destination-file's-initial-size
                     [,extension-size]]
```

HiRDB/Parallel Server

```
pdlogunld -d sys -s server-name [-u unit-identifier] -g
file-group-name
                 [-o output-destination-filename] [-n] [-f]
                 [-k output-destination-file-type]
                 [-i output-destination-file's-initial-size
                     [,extension-size]]
```

Options

- `-d sys`

Specifies that a system log file is to be unloaded.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the file group.

- `-g` *file-group-name* ～ <identifier> ((1-8))

Specifies the name of the file group containing the file to be unloaded.

- `-o` *output-destination-filename* ～ <pathname> ((up to 167 characters when `-k hirdb` is specified))

Specifies the name of the unload log file at the destination. This file is created in the host containing the server whose name is specified in the `-s` option. When this option is omitted, the standard output is assumed.

If you specify `-k hirdb`, specify the name of the destination file in the format

*HiRDB-file-system-area-name**HiRDB-filename*.

- `-n`

Specifies that the unload waiting status is to be retained without changing the file group status. When this option is specified, multiple unload files can be created, because the same generation of files can be unloaded one after another.

- `-f`

Specifies that the file is to be unloaded without checking the file group status. Specifying this option enables the current file group or an unload-completed file group to be unloaded. In this case, the status of the file group remains unchanged.

- `-k` *output-destination-file-type*

If the `-o` option is specified, this option specifies the type of unload log file at the destination. When this option is omitted, the system assumes `regular`.

`regular`

Outputs to a file.

`hirdb`

Outputs to a HiRDB file. Specify this option if you are using the system switchover facility.

If you specify `-k hirdb`, you must have created an HiRDB file system area in advance with the `pdfmkfs` command (`-k UTL` or `NUTL` specified).

- `-i` *output-destination-file's-initial-size*`[`,*extension-size*`]` ～ <unsigned integer> ((1-1,048,574)) <<100>>

If you specified `-k hirdb`, use this option to specify the destination file's initial size and extension size (used when the initial size is exceeded) in megabytes (MB).

When this option is omitted, the system assumes 100 MB for both initial and extension sizes.

Rules

1. Usually there is no need to specify this option because a default value of 100 MB is sufficient for the maximum size of unload log file (the maximum size of system log file is 1 gigabyte (GB)). If the system log file is smaller than 100 MB, specify the actual size in this option.
2. For the initial size, specify a value that is less than the value of the `-n` option used when the HiRDB file system area was created with the `pdfmkfs` command.
3. You cannot extend the size of the destination file when:
   - The number of extensions exceeds the specified value for the HiRDB file system area.

- The number of extensions exceeds the maximum extension count permitted for each HiRDB file (23).

■ `-u` *unit-identifier* ～ <identifier> ((4 characters))

When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk on which the system log file to be unloaded was created.

When the applicable server and system manager are running, the `-u` option is ignored, if specified.

Specifying the `-u` option results in an error if any of the following is true:

- The specified unit does not belong to the target HA group for the server specified in the `-s` option.
- The server specified in the `-s` option does not belong to the HA group.
- The standby-less system switchover (effects distributed) facility is not in effect.

The following table describes whether or not the `-u` option is required:

Server status	Host containing the disk where file is created	Specification of -u option
Active	Host containing the running back-end server	Optional
Inactive	Host containing the primary back-end server	
	Other host	Mandatory

Rules

1. The `pdlogunld` command can be executed under the following conditions:
   - If `pd_log_unload_check=N` is specified in the system definition, HiRDB is running, and the file group to be operated upon is open, execution of this command requires the specification of the `-f` option.
   - In all other cases, this command can be executed regardless of whether HiRDB is running (except when HiRDB is starting or terminating).
2. The `pdlogunld` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. The current file group or an unloaded-completed file group cannot be unloaded unless the `-f` option is specified.

Notes

1. The following are the `pdlogunld` command's return codes:

   `0`: Normal termination

4: Abnormal termination

8: Abnormal termination (such as an invalid option or rsh error)

12: Abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited)

2. A swappable file group unloaded with the -f option specified may become the current file group during unload processing. In this case, the command results in an error. Therefore, a swappable file group should first be closed by the pdlogcls command before it is unloaded by the pdlogunld command with the -f option specified.
3. The pdlogunld command must not be executed during HiRDB startup processing. HiRDB startup processing begins when the KFPS01800-I message is output and ends when the KFPS05210-I message is output. If the command is executed during this period, the swapped in system log file may not become effective in HiRDB. To make this system log file effective, the corresponding file group must be closed with the pdlogcls command after HiRDB startup processing is completed, then the file group can be opened with the pdlogopen command.
4. Because the pdlogunld command references the HiRDB system definition file, it may be impossible to unload the specified file group if the information in the HiRDB system definition file for the active HiRDB that collected the system log does not match the information in the HiRDB system definition file referenced by the pdlogunld command.
5. The result of the pdlogunld command can be checked by the following methods:
   - When HiRDB is active

     pdlogls command
   - When HiRDB is not active

     Completion message
6. A command execution error results if the pdlogunld command is executed when HiRDB is not running and one of the following conditions arises before the command terminates:
   - HiRDB starting
   - HiRDB online
   - HiRDB shutting down

   If this happens, any unloaded log files that are created cannot be used for database recovery processing. If this problem occurs, re-execute the pdlogunld command by checking the operating status of HiRDB.

7. In a server specifically shut down by using the `pdstop -s` command, the `pdlogunld` command cannot be executed without the `-f` option specification if `pd_log_unload_check=N` is specified for that server.
8. When an unload log file created in a HiRDB file system area by the `pdlogunld` command is migrated to another HiRDB file system area by the `pdfbkup` or `pdfrstr` command, the record length of the unload log file must be a multiple of the sector length of the HiRDB file system area that is used during the `pdfrstr` command's execution. The following table presents the relationship between the record length of an unload log file and the sector length of the HiRDB file system area that is used during execution of the `pdfrstr` command execution:

Record length of unload log file (bytes)	Sector length of the HiRDB file system area used during pdfrstr command execution (bytes)		
	1,024	2,048	4,096
1,024	Y	N	N
2,048	Y	Y	N
4,096	Y	Y	Y

Legend:

Y: `pdfrstr` command can be executed.

N: `pdfrstr` command cannot be executed.

You can use the `pdfls` command to check the record length of an unload log file. To check the current sector length, use the `pdfstatfs` command (the sector length of a HiRDB file system area is the value specified in the `-s` option during execution of the `pdsfmkfs` command).

pdls (Display HiRDB system status)

Common function

The pdls command displays the status of the HiRDB system for a specified display object.

Common executor

HiRDB administrator

Common format

HiRDB/Single Server

```
pdls [-d display-object-type [display-object-type-options]]
```

Note

The specifiable options depend on the display object type.

HiRDB/Parallel Server

```
pdls [-x host-name[,host-name]...]
     [-d display-object-type [display-object-type-options]]
```

Common options

- -x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host of the server for which status is to be displayed.

When this option is omitted, the items for the specified display object type are displayed for all servers in the system. If -d aud is specified, you cannot specify the -x option.

- -d *display-object* ～ <<svr>>

Specifies the type of object whose status is to be displayed:

act

User status

aud

Status of audit trail files

ha

System status when system switching facility is used

lck

Server lock status

`mem`

Information about common server memory

`prc`

Server processing status

`rpc`

Displays server communication control information

`rpl`

HiRDB Datareplicator linkage status

`scd`

Server scheduling status

`stj`

Displays whether types of statistical information output by unit and server are specified

`sts`

Status file status

`svr`

Unit and server status

`trn`

Server transaction status

Common rules

1. When `aud`, `ha`, `rpl`, or `ust` is specified in the `-d` option, you can execute the `pdls` command whether or not HiRDB is active. The `pdls` command with any other option specified can be executed only when HiRDB is running.
2. You can execute the `pdls` command with `-d ha`, `lck`, `svr`, `trn`, or `ust` specified from any server machine. Otherwise, this command must be executed at the server machine that contains the single server or where the system manager is located.
3. An option that is not applicable to the specified display object is ignored, if specified.

Common note

1. When the standby-less system switchover (effects distributed) facility is used, the server switched over to another unit is treated as the target unit's server.

pdls [-d act] (Display user status)

Function

Displays user status.

Format

```
pdls -d act [-s server-name]
```

Option

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies that the status of users in the specified server is to be displayed. When this option is omitted, the status of users associated with all servers in the HiRDB system is displayed.

Notes

1. Return code `0` for the `pdls -d act` command indicates normal termination, and return code `8` indicates abnormal termination. When the return code is `8`, see the previous message and eliminate the cause of the error.

Output format

```
HOSTNAME : aa...a(bbccdd)
SVID     PID      ACTID                LOCK-STATE
ee...e   ff...f    gggg-hhhh-ii...i     j
```

Explanation

aa...a

Host name (up to 32 characters)

bbccdd

Time when `pdls` command was executed (*hour:minute:second*)

ee...e

Server name (up to 8 characters)

ff...f

Process ID (up to 5 decimal digits)

`0` is displayed in the case of user information recovered by a unit restart.

gggg-hhhh-ii...i

User identification number (up to 16 decimal digits)

j

Lock hold status of `UNTIL DISCONNECT` specification:

`Y`

Lock is held

`N`

Lock is not held

pdls [-d aud] (Display the status of audit trail files)

Function

The `pdls -d aud` command displays the status of audit trail files.

Format

```
pdls -d aud [-u unit-identifier] [-U authorization-identifier]
```

Options

- `-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit whose information is to be displayed.

The `-u` option is applicable to a HiRDB/Parallel Server. When the `-u` option is omitted, the command displays the information for all units.

- `-U` *authorization-identifier*

Specifies the auditor's authorization identifier. When this option is specified, a message requesting entry of a password is displayed.

This option is effective when the `pdls -d aud` command is executed during HiRDB operation. When HiRDB is not running, you can execute this command without setting this option or the `PDUSER` environment variable.

Rules

1. When HiRDB is running, only the auditor can execute the `pdls -d aud` command.
2. The command uses the authorization identifier specified in this option and the password that is entered in the response-request message to connect to HiRDB and checks that this authorization identifier has the audit privilege.
3. You can specify this option only when you execute the `pdls -d aud` command in an environment in which you can enter a response to a response-request message.
4. When this option is omitted, the `pdls -d aud` command assumes the value set in the `PDUSER` environment variable during command execution. If this option is omitted and the `PDUSER` environment variable has not been set, the `pdls -d aud` command cannot be executed. If this option is omitted, the `PDUSER` environment variable must be set, as shown in the following example:

Example:

PDUSER=\"*authorization-identifier*\"/\"*password*\"

Notes

1. Return code 0 for the pdls -d aud command indicates normal termination, and return code 8 indicates abnormal termination.
2. The pdls -d aud command cannot be executed during unit startup or termination processing.
3. If the pd_aud_file_name operand has not been specified in the system definition or the audit trail file specified in the pd_aud_file_name operand is missing, executing the pdls -d aud command results in a command error.
4. If you execute the pdls -d aud command for a HiRDB/Parallel Server without specifying the -u option and an error occurs on the unit during command processing, the command displays the error message; if no error occurs on the unit, the command displays information about the audit trail files.
5. If you initialize a unit status file while the unit is inactive and then start the unit, the command displays - as the data loading status of the audit trail file generation that has been placed in shutdown status due to an error.

Output format

```
HOSTNAME      : aa...a(bbccdd)
UNITID        : eeee
Audit-Status  : ff...f
File-Name     : gg...g
Generation-Name    Status
hh...h             ijk
   :                 :
```

When information about all units is displayed, the command displays as many sets of the above information as there are units.

Explanation

aa...a:

Host name (1-32 characters)

bbccdd:

Time the pdls command was executed (*hour-minute-second*)

eeee:

Unit identifier (4 characters)

ff..f:

Audit trail acquisition status (1-8 characters)

`ACTIVE`: Acquiring

`INACTIVE`: Stopped

`********`: The audit trail acquisition status cannot be displayed because HiRDB is not active.

gg...g:

Name of the HiRDB file system area for audit trail files (absolute path name expressed in 1-117 characters)

If the name of the audit trail file exceeds 117 characters, the command displays the first 117 characters.

hh...h:

File name for each audit trail file generation (16 characters)

If the number of audit trail file generations is 1 or fewer, this information is not displayed.

ijk:

Status of each audit trail file generation (3 characters)

If the number of audit trail file generations is 1 or fewer, this information is not displayed.

i: Current status of the audit trail file:

`c`: Current. The current audit trail is being output to this generation. If audit trail acquisition has stopped, this file will be used as the current file the next time acquisition begins.

`s`: Standby. The current audit trail is not being output to this generation. In the case of an audit trail file that is in shutdown status due to an error, `s` is displayed while the unit is inactive.

j: Data loading status of the audit trail file:

`d`: Waiting for data loading. Data has not been loaded from the applicable audit trail file generation to the audit trail table.

`-`: Data loading has been completed. Data has been loaded from the applicable audit trail file generation to the audit trail table. In the case of an audit trail file that is in shutdown status due to an error, `-` is displayed while the unit is inactive.

k: Shutdown status of the audit trail file:

h: Shut down. An error occurred in the audit trail file and no audit trail has been output to the file since then.

-: Not shut down.

pdls [-d ha] (Display HiRDB system status)

Function

Displays the HiRDB system status when the system switching facility without IP address inheritance is used.

Format

```
pdls -d ha [{-u unit-identifier|-s server-name}] [-a]
```

Options

- `-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit for which system status is to be displayed. When this option is omitted, the system status is displayed for all units.

If you specify the identifier of a unit to which the standby-less system switchover (effects distributed) facility is applicable, only the host and guest BESs in that unit are subject to this status display.

- `-s` *server-name* ～ <identifier> ((1-8))

When the standby-less system switchover (effects distributed) facility is used, this option specifies the name of the server whose status is to be displayed. Specifying the server name of a unit to which the standby-less system switchover (effects distributed) facility is not applicable results in an error (`KFPS05041-E` message).

- `-a`

When the standby-less system switchover (effects distributed) facility is used, specifies that not only the execution status but also the detailed information (standby status, acceptable status, and running system start wait status) is to be displayed.

Rules

1. If the standby-less system switchover (effects distributed) facility is not being used, the `-a` option is ignored, if specified.
2. If the `-a` and `-u` options are both specified, detailed information about the host and guest BESs in the specified unit is displayed.
3. If the `-a` and `-s` options are both specified, detailed information is displayed for each unit in the HA group of the specified server.
4. When the `-a` option is omitted, information about the executing server is displayed.
5. To check the servers that are in acceptable status, specify the `-a` option.

6. The information to be displayed depends on the combination of the `-a`, `-u`, and `-s` options that are specified, as shown below:

Option			Active servers			Inactive servers
-a	-u	-s	Execution status	Standby or acceptable status	Running system start wait status	
Specified	Specified	Specified	—	—	—	—
		Omitted	D	D	D	D
	Omitted	Specified	D	D	D	D
		Omitted	D	D	D	D
Omitted	Specified	Specified	—	—	—	—
		Omitted	D	E	E	E
	Omitted	Specified	D	E	E	D*
		Omitted	D	E	E	D*

Legend:

D: Displayed

—: Not displayed

E: Results in an error

* Status of the inactive servers on all units in the HA group is displayed.

Notes

1. Return code `0` for the `pdls -d ha` command indicates normal termination, and return code `8` indicates abnormal termination.
2. If this command is specified when the system switching facility is not used or when the system switching facility is used but without IP address inheritance, a message is displayed and processing of the command terminates.

Output format

■ When the standby-less system switchover (effects distributed) facility is not used

```
UNITID  INITIAL-HOST  [STATUS]  RESERVED-HOST  [STATUS]
aaaa      bb...b        [ccc]      dd...d         [eee]
  :         :             :          :              :
```

Explanation

aaaa

Unit identifier (4 characters)

bb...b

Host name of current system (up to 32 characters)

ccc

Status of host of current system (3 characters):

In the monitor mode:

ONL: Executing (online)

SBY: On standby (including stopped)

STP: Both systems are stopped

In the server mode:

ONL: Executing (online)

SBY: On standby*

STP: Stopped

dd...d

Host name of spare system (up to 32 characters)

If the corresponding unit is not subject to system switchover or is using the system switchover facility with inheritance of IP addresses, ******* is displayed.

eee

Host status of spare system (3 characters):

In the monitor mode:

ONL: Executing (online)

SBY: On standby (including stopped)

STP: Both systems are stopped

***: Not subject to system switchover or using the system switchover facility with inheritance of IP addresses

In the server mode:

ONL: Executing (online)

SBY: On standby

STP: Stopped

***: Not subject to system switchover or using the system switchover facility with inheritance of IP addresses

* When the standby-less system switchover (1:1) facility is used, the alternate BES is placed in SBY status when its preparations have been completed, not when the alternate BES unit has started.

Supplementary description of the output format:

1. The following table shows the system statuses and display result:

System status		Display result			
		Monitor mode		Server mode	
Primary system	Secondary system	Primary system	Secondary system	Primary system	Secondary system
Stopped	Stopped	STP	STP	STP	STP
Stopped	Standby*	—	—	STP	SBY
Active	Stopped	ONL	SBY	ONL	STP
Active	Standby	ONL	SBY	ONL	SBY
Stopped	Active	SBY	ONL	STP	ONL
Standby	Active	SBY	ONL	SBY	ONL
Standby*	Stopped	—	—	SBY	STP

Legend:

—: Not applicable

* Although the secondary system cannot be started by itself, if Hitachi HA Toolkit Extension is used and the running system is terminated normally while both systems are active, the secondary system does not stop automatically. As a result, only the secondary system remains active.

2. If the names of the primary and secondary systems cannot fit on one line, the command displays the information about the secondary system's host on the next line. The following shows an example of this:

```
UNITID  INITIAL-HOST [STATUS]  RESERVED-HOST [STATUS]
aaaa    bb.....bb      [ccc]
        dd.....dd      [eee]
```

3. Figure 2-3 shows the correspondence between the mode (monitor or server) and the information displayed by the `pdls -d ha` command.

   Figure 2-3: Correspondence between the mode (monitor or server) and the information displayed by the pdls -d ha command

- When the primary system was used to start operations and the secondary system terminated normally after system switchover

  - System status changes

Primary system	Secondary system
Stopped	Stopped
Active	⋮
⋮	Standby
	⋮
System switchover	System switchover
Stopped	Active
⋮	⋮
	Stopped

  - Display by the `pdls -d ha` command

Monitor mode		Server mode	
Primary system	Secondary system	Primary system	Secondary system
STP	STP	STP	STP
ONL	SBY	ONL	⋮
⋮	⋮	⋮	SBY
			⋮
SBY	ONL	STP	ONL
⋮	⋮	⋮	⋮
STP	STP		STP

- When the primary system was used to start operations and the primary system terminated normally without executing system switchover

  - Change of system status

Primary system	Secondary system
Stopped	Stopped
Active	⋮
⋮	Standby
	⋮
Stopped	Stopped

  - Display by the `pdls -d ha` command

Monitor mode		Server mode	
Primary system	Secondary system	Primary system	Secondary system
STP	STP	STP	STP
ONL	SBY	ONL	⋮
⋮	⋮	⋮	SBY
			⋮
STP	STP	STP	STP

4. When the standby-less system switchover (1:1) facility is used, the terms primary system, secondary system, executing unit, and standby unit mean the following:

   Primary system: Normal BES unit

   Secondary system: Alternate BES unit

   Executing: Unit that is executing processing (normal BES unit that has not been switched over or alternate BES unit that has been switched over)

   Standby: Standby unit (alternate BES unit that has not been switched over or normal BES unit that has been switched over)

- When the standby-less system switchover (effects distributed) facility is used

```
UNITID  INITIAL-HOST [STATUS]  RESERVED-HOST [STATUS]
aaaa    bb...bb      [ccc]     dd...dd       [eee]
  :       :            :         :             :

SVID    UNIT  [STATUS] INITIAL-UNIT
ff..f   gggg  [hhh]    iiii
  :       :     :        :
```

Explanation

The values of *aaaa* through *eee* are the same as when the standby-less system switchover (effects distributed) facility is not used.

ff...f through *iiii* are sorted by the identifier of the unit that contains servers and are displayed for each server.

ff...f

Server name (up to 8 characters).

This is a server on the unit to which the standby-less system switchover (effects distributed) facility is applicable.

gggg

Identifier of the unit that contains the server (4 characters).

This is the identifier of the unit that contains server *ff...f*.

If the `-u` option is specified, this is the specified unit identifier.

If the `-a` option is specified but the `-u` option is omitted, this is the identifier of the unit whose status is to be displayed, among all units in the target HA group of server *ff...f*.

If the `-a` and `-u` options are both omitted, this is the identifier of the unit on which server *ff...f* is running.

If server *ff...f* is not running in any of the units of the HA group, `****` is displayed.

hhh

Server's status (3 characters):

`ONL`

Executing (online)

`SBY`

Standby status for a host BES; acceptable status for a guest BES

If the -a option is omitted, this information is not displayed.

STP

If the -a, -u, and -s options are omitted, this value means that the server is not running on any unit in the HA group.

If the -a option is specified, this value means that the server is stopped on the unit *gggg*.

WIT

Running system start wait status

If the -a option is omitted, this information is not displayed.

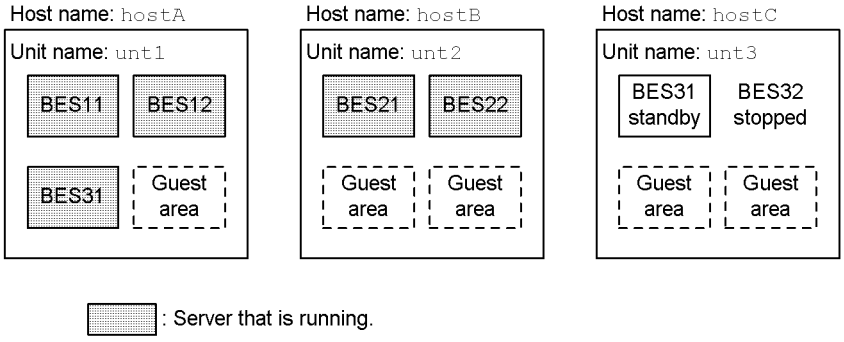
iiii

Identifier of the definition target unit (4 characters).

This is the unit identifier specified in the pdstart -u system definition for server *ff...f*.

Example of display when the standby-less system switchover (effects distributed) facility is used:

The following shows a configuration example when the standby-less system switchover (effects distributed) facility is used:



When the pdls -d ha command is executed in this configuration, the following information is displayed:

- When pdls -d ha is executed

```
SVID  UNIT  [STATUS]  INITIAL-UNIT
BES11 unt1  [ONL]     unt1
BES12 unt1  [ONL]     unt1
BES31 unt1  [ONL]     unt3
BES21 unt2  [ONL]     unt2
```

```
BES22 unt2  [ONL]     unt2
BES32 ****  [STP]     unt3
```

- When `pdls -d ha -a` is executed

```
SVID  UNIT  [STATUS]  INITIAL-UNIT
BES11 unt1  [ONL]     unt1
BES12 unt1  [ONL]     unt1
BES21 unt1  [SBY]     unt2
BES22 unt1  [SBY]     unt2
BES31 unt1  [ONL]     unt3
BES32 unt1  [STP]     unt3
BES11 unt2  [SBY]     unt1
BES12 unt2  [SBY]     unt1
BES21 unt2  [ONL]     unt2
BES22 unt2  [ONL]     unt2
BES31 unt2  [SBY]     unt3
BES32 unt2  [STP]     unt3
BES11 unt3  [SBY]     unt1
BES12 unt3  [STP]     unt1
BES21 unt3  [SBY]     unt2
BES22 unt3  [SBY]     unt2
BES31 unt3  [SBY]     unt3
BES32 unt3  [STP]     unt3
```

- When `pdls -d ha -u unt1` is executed

```
SVID  UNIT  [STATUS]  INITIAL-UNIT
BES11 unt1  [ONL]     unt1
BES12 unt1  [ONL]     unt1
BES31 unt1  [ONL]     unt3
```

- When `pdls -d ha -u unt1 -a` is executed

```
SVID  UNIT  [STATUS]  INITIAL-UNIT
BES11 unt1  [ONL]     unt1
BES12 unt1  [ONL]     unt1
BES21 unt1  [SBY]     unt2
BES22 unt1  [SBY]     unt2
BES31 unt1  [ONL]     unt3
BES32 unt1  [STP]     unt3
```

- When `pdls -d ha -s BES12` is executed

```
SVID  UNIT  [STATUS]  INITIAL-UNIT
BES12 unt1  [ONL]     unt1
```

- When `pdls -d ha -s BES12 -a` is executed

```
SVID  UNIT  [STATUS]  INITIAL-UNIT
BES12 unt1  [ONL]     unt1
BES12 unt2  [SBY]     unt1
BES12 unt3  [STP]     unt1
```

pdls [-d lck] (Display server lock control status)

Function

Displays server lock status.

Format

```
pdls -d lck [{-a [-s server-name] | -p}]
```

Options

- {-a [-s *server-name*] | -p}

Specifies whether both occupancy and wait information or the utilization factor of the tables for managing locked resources is to be displayed. When this option is omitted, only the wait information is displayed. If the -a and -p options are both specified, the -p option is effective.

-a

Specifies that both occupancy and wait information are to be displayed. When this option is omitted, only the wait information is displayed.

-s *server-name* ∼ <identifier> ((1-8))

Specifies the name of the server whose wait information only is to be displayed. When this option is omitted, information about all servers is displayed.

-p

Specifies that the utilization factor of the tables for managing locked resources is to be displayed.

Notes

1. Return code 0 for the pdls -d lck command indicates normal termination, and return code 8 indicates abnormal termination.

Output format

Options omitted or -a and -s options specified:

```
HOSTNAME : aa...a(bbccdd)
SVID          PID    TID KIND  RESOURCE
 WAIT  MODE  TRNBID              ACTID                LOCK-STATE  DPRIO
ee...e     ff...f gg...g hhhh   ii...i
jj...j kk     ll...l               mm...m-nn...n-oo...o p             qqq     . . .l
                        :
```

Explanation

`1` is displayed as many times as there are occupancy and wait information items.

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

ee...e

Server name (up to 8 characters).

ff...f

Process ID (32-bit mode: 5 decimal digits; 64-bit mode: 10 decimal digits)

gg...g

Thread ID (5 decimal digits).

hhhh

Resource type (4 hexadecimal digits).

ii...i

Resource name (28 hexadecimal digits).

jj...j

Wait time (5 decimal digits; in seconds).

`*****` is displayed for the occupancy information.

kk

Lock mode (2 characters).

ll...l

Transaction identifier (16 characters).

If `Y` is displayed for `p` (lock hold status of `UNTIL DISCONNECT` specification), this transaction branch may already have been terminated. Therefore, the `pdls -d trn` command may not be able to display the status, depending on the timing.

mmmm-nnnn-oo...o

User identification number (up to 16 decimal digits).

p

Lock hold status of `UNTIL DISCONNECT` specification:

`Y`

Lock is held

`N`

Lock is not held

qqq

Deadlock priority value (up to 3 decimal digits).

`-p` option specified:

```
HOSTNAME: aa...a(bbccdd)
SVID        TABLE-KIND  TOTAL  KIND  RATE
ee...e      RESOURCE    ff..f  gg...g  hh%                    1
:             :           :      :       :
```

Explanation

`1` is displayed as many times as there are servers.

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

ee...e

Server name (up to 8 characters).

ff..f

Maximum number of tables for managing locked resources that can be used (32-bit mode: 8 decimal digits; 64-bit mode: 10 decimal digits).

gg...g

Number of tables for managing locked resources currently in use (32-bit mode: 8 decimal digits; 64-bit mode: 10 decimal digits).

hh

Utilization factor of the tables for managing locked resources (%).

pdls [-d mem] (Display server shared memory status)

Function

Displays information about the shared server memory. The command displays the size of the shared memory being used in each unit corresponding to the host.

Format

```
pdls -d mem
```

Option

None

Notes

1. Return code `0` for the `pdls -d mem` command indicates normal termination, and return code `8` indicates abnormal termination.

Output format

```
HOSTNAME : aa...a(bbccdd)
SHM-ID  GET-SIZE  ACT-SIZE SHM-OWNER          POOL-ID
ee...e     ff..f     gg...g     hh...h            ii...i
  :          :         :          :                 :
```

Explanation

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

ee...e

Identifier of shared memory segment (10 decimal digits).

If the value exceeds 6 digits, the subsequent display will be shifted to the right by that number of digits.

ff..f

Allocation size (1-10 decimal digits) (bytes)

If the value exceeds 9 digits, the subsequent display will be shifted to the right by that number of digits.

gg...g

Amount of space being used (1-10 decimal digits) (bytes)

If the value exceeds 9 digits, the subsequent display will be shifted to the right by that number of digits.

hh...h

Attribute of the process that uses shared memory (up to 8 characters).

Type of shared memory		Value of hh...h
Shared memory for unit controller		MANAGER
Shared memory for global buffer	When the standby-less system switchover (effects distributed) facility is applicable to the unit	Unit name enclosed in parentheses
	When the standby-less system switchover (effects distributed) facility is not applicable to the unit	HiRDB server name using the corresponding shared memory
Shared memory for utility		UTILITY
Shared memory for security audit information buffer		AUDDEF

ii...i

Pool identifier (up to 16 characters).

Type of shared memory		Value of ii...i
Shared memory for unit controller		Not displayed
Shared memory for global buffer	When the standby-less system switchover (effects distributed) facility is applicable to the unit	Sequence number in the unit
	When the standby-less system switchover (effects distributed) facility is not applicable to the unit	Sequence number in the HiRDB server that uses the corresponding shared memory
Shared memory for utility		Unique character string
Shared memory for security audit information buffer		None

Note

The same value (allocation size) is always displayed for *ff...f* (allocation size) and *gg...g* (size being used).

pdls [-d prc] (Display server process status)

Function

Displays server processing status.

Format

```
pdls -d prc [{-a | -s server-name}] [-c]
```

Options

- {-a | -s *server-name*}

Specifies -a to display information about all servers in the HiRDB system (including servers used internally by HiRDB), or specifies -s and a server name to display information about the specified server only. When this option is omitted, information about all servers except for servers used internally by HiRDB is displayed.

-a

Specifies that the processing status of all servers in the HiRDB system is to be displayed, including servers used internally by HiRDB.

-s *server-name* ～ <identifier> ((1-8))

Specifies the name of a server whose processing status is to be displayed.

- -c

Specifies that the process's operating status (whether or not the process is scheduled to stop) is to be displayed.

Use this option to check the process's stop status after using the pdpfresh command to refresh the process.

Notes

1. Return code 0 for the pdls -d prc command indicates normal termination, and return code 8 indicates abnormal termination.

Output format

```
HOSTNAME:  aa...a(bbccdd)
STATUS PID    UID    GID   SVID  TIME   PROGRAM  C-PID      C-GRP
e      ff...f  gg...g  hh...h  ii...i  jjkkll  mm...m     nn...n(oo...o)  pp
:      :       :       :       :       :       :
```

Explanation

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

e

Server's activity status:

`D`

Server is engaged in startup or termination processing

`L`

Server is active

`C`

Process is stopping or scheduled to stop (displayed when the `-c` option is specified)

ff...f

Process ID (up to 10 decimal digits).

gg...g

User ID (up to 5 decimal digits).

hh...h

Group ID (up to 5 decimal digits).

ii...i

Server name (up to 8 characters).

jjkkll

Time the service request was accepted (*hour:minute:second*).

If the server is not connected to a UAP, `999999` is displayed.

mm...m

UAP identification information (up to 30 characters).

The UAP's identification name as specified in `PDCLTAPNAME` in the client environment definition is displayed.

For an object other than a server, `******` is displayed.

If `PDCLTAPNAME` is undefined, `Unknown` is displayed.

nn...n

Process ID of the client (UAP or utility) that was connected to the server process (up to 10 decimal digits).

This information is applicable only to the server process that was connected to the client; it is not displayed for a server process that was not connected to the client. This information is not displayed if the version of the client library connected to the UAP is earlier than 04-00.

oo...o

IP address of the client (UAP or utility) that was connected to the server process (up to 15 characters).

This information is applicable only to the server process that was connected to the client; it is not displayed for a server process that was not connected to the client. This information is not displayed if the version of the client library connected to the UAP is earlier than 04-00.

pp

Client group name (up to 2 characters) of the client connected to a server process. Only server processes that are connected to a client can be specified in this operand. Server processes that are not connected to a client are not displayed. Any client libraries linked to a UAP and having a version older than 04-00 are not displayed.

When using one character, specify a user-assigned client group name; when using two characters, specify a HiRDB-supplied client group name by client type. Following is a list of HiRDB-supplied client group names by client type:

Client group name	Client type
XA	X/Open XA interface
PC	PC client
WS	WS client
MF	Mainframe-based client

Notes

If a server process has just been activated, blanks and `******` may be displayed as the server name and the UAP identification information, respectively.

When the status of a HiRDB command process is being displayed, `0` is displayed as both the user ID and the group ID.

pdls [-d rpc] (Display server communication control information)

Function

Displays communication control information about a communication-established server. This command can be used when it is necessary to determine whether processing control is on the HiRDB side or on the client side if the client stops responding during UAP operation.

Format

```
pdls -d rpc [-a]
```

Options

- `-a`

Displays communication control information about all processes in the HiRDB system. This option can be used to determine details of communication control information.

Notes

1. Return code `0` for the `pdls -d rpc` command indicates normal termination, and return code `8` indicates abnormal termination.
2. Client information (process ID, IP address, UAP information, OS type, group name, XA interface usage status) is not displayed if the version of the client library that is linked with the UAP is earlier than 04-00.
3. If the `-a` option is specified, the client OS type, the client group name, and the XA interface usage status are displayed if the server type is `F` or `S`, and the client is connected.

Output format

Default option (UAP connected)

```
[aaaa]
  Client Information :
    PID       ADDR       NAME      OS    GROUP    XA
    bb...b    cc...c     dd...d    e     ff       g
  Server Information :
    PID     SVID    STATUS  SYS_EVENT USR_EVENT  CRITICAL(MASK)  SERVICE  SYS_CALL
    hh...h  ii...i  jj...j  0xkk...k  0xll...l   m(0xnn...n)     oo...o   0xtt...t
       :       :       :         :         :           :             :         :
[aaaa]
  Client Information :
  :
```

Default option (Utility connected)

```
[aaaa]
  Server Information :
    PID     SVID     STATUS   SYS_EVENT USR_EVENT    CLITICAL(MASK) SERVICE  SYS_CALL
   hh...h   ii...i    jj...j     0xkk...k  0xll...l      m(0xnn...n)    oo...o     0xtt...t
      :      :         :          :         :              :             :          :
[aaaa]
  Server Information :
  :
```

The -a option specified

```
HOSTNAME : ss...s
PID     SVID    STATUS   SYS_EVENT USR_EVENT CLITICAL(MASK) SERVICE OS GROUP XA  ..1
hh...h  ii...i  jj...j     0xkk...k  0xll...l     m(0xnn...n)    oo...o  e  ff  Y  ...2
CLTKIND SVRKIND CLTPID CLTADDR CLTNAME ACTID    SYS_CALL  .......................3
p       q       bb...b  cc...c  dd...d  0xrr...r  0xtt...t  .......................4
:        :        :       :       :         :        :
```

Explanation

Items 1-4 are displayed in one line each. Items 2 and 4 are displayed repetitively for each process.

aaaa

Item number

Bb...b

Process ID of the client connected to a server process (in decimal, up to 10 digits)

cc...c

IP address of the client connected to a server process (up to 15 characters)

dd...d

UAP identification information for the client connected to a server process (up to 30 characters)

Displays identification information about the UAP specified in the PDCLTAPNAME client environmental variable. If PDCLTAPNAME is not specified, the command displays Unknown.

e

Client operating system

W

Workstation-based client

`P`

PC-based client

`M`

Mainframe-based client

ff

Client group name (up to 2 characters) of the client connected to a server process.

When using one character, specify a user-assigned client group name; when using two characters, specify a HiRDB-supplied client group name by client type. Following is a list of HiRDB-supplied client group names by client type:

Client group name	Client type
XA	X/Open XA interface
PC	PC client
WS	WS client
MF	Mainframe-based client

g

Use of X/Open XA interface

`Y`

Used

`N`

Not used

hh...h

Server process ID (up to 10 decimal digits)

ii...i

Server name (up to 8 characters)

jj...j

Process status display (up to 12 characters)

`ACTIVE`

currently processing

SUSPEND (QUE)

processing suspended (service wait state: queued)

SUSPEND (CLT)

processing suspended (waiting for next service)

SUSPEND (SVR)

processing suspended (waiting for other messages)

kk...k

Display of process status details (26 characters)

Displays system internal code.

ll...l

Display of process status details (8 characters)

Displays system internal code.

mm...m

Critical status display

Y

Critical

N

Not critical

n

Critical information display (8 hexadecimal digits)

oo...o

Service name display (up to 32 characters)

Displays internal system information.

p

Client type

C

Client

U

Utility

Blank

Other than above

q

Server type (With `-a` option, displays a server type only when the client type is `C`.)

`F`

Front-end server (FES)

`B`

Back-end server (BES)

`D`

Dictionary server (DIC)

`S`

Single server (SDS)

rr...r

Activity ID (18 characters)

This is not displayed for a process that does not have an activity ID.

ss...s

Host name (up to 32 characters)

tt...t

Internal information used by the system (8 characters)

Example of information verification

SQL not responding

If the `STATUS` is `ACTIVE`, the server is processing.

To determine whether current control is with the client or the server:

This can be determined based on the information indicated in the `STATUS` display on the front-end server (single server). If the `STATUS` display on the front-end server is `SUSPEND (CLT)`, control is with the client.

pdls [-d rpl] (Display HiRDB Datareplicator linkage status)

Function

Displays the HiRDB Datareplicator linkage status. Also displays the extraction status of the extracted HiRDB Datareplicator system log in the system log file.

Format

```
pdls -d rpl [-j] [-u unit-identifier] [-s server-name]
```

Options

- `-j`

Specifies that the extraction status of the extracted HiRDB Datareplicator system log in the system log file is to be displayed. When this option is specified, the following information is displayed:

- Whether or not the HiRDB Datareplicator linkage facility is being used
- The extraction status of the system log from the system log file of the extracted HiRDB Datareplicator
- Internal information used by the system

When this option is omitted, the only information displayed is whether or not the HiRDB Datareplicator linkage facility is being used (the extraction status of the system log and the internal information are not displayed).

- `-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the unit identifier of a unit in order to display the extraction status of the extracted HiRDB Datareplicator system log for that unit. When this option is omitted, this information is displayed for all units in the system. When the `-j` option is omitted and this option is specified, whether or not the system and the specified unit are using the HiRDB Datareplicator linkage facility is displayed.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies a server name in order to display the extraction status of the extracted HiRDB Datareplicator system log for that server. When this option is omitted, this information is displayed for all servers in the system. When the `-j` option is omitted and this option is specified, whether or not the system and the unit containing the specified server are using the HiRDB Datareplicator linkage facility is displayed.

Notes

1. Return code `0` for the `pdls -d rpl` command indicates normal termination, and return code `8` indicates abnormal termination.

2. When the `-j` option is specified and the `-u` and `-s` options are omitted in a `pdls` command in which `-d rpl` is specified, the HiRDB Datareplicator linkage facility status is displayed for all servers of all units in the system.
3. A `pdls` command with `-d rpl` specified can be executed even if the server is inactive.

   However, an error results if the extracted HiRDB Datareplicator directory name is not specified in `pd_rpl_hdepath` in the system definition or if there is no link file for data linkage.

Output format

`-j` option omitted

```
SYSTEMID          :aaaa(bbccdd)                    1
Data Replication  :e                               1
UNITID            :ffff(bbccdd)                    2
Data Replication  :e                               2
```

1. If the `-u` option was specified for a HiRDB/Parallel Server, this information is displayed only when the specified unit is the server machine where the system manager is located.
2. In the case of a HiRDB/Parallel Server, this information is displayed as many times as there are units in the system. If the `-u` option was specified, this information is displayed only for the specified unit.

Explanation

aaaa

HiRDB identifier (4 characters)

bbccdd

Time (*hour:minute:second*) when the `pdls` command was entered

e

One character indicating whether or not the HiRDB Datareplicator linkage facility is being used at the HiRDB system and unit (`Y` is displayed, indicating that the HiRDB Datareplicator linkage facility is being used, even if there is no database to be linked to all back-end servers in the system and in the unit):

`Y`

HiRDB Datareplicator linkage facility is being used

`N`

HiRDB Datareplicator linkage facility is not being used

`*`

System linkage status cannot be displayed because the system manager is inactive (the status of each unit should be checked; if there is even one unit that is not using the HiRDB Datareplicator linkage facility, it will be shown that the HiRDB Datareplicator linkage facility is not being used by the system)

-

There is no server subject to HiRDB Datareplicator linkage in this unit (the server to be linked is a back-end server in the case of a HiRDB/Parallel Server and the single server in the case of a HiRDB/Single Server)

ffff

Unit identifier (4 characters)

`-j` option specified

```
SYSTEMID          :aaaa(bbccdd)                          1
Data Replication  :e                                     1
UNITID            :ffff(bbccdd)                          2
Data Replication  :e                                     2
SERVER NAME       :gg...g                                2,3
Extract Database  :h                                     2,3
Extract Status    :I                                     2,3
System Log Extract Point :                               2,3,4
Run ID Group   Gen No.  Block No.                        2,3,4
jj...j     kk...k     ll...l       mm...m                2,3,4
System Log Sync info   :                                 2,3,4
Run ID  Group   Gen No. Block No.                        2,3,4
nn...n     oo...o     pp...p    qq...q                  2,3,4
```

1. If the `-u` option is specified for a HiRDB/Parallel Server, this information is displayed only if the specified unit is the server machine where the system manager is located.
2. In the case of a HiRDB/Parallel Server, this information is output for only the number of units in the system.
3. In the case of a HiRDB/Parallel Server, this information is displayed as many times as there are servers in the unit. If the `-s` option was specified, this information is displayed only for the specified server. If the `pd_rpl_hdepath` operand is omitted in the system definition or the linkage file is invalid (for example, because the file is missing), this information is not displayed.
4. This information is displayed only when the specified HiRDB server contains the target database that is to be extracted. If the `-u` option was specified, this information is displayed only for the specified unit.

Explanation

aaaa

HiRDB identifier (4 characters)

bbccdd

Time (*hour:minute:second*) when the `pdls` command was entered.

e

One character indicating whether or not the HiRDB Datareplicator linkage facility is being used at the HiRDB system and unit (`Y` is displayed, indicating that the HiRDB Datareplicator linkage facility is being used, even if there is no database to be linked to all back-end servers in the system and in the unit):

`Y`

HiRDB Datareplicator linkage facility is being used

`N`

HiRDB Datareplicator linkage facility is not being used

`*`

System linkage status cannot be displayed because the system manager is inactive (the status of each unit should be checked; if there is even one unit that is not using the HiRDB Datareplicator linkage facility, it will be shown that the HiRDB Datareplicator linkage facility is not being used by the system)

`-`

There is no server subject to the HiRDB Datareplicator linkage in this unit (the server to be linked is a back-end server in the case of a HiRDB/Parallel Server and the single server in the case of a HiRDB/Single Server)

ffff

Unit identifier (4 characters)

gg...g

Server name (up to 8 characters)

h

Whether or not there is a database to be extracted:

`Y`

There is a database to be extracted at the indicated HiRDB server

`N`

There is no database to be extracted at the indicated HiRDB server

i

Internal information used by the system (one character)

The information below is displayed when the HiRDB Datareplicator linkage facility is being used.

- Extraction information from extracted HiRDB Datareplicator system log

jj...j

Run ID (run ID of log service when file was used) (8 hexadecimal digits)

kk...k

Group name of system log file currently being extracted (up to 8 characters)

ll...l

Generation number of system log file currently being extracted (8 hexadecimal digits)

mm...m

System log block number currently being extracted (8 hexadecimal digits)

- Internal information from extracted HiRDB system log

nn...n

Internal information used by the system (8 characters)

oo...o

Internal information used by the system (up to 8 characters)

pp...p

Internal information used by the system (up to 8 characters)

qq...q

Internal information used by the system (up to 8 characters)

Note

Items *j* through *q* are displayed as blanks or zeros when HiRDB Datareplicator linkage does not start after the link file for data linkage is initialized or when the link file for data linkage was initialized during execution of HiRDB Datareplicator linkage.

pdls [-d scd] (Display server schedule status)

Function

Displays server scheduling status.

Format

```
pdls -d scd [{-a | -s server-name}]
```

Options

- `-a`

Specifies that the service scheduling status of all servers that are active at the time the `pdls` command is executed is to be displayed.

- `-s` *server-name* ∼ <identifier> ((1-8))

Specifies the name of a server whose scheduling status is to be displayed.

Notes

1. Return code `0` for the `pdls -d scd` command indicates normal termination, and return code `8` indicates abnormal termination.

Output format

```
HOSTNAME: aa...a(bbccdd)
total server count ee...e                                          1
SVID  ST QUECNT MAXCNT USED_POL FREE_POL MAX_UPOL MAX_FPOL
LAST_RECV_TIME  2
ff...f  g  hh...h  ii...i  jj...j  kk...k  ll...l  mm...m  nnooppqqrrss
3
:    :    :       :       :       :       :       :
```

Explanation

1. Displayed only when the `-s` option is specified.
2. One line of information (headers); displayed in conjunction with 3.
3. When the `-a` option is specified and more than one server is operating under the scheduler, this information is displayed for each server. The scheduler controls the scheduling of a requested service and the process.

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

ee...e

Number of servers operating under the scheduler (6 decimal digits).

ff...f

Server name (up to 8 characters).

g

Server's activity status:

`S`: Being prepared

`A`: Schedulable

`E`: Engaged in termination processing

hh...h

Number of service requests currently queued (5 decimal digits).

ii...i

Maximum number of service requests that can be queued (5 decimal digits).

jj...j

Amount of transfer data buffer being used (8 decimal digits).

kk...k

Amount of transfer data buffer available for use (8 decimal digits).

ll...l

Maximum size of transfer data buffer that can be used (8 decimal digits).

mm...m

Maximum size of contiguous transfer data buffer that is currently available (8 decimal digits).

nnooppqqrrss

Last date and time messages were retrieved from the message queue (*year-month-date-hour-minute-second*)

If no messages have been retrieved yet, `000000000000` is displayed.

Note

When `E` is displayed for *g* (specified server's activity status), `0` may be displayed for *hh...h* through *mm...m*.

Utility status is also displayed, as well as server status.

pdls [-d stj] (Display whether or not statistical data output type by unit and server is specified)

Function

Displays the following types of information:

- Whether or not types of statistical information output are specified by the `pdstbegin` command for each unit or server or by the `pdstbegin` command for the system common definitions.
- Whether or not output of statistical information regarding system operation is specified by the `pd_statistics` operand for the system common definitions.

Format

```
pdls -d stj [-s server-identifier]
```

Option

- `-s` *server-identifier*

Specifies the name of the server that displays whether or not types of statistical information output are specified. The default is to display all servers.

Note

1. Return code `0` for the `pdls -d stj` command indicates normal termination, and return code `8` indicates abnormal termination.
2. If a file error occurs during output of statistical information, the display of whether the output of statistical information is specified by the `pdls -d stj` command results in a `no statistical information output` display.
3. For a HiRDB/Single Server, this command does not display statistical information about SQL object transfer, external server activities, or external server utilization status.
4. The headers for `fsv` and `hba` are displayed even when HiRDB External Data Access has not been installed.

Output format

```
HOSTNAME : aa...a(bbccdd)
UNITID      sys
eeee         h
INTERVAL : ff..f
 SVID       sys uap sql buf fil dfw idx sop dop pcd sqh obj fsv hba
 gg...g      i   j   k   l   m   n   o   p   q   r   s   t   u   v
   :                              :
```

Explanation

aa..a

Host name (up to 32 characters)

bbccdd

Time of execution of `pdls -d stj` command (*hour:minute:second*)

eeee

Unit identifier (4 characters)

ff..ff

Displays the time interval at which system activity statistical information is output, in minutes. If the output of system activity statistical information is not specified, a value of `0` is displayed (up to 4 decimal digits).

gg..g

Server name (up to 8 characters)

If no servers are found in the unit, this option displays `********`.

For items *h*-*v* as follows, if output of the applicable statistical information is specified, the system displays `*`; otherwise, it displays a blank. For the alternate BES unit when the standby-less system switchover (1:1) facility is used, the system displays `-` instead of `h`.

h

System activity statistical information by unit

i

System activity statistical information by server

j

UAP statistical information

k

SQL statistical information

l

Global buffer pool statistical information

m

Statistical information on HiRDB files for database manipulation

n

Deferred write processing statistical information

o

Index statistical information

p

SQL static optimization information

q

SQL dynamic optimization information

r

SQL object execution information

s

SQL statement statistical information

t

Statistics on SQL object transmission

u

Statistical information about external server activities

v

Statistical information about external server utilization status

pdls [-d sts] (Display status file status)

Function

Displays the status of status files.

Format

HiRDB/Single Server

- Unit status file

```
pdls -d sts [{-x host-name | -u unit-identifier}]
            [{-n logical-filename | -f physical-filename|-a|-l|-p}]
```

- Server status file

```
pdls -d sts -s server-name
          [{-n logical-filename | -f physical-filename|-a|-l|-p}]
```

HiRDB/Parallel Server

- Unit status file

```
pdls -d sts {-x host-name | -u unit-identifier}
            [{-n logical-filename | -f physical-filename|-a|-l|-p}]
```

- Server status file

```
pdls -d sts  -s server-name
            [{-n logical-filename | -f physical-filename|-a|-l|-p}]
```

Options

- {`-x` *host-name* | `-u` *unit-identifier*}

For unit status file status display, specifies the identifier of the unit or the name of the host corresponding to the unit status file. This option can be omitted for a HiRDB/Single Server because the host and unit are fixed.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the name of the host corresponding to the unit status file.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit corresponding to the unit status file.

- `-s` *server-name* ～ <identifier> ((1-8))

Specifies the name of the server corresponding to the status files for the server whose status is to be displayed.

- {`-n` *logical-filename*|`-f` *physical filename*|`-a`|`-l`|`-p`}

Specifies how the status of the status files is to be displayed. When this option is omitted, the command displays the status of all status files being used by the specified unit or server.

`-n` *logical-filename* ～ <identifier> ((1-8))

If the status of the status files for a specific logical file is to be displayed, specifies the name of the logical file.

`-f` *physical-filename* ～ <path name> ((up to 167 characters))

If the status of the status files for a specific physical file is to be displayed, specifies the absolute path name of the physical file. The physical file name must be specified in the format *HiRDB-file-system-area-name\HiRDB-filename*.

`-a`

Specifies that a summary of the status of all status files being used by the unit or server is to be displayed.

`-l`

Specifies that the status of each logical file being used by the unit or server is to be displayed.

`-p`

Specifies that the status of each physical file being used by the unit or server is to be displayed.

Notes

1. Return code `0` for the `pdls -d sts` command indicates normal termination, and return code `8` indicates abnormal termination.
2. Executing the `pdstsinit` or `pdstsrm` command for a status file whose logical file status is `CLOSE`, `NONE`, or `BLOCKADE` during HiRDB operation does not change the status information displayed by the `pdls` command. In this case, the current status of the status file can be displayed by executing the `pdstsopen` command prior to the `pdls` command.

Output format

```
HOSTNAME : aa...a(bbccdd)
 Logical file L_Status Factor Available Manage   ..........2
 ee...e          ff..f       ggg      hh...h     ii...i       ..........2
Sy  P_status R_length R_count P_name    ....................1,2
 j    kk...k     ll...l       mm...m   nn...n     ....................1,2
 j    kk...k     ll...l       mm...m   nn...n     ....................1,2
```

Explanation

1 displays the status of physical files, and 2 displays the status of a logical file.

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

ee...e

Name of logical file (up to 8 characters).

ff...f

Status of logical file (up to 8 characters):

`ACTIVE`

Current file

`BLOCKADE`

Error shutdown file

`CLOSE`

Reservation file

`NONE`

No physical file corresponds to the logical file

`STANDBY`

Reserve file

ggg

Record utilization in the file (%).

hh...h

Number of available consecutive records in the file (8 decimal digits).

ii...i

Number of management records in the file (8 decimal digits).

j

Whether the physical file is A or B:

`A`

A

`B`

B

kk...k

Displays a maximum of 11 characters denoting physical file status information; if fewer than 11 characters are displayed, the displayed characters are right-justified (up to 11 characters):

`a`

Current file

`b`

Error shutdown file

`c`

Closed file

`i`

Initialized status

`l`

Logical error

`n`

No physical file corresponds to the logical file

`o`

Open file

`p`

Physical error

`r`

Recovered from an error

`s`

Reserve file

`u`

Used file

ll...l

Record length (8 decimal digits).

mm...m

Number of records (8 decimal digits).

nn...n

Name of physical file (up to 63 characters).

pdls [-d svr] (Display status by unit and server)

Function

Displays unit and server status.

Format

```
pdls -d svr [-a [-b]]
```

Option

- `-a`

Specifies that the unit and server status is to be displayed in DAT format.

- `-b`

When the `-a` option is also specified, specifies that a title line is to be displayed.

Notes

1. Return code `0` for the `pdls -d svr` command indicates normal termination, return code `4` indicates warning termination, and return code `8` indicates abnormal termination. When the return code is `8`, see the previous message and eliminate the cause of the error.
2. For a HiRDB/Parallel Server, if a unit that does not contain the system manager is terminated abnormally or forcibly, the command may display the status at the time the unit was terminated until the system manager detects the unit's termination.
3. When the standby-less system switchover (effects distributed) facility is used, a server switched over to another unit is displayed as the target unit's server.
4. You can specify the `-a [-b]` option even if you omit `-d svr` (example: `pdls -a -b`).
5. When the standby-less system switchover (effects distributed) facility is used, a server subject to system switchover may be displayed on both source and target units.
6. Depending on the host and unit status, unit and server information may not be displayed (if all hosts and units are in error status, specifying the `-b` option displays only the title line).
7. Even though an error message is output, an error value return code may not be set. Therefore, you must check for both an error message and a value indicating abnormality in the return code to determine the execution result of the `pdls -d svr -a [-b]` command.

8. If HiRDB was started by the `pdstart -r` command, the `pdls -d svr` command terminates with return code `4`.
9. If all front-end servers have been terminated by the `pdstop -s` command since HiRDB started, the `pdls -d svr` command terminates with return code `0`.

Output format

- When the `-a` option is omitted

```
HOSTNAME(aabbcc)  UNITID   SVID     STATUS  STARTTIME
dd...d            eeee     ff...f   gg...g  hhiijj
 :                 :        :        :       :
```

- When the `-a` option is specified

```
"dd...d","eeee","ff...f","gg...g","hhiijj"
   :       :      :        :       :
```

- When the `-a` and `-b` options are both specified

```
"HOSTNAME","UNITID","SVID","STATUS","STARTTIME"
"dd...d",   "eeee",   "ff...f", "gg...g",  "hhiijj"
```

Explanation

aabbcc

Time of `pdls` command execution (*hour:minute:second*).

dd...d

Host name (up to 32 characters).

eeee

Unit identifier (4 characters).

ff..f

Server name (up to 8 characters).

If the status of a unit is being displayed, `********` is displayed.

gg...g

Status information (up to 8 characters):

ACTIVE

Active

STOP

Stopped (server)

STOP(N)

Terminated normally by the pdstop command

STOP(F)

Terminated forcibly by the pdstop command

(-f option specified)

STOP(A)

One of the following:

- Abnormal termination was detected during system operation.
- Communication with unit *bbbb* or data reception from unit *bbbb* was disabled.
- Client connection failed on unit *bbbb*.

Execute the ps command on unit *bbbb* to see if the HiRDB process is active.

If the HiRDB process is active:

A communication error or service slowdown may have occurred on unit *bbbb*. Check the cause of the error on the basis of the message that was output before the KFPS05289-E message, enter the pdstop -z command to terminate the unit, then restart the unit.

If the HiRDB process is not active:

The HiRDB process may have terminated abnormally. Take an appropriate action according to the message output during the abnormal termination. To restart, execute the pdstart -x command.

START(I)

Waiting for execution of the database initialization utility (dictionary not initialized)

SUSPEND

The front-end server is waiting for recovery of the data dictionary RDAREA or startup of the dictionary server (in this case, remove the cause of the wait, and then start the front-end server by entering the pdstart -a command).

STARTING

Preparing for startup

`STOPPING`

Preparing for shutdown

`TRNPAUSE`:

Scheduling of a new transaction is suppressed

Scheduling of a new transaction is suppressed at the corresponding server. If the `KFPS01160-E` message has been issued, eliminate the cause of the space shortage for the system log. For details about determining the cause of a space shortage for the system log and how to handle the error, see *Action to be taken by the HiRDB administrator when the available space is less than the warning value* in the *HiRDB Version 8 System Operation Guide*.

hhiijj

Activation time (*hour:minute:second*).

pdls [-d trn] (Display server transaction status)

Function

Displays server transaction status

Format

```
pdls -d trn [-t transaction-identifier] [-{a|c}] [-s server-name]
```

Options

- `-t` *transaction-identifier* ∼ <alphanumeric> ((16 characters))

Specifies the identifier of a transaction about which information is to be displayed. When this option is omitted, information about all transactions is displayed.

- `-{a|c}`

`-a`

Specifies that all transaction information is to be displayed.

`-c`

Specifies that the number of branches in a global transaction is to be displayed.

When this option is omitted, the command displays all transaction information (with some exceptions).

- `-s` *server-name* ∼ <identifier> ((1-8))

Specifies the name of the front-end server or of a back-end server. If the front-end server name is specified, the command displays information about all global transactions managed by the front-end server. If a back-end server name is specified, the command displays information about all transactions branched from global transactions managed by the specified back-end server. When this option is omitted, the command displays information about all transactions managed by the specified unit.

Notes

1. Return code `0` for the `pdls -d trn` command indicates normal termination, and return code `8` indicates abnormal termination.

Output format

```
-a option specified:
HOSTNAME: aa...a(bbccdd)
TRNGID        TRNBID        STATUS     PID     SVID    B-SVID
eeeeffffgg...g    eeeeffffhh...h    jj...j(k,l)    mm...m    ii...i    nn...n
```

```
PROGRAM              TIME     SYSINF
oo...o               ppqqrr   ss...s
S-PID=tt...t C-PID=uu...u(vv...v)
TMID=wwww XID=xx...x,yy...y  ENVGRP=zzzz
```

Explanation

When more than one transaction is to be displayed, this information is displayed for each such transaction. When `-a` is omitted, `TIME`, `SYSINF`, and `S-PID` through `ENVGRP` information are not displayed.

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

eeeeffffgg...g and *eeeeffffhh...h*

Transaction identifier (16 characters)

eeee

HiRDB identifier (4 characters)

ffff

Unit identifier (4 characters)

gg...g

Global transaction number (8 decimal digits)

hh...h

Transaction branch number (8 decimal digits)

ii...i

Server name (up to 8 characters).

jj...j

Transaction status 1 (up to 16 characters):

`BEGINNING`

Engaged in transaction startup processing

`ACTIVE`

Executing

`SUSPEND`

Suspended

`IDLE`

Shifting to synchronization point processing

`PREPARE`

Engaged in commit (phase 1) processing

`READY`

Engaged in commit (phase 2) processing

`H_COMMIT`

Engaged in system-selected[1] commit processing

`H_ROLLBACK`

Engaged in system-selected[1] rollback processing

`COMMIT`

Engaged in commit processing

`ROLLBACK_ACTIVE`

Waiting for rollback processing

`ROLLBACK`

Engaged in rollback processing

`H_FORGETTING`

Engaged in system-selected[1] transaction termination processing

`FORGETTING`

Engaged in transaction termination processing

[1] In the event of an error, HiRDB selects a settlement method without waiting for an instruction from the user.

k

Transaction status 2

`u`

Accessing data by means of a database access process

`r`

Executing transaction recovery processing by means of a transaction recovery process

`p`

Waiting for transaction recovery processing by means of a transaction recovery process

l

Transaction status 3

`s`

Communicating (transferring a transaction settlement instruction between servers)

`r`

Waiting for communication (waiting to transfer a transaction settlement instruction between servers)

`w`

Synchronizing (synchronizing the transaction between the transaction manager and the server)

`n`

Not communicating (neither `s` nor `r`)

mm...m

Process ID (up to 10 decimal digits).

If there is no process associated with executing the transaction, such as when the transaction is waiting for transaction recovery processing, `0` is displayed.

nn...n

Name of branch source server (up to 8 characters).

If the local server issued the request, `********` is displayed.

oo...o

UAP identification information (up to 30 characters).

The UAP identification name specified in `PDCLTAPNAME` in the client definition is displayed.

If `PDCLTAPNAME` is undefined, `Unknown` is displayed.

For a utility, the name of the utility command is displayed.

This information may be left blank if the utility was running under a utility server process or if it was impossible to obtain the UAP identification information because of an error.

ppqqrr

Time the service request was accepted (*hour:minute:second*).

ss...s

Internal information used by the system (up to 32 characters).

tt...t

Process ID of the server that accepted transaction processing as part of the UAP or utility processing (up to 10 decimal digits).

`0` may be displayed if the utility was running under a utility server process or if it was impossible to obtain the process ID because of an error.

uu...u

Process ID of the client (UAP or utility) that was connected to the server process (up to 10 decimal digits).

This information is applicable only to the server process that was connected to the client. For a service process that was not connected to the client, `0` is displayed. `0` is also displayed if the version of the client library connected to the UAP is earlier than 04-00.

vv...v

IP address of the client (UAP or utility) that was connected to the server process (up to 15 characters).

This information is applicable only to the server process that was connected to the client. For a service process that was not connected to the client, `0.0.0.0` is displayed. `0.0.0.0` is also displayed if the version of the client library connected to the UAP is earlier than 04-00.

wwww

OLTP identifier of the connected OLTP (up to 4 characters).[1]

xx...x, yy...y

Transaction identifier provided by OLTP (33 characters).[1]

zzzz

Environment variable group identifier provided by the OLTP system (four characters).[1]

This information is displayed when the multi-connection facility is used. When the multi-connection facility is not used, `****` is displayed.

[1] This information is not output for a utility or if the connected client is not an

application complying with X/Open.

`-c` option or `-t` and `-c` options specified:

```
HOSTNAME: aa...a(bbccdd)
TRNGID                SVID        B-SVNUM
eeeeffffgg...g        hh...h      ii...i
```

Explanation

When more than one transaction is to be displayed, this information is displayed for each such transaction.

aa...a

Host name (up to 32 characters).

bbccdd

Time of `pdls` command execution (*hour:minute:second*).

eeeeffffgg...g

Transaction identifier (16 characters):

eeee

HiRDB identifier (4 characters)

ffff

Unit identifier (4 characters)

gg...g

Global transaction number (8 decimal digits)

hh...h

Server name (up to 8 characters).

ii...i

Number of transactions branched from the corresponding server (5 decimal digits).

pdls [-d ust] (Display unit activity status)

Function

The `pdls -d ust` command displays a unit's activity status.

This command displays the activity status for only the unit where the command is executed.

Format

```
pdls -d ust
```

Option

None

Notes

1. The following are the `pdls -d ust` command's return codes:

   `0`

   Unit is active (the front-end server can be connected if it is located at the unit where the command is executed).

   `4`

   Unit is starting (the front-end server is located at the unit where the command is executed, but the front-end server cannot be connected), or unit is stopping.

   `8`

   Restart of the process service was cancelled.

   `12`

   Unit is stopped.

   `16`

   OS registration has been deleted.

2. You should take one of the following actions when the unit status is `PAUSE`, or when the front-end server status is `SUSPEND` and does not change to `ONLINE`, or when the setup status is `UNSETUP`:

   - If the unit status is `PAUSE`, an error has occurred. See the `KFPS00715-I` message and the previous message output to the event log, eliminate the

cause of the error, and then restart the service.

- If the front-end server status is SUSPEND and does not change to ONLINE, an error may have occurred. See the message output to the event log, eliminate the cause of the error, and then use the pdstart -a command to release the front-end server from SUSPEND status.
- If the setup status is UNSETUP, the service has stopped. Before starting HiRDB, start the service.

Output format

```
HOSTNAME   : aa...a(bbccdd)
SYSTEMID   : eeee
UNITID     : ffff
ENTRYHOST  : gg...g
PAIRHOST   : hh...h
UNIT-STAT FES-STAT  SETUP-STAT
iiiiiii    jjjjjjj  kkkkkkk
```

Explanation

aa...a

Host name of the primary system (up to 32 characters)

bbccdd

Time the pdls command was executed (*hour minute second*)

eeee

HiRDB identifier (4 characters)

ffff

Unit identifier (4 characters)

gg...g

Host name of the unit where the pdls command was executed (up to 32 characters)

hh...h

Host name of the remote system (up to 32 characters) when the standby system switchover facility without IP address inheritance or the standby-less system switchover (1:1) facility is used.

Otherwise, this is blank.

iiiiiii

Unit status (8 characters):

PAUSE△ △ △: Process server restart suspended

STOP△ △ △ △: Stopped

STARTING: Starting

ONLINE△ △: Running

STOPPING: Stopping

△: One single-byte space.

jjjjjjjj

Status of the front-end server (8 characters):

STOP△ △ △ △: Stopped

STARTING: Starting

SUSPEND△: Suspended

ONLINE△ △: Running

STOPPING: Stopping

********: Unit contains no front-end server

△: One single-byte space.

For a HiRDB/Single Server, ******** is always displayed.

kkkkkkk

Setup status (7 characters):

SETUP△ △: Service is running.

UNSETUP: Service is stopped.

△: One single-byte space.

pdntenv (Set up HiRDB operating environment)

Function

The pdntenv command displays or sets up a HiRDB operating environment.

Executor

HiRDB administrator

Format

```
pdntenv [{-hc {on|off}|-pn service-port-name|-a|-sc {auto|manual}
          |-c character-codes-type | -ro {on|off}
          |-shmfile {regular|page}|-k {none|force}}]
```

Options

- -hc

Specifies whether or not to use hardware compression during DAT access:

on: Use hardware compression.

off: Do not use hardware compression.

The information specified at the server machine connected to the DAT device takes effect.

This option is applicable only when the DAT drive supports hardware compression. For the same DAT, use the same setting for both write and read operations.

- -pn *service-port-name*

Specifies the service port name for remote commands. Express the service port name by prefixing pdrshsrv to the identifier (1-4 single-byte alphanumeric characters without spaces).

Example

```
pdrshsrvUNT1
```

Specify the port number and the service port name specified in the %windir%\system32\drivers\etc\SERVICES file.

The specified value takes effect the next time the service starts.

- -a

If multiple HiRDBs are installed (multi-HiRDB configuration), this option displays information about all such HiRDBs.

- -sc {auto|manual}

When a HiRDB/Single Server is to be terminated by the pdstop command, specifies whether or not to also stop Windows services at the same time.

auto

Specifies that Windows services are to be stopped when HiRDB is terminated. In such a case, you must start Windows services before you start HiRDB the next time.

manual

Specifies that Windows services are not to be stopped when HiRDB is terminated. In such a case, there is no need to start Windows services the next time you start HiRDB. This option can reduce the time required for locking the Windows service control manager.

If this option is omitted, the system assumes auto if the HiRDB version is earlier than 06-00 and manual if it is 06-00 or later. If you have installed HiRDB over an existing HiRDB, this option value is inherited.

The specified value takes effect the next time the service starts. You cannot specify this option for a HiRDB/Single Server.

Whenever you use the pdchgconf or pdprgrenew command, manual must be specified. If the pdchgconf or pdprgrenew command is executed in an environment with auto specified, an error results and HiRDB may stop.

- -c *character-codes-type* ～ <<sjis>>

Specifies the type of character codes to be used for national character data in HiRDB.

sjis: Shift JIS Kanji codes

lang-c: Single-byte character codes

chinese: EUC Chinese Kanji codes

utf-8: Unicodes (UTF-8)#

#

According to the ISO/IEC 10646 standard, a range of 1 to 4 bytes is assigned per character and bytes 5 and 6 are reserved for a future standard. Although HiRDB supports a range of 1 to 6 bytes per character, problems may be introduced in the future if you use the range of bytes 5 and 6 in which no characters have been assigned.

If the version of HiRDB Text Search Plug-in is 02-02 or earlier, utf-8 is not supported.

If you change the character codes type, you must re-initialize the database.

- -ro {on|off}

Specifies whether or not remote commands are to be used. This option is applicable to a HiRDB/Single Server.

`on`:

Specifies that remote commands are to be used. Specify this option when you use the system switchover facility with the HiRDB/Single Server. If you specify `on`, also specify the `-pn` option.

`off`:

Specifies that remote commands are not to be used.

This option's settings take effect the next time the service starts. This option is disabled for a HiRDB/Parallel Server.

- -shmfile {regular|page}

Specifies the allocation target for shared memory used in HiRDB. If this option is omitted, the previous allocation target is used.

regular:

Allocates shared memory to files under the HiRDB directory.

page:

Allocates shared memory to paging files (virtual memory).

The following describes how to change the shared memory allocation target.

How to change the allocation target

To change the shared memory allocation target:

1. Terminate HiRDB normally and then stop the HiRDB services.
2. Specify the shared memory allocation target in the `-shmfile` operand of the `pdntenv` command and then execute the command.
3. Extend the size of the paging file (virtual memory size) and restart Windows.#
4. Restart HiRDB and the HiRDB services.

#

If the estimation of paging file size shows that there is sufficient capacity, there is no need to perform this procedure.

Rules

1. If you have specified `regular`, you must add the size of the shared memory that is used in HiRDB# to the capacity of the drive that contains the HiRDB directory.

2. If you have specified `page`, you must add the size of the shared memory that is used in HiRDB# to the maximum capacity of the paging file (virtual memory) that is used. In this case, note that the shared memory part of the paging file is fixed for use.

#

For details about the size of the shared memory that is used in HiRDB, see the formula for determining the required memory size in the manual *HiRDB Version 8 Installation and Design Guide*.

- -k {none|force}

Specifies the HiRDB termination method when the OS shuts down.

none:

Does not terminate HiRDB when the OS shuts down.

force:

Terminates HiRDB explicitly and forcibly when the OS shuts down.

If OS shutdown occurs while HiRDB services are in the process of being stopped, HiRDB is terminated forcibly by the OS (unit down) even if `force` is specified.

For details about how to terminate HiRDB during an OS shutdown, see the manual *HiRDB Version 8 System Operation Guide*.

- No option specified

The `pdntenv` command displays the current settings.

Rules

1. You can execute the `pdntenv` command only while HiRDB or the HiRDB services are stopped.
2. Execute the `pdntenv` command at each server machine.

Notes

1. Return code `0` for the `pdntenv` command indicates normal termination, and return code `8` indicates abnormal termination.

Output format

No option specified

```
***** HiRDB setup information *****
Program Name         = HiRDB/SingleServer
Version              = 07-03
Character Code       = sjis
Dat Hard Compress    = on
Setup ID             = ----
```

```
Rsh Port Name       = pdrshsrv
Path Name           = C:\win32app\hitachi\hirdb_s
Service Complete    = manual
Rsh Online          = on
ShmFile             = regular
Shutdown mode       = none
```

Explanation

`Program Name`

Name of HiRDB program

`Version`

HiRDB version

`Character Code`

HiRDB's character codes type

`sjis`: Shift JIS Kanji codes

`lang-c`: Single-byte character codes

`chinese`: EUC Chinese Kanji codes

`utf-8`: Unicodes (UTF-8)

`Dat Hard Compress`

Setting for hardware compression during DAT access

`on`: Compress

`off`: Do not compress

`Setup ID`

Setup identifier (if standard installation is used, `----` is displayed)

`Rsh Port Name`

Service port name for remote commands

`Path Name`

HiRDB installation directory

`Service Complete`

Time at which HiRDB services startup is completed

`auto`: HiRDB services startup is completed when the `KFPS01001-I` message is displayed.

`manual`: HiRDB services startup is completed when the system waits for

execution of pdstart.

blank: Not applicable.

Rsh Online: Specification of remote commands

on: Uses remote commands.

off: Does not use remote commands.

blank: Not applicable.

Shmfile: Allocation target of the shared memory that is used in HiRDB

regular: Allocates to files under the HiRDB directory

page: Allocates to paging files

Blank: Not applicable

Shutdown mode: How to terminate HiRDB during OS shutdown

none: Does not terminate HiRDB when the OS shuts down.

force: Terminates HiRDB explicitly and forcibly when the OS shuts down.

Blank: Not applicable

-a option specified (applicable to multi-HiRDB)

```
***** HiRDB setup information *****
Program Name         = HiRDB/SingleServer
Version              = 06-02-/D
Character Code       = sjis
Dat Hard Compress    = on
Setup ID             = UNT1
Rsh Port Name        = pdrshsrvUNT1
Path Name            = C:\win32app\hitach\hirdb_sUNT1
Service Complete     = manual
Rsh Online           = on
ShmFile              =
Shutdown mode        =
***** HiRDB setup information *****
Program Name         = HiRDB/ParallelServer
Version              = 07-03
Character Code       = sjis
Dat Hard Compress    = on
Setup ID             = UNT2
Rsh Port Name        = pdrshsrvUNT2
Path Name            = C:\win32app\hitach\hirdb_pUNT2
Service Complete     = manual
Rsh Online           = on
ShmFile              = regular
Shutdown mode        = none
```

Explanation

See the previous explanation for *No option specified*.

pdobils (Display SQL object buffer statistics)

Function

The pdobils command displays statistical information about the SQL objects stored in the SQL object buffer. This information enables you to identify SQL objects that take a long time to process or that are input and output frequently; you can also obtain the object reuse status of SQL objects.

Executor

HiRDB administrator

Format

```
pdobils [-s server-name] [-r]
```

Operands

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of a server (single server or front-end server) whose information is to be displayed or whose statistical information counters are to be initialized.

- -r

Specifies that the statistical information counters are to be initialized. For example, this option is specified when overflow occurs in statistical information counters.*

For details about the items for which counters are initialized, see *Output format*.

* When an SQL object is reused many times, its counter in the SQL object buffer may overflow. When a counter overflows, * is displayed for the corresponding item in the statistical information.

Rules

1. The pdobils command can be executed only when HiRDB is running.
2. The pdobils command can be executed at any server machine.

Notes

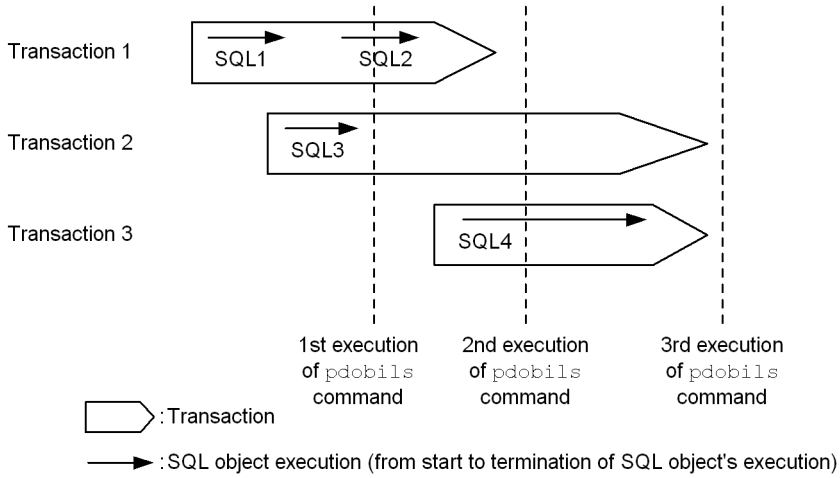
1. The execution results of the pdobils command can be confirmed from the return code after the command has executed. Return code 0 indicates normal termination, and return code 4 indicates abnormal termination.
2. Because the pdobils command manipulates the SQL object buffer, the overhead

associated with this manipulation may increase the HiRDB workload.

3. When the pdobils command is terminated forcibly, SQL processing on the SQL object buffer and post-processing may be placed in wait status for up to 1 minute. For this reason, it is unwise to terminate the pdobils command forcibly.

4. The amount of time required for an SQL object to execute is recorded in the SQL object buffer when the transaction is completed or the SQL object is released. Figure 2-4 illustrates the SQL object execution time (duration) that is displayed when the pdobils command is executed.

Figure 2-4: SQL object execution time (duration) displayed when the pdobils command is executed



Explanation

The display of the amount of time (duration) it takes an SQL object to execute depends on the timing of execution of the pdobils command. The following table shows the validity of the time that is displayed for each point of execution:

Execution sequence of pdobils command	SQL1	SQL2	SQL3	SQL4
1	N	N	N	N
2	Y	Y	N	N
3	Y	Y	Y	Y

Legend:

Y: The displayed execution duration time is valid for this SQL object.

N: The displayed execution duration time is not valid for this SQL object.

Note 1

When the cursor is used, the displayed execution duration time is the server's total processing time from opening to closing of the cursor.

Note 2

While the `pdobils` command is executing, the SQL object buffer is used by other UAPs. If there is an SQL object whose transaction is completed during execution of the `pdobils` command and the corresponding information is available, the information is applied to the output information.

Note 3

The execution time (duration) is displayed in seconds (rounded off to a whole second).

Output format

```
pdobils VV-RR-ZZ [70]
<< SQL OBJECT CACHE INFORMATION >> DATE: 2004/01/01 TIME: 00:00:00 [1]
HOST NAME                   : node01 [2]
SERVER NAME                 : fes01 [3]
CACHE SIZE(KB)              : 2048 [4]
SQL OBJECT TOTAL SIZE(B)    : 1932556 [5]


<< SQL OBJECT LIST >>
*SQL OBJECT NO               : 1 [6]
 STATUS                     : ACTIVE [7]
 TYPE                       : STATIC SQL [8]
 SIZE                       : 7792 [9]
 EXECUTE COUNT              : 10 [10]
 EXECUTE TIME AVG(s)        : 1.234567 [11]
 EXECUTE TIME MAX(s)        : 9.000000 [12]

 # SERVICE NAME             :          [13]
 # UAP NAME                 : Unknown [14]
 # CONNECT NO               : 123 [15]
 # SQL NO                   : 456 [16]
 # RECORD DATE/TIME         : 2004/01/01 00:00:00 [17]
 DB REFERENCE GET COUNT      : 55500 [28] AVG 5550 [29] MAX 5600 [30]
 DB UPDATE GET COUNT         : 0 [31] AVG 0 [32] MAX 0 [33]
 DB READ COUNT               : 800 [34] AVG 80 [35] MAX 85 [36]
 DB WRITE COUNT              : 0 [37] AVG 0 [38] MAX 0 [39]
```

```
 LOB REFERENCE GET COUNT      : 0 [40] AVG 0 [41] MAX 0 [42]
 LOB UPDATE GET COUNT         : 0 [43] AVG 0 [44] MAX 0 [45]
 LOB READ COUNT               : 0 [46] AVG 0 [47] MAX 0 [48]
 LOB WRITE COUNT              : 0 [49] AVG 0 [50] MAX 0 [51]
 LIST REFERENCE GET COUNT     : 0 [52] AVG 0 [53] MAX 0 [54]
 LIST UPDATE GET COUNT        : 0 [55] AVG 0 [56] MAX 0 [57]
 LIST READ COUNT              : 0 [58] AVG 0 [59] MAX 0 [60]
 LIST WRITE COUNT             : 0 [61] AVG 0 [62] MAX 0 [63]
 WKFILE READ COUNT            : 0 [64] AVG 0 [65] MAX 0 [66]
 WKFILE WRITE COUNT           : 0 [67] AVG 0 [68] MAX 0 [69]

 PREPROCESSOR USER            : user1 [18]
 PREPROCESSOR SOURCE          : abc.ec [19]
 SECTION NO                   : 123 [20]
 ISOLATION LEVEL              : 2 [21]
 OPTIMIZE LEVEL               : 10 [22]
 ADDITIONAL OPTIMIZE LEVEL    : 0 [23]
 DEFAULT SCHEMA               :   [24]
 ROUTINE ID                   : 123 [25]
 SQL                      : SELECT GNO,GNAME,KIKAKU,TANKA,SURYO,GENKA FROM ZAIKO [26]


<< SQL OBJECT STATUS COUNT >> [27]
STATUS     | TYPE      STATIC SQL        DYNAMIC SQL       ROUTINE           TOTAL
ACTIVE               :  0                 0                 0                 0
LRU                  :  0                 0                 0                 0
TEMPORARY            :  0                 0                 0                 0
COMPILE/TRANSFER     :  0                 0                 0                 0
PROCESS              :  -                 -                 -                 0
RELEASE              :  -                 -                 -                 0
TOTAL                :  0                 0                 0                 0
```

Note 1

The information that is displayed depends on the server's type and status.

Note 2

Items 6 through 69 (except for item 27) are repeated for each SQL object. The information that is displayed depends on the type and status. For details, see *Notes*.

Note 3

When the `-s` option is omitted, items 1 through 69 are displayed for each single server or front-end server.

Note 4

When the `-r` option is specified, items 10 through 17 and 28 through 69 are initialized.

Explanation

1. Statistical information display time

Displays the time the statistical information was displayed.

2. Host name

   Displays the name of the host for which statistical information was acquired.

3. Server name

   Displays the name of the server for which statistical information was acquired.

4. Size of SQL object buffer

   Displays the size of the SQL object buffer (in kilobytes).

5. Total size of SQL objects

   Displays the total size of SQL objects (in bytes) stored in the SQL object buffer. If the SQL object buffer is used by another UAP during execution of the `pdobils` command, the sum of the sizes of each SQL object may not match the total size of all SQL objects.

6. SQL object number

   Displays the management number of the SQL object.

7. Status

   Displays one of the following, indicating the status of the SQL object:

Status	Description
`ACTIVE`	A UAP is using the SQL object.
`LRU`	Although the SQL object is stored in the SQL object buffer, no UAP is currently using it.
`TEMPORARY`	The SQL object is not shared among processes. This is applicable when either of the following is true: • Creation of the SQL object is specified in the `PDVWOPTMODE` or `PDUAPREPLVL` client environment definition. • A dynamic SQL object is stored in the SQL object buffer.
`COMPILE/TRANSFER`	A static SQL object is being created, or the SQL object is being transferred from the front-end server to the back-end server or dictionary server.
`PROCESS`	The SQL object is being managed in the process memory.
`RELEASE`	The SQL object is being released or is no longer valid.

8. Type

   Displays the type of the SQL object:

   `STATIC SQL`: Static SQL object

DYNAMIC SQL: Dynamic SQL object

ROUTINE: Routine SQL object

9. Size

   Displays the size of the SQL object (in bytes).

10. Execution count[4]

    Displays the number of times the SQL object has executed.

11. Average execution time

    Displays the average execution time for the SQL object (in seconds).

12. Maximum execution time

    Displays the maximum execution time for the SQL object (in seconds).

13. Service name

    When UAPs were executed in an OLTP environment, displays the name of the service that recorded the maximum execution time.

14. UAP name

    Displays the name of the UAP that recorded the maximum execution time (not displayed if the UAP name cannot be identified).

15. CONNECT sequence number[4]

    Displays the CONNECT sequence number of the UAP that recorded the maximum execution time.

16. SQL number[4]

    Displays the number of the SQL that recorded the maximum execution time.

17. Recorded time

    Displays the time at which the maximum execution time was recorded.

18. Preprocessing user's authorization identifier

    Displays the authorization identifier of the user who executed preprocessing of the UAP source program. This item is displayed for static SQL objects.

19. Preprocessed source program name

    Displays the name of the source program that was preprocessed (this item is displayed for static SQL objects).

20. Section number

    Displays the section number (this item is displayed for static SQL objects).

21. Data guarantee level

    Displays the data guarantee level (this item is displayed for static and dynamic SQL objects).

22. SQL optimization option

    Displays the SQL optimization option (this item is displayed for static and dynamic SQL objects).

23. SQL extension optimizing option

    Displays the SQL extension optimizing option (this item is displayed for static and dynamic SQL objects).

24. Default schema name

    Displays the assumed authorization identifier when the authorization identifier is omitted from the SQL statement (this item is not displayed when an authorization identifier is specified in the SQL statement) (this item is displayed for dynamic SQL objects).

25. Routine's object ID

    Displays the routine's object ID (this item is displayed for routine SQL objects).

26. SQL statement

    Displays the SQL statement (this item is displayed for static and dynamic SQL objects).

27. Type and number of SQL objects for each status

    Displays for each status the number of SQL objects of each type that are being managed in the SQL object buffer.

28. Reference count for data, index, and directory pages (total)[2, 4]

    Displays the total number of times that buffer acquisition requests were issued to the global buffer using the SQL object for referencing purposes.

29. Reference count for data, index, and directory pages (average)[1, 2, 4]

    Displays for the SQL object the average number of times that acquisition requests for referencing were issued to the global buffer.

30. Reference count for data, index, and directory pages (maximum)[2, 4]

    Displays for the SQL object the maximum number of times that acquisition requests for referencing were issued to the global buffer.

31. Update count for data, index, and directory pages (total)[2, 4]

Displays the total number of times that buffer acquisition requests for updating were issued to the global buffer.

32. Update count for data, index, and directory pages (average)[1, 2, 4]

    Displays for the SQL object the average number of times that acquisition requests for updating were issued to the global buffer.

33. Update count for data, index, and directory pages (maximum)[2, 4]

    Displays the maximum number of times that buffer acquisition requests for updating were issued to the global buffer.

34. Real READ count for data, index, and directory pages (total)[2, 3, 4]

    Displays the total number of input operations that were performed on data, index, and directory pages. This value includes the number of input operations performed by the prefetch facility, but does not include input operations performed by the asynchronous READ facility.

35. Real READ count for data, index, and directory pages (average)[1, 2, 3, 4]

    Displays for the SQL object the average number of input operations that were performed on data, index, and directory pages. This value includes the number of input operations performed by the prefetch facility, but does not include input operations performed by the asynchronous READ facility.

36. Real READ count for data, index, and directory pages (maximum)[2, 3, 4]

    Displays for the SQL object the maximum number of input operations that were performed on data, index, and directory pages. This value includes the number of input operations performed by the prefetch facility, but does not include input operations performed by the asynchronous READ facility.

37. Real WRITE count for data, index, and directory pages (total)[2, 3, 4]

    Displays the total number of output operations that were performed on data, index, and directory pages. This value does not include the number of output operations performed by deferred write processing.

38. Real WRITE count for data, index, and directory pages (average)[1, 2, 3, 4]

    Displays for the SQL object the average number of output operations that were performed on data, index, and directory pages. This value does not include the number of output operations performed by deferred write processing.

39. Real WRITE count for data, index, and directory pages (maximum)[2, 3, 4]

    Displays for the SQL object the maximum number of output operations that

were performed on data, index, and directory pages. This value does not include the number of output operations performed by deferred write processing.

40. LOB column data page reference count (total)[4]

    Displays the total number of times that buffer acquisition requests were issued to reference LOB column data pages.

41. LOB column data page reference count (average)[1, 4]

    Displays for the SQL object the average number of times that acquisition requests were issued to reference LOB column data pages.

42. LOB column data page reference count (maximum)[4]

    Displays for the SQL object the maximum number of times that acquisition requests were issued to reference LOB column data pages.

43. LOB column data page update count (total)[4]

    Displays the total number of times that buffer acquisition requests were issued to update LOB column data pages.

44. LOB column data page update count (average)[1, 4]

    Displays for the SQL object the average number of times that buffer acquisition requests were issued to update LOB column data pages.

45. LOB column data page update count (maximum)[4]

    Displays for the SQL object the maximum number of times that buffer acquisition requests were issued to update LOB column data pages.

46. Real READ count for LOB column data pages (total)[4]

    Displays the total number of real READ operations that were performed on LOB column data pages.

47. Real READ count for LOB column data pages (average) [1, 4]

    Displays for the SQL object the average number of real READ operations that were performed on LOB column data pages.

48. Real READ count for LOB column data pages (maximum)[4]

    Displays for the SQL object the maximum number of real READ operations that were performed on LOB column data pages.

49. Real WRITE count for LOB column data pages (total)[4]

    Displays the total number of real WRITE operations that were performed on

LOB column data pages.

50. Real WRITE count for LOB column data pages (average)[1, 4]

    Displays for the SQL object the average number of real WRITE operations that were performed on LOB column data pages.

51. Real WRITE count for LOB column data pages (maximum)[4]

    Displays for the SQL object the maximum number of real WRITE operations that were performed on LOB column data pages.

52. List page reference count (total)[4]

    Displays the total number of times that buffer acquisition requests were issued to the global buffer to reference list pages.

53. List page reference count (average)[1, 4]

    Displays for the SQL object the average number of times that buffer acquisition requests were issued to the global buffer to reference list pages.

54. List page reference count (maximum)[4]

    Displays for the SQL object the maximum number of times that buffer acquisition requests were issued to the global buffer to reference list pages.

55. List page update count (total)[4]

    Displays the total number of times that buffer acquisition requests were issued to the global buffer to update list pages.

56. List page update count (average)[1, 4]

    Displays for the SQL object the average number of times that buffer acquisition requests were issued to the global buffer to update list pages.

57. List page update count (maximum)[4]

    Displays for the SQL object the maximum number of times that buffer acquisition requests were issued to the global buffer to update list pages.

58. Real READ count for list pages (total)[4]

    Displays the total number of real READ operations that were performed on list pages.

59. Real READ count for list pages (average)[1, 4]

    Displays for the SQL object the average number of real READ operations that were performed on list pages.

60. Real READ count for list pages (maximum)[4]

    Displays for the SQL object the maximum number of real READ operations that were performed on list pages.

61. Real WRITE count for list pages (total)[4]

    Displays the total number of real WRITE operations that were performed on list pages.

62. Real WRITE count for list pages (average)[1, 4]

    Displays for the SQL object the average number of real WRITE operations that were performed on list pages.

63. Real WRITE count for list pages (maximum)[4]

    Displays for the SQL object the maximum number of real WRITE operations that were performed on list pages.

64. READ count for work table files (total)[4]

    Displays the total number of READ operations that were performed on work table files.

65. READ count for work table files (average)[1, 4]

    Displays for the SQL object the average number of READ operations that were performed on work table files.

66. READ count for work table files (maximum)[4]

    Displays for the SQL object the maximum number of READ operations that were performed on work table files.

67. WRITE count for work table files (total)[4]

    Displays the total number of WRITE operations that were performed on work table files.

68. WRITE count for work table files (average)[1, 4]

    Displays for the SQL object the average number of WRITE operations that were performed on work table files.

69. WRITE count for work table files (maximum)[4]

    Displays for the SQL object the maximum number of WRITE operations that were performed on work table files.

70. HiRDB version

When there is no *ZZ*, only *VV*-*RR* is displayed.

[1] The average value is rounded off. If the total value overflows, `****` is displayed as the average value.

[2] This does not include accesses to data dictionary tables that store definition information, such as for tables and indexes.

[3] Deferred write processing involves writing pages updated in the global buffer to disk when the number of updated pages reaches a specified value, rather than when the `COMMIT` statement is issued. Deferred write processing outputs updated pages to HiRDB files asynchronously with the database processes.

The prefetch facility enables database processes to read multiple pages of table data on disk volumes in batch mode. The asynchronous READ facility reads multiple pages of table data on disk volumes in batch mode asynchronously with the database processes.

[4] The counter may overflow. If an overflow has occurred, `*` is displayed as the lead digit of the corresponding counter. Counting continues when an overflow occurs.

Notes

Whether or not items 6 through 26 are displayed depends on the combination of type and status. Table 2-4 shows whether or not each of these items is displayed depending on the combination of type and status. The numbers correspond to the numbers in the output format.

Table 2-4: Whether or not each item is displayed depending on the combination of type and status (applicable to static SQL objects)

No.	Type																	
	STATIC SQL						DYNAMIC SQL						ROUTINE					
	Status						Status						Status					
	A	L	T	C	P	R	A	L	T	C	P	R	A	L	T	C	P	R
6	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
8	Y	Y	—	—	—	—	Y	Y	—	—	—	—	Y	Y	—	—	—	—
9	Y	Y	—	—	—	—	Y	Y	—	—	—	—	Y	Y	—	—	—	—
10	Y	Y	—	—	—	—	Y	Y	—	—	—	—	Y	Y	—	—	—	—
11	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—

No.	Type																	
	STATIC SQL						DYNAMIC SQL						ROUTINE					
	Status						Status						Status					
	A	L	T	C	P	R	A	L	T	C	P	R	A	L	T	C	P	R
12	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
13	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
14	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
15	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
16	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
17	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
18	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
19	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
20	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
21	Y	Y	—	—	—	—	Y	Y	—	—	—	—	N	N	—	—	—	—
22	Y	Y	—	—	—	—	Y	Y	—	—	—	—	N	N	—	—	—	—
23	Y	Y	—	—	—	—	Y	Y	—	—	—	—	N	N	—	—	—	—
24	N	N	—	—	—	—	Y	Y	—	—	—	—	—	—	—	—	—	—
25	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	—	—	—	—
26	Y	Y	—	—	—	—	Y	Y	—	—	—	—	—	—	—	—	—	—
28	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
29	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
30	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
31	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
32	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
33	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
34	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
35	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
36	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—

No.	Type																	
	STATIC SQL						DYNAMIC SQL						ROUTINE					
	Status						Status						Status					
	A	L	T	C	P	R	A	L	T	C	P	R	A	L	T	C	P	R
37	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
38	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
39	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
40	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
41	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
42	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
43	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
44	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
45	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
46	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
47	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
48	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
49	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
50	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
51	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
52	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
53	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
54	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
55	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
56	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
57	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
58	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
59	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
60	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—

No.	Type																	
	STATIC SQL						DYNAMIC SQL						ROUTINE					
	Status						Status						Status					
	A	L	T	C	P	R	A	L	T	C	P	R	A	L	T	C	P	R
61	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
62	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
63	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
64	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
65	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
66	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
67	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
68	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—
69	C	C	—	—	—	—	C	C	—	—	—	—	C	C	—	—	—	—

Legend:

Y: Displayed

C: Displayed if the value of item 10 is `1` or greater. If the value of item 10 is `0`, this information is not displayed.

N: Not displayed

—: Not applicable

Statuses:

A: `ACTIVE`

L: `LRU`

T: `TEMPORARY`

C: `COMPILE/TRANSFER`

P: `PROCESS`

R: `RELEASE`

How to interpret statistical information

The following describes how to interpret statistical information.

- PROCESS in item 27

  If a TOTAL count is provided for SQL objects whose STATUS is PROCESS, the SQL objects are being managed for each process because SQL objects cannot be stored in the SQL object buffer.

  If SQL objects are always being counted, you should re-evaluate the size of the SQL object buffer.

- DYNAMIC in item 27

  - Part 1

    If a STATIC SQL count is provided for SQL objects whose STATUS is TEMPORARY, UAPs for which the PDUAPREPLVL client environment definition has been set are being executed.

    If PDUAPREPLVL is not needed for a UAP, you should delete the PDUAPREPLVL specification and then execute the UAP.

  - Part 2

    If a DYNAMIC SQL count is always provided for SQL objects whose STATUS is TEMPORARY, UAPs for which the PDUAPREPLVL and PDVWOPTMODE client environment definitions have been set are being executed.

    If PDUAPREPLVL and PDVWOPTMODE are not needed for a UAP, you should delete the PDUAPREPLVL and PDVWOPTMODE specifications and then execute the UAP.

- Identifying the SQL statement that is executing

  If the SQL object buffer is adequate, you can identify the SQL statement that is executing by referencing the statistical information by means of the procedure described below. To be able to identify the SQL statement that is executing, you must omit PDUAPREPLVL and PDVWOPTMODE and you must specify REUSE.

  1. Identify the SQL object whose STATUS is ACTIVE.
  2. Reference the SQL object identified in 1. In the case of a static SQL object, you can also identify the UAP that is executing the SQL object. Reference the SQL object's PREPROCESSOR USE and PREPROCESSOR SOURCE. The UAP using the preprocessing source is the one that is executing the SQL statement.

- About SQL objects subject to tuning

  Reference the items listed below. The SQL statements for which a large value is shown should be considered for tuning.

  - SQL object's execution count
  - SQL object's average execution time

- SQL object's maximum execution time
- Reference count for data, index, and directory pages
- Update count for data, index, and directory pages
- Real READ count for data, index, an directory pages
- Real WRITE count for data, index, and directory pages
- READ count for work table files
- WRITE count for work table files

For details about tuning, see the discussion of tuning SQL objects that take a long time to execute in the *HiRDB Version 8 System Operation Guide*.

pdopen (Open RDAREAs)

Function

The pdopen command opens specified RDAREAs.

Executor

HiRDB administrator

Format

```
pdopen -r {RDAREA-name[,RDAREA-name]...|ALL}
```

Options

- -r {*RDAREA-name*[,*RDAREA-name*]...|ALL}

-r *RDAREA-name* ～ <identifier> ((1-30))

Specifies the name of an RDAREA that is to be opened. You can also use batch specification of RDAREA names. For details about batch specification of RDAREA names, see *1.5.2 Batch specification of RDAREA names in operation commands*.

ALL

Specifies that all RDAREAs are to be opened, except for the master directory RDAREA.

Rules

1. An already open RDAREA cannot be opened again.
2. The same RDAREA name should not be specified more than once; if specified, a duplicate RDAREA name is ignored.
3. A maximum of 128 RDAREA names can be specified. If more than 128 RDAREA names are specified, the excess names are ignored.
4. If an RDAREA name is enclosed in \", the system treats it as being case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a space, enclose the entire name in \".
5. When a user RDAREA or user LOB RDAREA is specified, the corresponding data dictionary RDAREA must be in one of the following statuses:
   - Open and shutdown release status
   - Open and shutdown status placed by the pdhold command
6. If you specify data dictionary RDAREAs, user RDAREAs, and user LOB

RDAREAs at the same time, make sure that the data dictionary RDAREA names are specified last.

Rules

1. The `pdopen` command can be executed only while HiRDB is active.
2. The `pdopen` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. The `pdopen` command locks each specified RDAREA in the EX mode. If another transaction is accessing a specified RDAREA, the `pdopen` command goes onto wait status until the other transaction terminates.
4. If the `pdopen` command is executed on a shared RDAREA, all back-end servers are locked. If there can be multiple concurrent accesses to the corresponding RDAREA, global deadlock may occur, resulting in a timeout. If global deadlock has occurred, re-execute the `pdopen` command.

Notes

1. The result of the `pdopen` command can be checked by the `pddbls` command.
2. The following shows the `pdopen` command's return codes:

   `0`: Normal termination

   `4`: Warning termination (some RDAREA processing terminated with an error)

   `8`: Abnormal termination

   `12`: Abnormal termination (an event occurred that prevented output of an error message)

   If the error code is `12`, check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the command. If no error message has been output to the event log, contact the customer engineer.

pdpfresh (Refresh server process)

Function

The pdpfresh command refreshes the server process. Refreshing means terminating the active process and starting a new process.

This command is applicable to single servers, back-end servers, and dictionary servers.

Executor

HiRDB administrator

Format

```
pdpfresh [-s server-name] [{-f|-c}]
```

Options

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server to be refreshed.

You can specify a single server for a HiRDB/Single Server or a back-end server or dictionary server for a HiRDB/Parallel Server (you cannot specify a front-end server).

When this option is omitted, the command assumes all servers (except front-end servers).

- -f

Specifies that the server is to be refreshed forcibly.

Table 2-5 describes the process termination timing depending on whether or not the -f option is specified and the difference in possible errors.

Table 2-5: Process termination timing depending on whether or not the -f option is specified and difference in possible errors (pdpfresh command)

Process	-f option	
	Not specified	Specified
Single server, back-end server, or dictionary server process during utility's internal processing	If the pdpfresh command is executed during utility execution, the server process terminates with an error (return code = 6) without being refreshed. This has no effect on the utility's processing.	The server process is forcibly terminated when the transaction ends. In this case, the utility may terminate abnormally.

Process	-f option	
	Not specified	Specified
Processes of the following servers: • Single server locked with UNTIL DISCONNECT • Single server using a holdable cursor or back-end server • Back-end server using local buffer for each AP	The server process is not forcibly terminated. The process is terminated when it no longer satisfies the conditions listed at left or the next time DISCONNECT occurs. In the case of a single server, a connection error may occur between HiRDB client and HiRDB server. For a back-end server, no error occurs. Note that server process refreshing is not completed as long as the these processes exist.	The server process is terminated forcibly when the transaction ends. Thereafter, all SQL statements executed on this process will result in a connection error.
Process of the back-end server that is using the BES connection holding facility	If a transaction is being processed, the process is refreshed after the transaction ends. If no transaction is being processed, the process is refreshed after the next transaction ends.	The server process is terminated forcibly when the transaction ends. Thereafter, all SQL statements executed on these processes will result in an SQL connection error.

■ -c

Specifies that the command is to check whether or not all processes that received the process termination request have terminated. Specify this option when you want to monitor the process refreshing status using a batch file.

You can obtain the result from the pdpfresh command's return code (the result is also displayed in the KFPS00730-I message).

To check each process's refreshing status, execute the pdls -d prc -c command.

Rules

1. The pdpfresh command can be executed only while HiRDB is active.
2. The pdpfresh command must be executed at the server machine that contains the single server or where the system manager is located.
3. The pdpfresh command terminates without waiting for server process refreshing to be completed. If an error occurs during refreshing, an error message may be displayed after the pdpfresh command has terminated with return code 0.
4. If you execute one pdpfresh command and then another before process refreshing is completed, the second pdpfresh command refreshes the process that is active during the re-execution of the command

5. If the name of a front-end server is specified for a HiRDB/Parallel Server, the command terminates normally without refreshing.
6. A server process executing a transaction is terminated when the transaction is completed; it is not terminated immediately.
7. For a HiRDB/Single Server, a connection error may occur between HiRDB client and server during refreshing. To avoid this, add the processing for re-establishing connection to the UAP. If you use the automatic reconnect facility, connection is re-connected automatically; this error is not returned to the UAP. For details about the automatic reconnect facility, see the *HiRDB Version 8 UAP Development Guide*.

Notes

1. The result of the `pdpfresh` command can be checked with the `pdls -d prc` command or on the basis of the return code from execution of the command. The following table describes the return codes from command execution:

Return code	Execution result		Remarks
`0`	`-c` option specified	Refresh processing has started.	N/A
	`-c` option not specified	There is no process to be refreshed.	
`1`	`-c` option specified	N/A	Output only when the `-c` option is specified.
	`-c` option not specified	A process is being refreshed.	
`4`	Terminated with warning (error occurred at one ore more servers).*		Output only for a HiRDB/ Parallel Server.
`6`	Terminated without executing refresh processing because a utility was executing.		Output only when the `-f` option is not specified.
`8`	Abnormal termination		N/A

Legend: N/A: Not applicable

* If there is no response from one or more servers for some reason, the command continues processing for other servers and then terminates itself with return code `4`. In such a case, check the status of each server that resulted in an error.

If a server resulting in an error is active, RDAREAs may continue to be accessed by the applicable server process. If this is the case, re-execute the `pdpfresh` command specifying the name of that server in the `-s` option.

2. The `pdpfresh` command terminates without waiting for server process refreshing to be completed.

3. Refreshing is completed if one `pdpfresh` command with the `-c` option omitted terminates with return code `0` and then another `pdpfresh` command with the `-c` option specified terminates with return code `0`.
4. Termination of the `pdpfresh` command with the `-c` option specified with return code `0` means that no process was being refreshed during the command's execution (it does not mean that refresh processing was executed and then was completed). If you execute the `pdpfresh` command with the `-c` option specified without executing the `pdpfresh` command with the `-c` option omitted, the command terminates with return code `0`, but refresh processing does not occur.
5. When you use the `pdpfresh` command for server process refreshing, the `KFPS01820-E`, `KFPS04620-I`, and `KFPS01819-I` messages are not displayed.
6. During refresh processing, the system temporarily sets the number of resident processes to 0. Therefore, do not execute the `pdpfresh` command on a server while it is being refreshed. If you do, refresh processing may not be completed. If this happens, re-execute the `pdpfresh` command, wait until refresh processing is completed, and then execute the `pdchprc` command.

pdplgrgst (Register and delete plug-ins)

Function

- Registers plug-ins with HiRDB.

  Specifies the plug-in definition and the PIC files for the plug-in to be registered.

  Executing the `pdplgrgst` command registers plug-in-related definition information, as well as abstract data type definitions or index-type definitions, in a dictionary.

  Each execution of the `pdplgrgst` command registers one plug-in.

- Deletes a plug-in from HiRDB.

  The command deletes the following items:

  - Definition information about plug-ins that are registered in a data dictionary
  - Functions, abstract data types, and index data types that are supplied by a plug-in

- Upgrades plug-in versions.

  Updates the definition information, functions, abstract data type definitions, and index data type definitions of previously registered plug-ins.

Executor

Users with `DBA` privileges can execute this command.

Format

```
pdplgrgst [-{d|a} [-u]
          plug-in-definition-filename  PIC-filename
```

Options

- `-{d|a}`

Specify this option when deleting a plug-in or upgrading a plug-in to a new version. The default is to register a plug-in with HiRDB.

`-d`

This option deletes a plug-in.

`-a`

This option upgrades a plug-in to a new version.

- `-u`

Specifies that the plug-in's owner is to be set to the executor of the pdplgrgst command (to the authorization identifier specified in PDUSER in the client environment definitions) rather than to MASTER.

When specifying a plug-in owner other than MASTER, note the following:

1. The pdplgrgst command executor's schema must already be defined.
2. If the plug-in provides both abstract data and index types, be sure to specify the same owner.
3. Only a plug-in owner can delete or upgrade his/her plug-in. To delete or upgrade a plug-in, specify the -u option in the pdplgrgst command.
4. If you change the plug-in owner during upgrading, the functions may become invalid. In such cases, use ALTER ROUTINE to re-create the SQL objects for the functions.
5. If you delete a plug-in owner's schema, his/her plug-in is also deleted. In this case, delete the pdplugin operand from the system common definitions.

Command arguments

- *plug-in-definition-filename*

Specifies the file name of a plug-in definition file.

Specify a file name in the command input current directory or a file name including its absolute path.

- *PIC-filename*

Specifies the file name of a PIC file.

Specifies a file name in the command input current directory or a file name including its absolute path.

Rules

1. The pdplgrgst command can be executed only when HiRDB is running.
2. Execute the pdplgrgst command on an appropriate server machine.
3. To execute the pdplgrgst command, you must have already set PDHOST, PDNAMEPORT, and PDUSER in the client environment definition. For PDUSER, specify the authorization identifier and password of a user with DBA privileges.

Notes

1. The results of execution of the pdplgrgst command can be checked by reviewing the return code generated from the command execution process. A command execution return code of 0 indicates normal termination; 8 indicates an abnormal termination.

2. When executing the pdplgrgst command, be sure to install the plug-in in all the HiRDB utilization environments.
3. To use a plug-in that was registered in the HiRDB system using the pdplgrgst command, code the name of the plug-in used in system common definitions, and then restart the HiRDB system.
4. When a plug-in is being registered, no other operations can be performed on the schema associated with the plug-in.
5. If the pdplgrgst command terminates abnormally during plug-in registration, a part of the type definition process may terminate prematurely. If this happens, delete the plug-in by executing the pdplgrgst command with the -d option and re-register the plug-in.
6. When deleting a plug-in, delete the following items before executing the pdplgrgst command:
   - Any tables that used this plug-in
   - Any indexes that used this plug-in
   - Any functions, procedures, and abstract data types that used the abstract data types, index types, and functions supplied by this plug-in
7. When deleting a data-type plug-in, if the index-data-type plug-in that supplies the indexing function of the data type is also registered, delete the index-data-type plug-in first. For details, refer to the respective plug-in manuals.
8. After deleting a plug-in, delete coding of the system common definition pdplugin.
9. When upgrading a plug-in to a new version, back up the master directory RDAREA, the data dictionary RDAREA, the data directory RDAREA, and the data dictionary LOB RDAREA before executing the pdplgrgst command.
10. A plug-in cannot be reset to a version older than the plug-in version registered.
11. When upgrading both a data-type plug-in and an index-type plug-in that supply the indexing function for that data type, upgrade the data-type plug-in first.
12. If the HiRDB/Single Server structure is changed to a HiRDB/Parallel Server structure by using the database structure modification utility (pdmod), plug-in-supplying functions may become invalidated depending on the specific plug-in registered. If this happens, re-execute the pdplgrgst command by specifying the -a option.

pdprgcopy (Copy a HiRDB update version)

Function

Before the pdprgrenew command is executed, the pdprgcopy command is used to copy HiRDB programs from the HiRDB update version installation directory to the update directory (%PDDIR%\renew). For details about upgrading to a HiRDB update version, see the *HiRDB Version 8 Installation and Design Guide*. The following files are copied to the update directory:

HiRDB update version installation directory (source)	Update directory (target)	Description
installation-directory\bin	%PDDIR%\renew\bin	HiRDB operation commands and utilities
installation-directory\lib	%PDDIR%\renew\lib	HiRDB libraries, message text files, and system definition analysis files
installation-directory\client	%PDDIR%\renew\client	HiRDB client libraries, operation commands, and utilities
installation-directory\include	%PDDIR%\renew\include	Header files used during UAP creation
installation-directory\HiRDEF	%PDDIR%\renew\HiRDEF	HiRDB definition support
installation-directory\sample	%PDDIR%\renew\sample	Sample database facility
installation-directory\pdistup	%PDDIR%\renew\pdistup	HiRDB setup tool

Executor

HiRDB administrator

Format

```
- HiRDB-update-version-installation-directory\bin\pdprgcopy -HiRDB-directory-
```

Command arguments

- *HiRDB-update-version-installation-directory* ～ <path name> ((1-200))

Specifies the HiRDB directory of the HiRDB that is to be updated to the HiRDB update version.

Rules

1. The pdprgcopy command can be executed at any time, whether or not HiRDB is active.

2. The pdprgcopy command must be executed at the server machine that contains the single server or where the system manager is located.
3. You cannot execute the pdprgcopy command if the following information does not match between the existing HiRDB and HiRDB update version:
   - Type of HiRDB (HiRDB/Single Server or HiRDB/Parallel Server)
   - Version and revision
   - Addressing mode (32-bit or 64-bit mode)

Notes

1. The pdprgcopy command creates an update directory (%PDDIR%\renew). If this directory already exists, rename it beforehand.
2. Before executing the pdprgcopy command, make sure that the HiRDB directory has enough space.
3. Before executing the pdprgcopy command, you must install the HiRDB update version on all units.
4. The existing HiRDB and the HiRDB update version will have the same character codes type.
5. The result of the pdprgcopy command can be checked on the basis of the return code from execution of the command. A return value of 0 for this command indicates normal termination; 8 indicates abnormal termination. When the return code is 8, see the previous message and eliminate the cause of the error.

pdprgrenew (Update a HiRDB update version)

Function

The pdprgrenew command updates the active HiRDB to the HiRDB update version (HiRDB in the update directory that has been copied by the `pdprgcopy` command (`%PDDIR%\renew`)). For details about upgrading to a HiRDB update version, see the *HiRDB Version 8 Installation and Design Guide*.

The command updates the files in the following HiRDB directories on the active HiRDB:

Directory	Description
`%PDDIR%\renew\bin`	HiRDB operation commands and utilities
`%PDDIR%\renew\lib`	HiRDB libraries, message text files, and system definition analysis files
`%PDDIR%\renew\client`	HiRDB client libraries, operation commands, and utilities
`%PDDIR%\renew\include`	Header files used during UAP creation
`%PDDIR%\renew\HiRDEF`	HiRDB definition support
`%PDDIR%\renew\sample`	Sample database facility
`%PDDIR%\renew\pdistup`	HiRDB setup tool

Executor

HiRDB administrator

Format

```
- HiRDB-update-version-installation-directory\bin\pdprgrenew-- [-b] HiRDB-directory-
```

Options

- `-b`

Specifies that HiRDB is to be restored to the status in effect before the `pdprgrenew` command was executed, such as when the `pdprgrenew` command terminates abnormally with `KFPS04646-I` (return code = `12`).

Command arguments

- *HiRDB-update-version-installation-directory* ～ <path name> ((1-200))

Specifies the HiRDB directory of the HiRDB that is to be updated to the HiRDB update version.

Rules

1. The pdprgrenew command can be executed only while HiRDB is active.
2. The pdprgrenew command must be executed at the server machine that contains the single server or where the system manager is located.
3. Before executing the pdprgrenew command, you must use the pdprgcopy command to create an update directory (%PDDIR%\renew). You cannot execute the pdprgrenew command if you haven't created this directory. Note that the update directory (%PDDIR%\renew) is deleted when the pdprgrenew command terminates normally.

Notes

1. The pdprgrenew command creates a backup directory (%PDDIR%\renew_bak). If this directory already exists, rename it beforehand.
2. Before executing the pdprgrenew command, exit any of the following programs that are running:
   - HiRDB command prompt
   - Event viewer
3. When you execute the pdprgrenew command, system log files are swapped. Before executing the pdprgrenew command, make sure that there is enough space for operating with the updated HiRDB. If you execute the pdprgrenew command while there is no swappable system log file, HiRDB displays the KFPS01256-E message and stops (displays abort code Psjnf07 or Psjn381). If HiRDB has stopped, provide a swappable system log file and then execute the pdprgrenew -b command. If there is no swappable system log file, you must add one. For details about how to handle when there is no swappable file, see the *HiRDB Version 8 System Operation Guide*.
4. Although system log files are swapped when you execute the pdprgrenew command, no message to that effect is displayed. Therefore, if you need the message log in the message log file, back up the message log file before executing the pdprgrenew command.
5. If the active HiRDB is in the following status, you may not be able to update your HiRDB:
   - For a HiRDB/Parallel Server, some of the units or servers have terminated (including units that have not started due to reduced activation).
   - For a HiRDB/Parallel Server, a communication error has occurred in the network between units.

- A transaction or utility remains active 15 minutes after the `pdprgrenew` command was executed.

6. The following limitations apply while the `pdprgrenew` command is executing:
   - A UAP's response time may be delayed by the `pdprgrenew` command's execution time.
   - If you execute another operation command or utility, execution of the `pdprgrenew` command may fail or an error message indicating that HiRDB has stopped may be displayed.
7. All files in the HiRDB directory must be in the same HiRDB file system area as the HiRDB directory.
8. The result of the `pdprgrenew` command can be checked on the basis of the return code from execution of the command. A return value of `0` for this command indicates normal termination; `8` indicates abnormal termination (HiRDB status before updating); and `12` indicates abnormal termination (termination of HiRDB).
9. During execution of the `pdprgrenew` command, error messages and abort codes relating to HiRDB startup and termination may be displayed. In this happens, there may be a problem in the environment, such as a shortage of resources required for starting the updated HiRDB. Take appropriate action according to the displayed error messages and abort code.

pdrbk (Roll back transactions)

Function

The pdrbk command forcibly rolls back transactions when none of the transactions that are part of a global transaction could be concluded (for example, because of a communications error). All the transactions in the global transaction must be rolled back, otherwise they will not be synchronized.

The pdrbk command is used to forcibly roll back transactions that are shown in the status information displayed by the pdls -d trn command as being in READY status (Status 1) and p status (Status 2).

Executor

HiRDB administrator

Format

```
pdrbk {-x host-name|-u unit-identifier} [-s server-name]
      {-A|-t transaction-identifier}
```

Options

- {-x *host-name*|-u *unit-identifier*}

Specifies the identifier of the unit or the name of the host in which the transactions to be rolled back are located.

-x *host-name* ～ <identifier> ((1-32))

Specifies the name of the host in which the transactions to be rolled back are located.

-u *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit in which the transactions to be rolled back are located.

- -s *server-name* ～ <identifier> ((1-8))

Specifies the server name when the unit has multiple servers. If this option is omitted, all servers in the object host are subject to the pdrbk command.

- -A

Specifies that all transactions in the server specified by the -s option that are in READY status and recovery waiting status are to be rolled back.

- -t *transaction-identifier* ～ <alphanumeric characters> ((16 characters))

Specifies that the transaction identifier of a transaction in READY status and recovery

processing waiting status is to be rolled back.

Rules

1. The `pdrbk` command can be executed only while HiRDB is active.
2. The `pdrbk` command must be executed at each server machine.

Note

1. The results of the `pdrbk` command can be checked on the basis of the return code from execution of the `pdls` command (with `-d trn` specified) or by checking for any error messages. Return code `0` for the `pdrbk` command indicates normal termination, and return code `1` indicates abnormal termination (such as an invalid option or `rsh` error).
2. Forced rollback may not be possible for a transaction that is connected to a recovery-unnecessary front-end server and executed, even if transaction status 1 is `READY` and transaction status 2 is `p`. In such a case, complete the transaction automatically. For details about how to complete a transaction that is in uncompleted status for transaction determination, see the manual *HiRDB Version 8 System Operation Guide*.

pdrdrefls (Display information about related RDAREAs)

Function

The pdrdrefls command displays information about related RDAREAs.

The pdrdrefls command has the following two functions:

1. Displaying information about related RDAREAs
2. Checking the relationships among RDAREAs

Executor

HiRDB administrator

Format

- Displaying information about related RDAREAs

```
pdrdrefls [-k dsp] -e org
        {-r RDAREA-name[,RDAREA-name]...
          |-t [authorization-identifier.]table-identifier}
        [-s server-name[,server-name]...] [-l [-d delimiter]] [-a][-c constraint-type]
```

- Checking the relations among RDAREAs

```
pdrdrefls -k chk -e org -r RDAREA-name[,RDAREA-name]...
        [-s server-name[,server-name]...][-c constraint-type]
```

Options

- -k {dsp|chk}

Specifies the function of the pdrdrefls command that is to be executed.

dsp:

Display information about all RDAREAs that store a specified table or all RDAREAs that are related to a specified resource (table, index, LOB column).

chk:

Check the relationships among RDAREAs. If related RDAREAs are missing, this

option displays a message providing information about the missing RDAREAs.

- `-e org`

This option is fixed.

- `-r` *original-RDAREA-name*`[,`*original-RDAREA-name*`]...` ～ <identifier> ((1-30))

Specifies the names of the RDAREAs to be processed. Specify user RDAREAs or user LOB RDAREAs.

If an RDAREA name is enclosed in double quotation marks (`"`), the system treats it as being case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a space, enclose the entire name in double quotation marks (`"`). If you are using `sh` (Bourne shell), `csh` (C shell), or `ksh` (Korn shell), you must enclose the entire set of RDAREA names in single quotation marks (`'`).

- `-t [`*authorization-identifier*`.]`*table-identifier*}

Specifies the name of a table to be processed.

When the authorization identifier is omitted, the command assumes the authorization identifier of the `PDUSER` environment variable at the time of command execution. If the `PDUSER` environment variable has not been set, the command assumes the user name in the logon window.

If an authorization identifier and a table identifier are enclosed in double quotation marks (`"`), the command treats them as being case sensitive. If they are not enclosed in double quotation marks (`"`), the command treats them as in all uppercase letters. If you are using `sh` (Bourne shell), `csh` (C shell), or `ksh` (Korn shell), you must enclose this information in single quotation marks (`'`).

- `-s` *server-name*`[,`*server-name*`]...` ～ <identifier> ((1-8))

Specifies the names of servers to be processed. You can specify a maximum of 128 server names.

When `-k dsp` is specified, the command displays only information that has been defined in the specified servers.

When `-k chk` is specified, the command checks to see whether all the RDAREAs specified in the `-r` option have been defined in the specified servers. If any of the RDAREAs specified in the `-r` option does not satisfy this condition, the command displays the applicable RDAREA name in a message and then terminates with an error.

You can check the server names to be specified using `pddbst`'s condition analysis in units of RDAREAs (logical analysis) or in units of tables.

- `-l`

Specifies that the output information is to be displayed without linefeeds.

- `-d` *delimiter*

Specifies a single-byte character as the delimiter for the information that is output when the `-l` option is specified.

When this option is omitted, the command assumes a space.

To specify a special character as the delimiter, such as a single quotation mark (`'`) or double quotation mark (`"`), specify `\'` or `\"`. To specify the escape character as the delimiter, specify `\\`.

- `-a`

Specifies that all related information (server names, RDAREA names, RDAREA types, and resource information) is to be displayed. When this option is omitted, the command display only the RDAREA names.

- `-c` *constraint-type*

Specifies that referential RDAREA information associated with referential constraints is to be displayed or checked.

`ref`

Specifies that the RDAREA information associated with referential constraints is to be displayed or checked. For details about how to handle referential constraints, see the manual *HiRDB Version 8 Installation and Design Guide*.

Rules

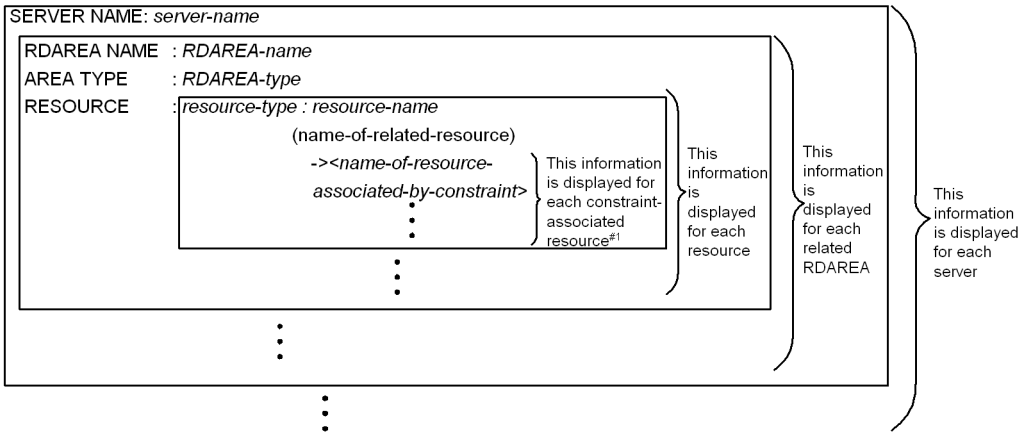
1. The `pdrdrefls` command can be executed only while HiRDB is active.
2. The `pdrdrefls` command must be executed at the server machine that contains the single server or where the system manager is located.

Notes

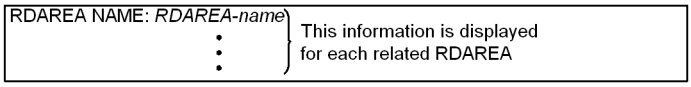
1. The result of the `pdrdrefls` command can be checked on the basis of the return code from execution of the command. Return code `0` for the `pdrdrefls` command indicates normal termination, return code `4` indicates warning termination (there is no resource to be analyzed), and return code `8` indicates abnormal termination.
2. Do not add to, delete, or change the definition of a table or index for an RDAREA that is to be processed by the `pdrdrefls` command. If you do so, the command's execution result may be invalid or the command may terminate abnormally.

Output format

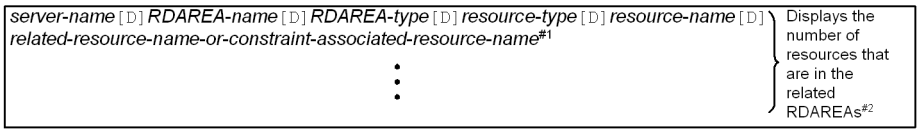
・When -l is omitted and -a is specified



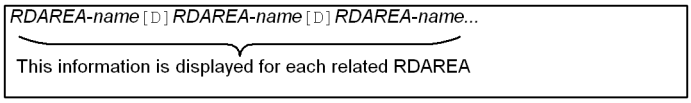
・When -l and -a are both omitted



・ When -l and -a are both specified



・When -l is specified and -a is omitted



Legend: [D] : Delimiter
#1: This information is displayed when the -c option is specified.
#2: When there are multiple related resources or constraint-associated resources, this information is displayed for each resource.

Explanation

server-name (maximum of 8 bytes):

Name of a server that contains related RDAREAs.

RDAREA-name (maximum of 32 bytes):

Name of a related RDAREA. The RDAREA name is enclosed in double quotation marks (").

RDAREA-type (maximum of 8 bytes):

Type of RDAREA:

USER: User RDAREA

USER_LOB: User LOB RDAREA

resource-type (3 bytes):

Type of resource:

TBL: Table

IDX: Index

LOB: LOB column

resource-name (maximum of 43 bytes):

Name of the resource that is stored in the corresponding RDAREA. The resource name is enclosed in double quotation marks.

When the resource type is TBL: *authorization-identifier*.*table-identifier*

When the resource type is IDX: *authorization-identifier*.*index-identifier*

When the resource type is LOB: LOB column name or abstract data type column name of LOB attribute

name-of-related-resource (maximum of 45 bytes):

Name of the resource related to the resource with type IDX or LOB. The related resource name is displayed in the format *authorization-identifier*.*table-identifier* and is enclosed in double quotation marks.

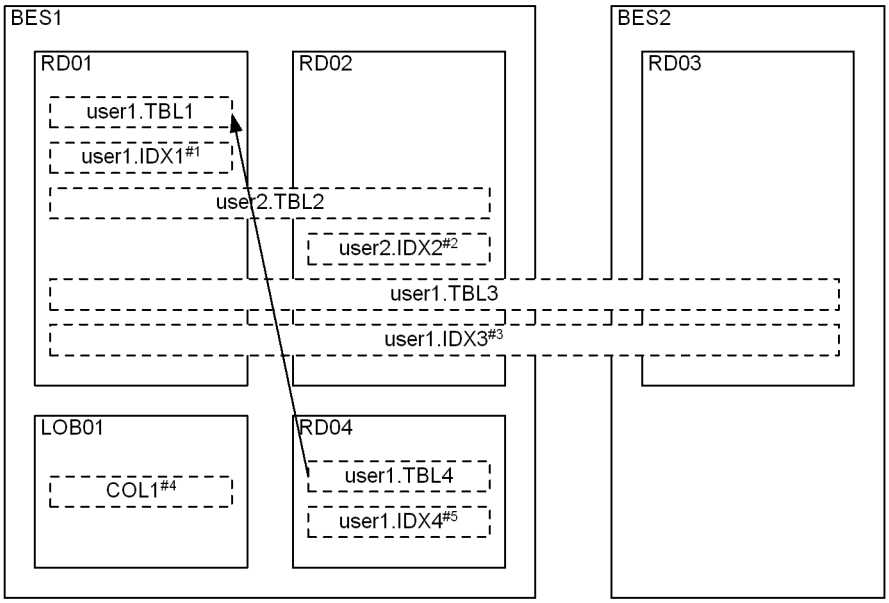
name-of-resource-associated-by-constraint (maximum of 47 bytes):

Name of the resource with type TBL that is associated by the constraint definition. This information is displayed only when the -c option is specified. When -c ref is specified, the name of the referenced table is displayed. A constraint-associated resource name is displayed enclosed in double quotation marks (").

Examples

This example displays information about related RDAREAs. The example omits the -d option. In the execution result, Δ indicates a single-byte space.

- RDAREA configuration



Legend:

→ : Indicates a referential constraint. The arrow points from a referencing table to a referenced table.

#1: Index defined for `user1.TBL1`
#2: Index defined for `user2.TBL2` (non-partitioning key index)
#3: Index defined for `user1.TBL3` (partitioning key index)
#4: LOB column structure base table is `user1.TBL1`.
#5: Index defined for `user1.TBL4`

■ Execution result

- When `pdrdrefls -e org -r RD01 -a` is specified

```
SERVER NAME : BES1
  RDAREA NAME : "RD01"
  AREA TYPE : USER
    RESOURCE  : TBL : "user1"."TBL1"
                      "user2"."TBL2"
                      "user1"."TBL3"
                IDX : "user1"."IDX1"
                      ("user1"."TBL1")
                      "user1"."IDX3"
                      ("user1"."TBL3")
  RDAREA NAME : "RD02"
    AREA TYPE : USER
    RESOURCE  : TBL : "user2"."TBL2"
                      "user1"."TBL3"
```

```
                IDX : "user2"."IDX2"
                      ("user2"."TBL2")
                      "user1"."IDX3"
                      ("user1"."TBL3")
RDAREA NAME : "RD04"                    -
  AREA TYPE : USER                       |
  RESOURCE  : TBL : "user1"."TBL4"      |-#
                    -><"user1"."TBL1"> |
                IDX : "user1"."IDX4"     |
                     ("user1"."TBL4")   -
RDAREA NAME : "LOB01"
  AREA TYPE : USER_LOB
  RESOURCE  : LOB : "COL1"
                    ("user1"."TBL1")
```

Displayed only when `-c ref` is specified.

- When `pdrdrefls -e org -r RD01` is specified

```
RDAREA NAME : "RD01"
RDAREA NAME : "RD02"
RDAREA NAME : "RD04" #
RDAREA NAME : "LOB01"
```

Displayed only when `-c ref` is specified.

- When `pdrdrefls -e org -r RD01 -l -a` is specified

```
BES1△"RD01"△USER△TBL△"user1"."TBL1"
BES1△"RD01"△USER△TBL△"user2"."TBL2"
BES1△"RD01"△USER△TBL△"user1"."TBL3"
BES1△"RD01"△USER△IDX△"user1"."IDX1"△"user1"."TBL1"
BES1△"RD01"△USER△IDX△"user1"."IDX3"△"user1"."TBL3"
BES1zueng020.tif"LOB01"zueng020.tifUSER_LOBzueng020.tifLOBz
ueng020.tif"COL1"zueng020.tif"user1"."TBL1"
BES1△"RD02"△USER△TBL△"user2"."TBL2"
BES1△"RD02"△USER△TBL△"user1"."TBL3"
BES1△"RD02"△USER△IDX△"user2"."IDX2"△"user2"."TBL2"
BES1△"RD02"△USER△IDX△"user1"."IDX3"△"user1"."TBL3"
BES1zueng020.tif"RD04"zueng020.tifUSERzueng020.tifTBLzueng0
20.tif"user1"."TBL4"zueng020.tif
"user1"."TBL1" #
BES1zueng020.tif"RD04"zueng020.tifUSERzueng020.tifIDXzueng0
20.tif"user1"."IDX4"zueng020.tif
"user1"."TBL4" #
```

Displayed only when `-c ref` is specified.

- When `pdrdrefls -e org -r RD01 -l` is specified

```
"RD01"Δ"RD02"Δ"LOB01"Δ"RD04" #
```

Displayed only when `-c ref` is specified.

- When `pdrdrefls -e org -r RD02 -a` is specified

```
SERVER NAME : BES1
  RDAREA NAME : "RD02"
    AREA TYPE : USER
    RESOURCE  : TBL : "user2"."TBL2"
                      "user1"."TBL3"
                IDX : "user2"."IDX2"
                      ("user2"."TBL2")
                      "user1"."IDX3"
                      ("user1"."TBL3")
  RDAREA NAME : "RD01"
    AREA TYPE : USER
    RESOURCE  : TBL : "user1"."TBL1"
                      "user2"."TBL2"
                      "user1"."TBL3"
                IDX : "user1"."IDX1"
                      ("user1"."TBL1")
                      "user1"."IDX3"
                      ("user1"."TBL3")
  RDAREA NAME : "RD04"                    -
    AREA TYPE : USER                       |
    RESOURCE  : TBL : "user1"."TBL4"       |-#
                      -><"user1"."TBL1">   |
                IDX : "user1"."IDX4"       |
                     ("user1"."TBL4")    -
  RDAREA NAME : "LOB01"
    AREA TYPE : USER_LOB
    RESOURCE  : LOB : "COL1"
                      ("user1"."TBL1")
```

Displayed only when `-c ref` is specified.

- When `pdrdrefls -e org -r RD03 -a` is specified

```
SERVER NAME : BES2
  RDAREA NAME : "RD03"
    AREA TYPE : USER
```

```
    RESOURCE  : TBL : "user1"."TBL3
                IDX : "user1"."IDX3"
                      ("user1"."TBL3")
```

- When `pdrdrefls -e org -r RD04 -a` is specified

```
SERVER NAME : BES1
  RDAREA NAME : "RD04"
    AREA TYPE : USER
    RESOURCE  : TBL : "user1"."TBL4"
                      -><"user1"."TBL1"> #
                IDX : "user1"."IDX4"
                      ("user1"."TBL4")
  RDAREA NAME : "RD01"                  -
    AREA TYPE : USER                     |
    RESOURCE  : TBL : "user1"."TBL1"     |
                      "user2"."TBL2"     |
                      "user1"."TBL3"     |
                IDX : "user1"."IDX1"     |
                      ("user1"."TBL1")   |
                      "user1"."IDX3"     |
                      ("user1"."TBL3")   |
  RDAREA NAME : "RD02"                   |-#
    AREA TYPE : USER                     |
    RESOURCE  : TBL : "user2"."TBL2"     |
                      "user1"."TBL3"     |
                IDX : "user2"."IDX2"     |
                      ("user2"."TBL2")   |
                      "user1"."IDX3"     |
                      ("user1"."TBL3")   |
  RDAREA NAME : "LOB01"                  |
    AREA TYPE : USER_LOB                 |
    RESOURCE  : LOB   : "COL1"           |
                ("user1"."TBL1")       -
```

Displayed only when `-c ref` is specified.

pdrels (Release RDAREAs from shutdown status)

Function

The pdrels command releases specified RDAREAs from shutdown status.

Executor

HiRDB administrator

Format

```
pdrels -r {RDAREA-name[,RDAREA-name]...|ALL} [-o]
```

Options

- -r {*RDAREA-name*[,*RDAREA-name*]...|ALL}

-r *RDAREA-name* ～ <identifier> ((1-30))

Specifies the name of an RDAREA that is to be released from shutdown status. You can also use batch specification of RDAREA names. For details about batch specification of RDAREA names, see *1.5.2 Batch specification of RDAREA names in operation commands*.

ALL

Specifies that all RDAREAs are to be released from shutdown status, except the master directory RDAREA.

Rules

1. The master directory RDAREA cannot be specified.
2. The same RDAREA name cannot be specified more than once; if an RDAREA name is duplicated, the duplicate RDAREA name is ignored.
3. A maximum of 128 RDAREA names can be specified. If more than 128 RDAREA names are specified, the excess names are ignored.
4. If an RDAREA name is enclosed in \", the system treats it as being case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a space, enclose the entire name in \".
5. When a user RDAREA or user LOB RDAREA is specified, the corresponding data dictionary RDAREA must be in one of the following statuses:
   - Open and shutdown release status
   - Open and shutdown status placed by the pdhold command

- -o

Specifies that the RDAREA is to be opened after being released from shutdown status. When this specification is omitted, the RDAREA is only released from shutdown status.

Rules

1. The `pdrels` command can be executed only while HiRDB is active.
2. The `pdrels` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. The `pdrels` command locks each specified RDAREA in the PR mode. If another updating transaction is accessing a specified RDAREA, the `pdrels` command goes onto wait status until the other transaction terminates.
4. If the `pdrels` command is executed on a shared RDAREA, all back-end servers are locked. If there can be multiple concurrent accesses to the corresponding RDAREA, global deadlock may occur, resulting in a timeout. If global deadlock has occurred, re-execute the `pdrels` command.

Notes

1. The result of the `pdrels` command can be checked by the `pddbls` command.
2. If the contents of the update buffer take effect on the RDAREA on updatable backup-hold (`WAIT` mode), the command issues a warning message (`KFPH00157-W`) when releasing the RDAREA from shutdown status.
3. The following shows the `pdrels` command's return code:

   `0`

   Normal termination

   `1`

   Warning termination

   The contents of the update buffer took effect on the RDAREA on updatable backup-hold (`WAIT` mode). Shutdown release processing terminated normally.

   `4`

   Warning termination

   At least one of the specified RDAREAs resulted in an error.

   `8`

   Abnormal termination

   `12`

Abnormal termination

Check the error message in the event log at the host where the single server or dictionary server is located, eliminate the cause of the error, and then re-execute the command. If no error message has been output to the event log, contact the customer engineer.

4. This command cannot release an RDAREA from error shutdown status unless you recover the RDAREA. However, you can release an RDAREA from shutdown status if the status was caused by an input or output error during a read or write operation or by a UAP or utility in the no-log mode.

5. This command cannot release from command shutdown status an RDAREA that contains a falsification prevented table and is in reload-not-completed data status. When you execute table reloading with `pdrorg` on such a falsification prevented table, the reload-not-completed data status is released. After that, release the RDAREA from shutdown status.

pdrplstart (Start HiRDB Datareplicator linkage)

Function

The pdrplstart command starts HiRDB Datareplicator linkage.

Executor

HiRDB administrator

Format

```
pdrplstart
```

Option

None

Rules

1. The pdrplstart command can be executed only while all servers and all units in the HiRDB system are active. For example, this command cannot be executed for a HiRDB/Parallel Server if some units or servers are inactive or are performing processing in order to become inactive. However, you can execute the pdrplstart command if the only inactive units are ones that are not started due to reduced activation.
2. The pdrplstart command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. To execute the pdrplstart command, the HiRDB directory name should be set in the PDDIR environment variable. If the directory name is not set, no error message will be output if a command error occurs.
4. An error results if the pdrplstart command is entered again while HiRDB Datareplicator linkage is executing.
5. Once you execute the pdrplstart command, do not execute the pdrplstart, pdrplstop, or pdstop command (except when the -f or -z option is specified) while command processing is underway.
6. If you execute the pdrplstart command while database updating is underway, the update data may not be applied at the intended time. Execute the pdrplstart command after all transactions have terminated. Do not execute any new transactions until the current pdrplstart command has terminated.

Notes

1. The results of the pdrplstart command can be checked on the basis of the return code and by whether or not there are any error messages.

A return code of 0 for this command indicates normal termination; a return value of 8 indicates abnormal termination.

2. If a communication error occurs while the pdrplstart command is executing, some units may be left in a status with HiRDB Datareplicator linkage started, and the command may terminate abnormally with a return value of 8. In such a case, execute the pdrplstop -f command to stop HiRDB Datareplicator linkage at all units, and then re-execute the pdrplstart command.
3. Agreement between the extraction-side and target-side databases must be maintained when the pdrplstart command is executed. If HiRDB Datareplicator linkage is started with the databases not in agreement, the contents of the target database cannot be guaranteed.
4. Once the pdrplstart command is entered, pdrplstart and pdrplstop commands must not be entered until the entered pdrplstart command's processing has been completed.
5. If you execute the pdrplstart command while HiRDB Datareplicator cannot be linked, the unit that cannot start data linkage may terminate abnormally with abort code Psjiz09. Therefore, do not execute the pdrplstart command when linkage to HiRDB Datareplicator cannot be established.

pdrplstop (Stop HiRDB Datareplicator linkage)

Function

The pdrplstop command stops HiRDB Datareplicator linkage.

Executor

HiRDB administrator

Format

```
pdrplstop [-f]
```

Option

- -f

Specifies that HiRDB Datareplicator linkage is to be stopped forcibly regardless of the system log extraction status. When this option is specified, the command stops HiRDB Datareplicator linkage even if there are inactive units.

Rules

1. The pdrplstop command can be executed only while HiRDB is active.
2. The pdrplstop command must be executed at the server machine that contains the single server or where the system manager is located.
3. To execute the pdrplstop command, the HiRDB directory name should be set in the PDDIR environment variable. If the directory name is not set, no error message is output if a command error occurs.
4. The pdrplstop command with no option specified terminates HiRDB Datareplicator linkage once extraction by HiRDB Datareplicator of system log information has been completed. This command results in an error if there is an inactive unit. However, if all such inactive units are one that were not started due to reduced activation, the command terminates HiRDB Datareplicator linkage on the active units. For the units that have not been started due to reduced activation, the command terminates their HiRDB Datareplicator linkage the next time the units are started and extraction of system log information is completed.
5. If data is to be extracted from a HiRDB/Parallel Server, you can execute the pdrplstop -f command, even when other units are inactive, as long as the unit containing the system manager is active. For a unit that is inactive during execution of the pdrplstop -f command, the HiRDB Datareplicator linkage is cancelled the next time the unit is started.
6. An error results if the pdrplstop command is entered while HiRDB Datareplicator linkage is not executing.

Notes

1. The result of the pdrplstop command can be checked on the basis of the return code from execution of the command or by checking for error messages. The following table describes the return codes from execution of the command:

Return code	Termination status	Description
0	Normal termination	• When the -f option is omitted: HiRDB Datareplicator linkage was stopped. • When the -f option is specified: Termination of HiRDB Datareplicator linkage was instructed.
4	Abnormal termination	The command was unable to stop HiRDB Datareplicator linkage for one of the following reasons: • Some units were inactive (except ones that had not been started due to reduced activation) • Some log information was not extracted by HiRDB Datareplicator.
8	Abnormal termination	The command was unable to stop HiRDB Datareplicator linkage due to an error, or the pdrplstop command was executed illegally.

2. The pdrplstop command does not check the HiRDB server subject to HiRDB Datareplicator linkage for active transactions. Therefore, before executing the pdrplstop command, make sure that there are no active transactions.
3. If the source database to be extracted is updated during or after execution of the pdrplstop command, inconsistency occurs in the data between the source and target databases. The same also applies if the pdrplstop -f command is executed while there are active transactions or some log information has not been extracted for the source database.

   If inconsistency occurs in the data between the source and target databases, execute the pdrplstop command to terminate HiRDB Datareplicator linkage. Use the pdls -d rpl command to make sure that HiRDB Datareplicator linkage has stopped on all units, and then re-create the target database before restarting HiRDB Datareplicator linkage. For details about re-creating the target database, see the manual *HiRDB Datareplicator Version 8 Description, User's Guide and Operator's Guide*.
4. Once you have entered the pdrplstop command, do not enter the pdrplstart, pdrplstop, or pdstop command until the pdrplstop command processing terminates.
5. With a HiRDB/Parallel Server that consists of multiple units, normal termination of the pdrplstop -f command does not mean that HiRDB Datareplicator linkage has ended on all units. Once the pdrplstop command has terminated normally, use the pdls command (-d rpl) to make sure that HiRDB

Datareplicator linkage has ended on all units.

6. Once you have started HiRDB Datareplicator linkage, make sure that you use the `pdrplstop` command to stop it even if the reason for stopping the linkage is a missing database due to an error in the HiRDB system definitions.
7. If you are executing the `pdrplstop` command in order to restart HiRDB Datareplicator linkage, do not update the target database while HiRDB Datareplicator linkage is stopped. If you do, conformity will be lost between the databases. However, reorganization of a database that does not accompany logical updating of data does not cause inconsistency of data between databases even if it is executed while HiRDB Datareplicator linkage is stopped.
8. The table below describes the conditions and whether or not the `pdrplstop` command can be executed. If inconsistency occurs between the databases, you must re-create the target database.

Condition during execution of the pdrplstop command	Whether or not executable	
	-f not specified	-f specified
Unit containing the single server or system manager is inactive	N	N

<table>
<tr><th colspan="6" rowspan="2">Condition during execution of the pdrplstop command</th><th colspan="2">Whether or not executable</th></tr>
<tr><th>-f not specified</th><th>-f specified</th></tr>
<tr><td rowspan="6">Active</td><td colspan="5">There is unextracted log data after command execution</td><td>YI</td><td>YI</td></tr>
<tr><td rowspan="5">There is no unextracted log data</td><td colspan="4">There is unextracted log data at an active back-end server</td><td>N</td><td>YI</td></tr>
<tr><td rowspan="4">There is no unextracted log data</td><td colspan="3">There is no inactive back-end server</td><td>Y</td><td>Y</td></tr>
<tr><td rowspan="3">There are inactive back-end servers</td><td rowspan="2">Unit contains an inactive back-end server that was not been started because of reduced activation</td><td>There is unextracted log data for the inactive back-end server</td><td>Y[1]</td><td>YI[2]</td></tr>
<tr><td>There is no unextracted log data</td><td>Y[3]</td><td>Y[3]</td></tr>
<tr><td>Other</td><td>There is unextracted log data for the inactive back-end server</td><td>N</td><td>YI[2]</td></tr>
<tr><td>There is no unextracted log data</td><td>N</td><td>Y[3]</td></tr>
</table>

Legend:

Y: Can be executed without causing inconsistency.

YI: Can be executed, but inconsistency occurs.

N: Cannot be executed (inconsistency does not occur).

[1] For a unit that has not been started because of reduced activation, the command ends HiRDB Datareplicator linkage the next time the unit is started and system log data has been extracted.

[2] For a unit that has not been started because of reduced activation, the command ends HiRDB Datareplicator linkage the next time the unit is started without waiting for extraction of remaining system log data.

[3] For a unit that has not been started because of reduced activation, the command ends

HiRDB Datareplicator linkage the next time the unit is started.

pdstart (Start HiRDB system, unit, or server)

Function

The `pdstart` command starts a HiRDB system, unit, or server. This command is also used to initiate a restart.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

```
pdstart [{-i|-r|dbdestroy}]
```

- HiRDB/Parallel Server

Starting the HiRDB system

```
pdstart [{-i|-r|dbdestroy}]
```

Starting a unit

```
pdstart {-x host-name|-u unit-identifier}[{-r|dbdestroy}]
```

Starting a server

```
pdstart -s server-name
```

Starting the system by resetting the front-end server from its SUSPENDed status

```
pdstart -a [-s front-end-server-name]
```

Starting a unit from a unit

```
pdstart -q [-r]
```

Starting the alternate system when using the standby-less system switchover (1:1) facility (placing the alternate system in standby status)

```
pdstart -q -c
```

Starting a server on the specified target unit when using the standby-less system switchover (effects distributed) facility

```
pdstart [{-x host-name|-u unit-identifier}] -s server-name
```

Options

- `{-i|-r|dbdestroy}`

Specifies the startup mode for the HiRDB system. When this option is omitted, the HiRDB system performs normal startup if the previous termination was normal termination or performs a restart if the previous termination was forced or abnormal

termination. In the case of a restart, database recovery processing is performed.

`-i`

If databases have been initialized by the database initialization utility (`pdinit`), specifies that the databases are to be reinitialized.

`-r`

Specifies that the master directory RDAREA is to be backed up by the database copy utility (`-M x` specified). Also specifies that if HiRDB will not start due to an error in the master directory RDAREA, the database recovery utility (`pdrstr`) is to be executed.

`dbdestroy`

Specifies that HiRDB is to be restarted by forced restart because it would not restart. When HiRDB is restarted with this option specified, all RDAREAs (including system RDAREAs) are destroyed.

- {`-x` *host-name*|`-u` *unit-identifier*} `[-r|dbdestroy]`

Specifies that a unit is to be started.

`-x` *host-name* ～ <identifier> ((1-32))

When a unit is to be started, specifies the name of the host containing the unit. This option cannot be specified together with the `-i` option.

When the standby-less system switchover (effects distributed) facility is used, the command starts the host and guest BESs on the specified host's unit. The command starts the server specified as the running system by the cluster software as the running system and places the server specified as the standby system in acceptable status. The server started as the running system is in acceptable status when viewed from another unit in the HA group.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit to be started. This option cannot be specified simultaneously with the `-i` option.

When the standby-less system switchover (effects distributed) facility is used, the command starts the host and guest BESs on the specified unit.

The command starts the server specified as the running system by the cluster software as the running system and places the server specified as the standby system in acceptable status. The server started as the running system is in acceptable status when viewed from another unit in the HA group.

For details about `-r` and `dbdestroy`, see the description of {`-i`|`-r`|`dbdestroy`}.

- `-s` *server-name* ～ <identifier> (1-8))

When a server is to be started, specifies the name of the server.

Notes when the standby-less system switchover (effects distributed) facility is used:

When the `-u` option is omitted, the command starts the specified servers on all active units in the HA group. Note that only the one server that was specified as the running system by the cluster software is started as the running system and that all other servers are started as being in acceptable status.

When the `-u` option is specified, the command starts the specified servers on that unit. In this case, the command starts the server specified as the running system by the cluster software as the running system and the server specified as the standby system as being in acceptable status. When viewed from other units in the HA group, the server started as the running system is in acceptable status.

For details about the standby-less system switchover (effects distributed) facility, see the *HiRDB Version 8 System Operation Guide*.

- `-a [-s` *front-end-server-name*`]`

Restarts a front-end server after resolving the cause of its `SUSPEND` status (waiting for the recovery of the data dictionary RDAREA or the starting of the dictionary server). Regardless of whether the `-s` option is specified, this option starts all the front-end servers in which an error occurred.

- `-q [-r]`

Specifies that the unit is to be started on a HiRDB/Parallel Server. When the standby-less system switchover facility is used, specify this option to start the standby unit.

When the `-q` option is specified, the host corresponding to the unit to be started (host with the `-x` or `-c` operand specified in the `pdunit` statement in the system common definition) is directly logged into and the command is executed.

If the `-q` option is specified, you must start all units before the `pd_reduced_check_time` operand value in the system definition is reached (when omitted, 20 minutes). If the command cannot start any of the units, HiRDB startup processing is cancelled.

For details about `-r`, see the description of `{-i|-r|dbdestroy}`.

- `-q -c`

When the standby-less system switchover (1:1) facility is used, specifies that the alternate system is to be placed in standby status. Specify this option in the following cases:

- If planned switchover is executed using a cluster software command, the alternate system is stopped. In such a case, use the `pdstart -q -c` command to place the alternate system in standby status as a safeguard in the event of an error.

- The alternate system is stopped if you stop the normal BES unit with a command such as `pdstop -u` and restart the normal BES unit with the `pdstart -q` command after the alternate system of the corresponding alternate BES unit has stopped. In such a case, use the `pdstart -q -c` command to place the alternate system in the standby status as a safeguard in the event of an error.

To execute the `pdstart -q -c` command, directly log onto the host of the alternate system that is to be placed in standby status (host specified in `-x` of the `pdunit` operand in the system definition).

Rules

1. The `pdstart` command can be executed to start a HiRDB system only while the HiRDB is shut down. To start a unit or server, this command can be executed only while the system manager is active.
2. The `pdstart` command must be executed on the server machine that contains the single server or the server machine where the system manager is located.
3. If the system manager terminates abnormally on a HiRDB/Parallel Server, the server must be restarted with the `pdstart` command (with no options specified).
4. When the `-i` option is specified, any operation command or any utility command other than for the database initialization utility can be executed after the database initialization utility has terminated normally and the HiRDB system has been placed in startup completed status. The HiRDB system does not check a utility or operation command executed before the database initialization utility has terminated. If executed, subsequent HiRDB system operations cannot be guaranteed.
5. To place the alternate system in standby status when the standby-less system switchover (1:1) facility is used, you cannot specify the `-i` option in the `pdstart` command.
6. When the `-r` option is specified, the HiRDB system or unit is started only in order to execute the database copy utility or the database recovery utility. Therefore, when the HiRDB system or unit is started by the `pdstart` command with the `-r` option specified, no utility other than the database copy utility or the database recovery utility and no command other than `pdls` or `pdstop` should be executed. If executed, subsequent HiRDB system operations cannot be guaranteed.
7. When the `dbdestroy` option is specified, all RDAREAs (including system RDAREAs) updated since entry of the `pdclose` command or since the previous HiRDB startup are destroyed. When a HiRDB forced startup is performed with this option specified, it is necessary to either use the database recovery utility to recover all RDAREAs destroyed or to reload the entire database.

   For details about forced startup by specifying the `dbdestroy` option and for recovery of RDAREAs, see the *HiRDB Version 8 System Operation Guide*.

8. If you specify the `-r` option when the system switchover facility is being used, before you execute the `pdstart` command, you must activate the disk and start the IP address in a unit to which the standby-less system switchover (effects distributed) facility is not applied. In this case, the action depends on the `-q` option specification in the `pdstart` command as described below:

   - When the `-q` option is omitted

     Check the disk connection status and IP address operating status, and then activate the disk and start the IP address in the primary system.

   - When the `-q` option is specified

     Check the disk connection status and IP address operating status, and then activate the disk and start the IP address in the system where the `pdstart` command is executed.

9. If you use the `pdstart -u` command in a HiRDB/Parallel Server to start a unit that does not inherit IP addresses from the system switchover facility, the system containing the unit to be started is determined by the following conditions:

   - If the unit specified in the `-u` option is inactive and the system manager unit has terminated or restarted, the unit specified in the `-u` option is always started in the primary system.
   - If the unit specified in the `-u` option is inactive and the system manager unit is running, the unit specified in the `-u` option is started in the system where the unit was running before it was terminated. If the unit was running previously in the primary system, then the unit in the primary system is started; if it was running previously in the secondary system, then the unit in the secondary unit is started.

   If you want to start the unit in a different system from the one where the unit was running immediately before it was terminated, execute the `pdstart -q` command in the system where you want to start the unit.

10. If you use the system switchover facility and execute the `pdstart -r` command, activate the disk in advance and start the IP address in the target unit for system switchover (except for a unit to which the standby-less system switchover (effects distributed) facility is applied). In the case of a HiRDB/Parallel Server, the HiRDB startup method depends on whether or not all units are to be started in the primary system, as described below. If the primary system cannot be started for a reason such as a server machine failure, start each unit separately as described under *When some or all units are started in the secondary system* below.

    - When all units are started in the primary system

      Check the disk connection status. If the disk in the primary system is not active, activate it. In the case of a unit that inherits IP addresses from the system switchover facility, make sure that the IP address is active in the

primary system. If the IP address is not active, start it in the primary system. Check all primary systems to make sure that the disk is active and that the IP address has started, and then execute the `pdstart -r` command on the unit containing the system manager.

- When some or all units are started in the secondary system

  Check the disk connection status. If the disk is not active in the system to be started, activate the disk in that system. In the case of a unit that inherits IP addresses from the system switchover facility, make sure that the IP address is active in the system that is to be started. If the IP address is not active, start it in that system. Make sure that the disk is active in all units and that the IP address has started, and then execute the `pdstart -q -r` command on each unit.

Notes

1. The following are the `pdstart` command's return codes:

   `0`: Normal termination

   `4`: A message was displayed because of a timeout.

   `8`: Abnormal termination (an invalid option was specified)

2. The result of the `pdstart` command can be checked by the `pdls` command (with `-d prc` or `-d svr` specified).

pdstbegin (Start output of statistical information)

Function

The pdstbegin command outputs to the statistics log file statistical information about a unit located at the server machine with a specified host name. When output of statistical information is to be stopped, the pdstend command must be executed.

The statistical information is output to the statistics log file named pdstj1 or pdstj2 at each host.

The obtained statistical information can be read by the statistics analysis utility. For details about the statistics analysis utility, see Chapter *14. Statistics Analysis Utility (pdstedit)*.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdstbegin [-k statistical-information-type[,statistical-information-type]...]
          [-m interval] [-a|-s server-name[,server-name]...}]
```

HiRDB/Parallel Server

```
pdstbegin [-k statistical-information-type[,statistical-information-type]...]
          [-m interval] [{-x host-name|-u unit-identifier}]
          [{-a|-s server-name[,server-name]...}]
```

Options

- -k *statistical-information-type* ～ <<sys>>

Specifies the type of statistical information to be output:

sys

System activity statistical information

uap

UAP statistical information

sql

SQL statistical information

sqh

SQL statement statistical information (statistical information about issued data

manipulation SQL, definition SQL, and the LOCK statement and SQL)

This statistical information is output when the output of SQL statistical information is specified.

buf

Global buffer pool statistical information

fil

Statistical information on HiRDB files for database manipulation

dfw

Deferred write processing statistical information

idx

Index statistical information

sop

SQL static optimization information

dop

SQL dynamic optimization information

pcd

SQL object execution information

obj

Statistics on SQL object transmission (applicable to HiRDB/Parallel Server only)

fsv

Statistical information about the activities of external servers

hba

Statistical information about the utilization status of external servers

all

All statistical information except sqh, pcd, obj, fsv, and hba

- -m *interval* ～ <unsigned integer> ((1-1440)) <<10>>

When system activity statistical information (-k sys specified) is to be output to the statistics log file, specifies in minutes the interval at which the information is to be output.

- {-x *host-name*|-u *unit-identifier*}

Specifies the identifier of the unit or the name of the host for which statistical

information is to be output. If this option is omitted, statistical information for the entire HiRDB system is output to the statistics log file.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the name of the host for which statistical information is to be output. If the standby-less system switchover (1:1) facility is used and the alternate system is in use, the command starts output of statistical information about both hosts, whether the specified host name belongs to the normal BES or to the alternate BES.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit for which statistical information is to be output. If the standby-less system switchover (1:1) facility is used and the alternate system is in use, the command starts output of statistical information about both units, whether the specified unit identifier belongs to the normal BES or to the alternate BES.

- `{-a|-s` *server-name*`[,`*server-name*`]...}<<-a>>`

Specifies the servers for which statistical information is to be output. When this option is omitted, the statistical information to be output depends on the server type, as shown in Tables 2-6 and 2-7.

`-a`

Specifies that statistical information about the entire system and all servers is to be output.

This option and the `-s` option are mutually exclusive.

`-s` *server-name*`[,` *server-name*`]` ～ <identifier> ((1-8))

Specifies the name of a server for which statistical information is to be output.

This option and the `-a` option are mutually exclusive. If the standby-less system switchover (1:1) facility is used and the alternate system is in use, the command starts output of statistical information about both servers, whether the normal BES or the alternate BES is specified.

Rules

1. The `pdstbegin` command can be executed only while HiRDB is active.
2. The `pdstbegin` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. Once output of statistical information begins, it continues until HiRDB is shut down or the `pdstend` command is executed.
4. The type of statistical information to be output depends on the specification of the `-s` option. Table 2-6 (for a HiRDB/Single Server) and Table 2-7 (for a HiRDB/

Parallel Server) show the relationship between the statistical information to be output and the specification of the -s option.

Table 2-6: Relationship between statistical information output and s option specification: HiRDB/Single Server

Types of statistical information	Servers specified with -s option (Single Server)
sys	Y
uap	Y
sql	Y
sqh	Y
buf	Y
fil	Y
dfw	Y
idx	Y
sop	Y
dop	Y
pcd	Y
obj	—
fsv	—
hba	—

Y:Statistical information is output.

—: statistical information is not output.

Table 2-7: Relationship between statistical information output and -s option specification: HiRDB/Parallel Server

Types of statistical information	Servers specified with -s option		
	FES	DS	BES
sys	Y	Y	Y
uap	Y	—	—
sql	Y	—	—

Types of statistical information	Servers specified with -s option		
	FES	DS	BES
sqh	Y	—	—
buf	—	Y	Y
fil	—	Y	Y
dfw	—	Y	Y
idx	—	Y	Y
sop	Y	—	—
dop	Y	—	—
pcd	Y	Y	Y
obj	—	Y	Y
fsv	—	—	Y
hba	—	—	Y

Y: Statistical information is output.

—: Statistical information is not output.

FES: Front-end server

DS: Dictionary server

BES: Back-end server

Notes

1. The following are the pdstbegin command's return codes:

   0: Normal termination

   8: Abnormal termination (such as an invalid option or rsh error)

2. The pdstbegin command can also be executed using pdststart as the command name.

3. If none of the servers in the unit has started, statistical information about the unit is not acquired.

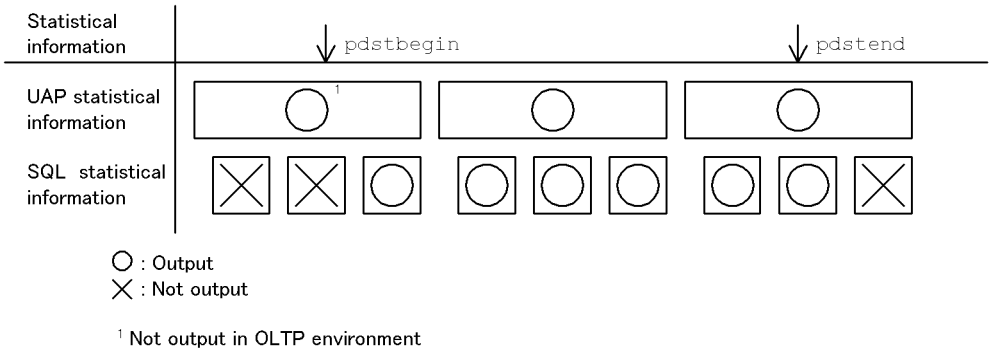
4. When acquisition of a server's sys (statistical information about system activities) is started, acquisition of the unit's sys also begins.

5. When HiRDB is terminated and then restarted, the statistical information output

operation is not inherited. To output statistical information after HiRDB has started, you must re-execute the pdstbegin command.

6. Once a server is terminated, the statistical information output operation is not inherited when the server is started normally. To output statistical information for a server that was terminated, the pdstbegin command must be re-executed after the server has been started normally.
7. Execute the pdls -d stj command to determine whether statistical information is being output.
8. UAP-related statistical information is output only for UAPs that executed between the time the pdstbegin command was entered and the time the pdstend command was entered. For UAPs executing in the OLTP environment, UAP-related statistical information is output only for transactions that executed between the time the pdstbegin command was entered and the time the pdstend command was entered.
9. SQL-related statistical information is output only for SQLs that executed between the time the pdstbegin command was entered and the time the pdstend command was entered.
10. UAP-related statistical information and SQL-related statistical information may not be in agreement, depending on the timing of the pdstbegin command and pdstend command entry, as shown in Figure 2-5.

Figure 2-5: Relationship between output statistical information and entry of pdstbegin and pdstend commands



11. Following are the times at which the different types of statistical information are output:

sys

Output at the time interval specified in the -m option

uap

Output when connected to HiRDB and when disconnected from HiRDB

`sql`

Output when SQL execution starts and when SQL execution terminates

`sqh`

Output information during `PREPARE` and during execution of an embedded data-type SQL statement.

`buf`, `fil`, `dfw`, `idx`

Output at synchronization points

`sop`

Output when the specified SQL object was not found in the SQL object buffer during dynamic and static SQL preprocessing

`dop`

Output when an SQL statement other than `FETCH` or `CLOSE` was executed

`pcd`

Output when an SQL object is executed

`obj`

Output when an SQL object execution request is issued at the dictionary server and back-end server.

`fsv`

Output when the transaction has terminated.

`hba`

Output when an SQL statement is executed on an external server.

12. The statistical information listed below is collected at synchronization points, and the information that exists between synchronization points is edited. Therefore, this statistical information is not collected unless there are at least two synchronization points:
    - Global buffer pool statistical information
    - Statistical information on HiRDB files for database manipulation
    - Deferred write processing statistical information
    - Index statistical information

    To collect this statistical information, synchronization points must be created by executing the `pdlogswap` command immediately after the `pdstbegin`

command and immediately before the pdstend command.

13. SQL statement statistical information is output when the output of SQL statistical information is specified.
14. When the standby-less system switchover (effects distributed) facility is used, if restart occurs after execution of the pdstbegin command but before execution of the pdstend command, the same statistical information is output after the restart as before the restart. If system switchover occurs after execution of the pdstbegin command but before execution of the pdstend command (as in the case of a server that was being used as the running system on the erroneous unit), the same statistical information is output after the switchover at the target accepting unit as before the switchover. To stop output of statistical information, you must execute the pdstend command.
15. When the standby-less system switchover (effects distributed) facility is used and the -x or -u option is specified, the target of the pdstbegin command execution is different before and after system switchover, as shown below:

Server type	Server status	Target
Host BES	Executing	Y
	Standby	N
Guest BES	Executing	Y
	Acceptable status	N

Legend:

Y: Can be executed

N: Cannot be executed

16. You should note the following when the standby-less system switchover (effects distributed) facility is used and the -x or -u option is specified together with the -s option:
    - Acquisition of statistical information begins if the back-end server specified in the -s option on the unit specified in the -x or -u option is running when the pdstbegin command is executed.

pdstend (Stop output of statistical information)

Function

The pdstend command stops the output of specified statistical information on a unit on the server machine specified in the host name (or unit identifier). If the pdstend command is not executed, the system will continue to output statistical information until the HiRDB system stops.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdstend [-k statistical-information-type[,statistical-information-type]...]
        [-a|-s server-name[server-name]...]
```

HiRDB/Parallel Server

```
pdstend [-k statistical-information-type[,statistical-information-type]...]
        [{-x host-name|-u unit-identifier}]
        [-a|-s server-name[server-name]...]
```

Options

- -k *statistical-information-type*

Specifies the type of statistical information to be output. The default is to stop the output of all statistical information.

sys

System activity statistical information

uap

UAP statistical information

sql

SQL statistical information

sqh

SQL statement statistical information (statistical information about issued data manipulation SQL, definition SQL, and the LOCK statement and SQL)

This statistical information is output when the output of SQL statistical information is specified.

`buf`

Global buffer pool statistical information

`fil`

Statistical information on HiRDB files for database manipulation

`dfw`

Deferred write processing statistical information

`idx`

Index statistical information

`sop`

SQL static optimization information

`dop`

SQL dynamic optimization information

`pcd`

SQL object execution information

`obj`

Statistics on SQL object transmission (applicable to HiRDB/Parallel Server only)

`fsv`

Statistical information about the activities of external servers

`hba`

Statistical information about the utilization status of external servers

`all`

All statistical information except `sqh`, `pcd`, `obj`, `fsv`, and `hba`

- {`-x` *host-name*| `-u` *unit-identifier*}

Specifies the host name of the host or the unit identifier of the unit on which statistical information is to be output. The default is to stop the output of statistical information on the entire HiRDB system.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the host name of the host on which statistical information is to be output. If the standby-less system switchover (1:1) facility is used and the alternate system is in use, the command stops output of statistical information about both hosts, whether the specified host name belongs to the normal BES or to the alternate BES.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the unit identifier of the unit on which statistical information is to be output. If the standby-less system switchover (1:1) facility is used and the alternate system is in use, the command stops output of statistical information about both units, whether the specified unit identifier belongs to the normal BES or to the alternate BES.

- `{-a|-s [`*server-name* `[,` *server-name*`] ...]}` ～ <<-a>>

Specifies the server on which the output of statistical information is to be stopped. The default is the option `-a`.

`-a`

Stops the output of statistical information for system services and all servers. This option and the `-s` option are mutually exclusive.

`-s [`*server-name* `[,` *server-name*`] ...]` ～ <identifier> ((1-8))

Specifies the name of the server for which the output of statistical information is to be stopped. This option and the `-a` option are mutually exclusive. If the standby-less system switchover (1:1) facility is used and the alternate system is in use, the command stops output of statistical information about both servers, whether the normal BES or the alternate BES is specified.

Rules

1. The `pdstend` command can be executed only while HiRDB is active.
2. The `pdstend` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. When statistical information is being output, the output continues until HiRDB is shut down unless the `pdstend` command is executed.

Note

1. The following shows the `pdstend` command's return codes:

   `0`: Normal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)
2. Executing the `pdstend` command causes statistics log buffer information to be output to the statistics log file. To acquire the most recent statistical information, you must execute the `pdstjsync` command before you execute the `pdstend` command.
3. If you stop acquisition of a server's `sys` (statistical information about system activities), acquisition of the unit's `sys` also stops.
4. SQL statement statistical information is not output if the output of SQL statistical

information is specified.

5. When the standby-less system switchover (effects distributed) facility is used, if restart occurs after execution of `pdstend`, no statistical information is output after the restart. If system switchover occurs after execution of `pdstend`, no statistical information is output at the target accepting unit for the server that was being used as the running system on the erroneous unit. To restart output of statistical information, you must execute the `pdstbegin` command.

6. When the standby-less system switchover (effects distributed) facility is used and the `-x` or `-u` option is specified, the target of the `pdstend` command execution is different before and after system switchover, as shown below:

Server type	Server status	Target
Host BES	Executing	Y
	Standby	N
Guest BES	Executing	Y
	Acceptable status	N

Legend:

Y: Can be executed

N: Cannot be executed

7. You should note the following when the standby-less system switchover (effects distributed) facility is used and the `-x` or `-u` option is specified together with the `-s` option:

   - Acquisition of statistical information stops if the back-end server specified in the `-s` option on the unit specified in the `-x` or `-u` option is running when the `pdstend` command is executed.

pdstjswap (Swap statistics log files)

Function

The pdstjswap command swaps statistics log files and restores the management information on the statistics log file that has resulted in an error.

Executor

HiRDB administrator

Format

HiRDB/Single Server

```
pdstjswap [{-x host-name|-u unit-identifier}]
```

HiRDB/Parallel Server

```
pdstjswap {-x host-name|-u unit-identifier}
```

Options

- -x *host-name* ～ <identifier> ((1-32))

Specifies the host name of the unit that contains the statistics log files.

- -x *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit that contains the statistics log files.

Rules

1. You can execute the pdstjswap command only when HiRDB is active.
2. Execute the pdstjswap command at the server machine containing the single server or the server machine where the system manager is located.

Note

1. The following are the pdstjswap command's return codes:

   0: Normal termination

   8: Abnormal termination (such as an invalid option or rsh error)

2. If you execute the pdstjswap command, the immediately preceding generation of the statistics log file is overwritten. If you need this statistics log file, back it up before executing the pdstjswap command.

pdstjsync (Apply statistics log buffer to statistics log file)

Function

The pdstjsync command outputs information from the statistics log buffer to the statistics log file.

Executor

HiRDB administrator

Format

```
pdstjsync
```

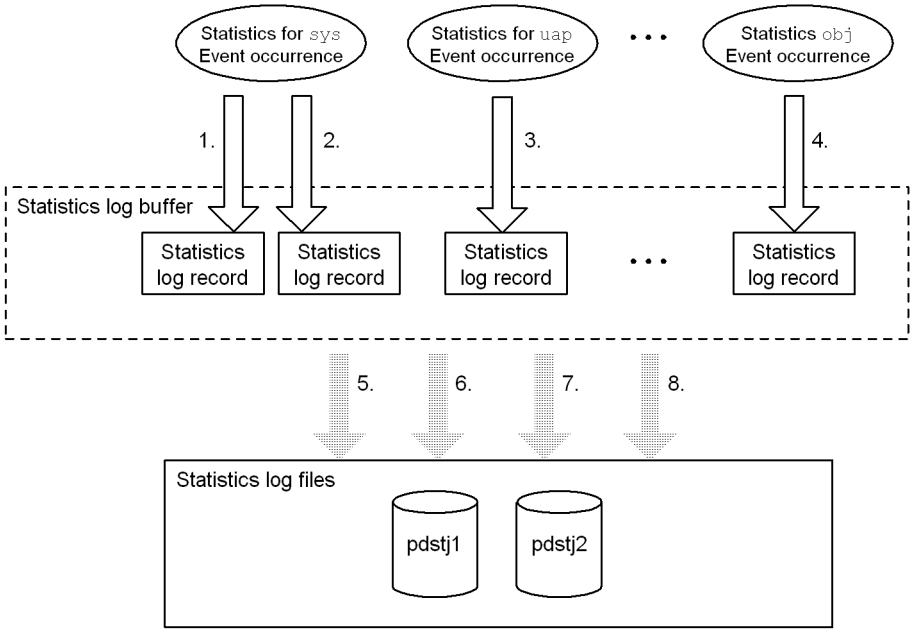
Option

None

Rules

1. You can execute the pdstjsync command only when HiRDB is active and statistical information is being acquired.
2. Execute the pdstjsync command at the applicable server machine (on the unit where you wish to output information from the statistics log buffer to the statistics log file).
3. When statistical information about system activities is being acquired, the pdstjsync command also outputs to the statistics log file statistical information about the system activity at the time of this command's execution. Figure 2-6 shows the statistics record output timing and the timing of output to the statistics log file.

Figure 2-6: Statistics record output timing and timing of output to the statistics log file



Legend:

`sys`: Statistical information about system activities

`uap`: Statistical information about UAP

`obj`: Statistical information about SQL object transfer

Explanation

1. Statistics log records are output to the statistics log buffer at the interval specified in the `pdstbegin -m` command.

2. Statistics log records are output to the statistics log buffer when the `pdstjsync` command is executed.

3. Statistics log records are output to the statistics log buffer when connection is established with HiRDB and when it is released from HiRDB.

4. Statistics log records are output to the statistics log buffer when an SQL object execution request is issued.

5. Information from the statistics log buffer is output to the statistics log file when the statistics log buffer becomes full.

6. Information from the statistics log buffer is output to the statistics log file

when acquisition of statistical information is stopped (when the pdstend command is executed).

7. Information from the statistics log buffer is output to the statistics log file when the pdstop command is executed (normal or planned termination).

8. Information from the statistics log buffer is output to the statistics log file when the pdstjsync command is executed.

Note

Nos. 2 and 8 are performed by a single execution of the pdstjsync command.

4. The statistics log records containing statistical information about system activities are not only output at the interval specified in the pdstbegin command's -m option but also output to the statistics log buffer when the pdstjsync command is executed.

Notes

1. If you execute the pdstjsync command while statistical information about system activities is being acquired, the amount of statistics log information to be output to the statistics log file increases. Therefore, when determining the value for the pd_stj_file_size operand in the system definition, you must take into account the number of times the pdstjsync command will execute. No other type of statistical information (other than system activity information) increases the amount of statistics log information.
2. Executing the pdstjsync command has no effect on the editing of statistical information by pdstedit.
3. For the pdstjsync command, return code 0 indicates normal termination, and return code 8 indicates abnormal termination (invalid option or nonavailability of statistical information).

pdstop (Terminate HiRDB system, unit, or server)

Function

The pdstop command terminates a HiRDB system, unit, or server.

Executor

HiRDB administrator

Format

- Normal termination

HiRDB/Single Server
```
pdstop [-d]
```

HiRDB/Parallel Server

```
pdstop [{-x host-name|-u unit-identifier|-s server-name
          |-u unit-identifier -s server-name}] [-d]
```

- Planned termination

```
pdstop -P[-d]
```

- Forced termination

HiRDB/Single Server
```
pdstop -f[{-d|-q}]
```

HiRDB/Parallel Server

```
pdstop {-f [{-x host-name|-u unit-identifier|-s server-name
              |-u unit-identifier -s server-name}[-d]]
         |-z[{-d|-q|-c|-s server-name}]}
```

Options

- -p

Specifies that the termination is to be by planned termination.

Difference between planned termination and normal termination is as follows:

Connected to HiRDB	Transaction	Normal termination	Planned termination
No	No	Y	Y
	Yes	N	N
Yes	No	N	Y
	Yes	N	N

Legend:

Y: Can be executed.

N: Cannot be executed.

- `-f`

Specifies that the termination is to be by forced termination. When this option is omitted, normal termination is executed.

- {`-x` *host-name*|`-u` *unit-identifier*|`-s` *server-name*}

Specifies the unit or server that is to be terminated. When this option is omitted, the HiRDB system is terminated.

`-x` *host-name* ～ <identifier> ((1-32))

When a unit is to be terminated, specifies the name of the host containing the unit. The host containing the system manager cannot be specified.

When the standby-less system switchover (effects distributed) facility is used, the command terminates the host and guest BESs on the specified host's unit. If the terminated server was the running system, it is released from acceptable status when viewed from another unit in the HA group.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

When a unit is to be terminated, specifies the identifier of the unit. The unit identifier of the unit in which the system manager is located cannot be specified.

When the standby-less system switchover (effects distributed) facility is used, the command terminates the host and guest BESs on the specified unit. If the terminated server was the running system, it is released from acceptable status when viewed from another unit in the HA group.

`-s` *server-name* ～ <identifier> ((1-8))

When a server is to be terminated, specifies the name of the server.

When the standby-less system switchover (effects distributed) facility is used:

When the -u option is omitted, the command terminates the specified servers on all active units in the HA group.

When the -u option is specified, the command terminates the specified server on that unit. If the terminated server was the running system, it is released from acceptable status as viewed from other units in the HA group.

For details about the standby-less system switchover (effects distributed) facility, see the *HiRDB Version 8 System Operation Guide*.

- -z

Specifies that a unit other than the system manager is to be directly logged into and terminated forcibly; this option is specified when the system manager in a HiRDB/Parallel Server terminates abnormally and cannot be restarted.

If the pdstop command with the -z option specified is executed by mistake during HiRDB system startup or normal termination, the unit may terminate abnormally or the command may not be executed.

- -d

Specifies that a dump of shared memory is to be output at the time of termination. If the -s option is specified, this option is ignored. The shared memory dump is output to %PDDIR%\spool\pdshmdump\shmdump.

- -q

If the system switchover facility is used, this option specifies that the specified system is to be terminated forcibly. You can specify this option if use is specified in the pd_ha operand, and the pd_ha_unit operand is also specified in the system definitions.

- -c

When the standby-less system switchover (1:1) facility is being used, specifies that the alternate system is to be terminated forcibly. Specify this option in the following cases:

- When Hitachi HA Toolkit Extension is being used and the alternate system is to be switched over to the primary system (specify the pdstop -q -c command in the termcommand operand of Hitachi HA Toolkit Extension).
- When only the alternate system is to be terminated forcibly without stopping the alternate BES unit. While the alternate system is running, you can also terminate the alternate system forcibly by executing one of the following commands from the unit that contains the system manager: pdstop -f -u *unit-identifier-of-normal-BES-unit* or pdstop -f -x *host-name-of-normal-BES-unit*

To execute the pdstop -q -c command, directly log onto the host of the alternate system that is to be terminated forcibly (host specified in -x of the pdunit operand in the system definition).

Rules

1. The pdstop command can be executed only while HiRDB is active.
2. The pdstop command must be executed on the server machine that contains the single server or the server machine where the system manager is located.
3. Once the HiRDB system has been started by the pdstart command with the -r option specified, it cannot be terminated in units by specifying the -x option. In this case, either all options must be omitted or the -f option must be specified to terminate the HiRDB system.
4. If the HiRDB system was started by the pdstart command with the -i option specified and the database initialization utility has been terminated, the -x option can be specified to terminate each unit of the HiRDB system. If the database initialization utility has not been terminated, the HiRDB system cannot be terminated in units by specifying the -x option. In this case, the -f option must be specified to terminate the HiRDB system.
5. When the -z option is specified, the specified unit is stopped forcibly and any other options that are specified are ignored.
6. When HiRDB is being terminated by the pdstop or pdstop -P command, pdstop -f -x *host-name* or pdstop -f -u *unit-identifier* cannot be used to execute forced termination.
7. While the pdrplstart or pdrplstop command is executing, do not execute the pdstop command (except when the -f or -z option is specified).
8. While a utility is executing, the pdstop command cannot terminate a system or unit normally. You can use the pdstop command to execute planned or forced termination.

Notes

1. The following are the pdstop command's return codes:

   0: Normal termination

   4: Abnormal termination (communication timeout during planned termination)

   8: Abnormal termination (invalid option)
2. The result of the pdstop command can be checked by the pdls command (with -d svr specified).
3. Executing the pdstop command (with -f specified) during HiRDB startup or termination processing or during transaction processing may result in abnormal termination of the unit.
4. If the pdstop command is entered with the -z option specified at a unit other than the system manager during normal operation of the system manager, that unit

will be terminated abnormally.

5. If HiRDB is linked to an OLTP system, OLTP must be terminated before HiRDB is terminated; otherwise, normal or planned termination of OLTP may fail because OLTP cannot process transactions.
6. If planned termination does not end 15 minutes after the execution of the `pdstop -P` command because either a utility or a transaction was being executed, a `KFPS05072-W` message appears, indicating that the planned termination process is taking time. The planned termination process continues after the message.
7. If you have specified the `pd_log_auto_unload_path` operand in the system definitions and executed the `pdstop` command while the automatic log unloading facility is running, you cannot terminate HiRDB until all the system log files have been unloaded. If you want to terminate HiRDB immediately, terminate the automatic log unloading facility using the `pdlogatul -t` command before executing the `pdstop` command.

pdstscls (Close open status file)

Function

The `pdstscls` command closes an open status file. This command is applicable only to status files defined in the HiRDB system definition and can be executed only while HiRDB is active; it cannot close the current file.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

Unit status file
```
pdstscls [{-x host-name|-u unit-identifier}]
          {-n logical-filename|-f physical-filename}
```

Server status file
```
pdstscls -s server-name
         {-n logical-filename|-f physical-filename}
```

- HiRDB/Parallel Server

Unit status file
```
pdstscls {-x host-name|-u unit-identifier}
         {-n logical-filename|-f physical-filename}
```

Server status file
```
pdstscls -s server-name
         {-n logical-filename|-f physical-filename}
```

Options

- {`-x` *host-name*|`-u` *unit-identifier*}

When a unit status file is to be closed, specifies the identifier of the unit or the name of the host corresponding to the unit status file. In the case of a HiRDB/Single Server, this option can be omitted because the host and unit are fixed.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the name of the host corresponding to the unit status file.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit corresponding to the unit status file.

- `-s` *server-name* ～ <identifier> ((1-8))

When a server status file is to be closed, specifies the name of the server corresponding to the server status file.

- `-n` *logical-filename* ～ <identifier> ((1-8))

When a status file corresponding to a specific logical file is to be closed, specifies the name of the logical file. Both versions (A and B) of the specified logical file will be closed at the same time. However, to close both files (A and B), one of the following must apply:

- One of the files (file A or file B) is in `BLOCKADE` or `STANDBY` status
- Files A and B are both in `BLOCKADE` or `STANDBY` status

- `-f` *physical-filename* ～ <pathname> ((up to 167 characters))

When the status file corresponding to a specific physical file is to be closed, specifies the absolute path name of the physical file. A physical file name must be specified in the format *HiRDB-file-system-area-name**HiRDB-filename*. This option is used in the event of an error in order to close the file version in `STANDBY` status, not the file version in shutdown status.

Rules

1. The `pdstscls` command can be executed only while HiRDB is active.
2. The `pdstscls` command must be executed at the server machine containing the single server or the server machine where the system manager is located.

Note

1. The following are the `pdstscls` command's return codes:

   `0`: Normal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)

2. The result of the `pdstscls` command can be checked by the `pdls` command (with `-d sts` specified).

pdstsinit (Initialize status file)

Function

The pdstsinit command creates a physical file that is to comprise a status file (logical file) in the HiRDB file system and initializes it so that it can be used during HiRDB operation. This command cannot initialize the current or a spare status file being used in the active HiRDB.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

Unit status file

```
pdstsinit [{-x host-name|-u unit-identifier}]
          -f physical-file-name [-l record-length][-c records-count]
```

Server status file

```
pdstsinit -s server-name
          -f physical-file-name [-l record-length][-c records-count]
```

- HiRDB/Parallel Server

Unit status file

```
pdstsinit {-x host-name|-u unit-identifier}
          -f physical-file-name [-l record-length][-c records-count]
          [-D]
```

Server status file

```
pdstsinit -s server-name [{-x host-name|-u unit-identifier}]
          -f physical-file-name [-l record-length] [-c records-count]
          [-D]
```

Options

- {-x *host-name*|-u *unit-identifier*}

When a unit status file is to be initialized, specifies the identifier of the unit or the name of the host corresponding to the unit status file. For a HiRDB/Single Server, this option can be omitted because the host and unit are fixed.

-x *host-name* ～ <identifier> ((1-32))

- For a unit status file

Specifies the name of the host corresponding to the unit status file.

- For a server status file

  When the standby-less system switchover (effects distributed) facility is used, specifies the name of the host that contains the disk where the file is to be created. This host must contain a unit that belongs to the HA group to which the server specified in the `-s` option is to be moved.

  When the server specified in the `-s` option and the system manager are running, this option is ignored, if specified.

  When the standby-less system switchover (effects distributed) facility is not used and this option is specified together with the `-s` option, the command checks the host specified in this option and the primary system's host at the server specified in the `-s` option to determine if they match. If the host names do not match, an error results.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

- For a unit status file

  Specifies the identifier of the unit corresponding to the unit status file.

- For a server status file

  When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk where the file is to be created. This unit must belong to the HA group to which the server specified in the `-s` option is to be moved.

  When the server specified in the `-s` option and the system manager are running, this option is ignored, if specified.

  When the standby-less system switchover (effects distributed) facility is not used and this option is specified together with the `-s` option, the command checks the primary system's host on the unit specified in this option and the primary system's host at the server specified in the `-s` option to determine if they match. If the host names do not match, an error results.

■ `-s` *server-name* ～ <identifier> ((1-8))

When a server status file is to be initialized, specifies the name of the server corresponding to the server status file.

■ `-f` *physical-file-name* ～ <pathname> ((up to 167 characters))

Specifies the absolute path name of the physical file to be initialized. A physical file name must be specified in the format *HiRDB-file-system-area-name**HiRDB-file-name*. A name beginning with `pl` cannot be specified as an HiRDB filename. If the specified physical file has already been initialized, it will be reinitialized.

- `-l` *record-length* ～ <unsigned integer> ((1024-32768)) <<4096>>

Specifies in bytes the record length for the status file. If the records are short, the number of I/O operations on the status file increases, but the utilization efficiency becomes high. On the other hand, if the records are long, the number of I/O operations decreases, but the utilization efficiency becomes low. This must be taken into account when the record length is specified.

- `-c` *records-count* ～ <unsigned integer> ((32-2096107)) <<256>>

Specifies the number of records in the status file.

For details about how to estimate the number of records, see the *HiRDB Version 8 Installation and Design Guide*. The standard value is the estimated number of records × 1.2.

- `-D`

Specifies that a unit or server is to be added by the `pdchgconf` command.

The command initializes the status files that are required to add a unit or server. When this option is specified, the command does not check the unit identifier or server name.

When you specify this option, make sure that the `pdchgconf` command is executed on the unit that contains the status file that is to be created.

Rules

1. The `pdstsinit` command can be executed at any time, whether or not HiRDB is active (except during HiRDB startup or termination processing).
2. The `pdstsinit` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. The physical files A and B comprising one logical file must be initialized so that they have the same number of records and the same record length. When a logical file is to be initialized, the numbers of records need not be the same, but the record length must be the same.
4. The number of records in a status file depends on the HiRDB system configuration; the value can be checked with the `pdls` command (with `-d sts` specified). When the `pdls` command executes, the record utilization factor is displayed, enabling the number of records to be changed on the basis of the displayed record utilization factor. For example, if the record utilization factor is high, a status file should be initialized to contain more records than the current file.

Note

1. The result of the `pdstsinit` command can be determined from the command's return code or the existence of an error message. Return code `0` for this command

indicates normal termination, `8` indicates abnormal termination (such as invalid option or `rsh` error), and `12` indicates abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited).

2. The following table describes how to specify the `-u` and `-s` options when the standby-less system switchover (effects distributed) facility is used:

Initialization target	Unit or server status	Host containing the disk where the file was created	Specification of -u and -s options
Unit status file	Active	Host where the unit is located	Specify one of them
	Inactive		
Server status file	Active	Host containing the running back-end server	Optional
	Inactive	Host containing the primary back-end server	
		Other host	Specify one of them

Examples

Create a unit status file and a server status file.

HiRDB/Single Server

```
pdstsinit -f c:\sysarea\rdsys011\systs01a          1
          -c 512                                   2
pdstsinit -f c:\sysarea\rdsys012\systs01b          1
          -c 512                                   2
pdstsinit -s sds1                                  3
          -f c:\sysarea\rdsys011\stsf01a           4
          -c 512                                   5
pdstsinit -s sds1                                  3
          -f c:\sysarea\rdsys012\stsf01b           4
          -c 512                                   5
```

Explanation

1. Names of the physical files that constitute the unit status file:

   `c:\sysarea\rdsys011\systs01a,c:\sysarea\rdsys012\systs01b`

2. Number of records per physical file:

   512

3. Name of the server corresponding to these server status files:

   `sds1`

4. Names of the physical files that constitute the server status file:

   ```
   c:\sysarea\rdsys011\stsf01a,c:\sysarea\rdsys012\stsf01b
   ```

5. Number of records per physical file:

   512

HiRDB/Parallel Server

```
pdstsinit -x host1                                    1
          -f c:\sysarea\rdsys011\systs11a             2
          -c 512                                      3
pdstsinit -x host1                                    1
          -f c:\sysarea\rdsys012\systs11b             2
          -c 512                                      3
pdstsinit -x host2                                    1
          -f c:\sysarea\rdsys013\systs21a             2
          -c 512                                      3
pdstsinit -x host2                                    1
          -f c:\sysarea\rdsys014\systs21b             2
          -c 512                                      3
pdstsinit -x host3                                    1
          -f c:\sysarea\rdsys015\systs31a             2
          -c 512                                      3
pdstsinit -x host3                                    1
          -f c:\sysarea\rdsys016\systs31b             2
          -c 512                                      3
pdstsinit -s fes1                                     4
          -f c:\sysarea\rdsys011\stsf11a              5
          -c 512                                      6
pdstsinit -s fes1                                     4
          -f c:\sysarea\rdsys012\stsf11b              5
          -c 512                                      6
pdstsinit -s ds1                                      4
          -f c:\sysarea\rdsys013\stsf21a              5
          -c 512                                      6
pdstsinit -s ds1                                      4
          -f c:\sysarea\rdsys014\stsf21b              5
          -c 512                                      6
pdstsinit -s bes1                                     4
          -f c:\sysarea\rdsys015\stsf31a              5
          -c 512                                      6
pdstsinit -x bes1                                     4
          -f c:\sysarea\rdsys016\stsf31b              5
          -c 512                                      6
```

Explanation

1. Names of the hosts corresponding to the unit status file:

host1, host2, and host3

2. Names of the physical files that constitute the unit status file:

   c:\sysarea\rdsys011\systs11a

   c:\sysarea\rdsys012\systs11b

   c:\sysarea\rdsys013\systs21a

   c:\sysarea\rdsys014\systs21b

   c:\sysarea\rdsys015\systs31a

   c:\sysarea\rdsys016\systs31b

3. Number of records per physical file:

   512

4. Names of the servers corresponding to the server status file:

   fes1, ds1, and bes1

5. Names of the physical files that constitute the server status file:

   c:\sysarea\rdsys011\stsf11a

   c:\sysarea\rdsys012\stsf11b

   c:\sysarea\rdsys013\stsf21a

   c:\sysarea\rdsys014\stsf21b

   c:\sysarea\rdsys015\stsf31a

   c:\sysarea\rdsys016\stsf31b

6. Number of records per physical file:

   512

pdstsopen (Open status file)

Function

The `pdstsopen` command opens a status file initialized by the `pdstsinit` command or a status file closed by the `pdstscls` command. This command is applicable only to status files defined in the HiRDB system definition and can be executed only while HiRDB is active.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

Unit status file

```
pdstsopen [-x host-name|-u unit-identifier}]
          {-n logical-filename|-f physical-filename}
```

Server status file

```
pdstsopen -s server-name
          {-n logical-filename|-f physical-filename}
```

- HiRDB/Parallel Server

Unit status file

```
pdstsopen {-x host-name|-u unit-identifier}
          {-n logical-filename|-f physical-filename}
```

Server status file

```
pdstsopen -s server-name
         {-n logical-filename|-f physical-filename}
```

Options

- {`-x` *host-name*|`-u` *unit-identifier*}

When a unit status file is to be opened, specifies the identifier of the unit or the name of the host corresponding to the unit status file. For a HiRDB/Single Server, this option can be omitted because the host and unit are fixed.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the name of the host corresponding to the unit status file.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit corresponding to the unit status file.

- `-s` *server-name* ～ <identifier> ((1-8))

When a server status file is to be opened, specifies the name of the server corresponding to the server status file.

- `-n` *logical-filename* ～ <identifier> ((1-8))

When a status file corresponding to a specific logical file is to be opened, specifies the name of the logical file. Both versions (A and B) of the specified logical file will be opened at the same time. However, to open both files (A and B), one of the following must apply:

- One of the files (file A or file B) is in `CLOSE`, `NONE`, or `BLOCKADE` status
- Files A and B are both in `CLOSE`, `NONE`, or `BLOCKADE` status

- `-f` *physical-filename* ～ <pathname> ((up to 167 characters))

When the status file corresponding to a specific physical file is to be opened, specifies the absolute pathname of the physical file. A physical filename must be specified in the format *HiRDB-file-system-area-name\HiRDB-filename*. This option is specified to open a status file that has been placed in shutdown status due to an error or a physical file initialized by the `pdstsinit` in order to extend its capacity.

Rules

1. The `pdstsopen` command can be executed only while HiRDB is active.
2. The `pdstsopen` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. If the current status file is in single-operation mode, the physical file version in shutdown status can be used as the current status file by initializing it with the `pdstsinit` command and then executing the `pdstsopen` command.

Note

1. The following are the `pdstsopen` command's return codes:

   `0`: Normal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)

2. The result of the `pdstsopen` command can be checked by the `pdls` command (with `-d sts` specified).

pdstsrm (Delete status file)

Function

The pdstsrm command deletes a physical file comprising a status file in the HiRDB file system. This command cannot delete the current or a spare physical file being used during HiRDB operation. This command is used to delete a physical file that has been placed in shutdown status due to an error.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

Unit status file
pdstsrm [{-x *host-name*|-u *unit-identifier*}] -f *physical-file-name*

Server status file
pdstsrm -s *server-name* -f *physical-file-name*

- HiRDB/Parallel Server

Unit status file
pdstsrm {-x *host-name*|-u *unit-identifier*} -f *physical-file-name* [-D]

Server status file
pdstsrm -s [{-x *host-name*|-u *unit-identifier*}] *server-name* -f *physical-file-name* [-D]

Options

- {-x *host-name*|-u *unit-identifier*}

When a unit status file is to be deleted, specifies the identifier of the unit or the name of the host corresponding to the unit status file. In the case of a HiRDB/Single Server, this option can be omitted because the host and unit are fixed.

-x *host-name* ～ <identifier> ((1-32))

- For a unit status file

  Specifies the name of the host corresponding to the unit status file.

- For a server status file

  When the standby-less system switchover (effects distributed) facility is used, specifies the name of the host that contains the disk where the status

file to be deleted was created. This host must contain a unit that belongs to the HA group to which the server specified in the `-s` option is to be moved.

When the server specified in the `-s` option and the system manager are running, this option is ignored, if specified.

When the standby-less system switchover (effects distributed) facility is not used and this option is specified together with the `-s` option, the command checks the host specified in this option and the primary system's host at the server specified in the `-s` option to determine if they match. If the host names do not match, an error results.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

- For a unit status file

  Specifies the identifier of the unit corresponding to the unit status file.

- For a server status file

  When the standby-less system switchover (effects distributed) facility is used, specifies the unit identifier of the host that contains the disk where the status file to be deleted was created. This unit must belong to the HA group to which the server specified in the `-s` option is to be moved.

  When the server specified in the `-s` option and the system manager are running, this option is ignored, if specified.

  When the standby-less system switchover (effects distributed) facility is not used and this option is specified together with the `-s` option, the command checks the primary system's host on the unit specified in this option and the primary system's host at the server specified in the `-s` option to determine if they match. If the host names do not match, an error results.

■ `-s` *server-name* ～ <identifier> ((1-8))

When a server status file is to be deleted, specifies the name of the server corresponding to the server status file.

■ `-f` *physical-filename* ～ <pathname> ((up to 167 characters))

Specifies the absolute pathname of the physical file to be deleted. A physical filename must be specified in the format *HiRDB-file-system-area-name\HiRDB-filename*.

■ `-D`

Specifies that a unit or server is to be deleted by the `pdchgconf` command.

To delete a unit or server, the command deletes the unit's or server's status files. When this option is specified, the command does not check the unit identifier or server name.

When you specify this option, make sure that the `pdstsrm` command is executed on the unit that contains the status files that are to be deleted.

Rules

1. The `pdstsrm` command can be executed at any time, whether or not HiRDB is active (except during HiRDB startup or termination processing).
2. The `pdstsrm` command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. The current status file being used during HiRDB operation must not be deleted.

Notes

1. The result of the `pdstsrm` command can be determined from the command's return code or the existence of an error message. Return code `0` for this command indicates normal termination, `8` indicates abnormal termination (such as invalid option or `rsh` error), and `12` indicates abnormal termination (when retry was executed from a standby system in a configuration in which IP addresses are not inherited).
2. The following table describes how to specify the `-u` and `-s` options when the standby-less system switchover (effects distributed) facility is used:

Initialization target	Unit or server status	Host containing the file storage disk	Specification of -u and -s options
Unit status file	Active	Host where the unit is located	Specify one of them
	Inactive		
Server status file	Active	Host containing the running back-end server	Optional
	Inactive	Host containing the primary back-end server	
		Other host	Specify one of them

pdstsswap (Swap status files)

Function

The `pdstsswap` command swaps status files. It changes the current status file to a spare status file and a spare status file to the current status file.

Executor

HiRDB administrator

Format

- HiRDB/Single Server

Unit status file
`pdstsswap [-x` *host-name*`|-u` *unit-identifier*`}]`

Server status file
`pdstsswap -s` *server-name*

- HiRDB/Parallel Server

Unit status file
`pdstsswap {-x` *host-name*`|-u` *unit-identifier*`}`

Server status file
`pdstsswap -s` *server-name*

Options

- `{-x` *host-name*`|-u` *unit-identifier*`}`

When unit status files are to be swapped, specifies the identifier of the unit or the name of the host corresponding to the unit status files. In the case of a HiRDB/Single Server, this option can be omitted because the host and unit are fixed.

`-x` *host-name* ～ <identifier> ((1-32))

Specifies the name of the host corresponding to the unit status files.

`-u` *unit-identifier* ～ <identifier> ((4 characters))

Specifies the identifier of the unit corresponding to the unit status files.

- `-s` *server-name* ～ <identifier> ((1-8))

When server status files are to be swapped, specifies the name of the server corresponding to the server status files.

Rules

1. The `pdstsswap` command can be executed only while HiRDB is active.
2. The `pdstsswap` command must be executed at the server machine containing the single server or the server machine where the system manager is located.

Note

1. The following are the `pdstsswap` command's return codes:

   `0`: Normal termination

   `8`: Abnormal termination (such as an invalid option or `rsh` error)
2. The result of the `pdstsswap` command can be checked by the `pdls` command (with `-d sts` specified).

pdsvhostname (Display server's host name)

Function

The pdsvhostname command displays a specified server's host name in DAT format.

Executor

HiRDB administrator

Format

```
pdsvhostname -s server-name [-b]
```

Options

- -s *server-name* ～ <identifier> ((1-8))

Specifies the name of the server whose host name is to be displayed.

Rules

1. If the specified server name is not defined in HiRDB, the command outputs the KFPS05041-E message and terminates.
2. If the specified server is not running (including when the unit is stopped), the command does not display a host name.
3. The command accepts the name of a single server, front-end server, dictionary server, or back-end server.

- -b

Specifies that a title line is to be displayed.

When this option is omitted, the command displays the host name without a title line.

Output format

- When -b is omitted

```
"aa...a"
```

- When -b is specified

```
"HOSTNAME"
"aa...a"
```

Explanation

aa...a

Name of the host where the server specified in the -s option is currently running (up to 32 characters).

The command displays one of the following host names:

- Host name specified in the -x or -c operand of the pdunit operand in the system definition.
- Host name specified in the -x option if the system switchover facility is used without IP address inheritance and the host is running as the primary system, or the host name specified in the -c option if the system switchover facility is used without IP address inheritance and the host is running as the secondary system.
- Name of the host whose server is running if the standby-less system switchover (effects distributed) facility is used and the specified server is currently running.
- Name of the host that is running in the alternate portion if the standby-less system switchover (1:1) facility is used and the specified server is running in the alternate portion.

Rules

1. The pdsvhostname command can be executed only when HiRDB is running. If the command is executed while HiRDB is stopped, the command outputs the KFPS01853-W message and terminates.
2. The pdsvhostname command must be executed at the server machine that contains the single server or where the system manager is located. If the command is executed from a server machine where the system manager is not located, the command outputs the KFPS05037-E message and terminates.

Notes

1. Depending on the execution timing, the pdsvhostname command may terminate normally even though HiRDB is inactive. This occurs in the following cases:
   - The system manager's shared memory is available.

- When the server stopped, the unit did not detect the stop.

2. The following are the `pdsvhostname` command's return codes:

   `0`: Normal termination.

   `1`: Specified server is inactive.

   `4`: HiRDB is inactive.

   `8`: Error termination (including memory shortage),* or the specified server was not found in HiRDB.

   * Check the error message in the event log or the standard error output, eliminate the cause of the error, and then re-execute the `pdsvhostname` command

pdtrndec (Settle unsettled transactions forcibly and automatically)

Function

If there are unsettled transactions when HiRDB is restarted after an error or after normal termination (when the KFPS00992-E message is displayed), the pdtrndec command settles those transactions forcibly.

For details about handling unsettled transactions, see the *HiRDB Version 8 System Operation Guide*.

Executor

HiRDB administrator

Format

```
pdtrndec -i input-file-name[,input-file-name]...
        [-r rollback-script-file-name] [-o output-destination-directory-name]
```

Options

- -i *input-file-name*[,*input-file-name*]... ～ <path name> ((1-4096))

Specifies the absolute path names of the text files to which the event log was saved.

Rules

1. When you specify multiple files, the total specification length, including the separator commas, cannot exceed 4,096 characters.
2. The execution results of this command cannot be guaranteed if a specified file is not a file containing the event log.
3. The input file names specification cannot contain any blanks.
4. If information about an applicable transaction is missing from the input file (including when an invalid input file was specified), the transaction will be placed on hold without being settled.

- -r *rollback-script-file-name* ～ <path name> ((1-4107))

Specifies the output file (pdtrnrbk.bat) from the previous execution of the pdtrndec command.

When this option is specified, the command terminates forcibly only the transactions in the rollback script file that remain to be settled. If the specified file is not a pdtrndec command output file, the command's execution results are not guaranteed.

- `-o` *output-destination-directory-name* ～ <path name> ((1-4096))

Specifies that an execution history (`pdtrndecout`) is to be output when the `pdtrndec` command executes, and specifies the absolute path name of the directory to which the batch file (`pdtrnrbk.bat`) is to be output.

Rules

1. When this option is omitted, the current directory is assumed as the output target.
2. If the specified output target directory already contains the execution history and a batch file, the new execution history is added and the batch file is overwritten.
3. If the `-r` option is also specified, the command does not output a batch file.
4. The specification of the output target directory cannot contain any blanks.

Rules

1. The `pdtrndec` command can be executed only when HiRDB is running.
2. The `pdtrndec` command must be executed at the server machine that contains the single server or where the system manager is located.
3. The `pdtrndec` command is executed when there are any unsettled transactions at the time HiRDB is restarted after an error or after having terminated normally. If you do not know whether or not there are any unsettled transactions, executing this command may settle transactions illegally.
4. Do not execute the `pdtrndec` command more than once at a time. If multiple `pdtrndec` commands are executed concurrently, the execution results cannot be guaranteed.

Notes

1. When you execute the `pdtrndec` command, all HiRDB servers (except recovery-unnecessary front-end servers) must be running. If there is any inactive server, start it with the correct procedure.
2. When you execute the `pdtrndec` command, make sure that any HiRDB client products (such as OLTP system and HiRDB Datareplicator) are running normally. If there is any inactive HiRDB client product, start it with the correct procedure. You should note the following about settling transactions forcibly while there is an inactive client product:
   - If you execute the batch file (`pdtrnrbk.bat`) that is output after command execution, any transaction whose settlement was placed on hold by execution of the command will be rolled back forcibly. If you then start the inactive client product, synchronization may be lost between HiRDB and the

client product.

3. If executing the `pdtrndec` command with an input file specified still results in an unsettled transaction, execute the batch file (`pdtrnrbk.bat`) to settle the transaction.
4. The `pdtrndec` command's processing may take a long time because it synchronizes with completion of other processing, such as transaction commit and rollback commands.
5. After executing the `pdtrndec` command, delete the file under the output target directory specified in the `pdtrndec` command.
6. The `pdtrndec` command's return codes are as follows:

   `0`: Normal termination

   `4`: Warning termination (there is still at least one unsettled transaction)

   `8`: Abnormal termination
7. If you select `utf-8` as the character encoding in the `pdntenv` command, you can use a file with a BOM as the input file for `pdtrndec`. Note that even when a file with a BOM is used as the input file for `pdtrndec`, the BOM is skipped. No BOM is included in the file that is output by `pdtrndec`.

Output format

- Execution history (`pdtrndecout`)

```
pdtrndec result                     [1]START TIME:2002/09/05 17:20:08
[2]HOSTNAME [3]TRNGID        [4]TRNBID        [5]SVID   [6]DECISION [7]TIME
host1       HRD1unt100020b4d HRD1unt100020034 bes1      COMMIT      17:20:08
host1       HRD1unt100020b22 HRD1unt100020035 bes1      ********    --:--:--
host1       HRD1unt100020b51 HRD1unt100020038 bes1      FORGET      17:20:09
pdtrndec result                     [1]START TIME:2002/09/05 17:36:47
[2]HOSTNAME [3]TRNGID        [4]TRNBID        [5]SVID   [6]DECISION [7]TIME
host1       HRD1unt100020b22 HRD1unt100020035 bes1      ********    --:--:--
```

Explanation

1. Command start date and time (*year*/*month*/*date* *hour*:*minute*:*second*)
2. Name of the host that contains an unsettled transaction (up to 32 characters)
3. Global transaction ID that contains an unsettled transaction (16 characters)
4. Branch transaction ID that contains an unsettled transaction (16 characters)
5. Name of the server that contains an unsettled transaction (up to 8 characters)
6. Settlement type of the unsettled transaction (up to 8 characters)

The types are as follows:

Type	Description	Action
COMMIT	Settlement by forced commit	None
ROLLBACK	Settlement by forced rollback	None
FORGET	Forced termination	None
FAIL	Settlement by forced commit or rollback, or forced termination resulted in an error	Eliminate the cause of the error and then re-execute the pdtrndec command
********	Settlement on hold	• For a HiRDB/Single Server, or for a HiRDB/Parallel Server when the pdtrndec command was executed on an input file of the system manager unit Change the input file and then re-execute the pdtrndec command. • For a HiRDB/Parallel Server when the pdtrndec command was executed on an input file of a unit where the system manager is not located Specify the batch file (pdtrnrbk.bat) and then re-execute the pdtrndec command.

7. Time forced settlement was executed (*hour*:*minute*:*second*)

■ Batch file (pdtrnrbk.bat)

```
[1]pdrbk -x host1 -s bes1 -t HRD1unt100020b22
```

Explanation

1. Command format for settling the transaction by rollback

pdtrnqing (Start or release the transaction queuing facility)

Function

The `pdtrnqing` command starts or releases transaction queuing for a back-end server. This command is applicable to a HiRDB/Parallel Server that is using the system switchover facility.

If a transaction is executing when transaction queuing is to start, execution of this command is placed on hold until the transaction terminates. If the back-end server is starting when transaction queuing is to be released (except when the `-f` option is specified), execution of this command is placed on hold until the server startup processing is completed.

Executor

HiRDB administrator

Format

```
pdtrnqing [-d [-f]] -s server-name
```

Options

- `-d`

Specifies that transaction queuing is to be released. When this option is omitted, startup of transaction queuing is assumed.

- `-f`

When transaction queuing is to be released (when the `-d` option is specified), specifies that forcible release of transaction queuing is to be performed (such as when an error has occurred). You can execute the `pdtrnqing` command with this option specified from the following locations:

- Inactive back-end server
- Inactive unit (such as the unit containing the system manager)

- `-s` *server-name* ～ <identifier>((1-8))

Specifies the name of the back-end server that is to be the target of transaction queuing startup or release. You must specify a back-end server that is using the standby-less system switchover (effects distributed) facility.

Rules

1. Whether or not the `pdtrnqing` command can be executed depends on the options that are specified and server's operating status. Note that the unit must be running.

Specified options	Status of back-end server				
	Inactive	Standby	Starting	Running	Stopping
None	—	—	—	Y[1]	—
`-d`	—	W[2,3]	Y[2]	Y	—
`-d -f`	Y	Y	Y	Y	Y

Legend:

Y:

Transaction queuing can be started or released.

—:

Transaction queuing cannot be started or released.

W:

Waits for completion of server startup processing. If server startup processing is completed in the executing unit within the time specified in the `pd_ha_trn_queuing_wait_time` operand in the system definition, transaction queuing is released. If the startup processing is not completed, the command terminates with an error.

1

This command is placed on hold until the transaction in the back-end server specified in `-s` is completed. If the transaction is not completed within the time specified in the `pd_ha_trn_queuing_wait_time` operand in the system definition, the command results in an error.

2

If the back-end server specified in `-s` is in standby status or is starting in the unit where this command is executed, the command waits for completion of the server startup processing. If the server startup processing is not completed within the time specified in the `pd_ha_trn_queuing_wait_time` operand in the system definition, the command results in an error.

3

When the system switchover facility is being used, if this command is executed from the point in time immediately after the server is switched over to the running system by means of the cluster software command until the server status managed by HiRDB is set to *Starting*, command execution is placed on hold in order to prevent errors.

2. The `pdtrnqing` command can be executed from any server machine.
3. If `error` is specified in the `pd_ha_transaction` operand in the system definition, the command terminates itself without starting or releasing transaction queuing. In such a case, the return code is `4`.

Notes

1. If an attempt is made to start transaction queuing while transaction queuing is already being started, or if transaction queuing is released during normal operation (when transactions are not queued), this command terminates normally without changing the status.
2. Release of transaction queuing is also executed as an extension of server startup processing.
3. If transaction queuing is released while the server is in standby wait status, the command waits for completion of the server startup processing. Therefore, release of transaction queuing executed for the wrong unit (a unit in standby status) does not immediately result in an error.
4. The following are the `pdtrnqing` command's return codes:

   `0`: Normal termination

   `4`: Warning termination (transaction queuing cannot be achieved because `pd_ha_transaction=queuing` is not specified in the system definition)

   `8`: Abnormal termination (invalid execution environment, operation error, etc.)

   `12`: Abnormal termination

pdusrchk (Check the Directory Server for user information)

Function

The pdusrchk command checks whether or not the users registered in HiRDB are registered in the Directory Server. The command outputs the result to the standard output.

When user management is changed from HiRDB to the Directory Server or when users are registered into or deleted from the Directory Server, inconsistencies may occur between the users registered in HiRDB and the users registered in the Directory Server. The pdusrchk command is used to achieve consistency in user information.

Executor

HiRDB administrator

Format

```
pdusrchk [-k display-target-type]
```

Options

- -k *display-target-type*

Specifies the type of information to be displayed. When this option is omitted, the command displays all information.

usr:

Display whether or not the users registered in the dictionary table SQL_USERS are registered in the Directory Server.

rdp:

Display whether or not the users with RDAREA usage privileges that are registered in the SQL_RDAREA_PRIVILEGES dictionary table are registered in the Directory Server.

tbp:

Display whether or not the users with table access privilege that are registered in the SQL_TABLE_PRIVILEGES dictionary table are registered in the Directory Server.

Rules

1. The pdusrchk command can be executed only while HiRDB is active.

2. The pdusrchk command must be executed at the server machine that contains the single server or where the system manager is located.
3. You can execute the pdusrchk command only while you are using the Directory Server linkage facility.
4. To execute the pdusrchk command, place system RDAREAs in open and shutdown-release status.

Notes

1. Not all users registered in the Directory Server can connect to HiRDB.
2. When the pdusrchk command is executed, the data dictionary RDAREAs (SQL_USERS, SQL_RDAREA_PRIVILEGES, and SQL_TABLE_PRIVILEGES) are locked.
3. The following shows the pdusrchk command's return codes:

   0: Normal termination

   8: Abnormal termination
4. If DIR_SRV in the output format is N, register or delete the user as shown below:
   - When -k usr is specified

     Delete from HiRDB a user who does not have the DBA, schema definition, or audit privilege. Also delete a user from HiRDB who does not need to connect to HiRDB. Register into the Directory Server a user who needs to connect to HiRDB.
   - When -k rdp or -k tbp is specified

     Delete from HiRDB a user who does not need to connect to HiRDB. Register into the Directory Server a user who needs to connect to HiRDB.
5. When it is executed, the pdusrchk locks the following resources in the SR mode:
   - Dictionary RDAREAs that contains SQL_USERS, SQL_RDAREA_PRIVILEGES, and SQL_TABLE_PRIVILEGES
   - Dictionary tables (SQL_USERS, SQL_RDAREA_PRIVILEGES, and SQL_TABLE_PRIVILEGES)

Output format

- When -k usr is specified

```
***** SQL_USERS INFORMATION *****
NAME      DBA   SCHEMA   AUDIT    DIR_SRV
aa...a    b     c        d        e
  :       :     :        :        :
```

Explanation

aa...a:

User ID (up to 8 characters)

b:

Whether or not the user has the DBA privilege:

Y: The user has the DBA privilege.

N: The user does not have the DBA privilege.

c:

Whether or not the user has the schema definition privilege:

S: The user has a schema.

Y: The user has the schema definition privilege.

N: The user does not have the schema definition privilege.

d:

Whether or not the user has the audit privilege:

Y: The user has the audit privilege.

N: The user does not have the audit privilege.

e:

Whether or not the user is registered in the Directory Server:

Y: The user is registered in the Directory Server.

N: The user is been registered in the Directory Server.

- When `-k rdp` is specified

```
***** SQL_RDAREA_PRIVILEGES INFORMATION *****
NAME      RDAREA_NAME   DIR_SRV
aa...a    bb...b        c
  :         :           :
```

Explanation

aa...a:

User ID (up to 8 characters)

bb...b:

RDAREA name (up to 30 characters)

c:

Whether or not the user is registered in the Directory Server:

`Y`: The user is registered in the Directory Server.

`N`: The user is been registered in the Directory Server.

Note

If there is no information to be output, the command displays `*****SQL_RDAREA_PRIVILEGES INFORMATION NOTHING *****`.

- When `-k tbp` is specified

```
***** SQL_TABLE_PRIVILEGES INFORMATION *****
KIND  NAME    TABLE_NAME  SEL  INS  DEL  UPD  DIR_SRV
a     bb...b  cc...c      d    e    f    g    h
:       :       :         :    :    :    :    :
```

Explanation

a:

Type of user name:

`G`: Role name

blank: User ID

bb...b:

User ID, or role name (up to 30 characters)

cc...c:

Table name (up to 30 characters)

d:

Whether or not the user has the `SELECT` privilege:

`G`: Table owner

`Y`: The user has the `SELECT` privilege.

N: The user does not have the SELECT privilege.

e:

Whether or not the user has the INSERT privilege:

G: Table owner

Y: The user has the INSERT privilege.

N: The user does not have the INSERT privilege.

f:

Whether or not the user has the DELETE privilege:

G: Table owner

Y: The user has the DELETE privilege.

N: The user does not have the DELETE privilege.

g:

Whether or not the user has the UPDATE privilege:

G: Table owner

Y: The user has the UPDATE privilege.

N: The user does not have the UPDATE privilege.

h:

Whether or not the user is registered in the Directory Server:

Y: The user is registered in the Directory Server.

N: The user is been registered in the Directory Server.

Note

If there is no information to be output, the command displays ***** SQL_TABLE_PRIVILEGES INFORMATION NOTHING *****.

pdvrup (Upgrade HiRDB version)

Function

The pdvrup command maintains the master directory RDAREA and data dictionary RDAREAs when the HiRDB version has been upgraded in order to make the database conform to the new version.

You can execute this command only when pd_auto_vrup=N is specified in the system definitions. If you have specified pd_auto_vrup=Y, HiRDB automatically executes the pdvrup command.

Executor

HiRDB administrator

Format

```
pdvrup
```

Option

None

Rules

1. The pdvrup command is executed when HiRDB is started after a new HiRDB version has been installed. If pd_auto_vrup=N is specified in the system definitions, the system issues a message requesting an entry of the pdvrup command. An attempt to execute this command at any other time results in an error. For details about how to upgrade, see the *HiRDB Version 8 Installation and Design Guide*.
2. The pdvrup command must be executed at the server machine containing the single server or the server machine where the system manager is located.
3. UAPs and utilities cannot be executed during execution of the pdvrup command.

Note

The execution results of the pdvrup command can be confirmed by the return code after the command has executed and by whether or not there are any error messages. A return code of 0 for this command indicates normal termination; a return value of 4 indicates that the upgrade terminated normally (in this case, restart HiRDB); a return value of 12 indicates abnormal termination.

Chapter

3. Database Initialization Utility (pdinit)

This chapter describes the database initialization utility (`pdinit`) that defines a physical structure to permit use of files as a HiRDB database.

This chapter contains the following sections:

3.1 Overview

Executor: HiRDB administrator

3.1.1 Function

The database initialization utility is used to define a physical structure for use of files as a HiRDB database. A HiRDB database consists of various RDAREAs. One RDAREA consists of one or more HiRDB files.

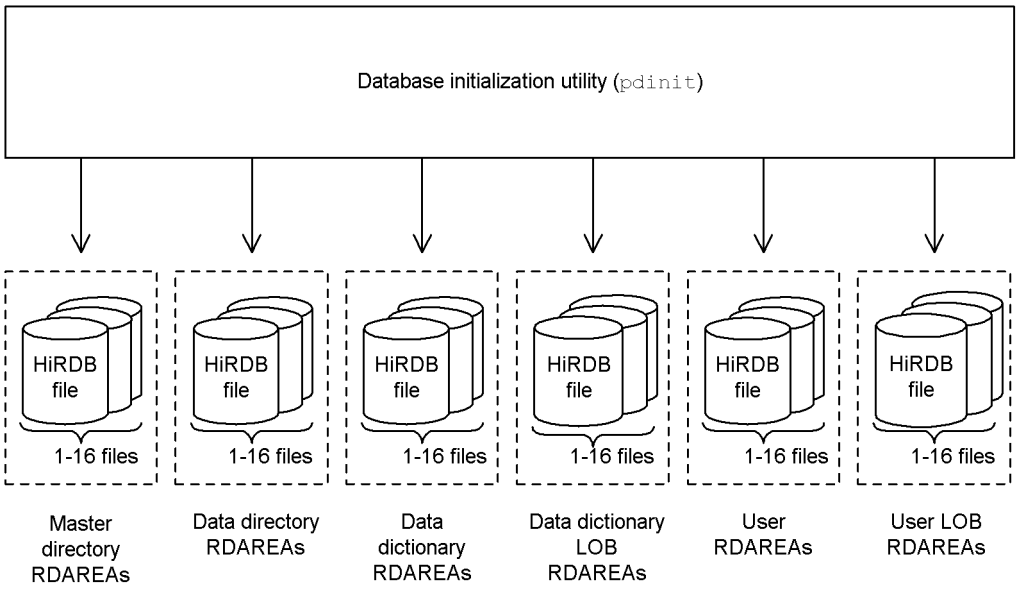
Before executing the database initialization utility, you must initialize the HiRDB file system areas in which HiRDB files are to be created. For details about initialization of HiRDB file system areas, see the *pdfmkfs command*.

The database initialization utility provides the following two functions:

- Initializes RDAREAs and simultaneously creates the master directory and data directory; also creates HiRDB-specific resources in data dictionary RDAREAs and data dictionary tables for managing the various privileges
- Defines the database system

Figure 3-1 provides an overview of the database initialization utility (`pdinit`).

Figure 3-1: Overview of database initialization utility (pdinit)



3.2 Command format

3.2.1 Format

```
pdinit -d control-statements-filename [-u authorization-identifier [-p password]]
```

3.2.2 Options

- `-d` *control-statements-filename* ～ <pathname> ((up to 255 characters)

  HiRDB/Single Server

  Specifies the absolute path name of the file that contains control statements for the database initialization utility.

  HiRDB/Parallel Server

  Specifies the absolute path name of the file that contains control statements for the database initialization utility. The control statements file must be created in the server machine where the dictionary server is located.

- `-u` *authorization-identifier* ～ <identifier> ((1-8 characters))

  Specifies an authorization identifier to which DBA privileges are to be granted and that is to be registered in HiRDB.

  - When this option is omitted

    When this option is omitted, the system assumes that `root` is specified in this option.

  - Rules for option specification

    If the authorization identifier contains lowercase letters, enclose the entire identifier in `\"`; otherwise, the system treats the specification as being in all uppercase letters.

  - Remarks

    This option can also be specified when HiRDB is installed as an extension of open middleware that uses HiRDB.

- `-p` *password* ～ <identifier> ((1-30 characters))

  Specifies the password for the authorization identifier specified in the `-u` option.

  If the password contains lowercase letters, enclose the entire identifier in `\"`; otherwise, the system treats the specification as being in all uppercase letters.

When this option is omitted, the system registers the authorization identifier character string specified in the -u option as the password.

The following shows the relationship between the -u and -p option values and the authorization identifier and password that are registered in HiRDB:

-u option specification	-p option specification	Authorization identifier registered in HiRDB	Password registered in HiRDB
HIR001	sytg0421	HIR001	sytg0421
HIR001	Omitted	HIR001	HIR001
Omitted	Omitted	root	root

3.3 Control statement

3.3.1 define system statement

This section explains the `define system` statement's operands. Each number in the table as follows corresponds to each operand.

No.	Operand
1	`define system [dicinf {unlimited | limited}]`
2	`[dictionary datatype mchar {use | nouse}]`
-	`;`

(1) dicinf {unlimited|limited}

Specifies whether or not information retrieved when data dictionary tables are accessed is to be restricted on the basis of user's privileges:

`unlimited`

All information can be retrieved when a data dictionary table is accessed

`limited`

The information to be retrieved when a data dictionary table is accessed is to be restricted on the basis of the user's privileges (a user with DBA privilege can retrieve all information; a user without DBA privilege can retrieve only the information authorized for his or her privileges)

When `limited` is specified, the number of UAP access tables for making retrievals from dictionary tables increases. Therefore, the number of simultaneously accessible base tables (the value of `pd_max_access_tables`) should be increased. Five base tables should be accessible simultaneously for each data dictionary table.

For details about how to set data dictionary table referencing privileges, see the *HiRDB Version 8 System Operation Guide*.

(2) dictionary datatype mchar {use|nouse}

Specifies whether or not mixed-character data can be used in the data of the dictionary tables:

`use`

Data type for dictionary table columns is to be `MVARCHAR`

`nouse`

Data type for dictionary table columns is to be `VARCHAR`

Following are guidelines for selecting MVARCHAR or VARCHAR:

- Select MVARCHAR when Kanji is used in table or column names.
- Select MVARCHAR when DBSHARE/Server 04-00 or a later version is used and Kanji is used in table or column names.
- Select VARCHAR in all other cases.

3.3.2 create rdarea statement

This section explains the create rdarea statement's operands. Each number in the following table corresponds to each operand.

No.	Operand
1	create [shared]
2	rdarea *RDAREA-name*
3	for {masterdirectory|datadirectory|datadictionary [of routines] |user used by {PUBLIC|*authorization-identifier* [,*authorization-identifier*]...} |LOB used by {HiRDB(*dictionary-table-name*)|PUBLIC |*authorization-identifier* [,*authorization-identifier*]...} |list}
4	[server name *server-name*]
5	[open attribute {INITIAL |DEFER |SCHEDULE}]
6	[page *page-length* characters]
7	[storage control segment *segment-size* pages]
8	[max entries *maximum-number-of-lists-that-can-be-registered*]
9	[extension {use *number-of-extension-segments* segments |nouse}]
10	file name "*HiRDB-file-system-area-name**HiRDB-filename*"
11	initial *HiRDB-file-segments-count* segments
-	[file name "*HiRDB-file-system-area-name**HiRDB-filename*"
-	initial *HiRDB-file-segments-count* segments]...
-	;

Note: Items 10 and 11 must not specified at the end of the create rdarea statement.

(1) shared

This operand is applicable only to a HiRDB/Parallel Server.

Specify this operand to create a shared RDAREA.

Rules

1. If you specify `shared`, you must specify `Y` in the `pd_sharedrdarea_use` operand in the system definition. If this specification is missing, an error occurs.
2. Specifying `shared` for a HiRDB/Single Server results in an error.
3. Only user RDAREAs can be created as shared RDAREAs.
4. You create a shared RDAREA in a HiRDB file system area whose purpose is SDB.
5. The created shared RDAREA can be referenced from any server other than one specified in the `server name` operand (it can be referenced from all back-end servers).

(2) RDAREA-name ～ <identifier> ((1-30))

Specifies the name of an RDAREA to be used by HiRDB. `ALL` cannot be specified as an RDAREA name. If the RDAREA name is enclosed in double quotation marks (`"`), it is treated as being case sensitive; otherwise, it is treated as all uppercase letters.

Examples

- create `rdarea "pdbuser01"` for...

  `pdbuser01` is used as the RDAREA name.

- create `rdarea pdbuser01` for...

`PDBUSER01` is used as the RDAREA name.

(3) for {masterdirectory|datadirectory|datadictionary [of routines]|user used by {PUBLIC|authorization-identifier [,authorization-identifier]...}|LOB used by {HiRDB(dictionary-table-name)|PUBLIC | authorization-identifier [,authorization-identifier]...}|list}

Specifies the type of RDAREA:

`masterdirectory`

Master directory RDAREA.

`datadirectory`

Data directory RDAREA.

`datadictionary`

Data dictionary RDAREA.

`of routines`

Specifies that the RDAREA stores data dictionary tables related to stored procedures, stored functions, and triggers separately from other data dictionary tables. This operand can be specified for only one data dictionary RDAREA. When this operand is specified, two data dictionary LOB RDAREAs must be specified.

`user used by {PUBLIC|`*authorization-identifier*`, [`*authorization-identifier*`] . . . }`

User RDAREA.

`PUBLIC`

Specifies that the user RDAREA to be added is a public RDAREA.

authorization-identifier`[,`*authorization-identifier*`] . . .` ～ <identifier> ((1-8))

Specifies that the user RDAREA to be added is a private RDAREA, and specifies the authorization identifiers of the users who can use the user RDAREA.

Rules

1. If an authorization identifier is enclosed in double quotation marks (`"`), it is treated as being case sensitive; otherwise, it is registered in the HiRDB system as all uppercase letters.
2. If you specify an authorization identifier, you can assign RDAREA usage privileges to the private RDAREA.
3. You can specify a maximum of 255 authorization identifiers.
4. You cannot specify `PUBLIC`, `MASTER`, `HiRDB`, or `ALL` as an authorization identifier.

`LOB used by {HiRDB (`*data dictionary-table-name*`)|PUBLIC|`*authorization-identifier*`[,`*authorization-identifier*`] . . . }`

Data dictionary LOB RDAREA or user LOB RDAREA

`HiRDB` (*data-dictionary-table-name*)

Specifies that the RDAREA is a data dictionary LOB RDAREA; `SQL_ROUTINES` must be specified as the data dictionary table name.

Rules

1. If you use stored procedures or stored functions, be sure to define this RDAREA.
2. There are separate RDAREAs for storing stored procedure and stored function definition source statements and for storing stored procedure and stored function objects; therefore, you need to specify this operand

twice (by specifying two separate `create rdarea` statements specifying these RDAREAs). Stored procedure and stored function definition source statements are stored in the first RDAREA, and stored procedure and stored function objects are stored in the second RDAREA.

3. If you have not defined these RDAREAs during the execution of the database initialization utility, but need to use stored procedures and stored functions, use the database structure modification utility to add these RDAREAs.

`PUBLIC`

Specifies that the user LOB RDAREA to be added is a public RDAREA.

authorization-identifier`[`,*authorization-identifier*`]` . . . ～ <identifier> ((1-8))

Specifies that the user LOB RDAREA to be added is a private RDAREA, and specifies the authorization identifiers of the users who can use the user LOB RDAREA.

Rules

1. If an authorization identifier is enclosed in double quotation marks (`"`), it is treated as being case sensitive; otherwise, it is registered in the HiRDB system as all uppercase letters.
2. If you specify an authorization identifier, you can assign RDAREA usage privileges to the private RDAREA.
3. You can specify a maximum of 255 authorization identifiers.
4. You cannot specify `PUBLIC`, `MASTER`, `HiRDB`, or `ALL` as an authorization identifier.

`list`

Specifies that the RDAREA is for lists.

(4) server name server-name ～ <identifier> ((1-8))

HiRDB/Single Server

This operand cannot be specified for a HiRDB/Single Server.

HiRDB/Parallel Server

Specifies the name of the server that is to manage the RDAREA.

For the master directory RDAREA, the data directory RDAREA, a data dictionary RDAREA, or a data dictionary LOB RDAREA, the name of the dictionary server must be specified. If any other server's name is specified, the name of the dictionary server is assumed.

For a user RDAREA or a user LOB RDAREA, the name of a back-end server must be specified. If any other server's name is specified, an error results.

When a shared RDAREA is created, the server specified in this operand becomes the updatable back-end server.

When multiple RDAREAs are to be managed by a single server, as many `create rdarea` statements as there are RDAREAs must be specified consecutively for that server name. Examples:

Example 1

```
create rdarea A ... server name BES1 ...;
create rdarea B ... server name BES1 ...;
create rdarea C ... server name BES2 ...;
```

This specification does not result in an error, because the two `create rdarea` statements with the same server name (`BES1`) are specified consecutively.

Example 2

```
create rdarea A ... server name BES1 ...;
create rdarea B ... server name BES2 ...;
create rdarea C ... server name BES1 ...;
```

This specification results in an error, because the `create rdarea` statements with the same server name (`BES1`) are not specified consecutively.

(5) open attribute {INITIAL|DEFER|SCHEDULE}

Specifies the RDAREA's open timing.

Criteria

Usually specify `INITIAL` unless a special operation mode is used.

If HiRDB startup takes time because there are many RDAREAs, specify `DEFER` to avoid the startup process.

Rules

1. This operand takes effect only when `Y` is specified in the `pd_rdarea_open_attribute_use` operand.
2. You can specify this operand for user RDAREAs, user LOB RDAREAs, and list RDAREAs. If specified for any other RDAREA, this operand is ignored (the RDAREA open timing for all other RDAREA types is `INITIAL`).
3. If this operand is omitted, the system assumes the value of the `pd_rdarea_open_attribute` operand.
4. If the HiRDB file system area allocated in the corresponding RDAREA is shared by another RDAREA, all such sharing RDAREAs must have the same open timing. Otherwise, expected results may not be obtained from

those RDAREAs.

INITIAL

The RDAREA's HiRDB file system area is to be opened at the time of HiRDB startup and the RDAREA information is to be made resident in memory; thus, high-speed operation is available from the first SQL.

Notes

1. The RDAREA's initial status at system startup is open and remains unchanged until an operation command is executed, unless it is placed in error shutdown status due to an error.
2. The INITIAL open attribute does not permit access to a closed RDAREA.

DEFER

The RDAREA's HiRDB file system area is not to be opened at the time of HiRDB startup; rather, it is to be opened and its RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed. Thus, high-speed operation is available beginning with the second access to the HiRDB file system area.

Notes

1. The RDAREA's initial status at system startup is closed and changes to open the first time an RDAREA in its HiRDB file system area is accessed. The HiRDB file system area's RDAREAs remain in open status thereafter until an operation command is executed, unless they are placed in error shutdown status due to an error.
2. The DEFER attribute permits access to an RDAREA in closed status.

SCHEDULE

The RDAREA's HiRDB file system area is not to be opened at the time of HiRDB startup; rather, it is to be opened and the RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed from within a transaction. When an HiRDB file system area has been opened by a transaction, it is closed when the transaction terminates. When this attribute is used, transaction workload increases because open processing must be performed on the HiRDB file system area even after the first-time access.

Notes

1. The RDAREA's initial status at system startup is closed and changes to open when an RDAREA in its HiRDB file system area is first accessed by a transaction. When the transaction is terminated, all the RDAREAs opened by it are closed; however, the pdopen command can be used to

keep RDAREAs in open status until they are placed in shutdown close status. You can also use other operation commands to change the RDAREA status. If an error is detected, RDAREAs are placed in error shutdown status.

2. The SCHEDULE attribute permits access to an RDAREA in closed status.

(6) page page-length characters ～ <unsigned integer> ((4096-30720)) <<4096 or 8192>>

Specifies in bytes as a multiple of 2048 the page length for the HiRDB files that are to comprise the RDAREA. If you specified the sector length when you created the HiRDB file system area, specify a multiple of 2,048 that is also a multiple of the sector length.

- Master directory or data directory RDAREA

  4096 must be specified for the master directory or data directory RDAREA; if any other value is specified, 4096 is assumed.

- Data dictionary LOB or user LOB RDAREA

  8192 must be specified for a data dictionary LOB or user LOB RDAREA; if any other value is specified, 8192 is assumed.

For details about the page lengths of other types of RDAREAs, see the *HiRDB Version 8 Installation and Design Guide* and specify the appropriate value. The page length specified in this operand is used as the smallest unit for I/O operations performed on the RDAREA by HiRDB.

(7) storage control segment segment-size pages ～ <unsigned integer> ((1-16000)) <<1 or 50>>

Specifies in pages the size of one segment.

- Master directory or data directory RDAREA

  50 must be specified for a data dictionary LOB or user LOB RDAREA; if any other value is specified, 50 is assumed.

- Data dictionary LOB or user LOB RDAREA

  1 must be specified for a data dictionary LOB or user LOB RDAREA; if any other value is specified, 1 is assumed.

For details about the segment sizes of other types of RDAREAs, see the *HiRDB Version 8 Installation and Design Guide* and specify the appropriate value.

(8) max entries maximum-number-of-lists-that-can-be-registered ～ <unsigned integer> ((500-50000 and a multiple of 500)) <<500>>

Specifies the maximum number of lists that can be created in the list RDAREA, expressed as a multiple of 500. If the specified value is not a multiple of 500, the

system rounds it up to the nearest multiple of 500.

If this operand is specified for an RDAREA other than the list RDAREA, an error results.

(9) extension {use number-of-extension-segments segments|nouse}

Specifies whether or not to apply automatic extension of RDAREA.

Automatic extension of RDAREA is a facility that is applied in the event of an RDAREA space shortage; it extends the RDAREA automatically if the HiRDB file system area has sufficient space. If you specify this facility for the RDAREA and a shortage of segments occur, unused segments are allocated to the RDAREA. These unused segments are added to the last HiRDB file that constitutes the RDAREA.

Prerequisites

1. The `-e` option (extension count) must be specified for the HiRDB file system area where the RDAREA is located.
2. The HiRDB file system area must have enough space to contain the last HiRDB file constituting the RDAREA.

`use` *number-of-extension-segments*

Applies automatic extension of RDAREA.

Automatic extension of RDAREA is applicable to data dictionary RDAREAs, data dictionary LOB RDAREAs, user RDAREAs, and user LOB RDAREAs. For other RDAREAs, the system assumes `nouse` even if `use` is specified.

For an RDAREA other than a LOB RDAREA, automatic extension occurs when there are no more used free segments or used segments in the RDAREA. For a LOB RDAREA, it occurs when there are no more unused segments.

number-of-extension-segments ～ <unsigned integer> ((1-64000))

Specifies the number of extension segments.

There can be a maximum of 24 extents for HiRDB files. Exceeding this results in an error. The maximum number of extensions per HiRDB file system area is determined by the value specified when the HiRDB file system area is created. Therefore, you should take into account the number of files in the HiRDB file system area and an extension frequency when defining the maximum number of extensions.

`nouse`

Specifies that automatic extension of RDAREA is not to be applied.

Notes

1. If the system is unable to allocate unused segments due to insufficient space

in the HiRDB file system area, extend or re-initialize the RDAREA or reorganize the table with the database reorganization utility.

2. If the number of extensions exceeds the maximum value, either integrate the extensions in the HiRDB file system area where the RDAREA is located or add another HiRDB file system area's HiRDB files to the RDAREA. To integrate extensions, make a backup copy with `pdfbkup`, initialize the HiRDB file system area with `pdfmkfs`, then restore the extensions from the backup copy using `pdfrstr`.
3. The last file is locked from the start to the end of automatic RDAREA extension processing.

(10) file name "HiRDB-file-system-area-name\HiRDB-filename"

～ ((up to 167 characters))

Specifies a file that is to be allocated to the RDAREA, in terms of the name of the HiRDB file system area and the name of the HiRDB file. *HiRDB-file-system-area-name**HiRDB-filename* must be enclosed in double quotation marks.

Rules

1. Do not specify linefeed characters inside the double quotation marks.
2. You can allocate a maximum of 16 HiRDB files to one RDAREA.
3. Make sure that *HiRDB-file-system-area-name**HiRDB-filename* is not duplicated in the same HiRDB system.
4. For a list RDAREA, use a HiRDB file system area with the `WORK` purpose.
5. If the master directory RDAREA consists of multiple HiRDB files, use the name specified in the `pd_master_file_name` operand of the system definitions as the name of the first HiRDB file.
6. In "*HiRDB-file-system-area-name**HiRDB-filename*", *HiRDB-file-system-area-name* is not case sensitive, but *HiRDB-filename* is case sensitive.

HiRDB-file-system-area-name ～ <pathname>

Specifies the name of a HiRDB file system area.

HiRDB-filename ～ <HiRDB filename> ((1-30 characters))

Specifies the name of a HiRDB file. This name cannot begin with `pl`.

(11) initial HiRDB-file-segments-count segments

Specifies the number of segments in the particular RDAREA file. The size of a HiRDB file cannot exceed 2 GB, regardless of the number of segments. The total number of

segments in all HiRDB files comprising each RDAREA must not be fewer than the applicable value in the following table:

RDAREA type	Minimum number of segments
Master directory RDAREA	4 + ↑ total number of RDAREAs ÷ 800 ↑
Data directory RDAREA	1
Data dictionary RDAREA[1]	177[3]
Data dictionary LOB RDAREA[2]	1
User RDAREA	1
User LOB RDAREA	1
List RDAREA	1

[1] When there is more than one data dictionary RDAREA, one data dictionary table is stored in each RDAREA, one at a time, starting with the RDAREA that has the most segments. The index for a data dictionary table is stored in the same RDAREA that stores the data dictionary table. The minimum number of segments when there are multiple data dictionary RDAREAs is the number of stored data dictionary tables and the number of indexes for those tables.

Storage order	Name of data dictionary table	Number of indexes
1	SQL_PHYSICAL_FILES	2
2	SQL_RDAREAS	2
3	SQL_TABLES	2
4	SQL_COLUMNS	3
5	SQL_INDEXES	3
6	SQL_USERS	1
7	SQL_RDAREA_PRIVILEGES	2
8	SQL_TABLE_PRIVILEGES	2
9	SQL_DIV_TABLE	3
10	Table used by the system	2
11	SQL_INDEX_COLINF	2
12	SQL_TABLE_STATISTICS	1

Storage order	Name of data dictionary table	Number of indexes
13	SQL_COLUMN_STATISTICS	1
14	SQL_INDEX_STATISTICS	2
15	SQL_VIEW_TABLE_USAGE	3
16	SQL_VIEWS	2
17	Table used by the system	2
18	SQL_DIV_INDEX	2
19	SQL_DIV_COLUMN	2
20	SQL_REFERENTIAL_CONSTRANTS	3
21	SQL_ALIASES	1
22	SQL_DATATYPES	3
23	SQL_DATATYPE_RESOURCES	2
24	SQL_TABLE_RESOURCES	3
25	SQL_PLUGINS	3
26	SQL_PLUGIN_ROUTINES	4
27	SQL_PLUGIN_ROUTINE_PARAMS	3
28	SQL_INDEX_TYPES	2
29	SQL_INDEX_RESOURCES	2
30	SQL_INDEX_DATATYPE	2
31	SQL_INDEX_FUNCTION	2
32	SQL_TYPE_RESOURCES	3
33	SQL_INDEX_TYPE_FUNCTION	1
34	SQL_EXCEPT	3
35	SQL_FOREIGN_SERVERS	0
36	SQL_USER_MAPPINGS	1
37	Table used by the system	0
38	Table used by the system	0
39	SQL_TRIGGERS	4

Storage order	Name of data dictionary table	Number of indexes
40	Table used by the system	2
41	SQL_TRIGGER_COLUMNS	2
42	SQL_TRIGGER_DEF_SOURCE	2
43	SQL_TRIGGER_USAGE	4
44	SQL_PARTKEY	1
45	SQL_PARTKEY_DIVISION	1
46	SQL_AUDITS	3
47	Table used by the system	2
48	SQL_KEYCOLUMN_USAGE	2
49	SQL_TABLE_CONSTRAINTS	2
50	SQL_CHECKS	2
51	SQL_CHECK_COLUMNS	2
52	SQL_DIV_TYPES	1
53	SQL_SYSPARAMS	1
54	Table used by the system	1
55	SQL_ROUTINES	4
56	SQL_ROUTINE_RESOURCE	4
57	SQL_ROUTINE_PARAM	3

[2] If no data dictionary LOB RDAREAs are specified, SQL_ROUTINES, SQL_ROUTINE_RESOURCES, and SQL_ROUTINE_PARAMS are not created. If there are data dictionary RDAREAs for storing data dictionary tables related to stored procedures and stored functions (RDAREAs for which of routines is specified), SQL_ROUTINES, SQL_ROUTINE_RESOURCES, and SQL_ROUTINE_PARAMS are stored in those data dictionary RDAREAs.

[3] This is the total of the number of data dictionary tables and data dictionary table indexes created by HiRDB.

The formula *total number of pages/segment size* is used to obtain the number of segments. For details about how to estimate the total number of pages, see the *HiRDB Version 8 Installation and Design Guide*.

3.3.3 Notes

1. A comment specified in a control statement must be enclosed between `/*` and `*/`.
2. When specifying the `define system` and the `create rdarea` statements at the same time, specify the `define system` statement first.
3. The following table shows the specification order and the permitted numbers of `create rdarea` statements:

Specification order	RDAREA type	Number of statements
1	Statement for master directory RDAREA	1
2	Statement for data directory RDAREA	1
3	Statements for data dictionary RDAREAs	1 to number of data dictionary tables*
4	Statements for data dictionary LOB RDAREAs	0 or 2
5	Statements for user RDAREAs, user LOB RDAREAs, and list RDAREAs	Any number

Note 1

Check that the number of RDAREAs does not exceed the maximum permissible number of RDAREAs, as specified in the system common definition (value of `pd_max_rdarea_no`). Also check that the number of HiRDB files that constitute the RDAREAs do not exceeds the maximum permissible number of HiRDB files that can constitute RDAREAs (value of `pd_max_file_no`).

Note 2

A created shared RDAREA is counted as if it has been created at every back-end server. Therefore, if the number of RDAREAs or the number of HiRDB files constituting the RDAREAs exceeds the permitted maximum value at any back-end server, the shared RDAREA cannot be created.

* When there are data dictionary RDAREAs for which `of routines` is specified, the maximum number of data dictionary RDAREAs that can be specified is (number of data dictionary tables - 2).

3.4 Rules and notes

(1) Rules

1. The database initialization utility can be executed only while HiRDB is in end-of-initialization wait status (waiting after issuing the KFPS05201-Q message). HiRDB is placed in end-of-initialization wait status when no master directory RDAREA has been initialized or when HiRDB is started by the pdstart command with the -i option specified.
2. The database initialization utility must be executed at the server machine that contains the single server or the server machine where the system manager is located.
3. If an error occurs while the database initialization utility is executing, processing is cancelled immediately. The error must be corrected on the basis of the error messages output to the standard error output and to the message log file and then the database initialization utility must be re-executed.
4. If the database initialization utility is terminated abnormally (aborted or placed in process shutdown status), the initialization status file (pdinit) may remain in %PDDIR%\spool. This file must be deleted before the database initialization utility is re-executed.
5. If the standby-less system switchover (1:1) facility is used and the alternate BES is placed in standby status, pdinit cannot be executed. To execute pdinit while using the standby-less system switchover (1:1) facility, start all normal BESs as running systems.

(2) Note

1. The following are the pdinit utility's return codes:

   0: Normal termination

   4: Normal termination (warning-level error occurred, but processing terminated normally)

   8: Normal termination (initialization terminated normally, but the communication about termination of initialization resulted in an error)

   12: Abnormal termination

2. The pdinit results can be checked by the following methods:

   pddbls command (-r ALL -a specified)

   Retrieval of data dictionary table

   Database condition analysis utility

3. If you selected `utf-8` as the character encoding in the `pdntenv` command, you can use a control statements file that contains a BOM. However, only ASCII characters are permitted for comments in the control statements file. If character encoding other than ASCII is used, `pdinit` may not function correctly.

3.5 Examples

Example 1 shows an example of the use of the database initialization utility.

Example 1

Create the following RDAREAs:

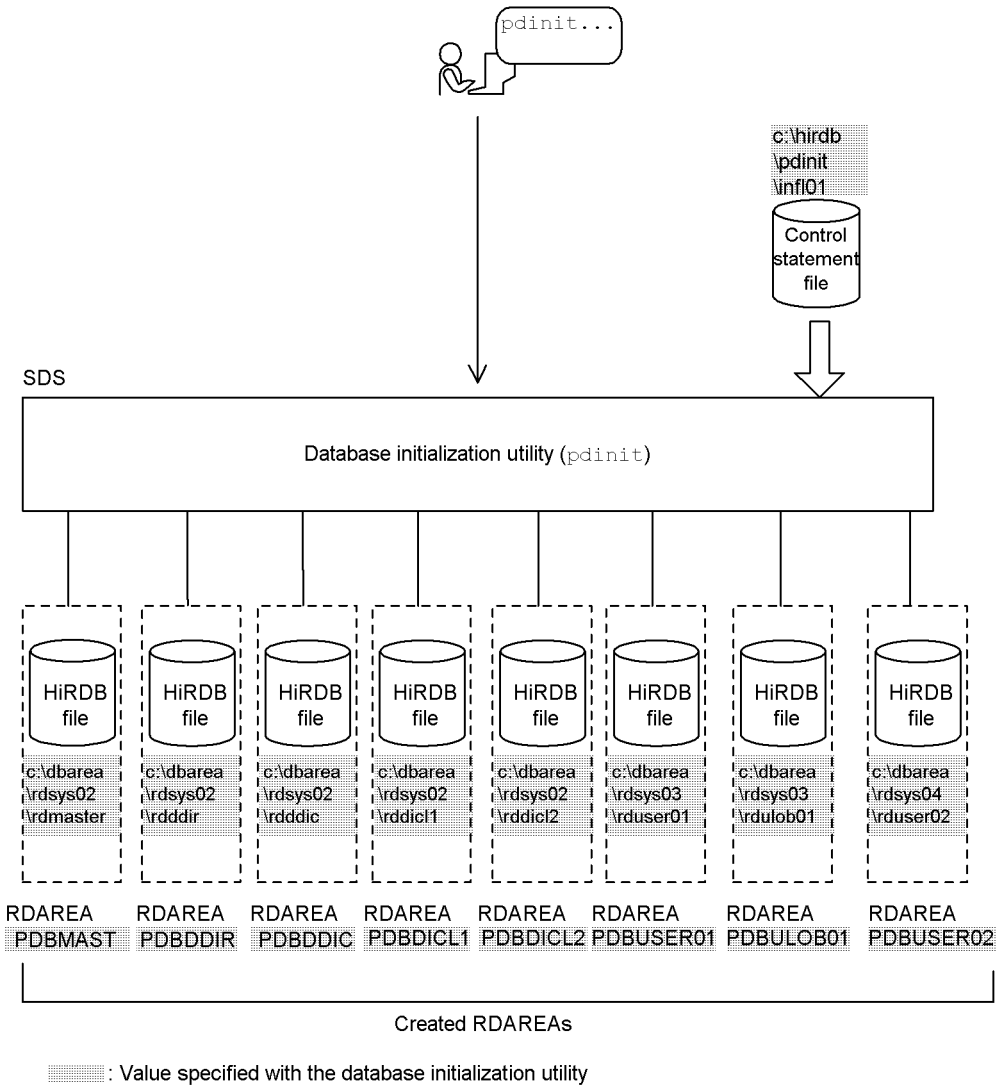
- PDBMAST (master directory RDAREA)
- PDBDDIR (data directory RDAREA)
- PDBDDIC (data dictionary RDAREA)
- PDBDICL1 (data dictionary LOB RDAREA)
- PDBDICL2 (data dictionary LOB RDAREA)
- PDBUSER01 (user RDAREA)
- PDBULOB01 (user LOB RDAREA)
- PDBUSER02 (user RDAREA)

The following HiRDB file system areas have already been created:

- c:\dbarea\rdsys02
- c:\dbarea\rdsys03
- c:\dbarea\rdsys04

HiRDB/Single Server

Overview



Command execution

```
pdinit -d c:\hirdb\pdinit\infl01
```

Contents of control statement file (c:\hirdb\pdinit\infl01)

```
/* Definition of master directory RDAREA */
 create rdarea PDBMAST   ..............................1
               for masterdirectory   ..................2
               page 4096 characters   .................3
               storage control segment 50 pages   ......4
               file name "c:\dbarea\rdsys02\rdmaster"
                   initial 10 segments ;   ............5


/* Definition of data directory RDAREA */
 create rdarea PDBDDIR   ..............................6
               for datadirectory   ....................7
               page 4096 characters   .................8
               storage control segment 50 pages   ......9
               file name "c:\dbarea\rdsys02\rdddir"
                   initial 5 segments ;   ............10


/* Definition of data dictionary RDAREA */
 create rdarea PDBDDIC   ..............................11
               for datadictionary   ...................12
               page 4096 characters   .................13
               storage control segment 30 pages   ......14
               file name "c:\dbarea\rdsys02\rdddic"
                   initial 150 segments ;   ...........15


/* Definition of data dictionary LOB RDAREA */
 create rdarea PDBDICL1   .............................16
               for LOB used by HiRDB(SQL_ROUTINES)   ...17
               page 8192 characters   .................18
               storage control segment 1 pages   .......19
               file name "c:\dbarea\rdsys02\rddicl1"
                   initial 200 segments ;   ...........20


/* Definition of data dictionary LOB RDAREA */
 create rdarea PDBDICL2   .............................21
               for LOB used by HiRDB(SQL_ROUTINES)   ...22
               page 8192 characters   .................23
               storage control segment 1 pages   .......24
               file name "c:\dbarea\rdsys02\rddicl2"
                   initial 2000 segments ;   ..........25


/* Definition of user RDAREA */
 create rdarea PDBUSER01   ............................26
               for user used by PUBLIC   ..............27
               page 4096 characters   .................28
               storage control segment 150 pages   .....29
               file name "c:\dbarea\rdsys03\rduser01"
                   initial 50 segments ;   ...........30
```

```
/* Definition of user LOB RDAREA */
 create rdarea PDBULOB01   ..............................31
               for LOB used by PUBLIC   ................32
               page 8192 characters   ..................33
               storage control segment 1 pages   .......34
               file name "c:\dbarea\rdsys03\rdulob01"
                   initial 50 segments ;   .............35


/* Definition of user RDAREA */
 create rdarea PDBUSER02   ..............................36
               for user used by USERID   ...............37
               page 4096 characters   ..................38
               storage control segment 50 pages   ......39
               file name "c:\dbarea\rdsys04\rduser02"
                   initial 100 segments ;   ............40
```

Explanation

1. Name of RDAREA: `PDBMAST`
2. Type of RDAREA: Master directory RDAREA
3. Page length: 4096 bytes
4. Segment size: 50 pages
5. HiRDB file comprising the RDAREA:
   - Name: `c:\dbarea\rdsys02\pdmaster`
   - Number of segments: 10
6. Name of RDAREA: `PDBDDIR`
7. Type of RDAREA: Data directory RDAREA
8. Page length: 4096 bytes
9. Segment size: 50 pages
10. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rdddir`
    - Number of segments: 5
11. Name of RDAREA: `PDBDDIC`
12. Type of RDAREA: Data dictionary RDAREA
13. Page length: 4096 bytes
14. Segment size: 30 pages
15. HiRDB file comprising the RDAREA:

- Name: `c:\dbarea\rdsys02\rdddic`
- Number of segments: 150

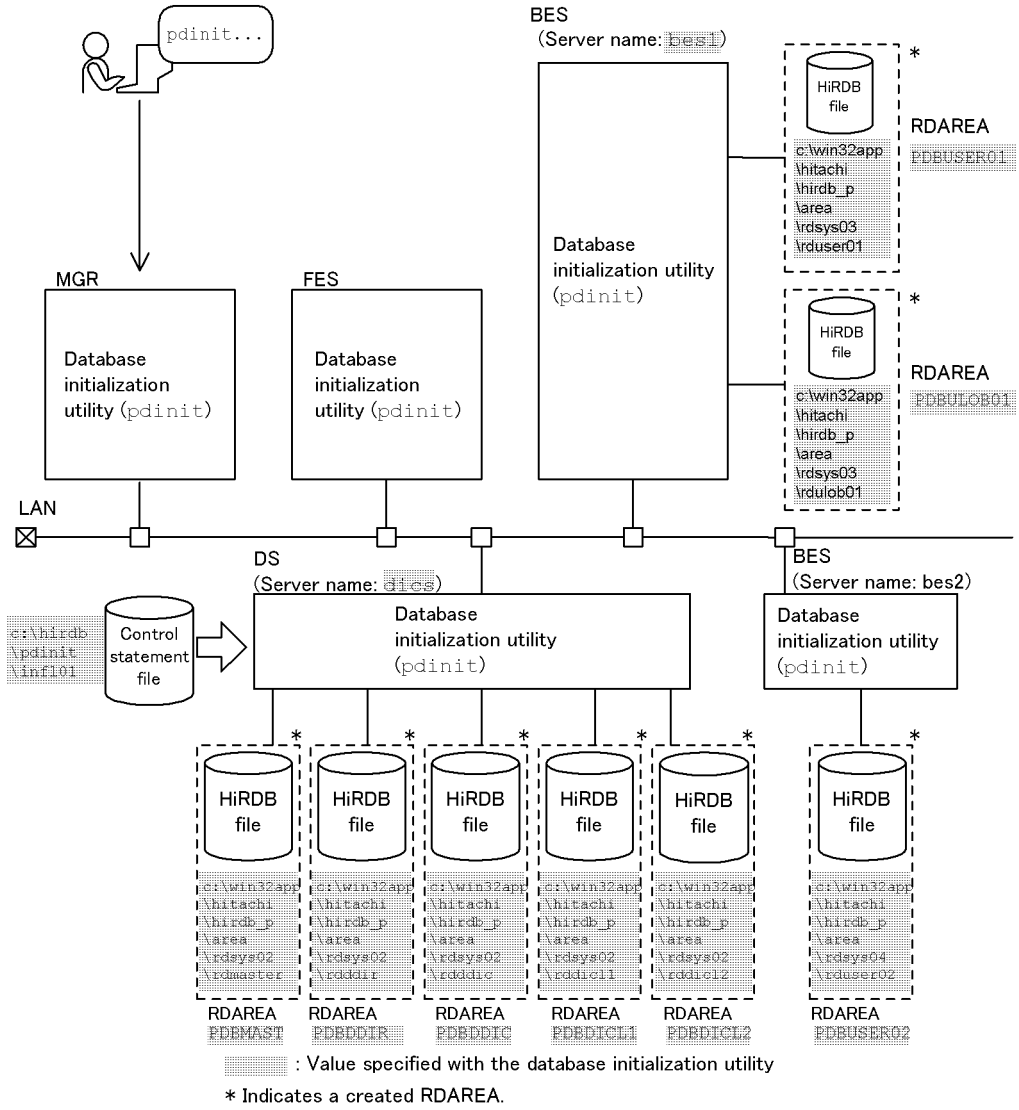
16. Name of RDAREA: `PDBDICL1`
17. Type of RDAREA: Data dictionary LOB RDAREA (stores stored procedure definition source statements)
18. Page length: 8192 bytes
19. Segment size: 1 page
20. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rddicl1`
    - Number of segments: 200
21. Name of RDAREA: `PDBDICL2`
22. Type of RDAREA: Data dictionary LOB RDAREA (stores stored procedure objects)
23. Page length: 8192 bytes
24. Segment size: 1 page
25. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rddicl2`
    - Number of segments: 2000
26. Name of RDAREA: `PDBUSER01`
27. Type of RDAREA: User RDAREA (public)
28. Page length: 4096 bytes
29. Segment size: 150 pages
30. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys03\rduser01`
    - Number of segments: 50
31. Name of RDAREA: `PDBULOB01`
32. Type of RDAREA: User LOB RDAREA (public)
33. Page length: 8192 bytes
34. Segment size: 1 page
35. HiRDB file comprising the RDAREA:

- Name: `c:\dbarea\rdsys03\rdulob01`
- Number of segments: 50

36. Name of RDAREA: `PDBUSER02`
37. Type of RDAREA: User RDAREA (private)
    Authorization identifier: `USERID`
38. Page length: 4096 bytes
39. Segment size: 50 pages
40. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys04\rduser02`
    - Number of segments: 100

HiRDB/Parallel Server

Overview



Command execution

```
pdinit -d c:\hirdb\pdinit\infl01
```

Contents of control statement file (c:\hirdb\pdinit\infl01)

```
/* Definition of master directory RDAREA */
 create rdarea PDBMAST   .....................................1
               for masterdirectory   .........................2
               server name dics   ............................3
               page 4096 characters   ........................4
               storage control segment 50 pages   ............5
               file name "c:\dbarea\rdsys02\rdmaster"
                   initial 10 segments ;   ...................6

/* Definition of data directory RDAREA */
 create rdarea PDBDDIR   .....................................7
               for datadirectory   ...........................8
               server name dics   ............................9
               page 4096 characters   ........................10
               storage control segment 50 pages   ............11
               file name "c:\dbarea\rdsys02\rdddir"
                   initial 5 segments ;   ....................12


/* Definition of data dictionary RDAREA */
 create rdarea PDBDDIC   .....................................13
               for datadictionary   ..........................14
               server name dics   ............................15
               page 4096 characters   ........................16
               storage control segment 30 pages   ............17
               file name "c:\dbarea\rdsys02\rdddic"
                   initial 150 segments ;   ..................18


/* Definition of data dictionary LOB RDAREA */
 create rdarea PDBDICL1   ....................................19
               for LOB used by HiRDB(SQL_ROUTINES)   .........20
               server name dics   ............................21
               page 8192 characters   ........................22
               storage control segment 1 pages   ............23
               file name "c:\dbarea\rdsys02\rddicl1"
                   initial 200 segments ;   ..................24


/* Definition of data dictionary LOB RDAREA */
 create rdarea PDBDICL2   ....................................25
               for LOB used by HiRDB(SQL_ROUTINES)   .........26
               server name dics   ............................27
               page 8192 characters   ........................28
               storage control segment 1 pages   ............29
               file name "c:\dbarea\rdsys02\rddicl2"
                   initial 2000 segments ;   .................30
```

```
/* Definition of user RDAREA */
 create rdarea PDBUSER01   .....................................31
               for user used by PUBLIC   .......................32
               server name bes1   ..............................33
               page 4096 characters   ..........................34
               storage control segment 150 pages   .............35
               file name "c:\dbarea\rdsys03\rduser01"
                   initial 50 segments ;   .....................36


/* Definition of user LOB RDAREA */
 create rdarea PDBULOB01   .....................................37
               for LOB used by PUBLIC   ........................38
               server name bes1   ..............................39
               page 8192 characters   ..........................40
               storage control segment 1 pages   ...............41
               file name "c:\dbarea\rdsys03\rdulob01"
                   initial 50 segments ;   .....................42


/* Definition of user RDAREA */
 create rdarea PDBUSER02   .....................................43
               for user used by USERID   .......................44
               server name bes2   ..............................45
               page 4096 characters   ..........................46
               storage control segment 50 pages   ..............47
               file name "c:\dbarea\rdsys04\rduser02"
                   initial 100 segments ;   ....................48
```

Explanation

1. Name of RDAREA: `PDBMAST`
2. Type of RDAREA: Master directory RDAREA
3. Name of server to manage the RDAREA: `dics`
4. Page length: 4096 bytes
5. Segment size: 50 pages
6. HiRDB file comprising the RDAREA:
   - Name: `c:\dbarea\rdsys02\pdmaster`
   - Number of segments: 10
7. Name of RDAREA: `PDBDDIR`
8. Type of RDAREA: Data directory RDAREA
9. Name of server to manage the RDAREA: `dics`
10. Page length: 4096 bytes

11. Segment size: 50 pages
12. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rdddir`
    - Number of segments: 5
13. Name of RDAREA: `PDBDDIC`
14. Type of RDAREA: Data dictionary RDAREA
15. Name of server to manage the RDAREA: `dics`
16. Page length: 4096 bytes
17. Segment size: 30 pages
18. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rdddic`
    - Number of segments: 150
19. Name of RDAREA: `PDBDICL1`
20. Type of RDAREA: Data dictionary LOB RDAREA (stores stored procedure definition source statements)
21. Name of server to manage the RDAREA: `dics`
22. Page length: 8192 bytes
23. Segment size: 1 page
24. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rddicl1`
    - Number of segments: 200
25. Name of RDAREA: `PDBDICL2`
26. Type of RDAREA: Data dictionary LOB RDAREA (stores stored procedure objects)
27. Name of server to manage the RDAREA: `dics`
28. Page length: 8192 bytes
29. Segment size: 1 page
30. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys02\rddicl2`
    - Number of segments: 2000

31. Name of RDAREA: `PDBUSER01`
32. Type of RDAREA: User RDAREA (public)
33. Name of server to manage the RDAREA: `bes1`
34. Page length: 4096 bytes
35. Segment size: 150 pages
36. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys03\rduser01`
    - Number of segments: 50
37. Name of RDAREA: `PDBULOB01`
38. Type of RDAREA: User LOB RDAREA (public)
39. Name of server to manage the RDAREA: `bes1`
40. Page length: 8192 bytes
41. Segment size: 1 page
42. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys03\rdulob01`
    - Number of segments: 50
43. Name of RDAREA: `PDBUSER02`
44. Type of RDAREA: User RDAREA (private)
    Authorization identifier: `USERID`
45. Name of server to manage the RDAREA: `bes2`
46. Page length: 4096 bytes
47. Segment size: 50 pages
48. HiRDB file comprising the RDAREA:
    - Name: `c:\dbarea\rdsys04\rduser02`
    - Number of segments: 100

Chapter

4. Database Definition Utility (pddef)

This chapter describes the database definition utility (`pddef`) that permits modifications of schema definitions and contents.

This chapter contains the following sections:

4.1 Overview

4.1.1 Function

The database definition utility (pddef) defines schemas and modifies existing schema definitions.

This utility executes a definition SQL entered from the standard input, defines or changes schema information, then outputs the execution result to the standard output. The definition information is stored in data dictionary tables.

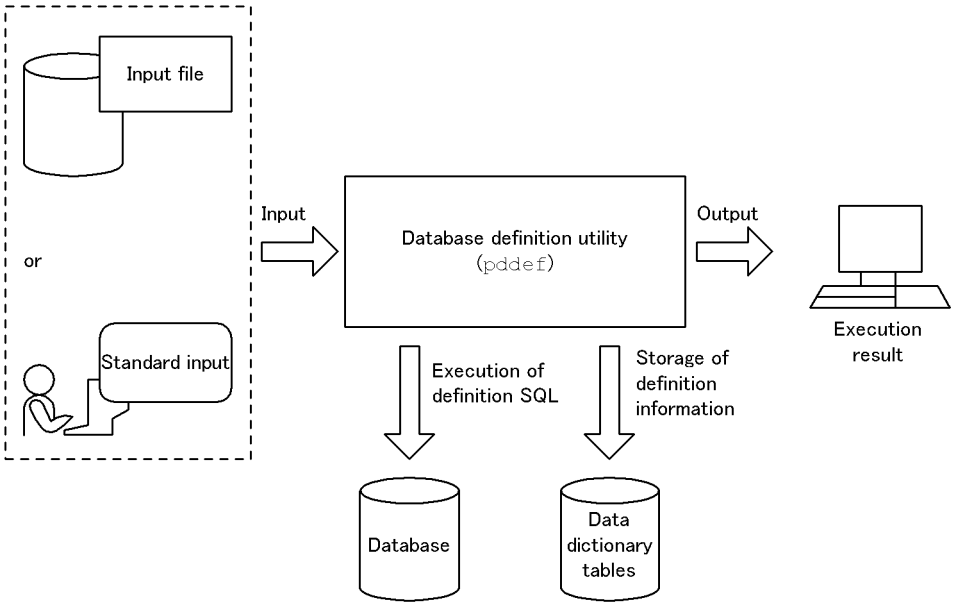
Before executing the database definition utility, you need to specify the client environment definitions. Specify at least PDHOST, PDNAMEPORT, and PDUSER. The following shows an example:

```
PDHOST=host1
PDNAMEPORT=20001
PDUSER=\"k916321\"/\"k916321\"
        :
```

For details about the client environment definitions, see the *HiRDB Version 8 UAP Development Guide*.

Figure 4-1 provides an overview of the database definition utility (pddef).

Figure 4-1: Overview of database definition utility (pddef)



4.1.2 Executor

User with the CONNECT privilege. However, the privilege required depends on the SQL to be executed. For details about the privileges required to execute SQLs, see the *HiRDB Version 8 SQL Reference*.

4.2 Command format

4.2.1 Format

```
pddef [-e] [-s] [-R] {definition-SQL|input-file}
```

4.2.2 Options

- `-e`

Specifies that `END_FUNC;` (for `CREATE FUNCTION`), `END_TYPE;` (for `CREATE TYPE`), or `END_TRIG` (for `CREATE TRIGGER`) is to be added to demarcate the end of the SQL statement.

For `CREATE FUNCTION`:

The system treats a line of code through `END_PROC;` or `END_FUNC;` as one SQL statement.

For `CREATE TYPE`:

The system treats a line of code through `END_PROC;` or `END_TYPE;` as one SQL statement.

For `CREATE TRIGGER`:

The system treats a line of code through `END_PROC;` or `END_TRIG;` as one SQL statement.

For the handling of continuation lines for SQL statements, see *4.3 Handling of continuation lines*.

- `-s`

Specifies that an input wait prompt is to be displayed when a definition SQL is entered directly.

There are two different prompts that are displayed according to the input status:

`COMMAND?`: Indicates a prompt for a normal entry.

`NEXT?`: Indicates a prompt for a continuation line of an SQL statement.

The following shows examples:

```
pddef -s
COMMAND ?
+----2----+----3----+----4----+----5----+----6----+----7----+
```

```
CREATE TABLE T1 (
NEXT ?
+----2----+----3----+----4----+----5----+----6----+----7----+

C1 INT,
NEXT ?
+----2----+----3----+----4----+----5----+----6----+----7----+

C2 CHAR(64)
NEXT ?
+----2----+----3----+----4----+----5----+----6----+----7----+

);
KFPA12000-I Processing of SQL completed
COMMAND ?
+----2----+----3----+----4----+----5----+----6----+----7----+
```

- {*definition-SQL*|*input-file*}

  Specifies a definition SQL statement.

  You can directly enter definition SQL statements or use an input file that contains definition SQL statements. For details about the definition SQL, see the *HiRDB Version 8 SQL Reference*.

Directly entering SQL statements:

When `pddef` is executed, the utility waits for entry of definition SQL statements, which should then be entered. You cannot enter an SQL statement immediately after `pddef`. Specify a semicolon (`;`) at the end of each SQL statement.

To terminate `pddef`, press the **CTRL** + **Z** keys, then press the **ENTER** key.

Specifying an input file:

Create an input file containing definition SQL statements using a text editor such as Notepad. Redirect the standard input to this input file.

Example

```
pddef <infile
```

- `R`

Specifies that the `pddef` return code is to be set to `4` if the `SQLCODE` for any SQL statement that executes is a negative value (including abnormal termination during HiRDB connection). You specify this option when you wish to detect errors in SQL statements during execution of `pddef`.

4.3 Handling of continuation lines

(1) SQL other than CREATE PROCEDURE, CREATE FUNCTION, CREATE TYPE, or CREATE TRIGGER

If a line ends with a semicolon (;), the system treats the line of code as one SQL statement (the system ignores any space or control character following the semicolon). A line that does not end with a semicolon is assumed to be continued on the next line.

Example

```
CREATE TABLE T1 (C1 INT,          (Continued)
            C2 CHAR(10),          (Continued)
            C3 CHAR(10));         (End of SQL statement)
```

If a line ends with a semicolon that is enclosed in double quotation marks ("), the system treats it as the end of the SQL statement.

(2) CREATE PROCEDURE, CREATE FUNCTION, CREATE TYPE, and CREATE TRIGGER

Table 4-1 shows the handling of continuation lines with CREATE PROCEDURE, CREATE FUNCTION, CREATE TYPE, and CREATE TRIGGER. If a line ends with a character string indicated in Table 4-1, the system treats the line of code up to that character string as an SQL statement (the system ignores any space or control character following the semicolon).

Table 4-1: Handling of continuation lines with CREATE PROCEDURE, CREATE FUNCTION, CREATE TYPE, and CREATE TRIGGER

SQL statement	-e option	
	Not specified	Specified
CREATE PROCEDURE	END_PROC;	END_PROC;
CREATE FUNCTION	END_PROC;	END_PROC; or END_FUNC;
CREATE TYPE	END_PROC;	END_PROC; or END_TYPE;
CREATE TRIGGER	END_PROC;	END_PROC; or END_TRIG;

The system assumes that CREATE PROCEDURE, CREATE FUNCTION, CREATE TYPE, or CREATE TRIGGER is continued on to the next line, even if it ends with a semicolon.

Example

```
CREATE PROCEDURE P1                    (Continued)
BEGIN                                  (Continued)
  INSERT INTO VALUES(10);              (Continued)
```

```
  UPDATE T1 SET C1 = 0 WHERE C1 < 0;        (Continued)
END                                         (Continued)
END_PROC;                                   (End of SQL statement)
```

If a line ends with `END_PROC;` that is, enclosed in double quotation marks (`"`), the system treats it as the end of the SQL statement.

(3) EOF detected in an SQL statement

If the system detects EOF in an SQL statement (before detecting a semicolon), the system treats it as the end of the SQL statement and executes the SQL statement read so far.

Example

```
CREATE TABLE T1(C1 INT,                     (Continued)
                C2 CHAR(10)                 (Continued)
EOF                                         (End of command)
```

4.4 Rules and notes

(1) Rules

1. The database definition utility can be executed only while HiRDB is active.
2. The database definition utility can be executed from any server machine.
3. When a definition SQL is executed, commit processing is executed for each SQL statement.
4. A maximum of 2 MB of SQL statements can be executed.
5. Multiple definition SQL statements cannot be specified on the same line.
6. When `CREATE INDEX` is executed by the database definition utility, the index is created in the batch mode at the time of execution if there is row data in the table for which the index is being created. If `PDDBLOG=NO` is specified in the client environment definition at this time, the index is created in the batch mode without collecting database updating log information. Therefore, the HiRDB administrator must perform operations in the no-log mode. For details about operations in the no-log mode, see the *HiRDB Version 8 System Operation Guide*
7. `ALTER TABLE` and `DROP TABLE` must not be executed for an extracted database subject to data linkage. If either of these commands is executed in such a case, database agreement is not guaranteed. However, the definition on the HiRDB side can be changed if the tables or columns in a database subject to data linkage are not being updated and all update information has already been transferred to the target side. If the HiRDB Datareplicator is then reactivated after using the `hdeprep` command to re-analyze the extraction definition of tables or columns to be extracted by the HiRDB Datareplicator side, database agreement is guaranteed. For details about the procedures, see the *HiRDB Datareplicator Version 8 Description, User's Guide and Operator's Guide*.
8. When executing `pddef`, make sure that the character codes are the same as for the HiRDB server in which the definitions are registered. You must also use the specified character codes for entering data.

(2) Note

The results of the database definition utility can be checked by the return code set by the utility or by referencing a data dictionary table. The return codes are as follows:

- When the `-R` option is omitted

Return code	Meaning	Action	HiRDB operation
0	Normal termination (including when an SQL statement resulted in an error)	Not applicable	Not applicable
8	Memory shortage	Re-execute after another process has terminated, or increase memory.	Cancels processing and terminates pddef
	Option specification error	Specify the correct option.	

- When the -R option is specified

Return code	Meaning	Action	HiRDB operation
0	Normal termination	Not applicable	Not applicable
4	HiRDB connection error	Check and, if necessary, revise the specified environment variables.	Cancels processing and terminates pddef
	SQL error	Check and, if necessary, revise the executed SQL statement.	Resumes processing
8	Memory shortage	Re-execute after another process has terminated, or increase memory.	Cancels processing and terminates pddef
	Option specification error	Specify the correct option.	

Note

If return code 4 is followed by return code 8, the system cancels processing and terminates with final return code 8.

(3) Information that is output in the event of an SQL error

If any of the SQL statements listed below results in an error, the GET DIAGNOSTICS statement can be executed to display error information (if the SQL statement contains a comment, ERROR_POSITION will display the values only without the comment):

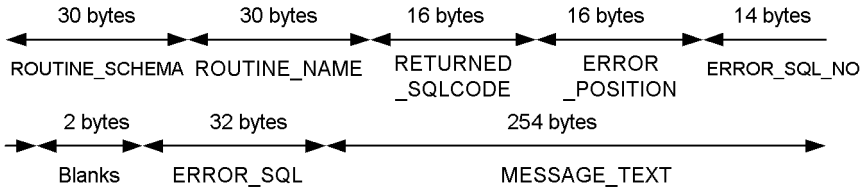
- CREATE PROCEDURE
- CREATE FUNCTION
- CREATE TYPE
- ALTER PROCEDURE
- ALTER ROUTINE

- CREATE TRIGGER

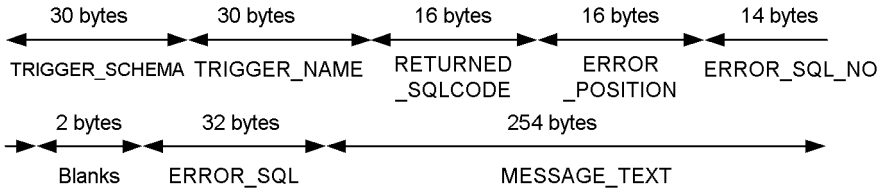
Figure 4-2 shows the format of error information.

Figure 4-2: Format of error information

- For CREATE PROCEDURE, CREATE FUNCTION, CREATE TYPE, ALTER PROCEDURE, and ALTER ROUTINE



- For CREATE TRIGGER



(4) Handling of comments

The system regards any data beginning with /* in an SQL statement as a comment. All characters following /* (including semicolons) are handled as part of the comment. The system regards the first */ that follows /* as the end of the comment.

The following shows examples:

```
Example 1:
CREATE TABLE T1 (C1 INT) ; /* CU ADDRESS TABLE */     ...Start and end
of comment

Example 2:
CREATE TABLE T1 /* CU ADDRESS TABLE */ (C1 INT) ;     ...Start and end
of comment

Example 3:
CREATE TABLE T1 /* CU ADDRESS     ...Start and continuation of comment
TABLE */ (C1 INT) ;    ..............End of comment

Example 4:
CREATE TABLE T1 /* CU ADDRESS TABLE * (C1 INT) ;     ...Start and
continuation of comment
```

```
Example 5:
CREATE TABLE T1 /* CU /* ADDRESS */ TABLE */ (C1 INT) ;
  ...Start and end of comment
```

Data from `/*` to `*/` that is enclosed in double quotation marks (`"`) or single quotation marks (`'`) is not handled as a comment. For details about specifying comments, see the description of the SQL specification format in the manual *HiRDB Version 8 SQL Reference*. Note that this utility's handling of comments is different from the interactive SQL execution utility (HiRDB SQL Executer).

(5) Using a file that contains a BOM

If you selected `utf-8` as the character encoding in the `pdntenv` command, you can use a file with a BOM as the input file for `pddef`. Note that even when a file with a BOM is used as the input file for `pddef`, the BOM is skipped.

4.5 Examples

Example 1 shows an example of the use of the database definition utility.

Example 1

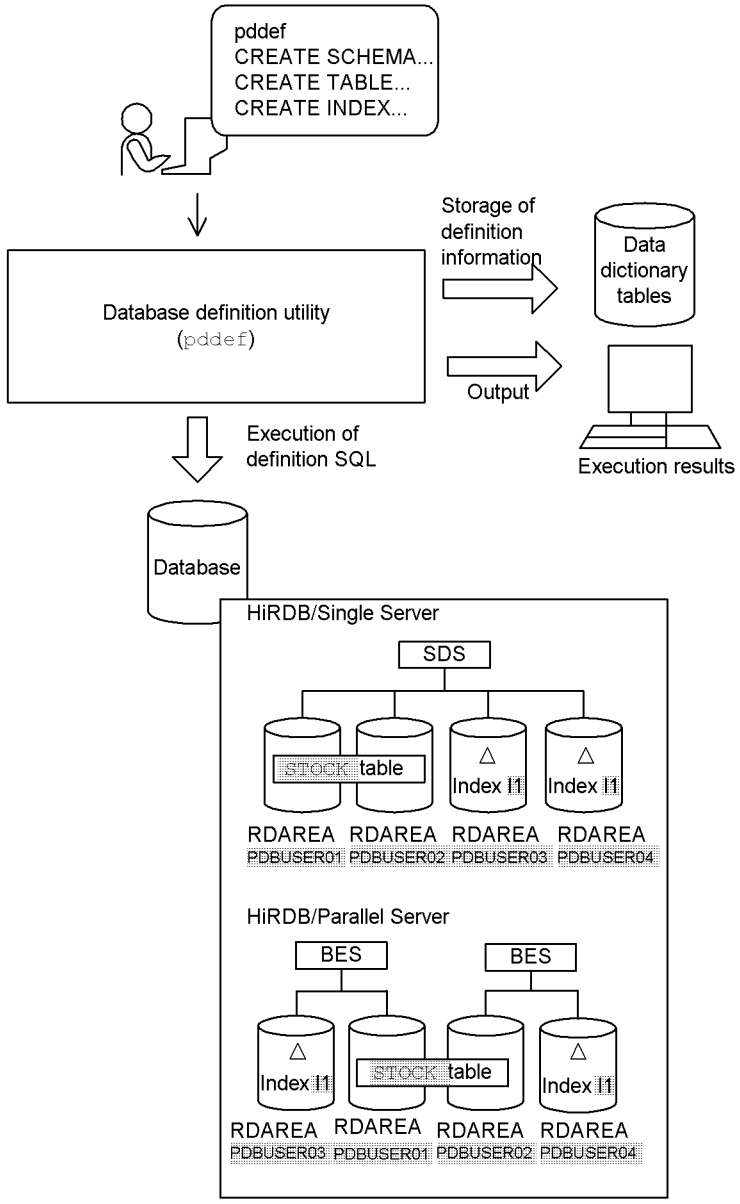
Define the following table and index:

- Table: `STOCK`
- Index: `I1`

The following RDAREAs have already been created:

- `PDBUSER01`
- `PDBUSER02`
- `PDBUSER03`
- `PDBUSER04`

Overview



: Values specified with the database definition utility's definition SQL

Command execution

- Entry of definition SQL from standard input

```
pddef
 CREATE SCHEMA;                                                    1
 CREATE TABLE STOCK(PCODE CHAR(4),PNAME NCHAR(8),                  2
                    COLOR NCHAR(1),PRICE INTEGER
                    SQUANTITY INTEGER)
             IN((PDBUSER01) PCODE<'302S',
                (PDBUSER02) PCODE>='302S');
 CREATE INDEX I1 ON STOCK(PCODE) IN ((PDBUSER03),(PDBUSER04));     3
<End by pressing CTRL + Z, then ENTER>
```

Explanation

1. Definition of schema.
2. Definition of table (`STOCK`).
3. Definition of index (`I1`).

- Entry of definition SQL from input file

```
pddef < c:\hirdb\pddef\crfile
```

Explanation

Specification of input file `c:\hirdb\pddef\crfile`.

Contents of input file c:\hirdb\pddef\crfile

```
CREATE SCHEMA;                                                     1
CREATE TABLE STOCK(PCODE CHAR(4),PNAME NCHAR(8),                   2
                   COLOR NCHAR(1),PRICE INTEGER
                   SQUANTITY INTEGER)
               IN((PDBUSER01) PCODE<'302S',
                  (PDBUSER02) PCODE>='302S');
CREATE INDEX I1 ON STOCK(PCODE) IN ((PDBUSER03),(PDBUSER04));      3
```

Explanation

1. Definition of schema.
2. Definition of table (`STOCK`).
3. Definition of index (`I1`).

Chapter 5. Database Load Utility (pdload)

This chapter describes the database load utility (`pdload`) that stores user-provided data in a table.

This chapter contains the following sections:

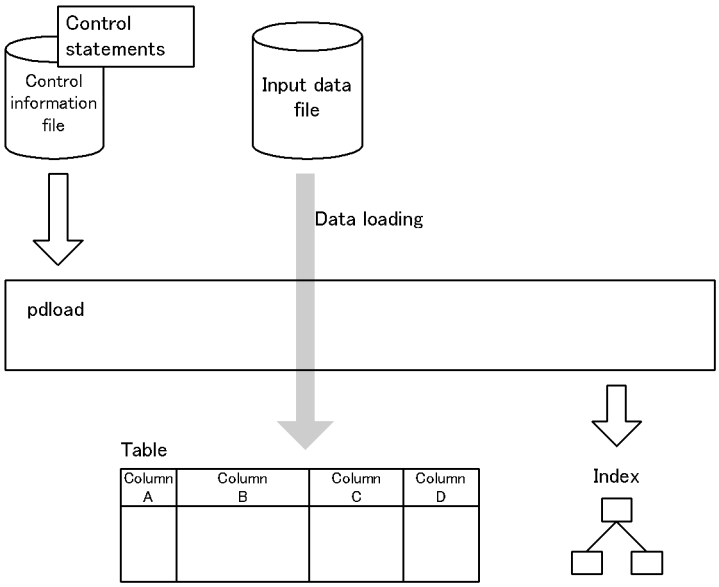
5.1 Function

The database load utility (pdload) loads user-provided data to a table.

5.1.1 What is data loading?

The database load utility reads user-provided data and stores it in a table. This is called data loading. Figure 5-1 shows an overview of data loading.

Figure 5-1: Overview of data loading



(1) Input data file

An input data file contains user-provided data. The database load utility supports four different formats of files. Basically, one of the following two formats is used:

DAT format

In the DAT format, column data is described in characters. This format is generally referred to as the CSV format.

- pdload converts column data to the internal HiRDB format before storing it; therefore, the DAT format is suitable for creating a table from data imported from a non-HiRDB system.

Example of data in the DAT format:

```
Jones,36,1958-10-15,Chicago
```

```
.
.
```

Additionally, there is the extended DAT format that enables you to use extended functions. Although the format is basically the same as the DAT format, the extended DAT format supports the extended functions, such as changing the enclosing characters.

Binary format

In the binary format, column data is stored in the internal HiRDB format.

- The binary format is excellent in terms of performance, because it matches the internal HiRDB format, thereby requiring no format conversion. This format is suitable when high performance is required, such as for storing a large amount of data.

Example of data in the binary format:

```
928691ba8140814081402400000003f0080000800796f6b6f68616d6
1
<------------------><------><------><------------------
>
        Jones          36       1          Chicago
      :
```

Note

The upper row of the input data indicates the data and the lower row indicates data contents.

Additionally, you can use a format in which fixed-size data is specified on each line, or a format that is output by `pdrorg`, shown as follows:

Fixed-size data format

In the fixed-size data format, all lines have the same length and all the data items in a column begin at the same location (the same offset from the beginning of the line). Input data can be specified either in the DAT format or in the binary format.

- The fixed-size data format is suitable for a table created from text data that is not delimited by separator characters, or created from binary-format data with a data storage sequence that needs to be changed.

pdrorg-generated binary format

This is an unload data file that is output by `pdrorg`, specifying the `-W` option.

- The `pdrorg`-generated binary format is used to migrate data from another

HiRDB system.

(2) Control information file

A control information file contains the `pdload` control statements. These control statements specify an input data file, index information, LOB column information, and other information.

(3) Table

You need to define a table before you can perform data loading.

(4) Data loading method by table attribute

(a) Index defined for the table

You can create an index at the same time as data loading. Or, you can output only the index information during data loading and use `pdrorg` to create the index later.

(b) Row-partitioned table

You can execute data loading in units of tables or RDAREAs.

5.1.2 Loading data to a table with LOB columns

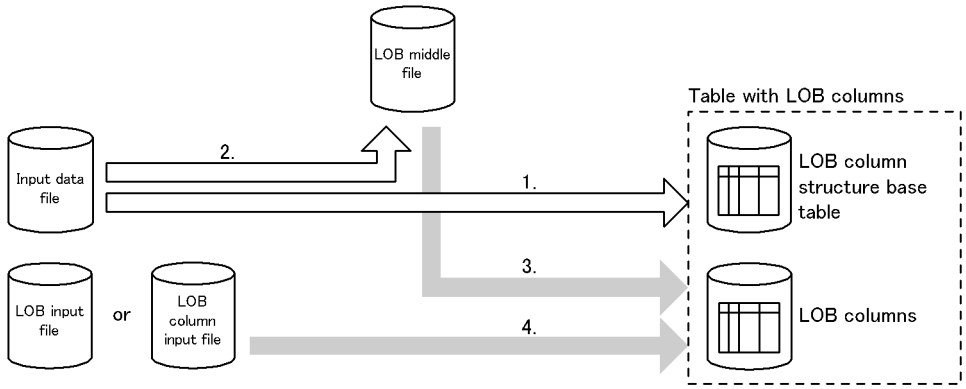
(1) Data loading method

If a table contains LOB columns, you can load data to the LOB column structure base table and LOB columns at the same time or separately. A LOB column structure base table is the part of the table without the LOB columns. A LOB column contains BLOB data.

When a table consists of multiple LOB columns, loading data only to the LOB columns later is achieved by a facility for concurrently executing data loading to multiple LOB columns. A database is not stable if data loading has been completed in a LOB column structure base table, but has not been completed in the LOB columns. Make sure that such tables are not updated in any manner, including execution of the `PURGE TABLE` statement. If you are executing data loading separately, we recommend that you place the applicable RDAREA in command shutdown status to suppress updating of the table by other users.

Figure 5-2 shows the method for loading data to a table with LOB columns.

Figure 5-2: Method for loading data to a table with LOB columns



Explanation

- Loading data to the LOB column structure base table and LOB columns at the same time

  `pdload` executes 1 through 4 during the first data loading.

- Loading data to the LOB column structure base table and LOB columns separately

  Execute steps (a) and (b) in this order:

(a) Loading data to the LOB column structure base table (1 and 2)

Execute data loading to the LOB column structure base table. In this case, load data to the LOB column structure base table and output information about the LOB data from the input data to the LOB middle data.

(b) Loading data to the LOB columns (3 and 4)

Execute data loading on the LOB columns. In this case, also load the information about LOB data from the LOB middle file, which was obtained in (a) previously.

A LOB input file is provided for each column and each row. This means that there must be as many LOB input files as there are LOB columns × rows.

A LOB column input file is provided for each column. This means that there are as many LOB column input files as there are LOB columns.

(2) Input data

To load data to a table with LOB columns, you need to provide the input data in one of the following forms:

Input data file	LOB input file
Containing the data other than the LOB columns, and the LOB input file names	As many LOB input files as there are data items to be stored
Containing the data other than the LOB columns, and the LOB data	Not needed

5.1.3 Loading data to a table with abstract data type columns (LOB attribute) provided by a plug-in

(1) Data loading method

If a table contains abstract data type columns (LOB attribute) provided by a plug-in, load data to the LOB column structure base table and LOB columns at the same time.

(2) Input HiRDB Datareplicator linkage facility

To load data to a table with abstract data type columns (LOB attribute) provided by a plug-in, you need to provide the input data in one of the following forms:

Input data file	LOB parameter file
Containing the data other than the LOB-attribute abstract data type columns, and the LOB parameter file names	As many LOB parameter files as there are data items to be stored
Containing the data other than the LOB-attribute abstract data type columns, and the LOB parameters	Not needed

A LOB parameter is the LOB data used as an input parameter for the constructor function that generates the values to be stored in abstract data type columns.

5.1.4 Data loading with the synchronization point specification

When loading data, usually a transaction is settled after all data has been stored. If the utility terminates abnormally during the processing, the transaction rolls back to the start point, in which case data loading must be re-executed from the beginning.

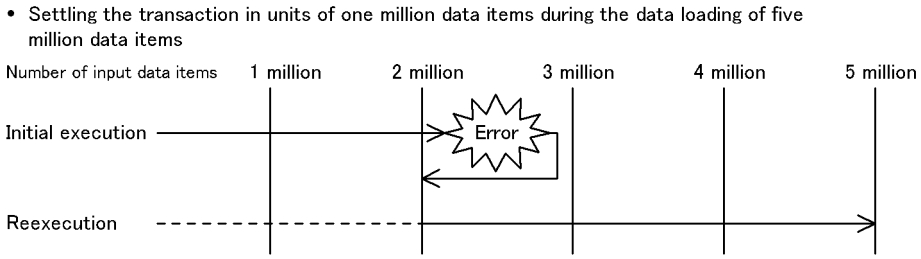
The data loading method with the synchronization point specification enables a transaction to be settled each time a specified number of data items has been stored. This method is suitable for loading a large amount of data because it reduces the rollback time, as well as the data loading time, in the event of an abnormal termination of the utility.

Data loading with synchronization point specification is not applicable to index creation (batch index creation mode) or LOB column data loading. In such cases, the transaction is settled after all processing is completed.

To execute data loading with the synchronization point specification, specify the `option` statement (`job` operand). Figure 5-3 shows an overview of data loading with

the synchronization point specification.

Figure 5-3: Overview of data loading with the synchronization point specification



Explanation

At the initial execution, the transaction rolls back after the occurrence of an error. This rollback is to the point of two million data items because the transaction has settled at that point.

The re-execution skips to the point of two million data items and starts storing the remaining data.

5.1.5 RDAREAs containing a table subject to data loading

To protect the UAP from unnecessarily being placed in wait status while accessing a table subject to data loading, you should use the `pdhold` command to shut down the RDAREAs that contain the table (and the RDAREAs containing indexes if the indexes are also defined for the table).

Parallel execution of data loading into the RDAREAs of a row-partitioned table reduces the length of time the table is locked (time required for data loading is reduced). If an input data file is not available for each RDAREA (only input data files for the table are available), specify the `src_work` statement. This statement creates an input data file for each RDAREA so that it is possible to perform parallel execution of data loading into the RDAREAs for the row-partitioned table. The files created by this statement are called *divided-input data files*. For details about the specifiable options and control statements, see *5.11(3) Whether or not options and control statements can be specified when pdload functions are used*.

5.1.6 Log acquisition mode during execution of pdload

When you execute `pdload`, we recommend that you use pre-update log acquisition (`-l p`) as the log acquisition mode.

In the pre-update log acquisition mode, the utility does not acquire an database update log after update processing, thereby reducing the processing time. Compared to when the log acquisition mode is used, you can shorten the utility's execution time.

5.1.7 Executor

Basically, a table creator executes data loading.

To enable another user execute data loading, you must grant the privileges shown below as appropriate to the data loading mode. For an audit trail table, only its auditor, who is the owner of the table, can execute data loading.

Addition mode (-d option omitted)

A user with the `INSERT` privilege can execute the data loading.

Creation mode (-d option specified)

A user with the `INSERT` and `DELETE` privileges can execute the data loading.

When the Directory Server linkage facility is used, the system permits data loading when the `pdload` executor's privileges and the executor's role privileges are consistent with the applicable condition above.

5.2 Typical examples of data loading

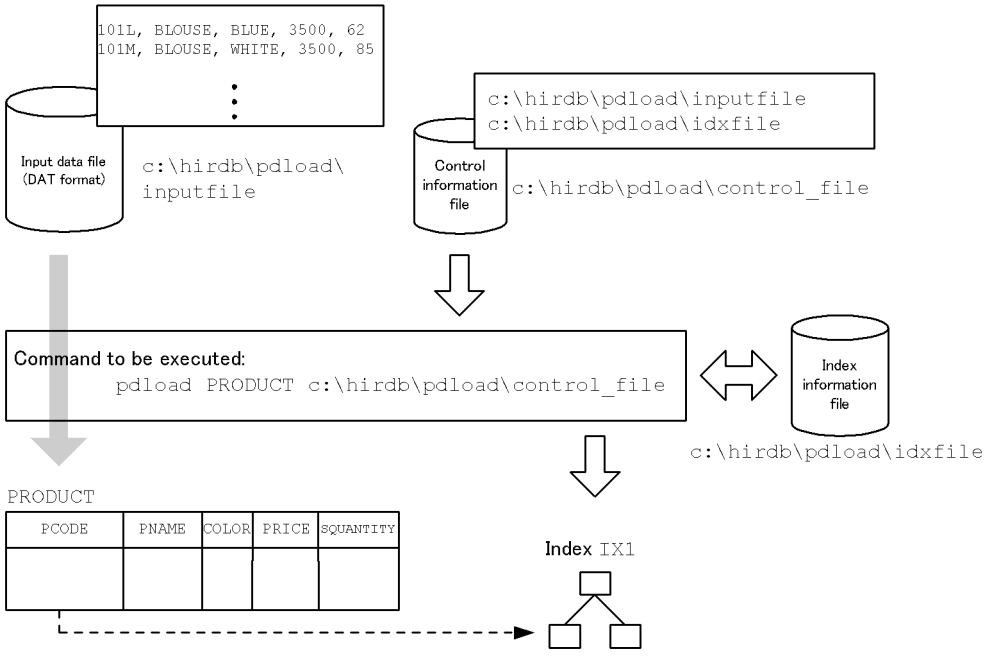
This section presents three typical examples of data loading:

- Loading data to a table with indexes defined
- Loading data to a table with LOB columns
- Loading data to a table with abstract data type columns provided by a plug-in

5.2.1 Loading data to a table with indexes defined

Figure 5-4 shows data loading into a table for which indexes are defined.

Figure 5-4: Loading data to a table with indexes defined



(1) Command to be executed

The following explains the options specified with `pdload`:

`PRODUCT`

This is the identifier of the table subject to data loading.

`c:\hirdb\pdload\control_file`:

This is the name of the control information file.

For the omitted options, the system assumes the following values:

`-d` omitted: Addition mode

`{-a|-b}` omitted: Input data file in the DAT format

`-i` omitted: Batch index creation mode

`-l` omitted: Pre-update log acquisition mode

(2) Control information file

(a) source statement

The `source` statement specifies the input data file named `c:\hirdb\pdload\inputfile`.

(b) index statement

The `index` statement specifies the index information file named `c:\hirdb\pdload\idxfile`.

(3) Input data file

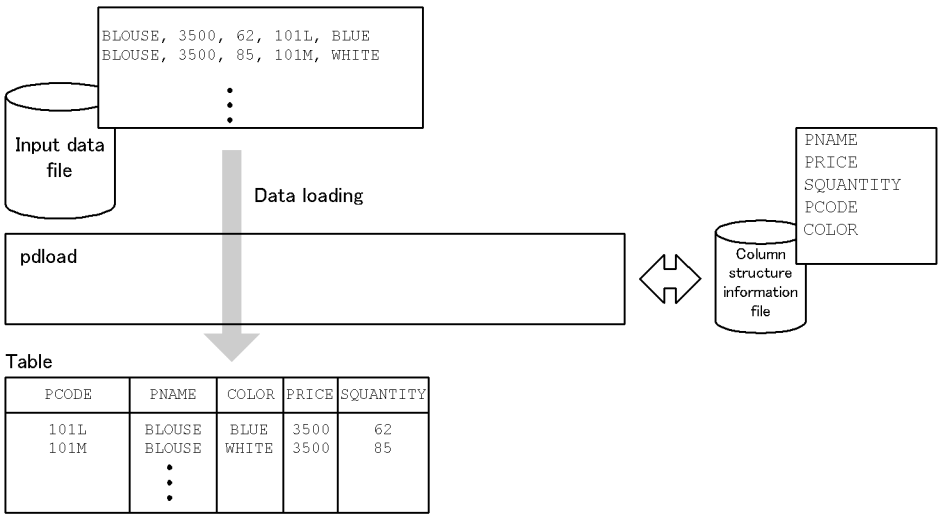
This example specifies the input data file in the DAT format.

(4) Inconsistent order of columns between input data and table

If the order of columns does not match between the input data file and the table, you can specify a column structure information file to define the correspondence of columns between input data and table.

You can use the column structure information files only when the input data file is in DAT, fixed-size data, or `pdrorg`-generated binary format (non-FIX).

The following shows an example:



Explanation

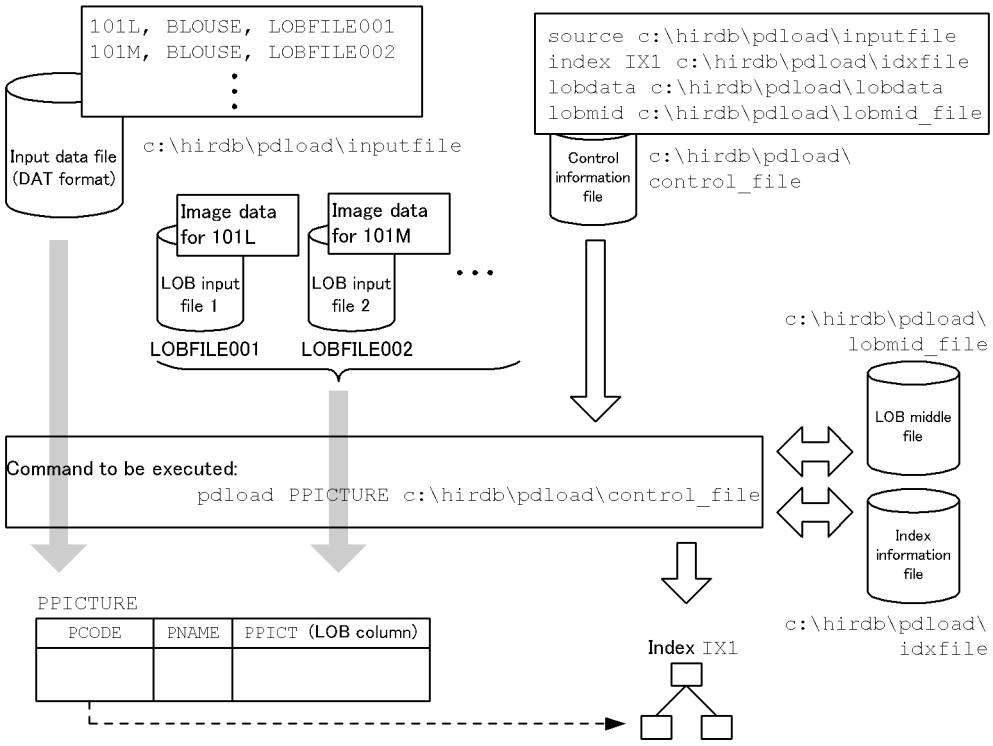
The order of input data items does not match the order of table columns.

In this case, use a column structure information file to specify column names in the order of input data items.

5.2.2 Loading data to a table with LOB columns

Figure 5-5 shows data loading into a table with LOB columns.

Figure 5-5: Loading data to a table with LOB columns



(1) Command to be executed

The following explains the options specified with `pdload`:

`PPICTURE`

This is the identifier of the table subject to data loading.

`c:\hirdb\pdload\control_file`:

This is the name of the control information file.

For the omitted options, the system assumes the following values:

`-d` omitted: Addition mode

`{-a|-b}` omitted: Input data file in the DAT format

`-i` omitted: Batch index creation mode

`-l` omitted: Pre-update log acquisition mode

`-k` omitted: `f` (Provide as many LOB input files as there are LOB columns × rows and specify the names of these LOB input files in the input data file.)

(2) Control information file

(a) source statement

The `source` statement specifies the input data file named `c:\hirdb\pdload\inputfile`.

(b) index statement

The `index` statement specifies the index information file named `c:\hirdb\pdload\idxfile`.

(c) lobdata statement

The `lobdata` statement specifies the LOB input file's directory named `c:\hirdb\pdload\lobdata`. This directory name is specified in such a manner that it results in the absolute path name when combined with the LOB input file name specified in the input data file.

(d) lobmid statement

The `lobmid` statement specifies the LOB middle file named `c:\hirdb\pdload\lobmid_file`.

(3) Input data file

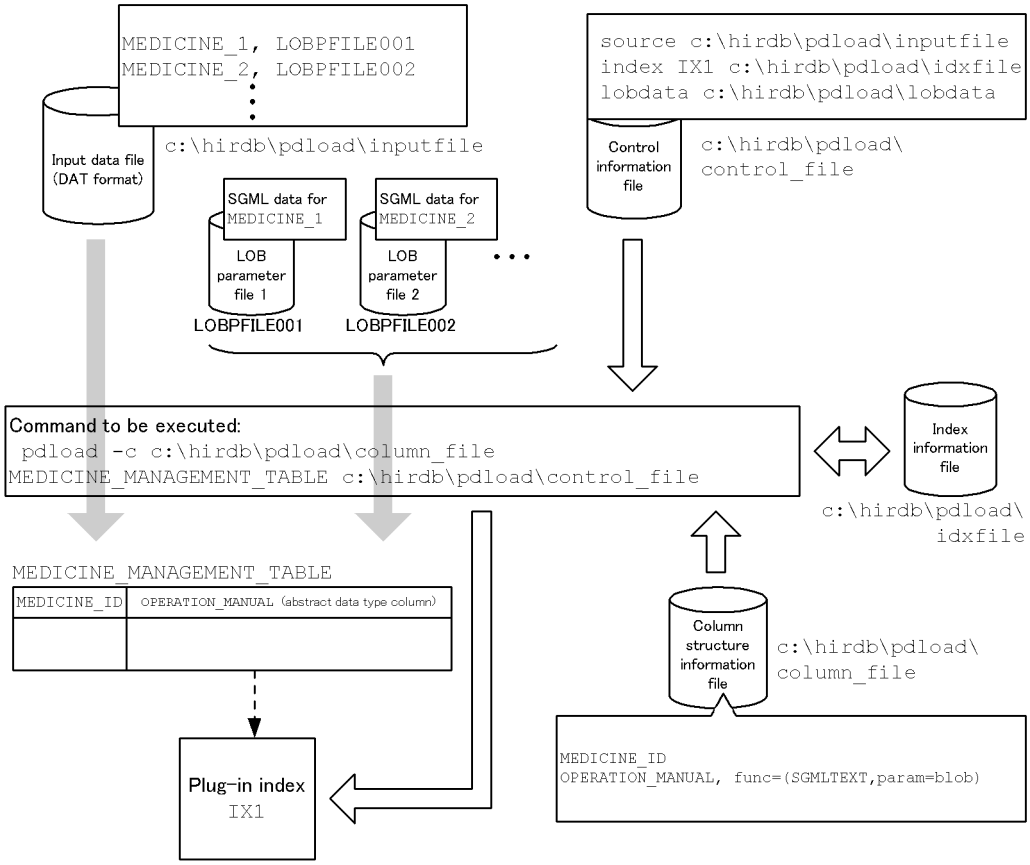
This example specifies the input data file in the DAT format.

Additionally, the example provides separate LOB input files and specifies their names in the input data file.

5.2.3 Loading data to a table with abstract data type columns provided by a plug-in

Figure 5-6 shows data loading into a table with an abstract data type provided by a plug-in. This example uses the HiRDB Text Search Plug-in.

Figure 5-6: Loading data to a table with abstract data type columns provided by a plug-in



(1) Command to be executed

The following explains the options specified with `pdload`:

`-c c:\hirdb\pdload\column_file`:

This is the name of the column structure information file.

`MEDICINE_MANAGEMENT_TABLE`

This is the identifier of the table subject to data loading.

`c:\hirdb\pdload\control_file`:

This is the name of the control information file.

For the omitted options, the system assumes the following values:

`-d` omitted: Addition mode

`{-a|-b}` omitted: Input data file in the DAT format

`-i` omitted: Batch index creation mode

`-l` omitted: Pre-update log acquisition mode

`-k` omitted: `f` (Provide the LOB parameter files and specify their file names in the input data file.)

(2) Control information file

(a) source statement

The `source` statement specifies the input data file named `c:\hirdb\pdload\inputfile`.

(b) index statement

The `index` statement specifies the index information file named `c:\hirdb\pdload\idxfile`.

(c) lobdata statement

The `lobdata` statement specifies the LOB parameter file's directory named `c:\hirdb\pdload\lobdata`. This directory name is specified in such a manner that it results in the absolute path name when combined with the LOB parameter file name specified in the input data file.

(3) Input data file

This example specifies the input data file in the DAT format.

Additionally, the example provides separate LOB parameter files and specifies their names in the input data file.

(4) Column structure information file

This example specifies the column structure information file named `c:\hirdb\pdload\column_file`.

`MEDICINE_ID`

This is a column name for the first data item in the input file.

`OPERATION_MANUAL`

This is a column name for the second data item in the input file.

`func=(SGMLTEXT,param=blob)`

This specifies information about the constructor function used to generate the values that are stored in the abstract data type columns.

`SGMLTEXT`: Name of the constructor function

`blob`: Type of argument that is passed to the constructor function

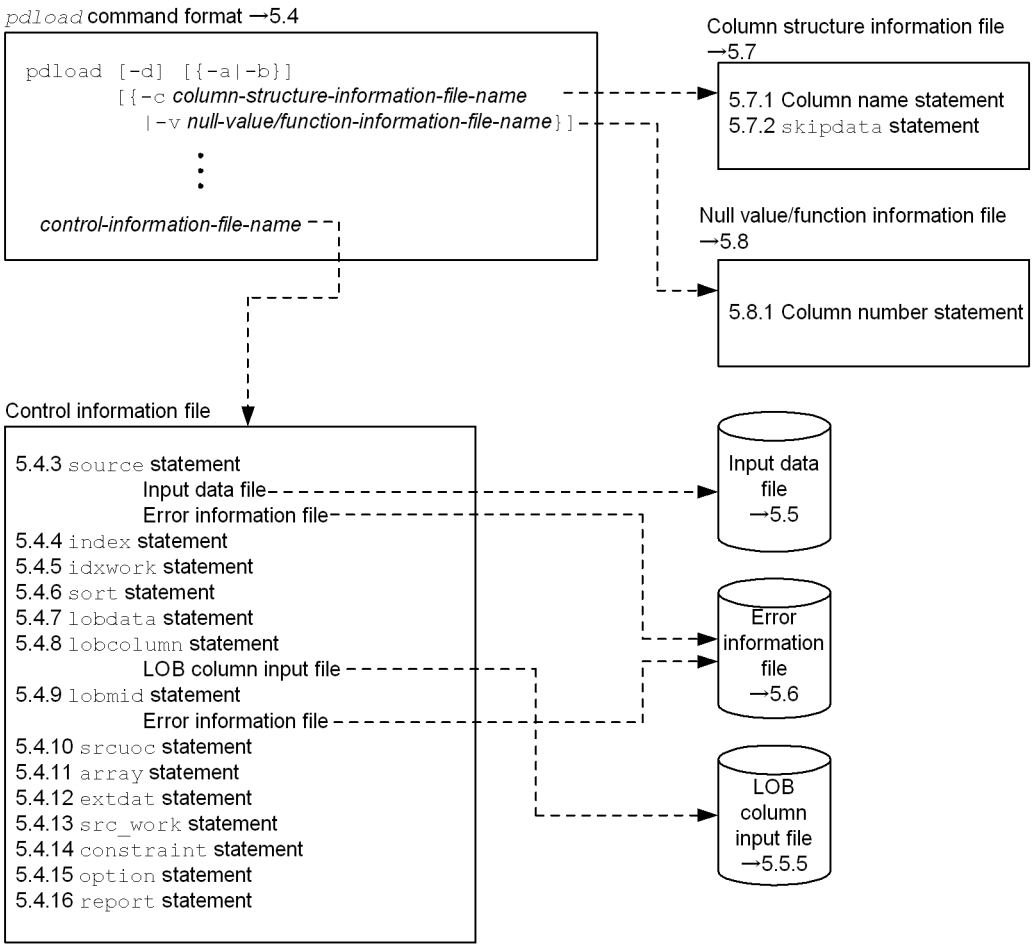
5.3 List of references by purpose

The options, control statements, and files to be specified depend on the data loading mode (type of table and type of data loading to be conducted).

sections *5.3.1 Required information* through *5.3.4 Information to be specified depending on the data loading method* present the tables of items and related options, control statements, and files, along with the corresponding sections. For options, see (1) through (21) in section *5.4.2 Options*.

Figure 5-7 shows the relationship among options, control statements, and files.

Figure 5-7: Relationship among options, control statements, and files



Note: For details about the command format, control statements, and files for pdload, see the section or subsection following → or the subsection indicated before each control statement.

5.3.1 Required information

To execute pdload, you need to specify the following information. For details, see the sections indicated.

Item		Reference target			
		Option		Control statement or file	
Name of the table subject to data loading		[*authorization-identifier*.]*table-identifier*	(20)	`source` statement	5.4.3
File containing control statements		*control-information-filename*	(21)		
Format of input data file	DAT format	Omitted	(2)	Input data file	5.5.1
	Binary format	`-b`	(2)	Input data file	5.5.2
	`pdrorg`-generated binary format	`-b`	(2)	Input data file	5.5.2
		`-W`	(5)		
	Fixed-length data format	`-a`	(2)	Input data file	5.5.3
		—	—	`column name` statement	5.7.1

—: Not applicable

5.3.2 Information to be specified depending on the attribute of the table subject to data loading

For details about an applicable item, see the section indicated.

Item		Reference target			
		Option		Control statement or file	
Index is defined		`-i`	(3)	`index` statement	5.4.4
		`-x`	(11)	`idxwork` statement	5.4.5
		`-o`	(17)	`sort` statement	5.4.6
There are LOB columns	Using LOB input file	`-k`	(6)	`lobdata` statement	5.4.7
				`lobmid` statement	5.4.9
	Using LOB column input file	`-k`	(6)	`lobcolumn` statement	5.4.8
				`lobmid` statement	5.4.9
				LOB input file by the column	5.5.5
There are abstract data type columns (LOB attribute) provided by a plug-in		`-k`	(6)	`lobdata` statement	5.4.7

Item	Reference target			
	Option		Control statement or file	
There are repetition columns	—	—	`array` statement	5.4.11
			Input data file	5.5.4
There are variable-length character strings with columns that have no corresponding input data, and data with a length of 0 is to be stored*	`-z`	(15)	—	—
There are constraint definitions	—	—	`constraint` statement	5.4.14
Partition storage conditions were changed	—	—	`option` statement	5.4.15

—: Not applicable

* If you specified `WITH DEFAULT` during table definition, a space equivalent to one byte or one character is stored. Otherwise, the null value is stored.

5.3.3 Information to be specified depending on the type of input data file

(1) DAT format

For details about an applicable item, see the section indicated.

Item	Reference target			
	Option		Control statement	
There is no correspondence between input data and table columns	`-c`	(7)	`column name` statement	5.7.1
Some of the input data is not to be stored in the table	`-c`	(7)	`skipdata` statement	5.7.2
Information about a constructor function is specified to generate values for the abstract data type columns	`-c`	(7)	`column name` statement	5.7.1
Separator used in the input data is not a comma (,)	`-s`	(12)	—	—
Code for a space is to be converted in the national character string or mixed character string type input data	—	—	`option` statement	5.4.15

Item	Reference target			
	Option		Control statement	
If the input data to be stored in character string or BINARY type column is longer than the defined column length, an input data error is to be detected	—	—	option statement	5.4.15
The input data file in DAT format satisfies one of the following conditions: • Data to be stored contains linefeed codes or NULL characters. • Character data is enclosed by a character that is not the double quotation mark (").	—	—	extdat statement	5.4.12
Data containing a decimal point is to be stored in INTEGER and SMALLINT columns	—	—	extdat statement	5.4.12

—: Not applicable

(2) Binary format

For details about an applicable item, see the section indicated.

Item	Reference target			
	Option		Control statement	
There is no correspondence between input data and table columns*	-c	(7)	column name statement	5.7.1
Some of the input data is not to be stored in the table*	-c	(7)	skipdata statement	5.7.2
Input data is converted to a null value, or information about a constructor function is specified to generate values for the abstract data type columns	-v	(8)	column number statement	5.8.1
Code for a space is to be converted in the national character string or mixed character string type input data	—	—	option statement	5.4.15

—: Not applicable

* Applicable if the input data file was created by pdrorg and data loading is to be on a non-FIX table.

(3) Fixed-size data format

For details about an applicable item, see the section indicated.

Item	Reference target			
	Option		Control statement	
There is no correspondence between input data and table columns	`-c`	(7)	`column name` statement	5.7.1
Some of the input data is not to be stored in the table	`-c`	(7)	`skipdata` statement	5.7.2
Input data is converted to a null value, or information about a constructor function is specified to generate values for the abstract data type columns	`-c`	(7)	`column name` statement	5.7.1
The double quotation mark (`"`) is to be stored as input data	—	—	`column name` statement	5.7.1
Code for a space is to be converted in the national character string or mixed character string type input data	—	—	`option` statement	5.4.15

—: Not applicable

5.3.4 Information to be specified depending on the data loading method

For details about an applicable item, see the section indicated.

Item	Reference target			
	Option		Control statement or file	
Database update log is acquired or not acquired	`-l`	(4)	—	—
Synchronization points are specified to load a large amount of data	—	—	`option` statement	5.4.15
The `pdload` execution time is to be monitored	—	—	`option` statement	5.4.15
Default value for the `DEFAULT` clause is to be stored in a table for which the `DEFAULT` clause is specified	—	—	`option` statement	5.4.15
UOC is used	—	—	`srcuoc` statement	5.4.10
Processing is to be cancelled if invalid input data is detected[1]	`-e`	(13)	Error information file	5.6
	`-r`	(14)	`option` statement	5.4.15
When invalid input data is detected, data loading is to be ignored	—	—	`option` statement	5.4.15

Item		Reference target			
		Option		Control statement or file	
Percentage of free table space is to be changed during data loading[2]	Storing data to unused area	`-y`	(16)	—	—
	Storing data in unused area based on the specified percentage of free space	—	—	`option` statement	5.4.15
Data loading status message is to be output in units of lines other than 100,000		`-m`	(18)	—	—
Response time for server-to-server communication is to be monitored		`-x`	(19)	—	—
Batch output data loading is to be conducted using a local buffer, not a global buffer[3]		`-n`	(9)	—	—
Authorization identifier of the user who executes data loading is to be changed to a value other than the one specified in the `PDUSER` environment variable[4]		`-u`	(10)	—	—
Tuning information is to be collected during the execution of `pdload`		—	—	`report` statement	5.4.16
The input data files for a table are to be divided into input data files that enable parallel data loading into each RDAREA		—	—	`src_work` statement	5.4.13

—: Not applicable

[1] Suppose that data loading is executed on a table with repetition columns using an input data file in the DAT format. The input data is created in VV format, but it is treated as being in FF format if the `elmtype` operand is omitted, in which case all input data results in an error, thereby creating an unneeded error information file and error data file. In this case, processing should be cancelled when an error is detected to avoid creating unneeded files or executing unneeded operations.

[2] During additional data loading, data is usually stored in unused pages according to the percentage of free table space. If the system runs out of unused pages, an error results, and rollback occurs. If this occurs, you must reorganize the table with the database reorganization utility or add or extend RDAREAs with the database structure modification utility before re-executing the data loading. In this case, if you change the percentage of free table space during data loading, you may be able to complete the processing without having to take such actions.

[3] If you specify the number of batch output pages, the system outputs the specified number of pages in batch mode, thereby reducing the number of input/output operations.

[4] If you specify nothing, the value of the PDUSER environment variable takes effect. If you have not specified the PDUSER environment variable, the system assumes the user name of the login window.

5.4 Command format

5.4.1 Format

This section describes the command format of the database load utility. The numbers in the table below correspond to the numbers in the explanation of the options.

Options in bold are important or mandatory.

No.	Format
1	pdload **[-d]**
2	**[{-a|-b}]**
3	**[-i *index-creation-method*]**
4	**[-l *log-acquisition-method*]**
5	[-W]
6	[-k *LOB-creation-type*]
7	[{-c *column-structure-information-filename*
8	|-v *null-value/function-information-filename*}]
9	[-n [*batch-output-local-buffer-sectors-count*],[div], [*local-buffer-sectors-count-for-random-access*]]
10	[-u *authorization-identifier*]
11	[-x]
12	[-s *separator-character*]
13	[-e]
14	[-r *input-begin-line*]
15	[-z]
16	[-y]
17	[-o]
18	[-m *progress-message-output-interval*]
19	[-X *response-monitoring-time-for-server-to-server-communication*]
20	**[*authorization-identifier*,] *table-identifier***
21	***control-information-filename***

Note

Be sure to specify [*authorization-identifier*,]*table-identifier* and *control-information-filename* as the last options.

5.4.2 Options

(1) -d

～ <<addition mode>>

Specifies that the database load utility is to be executed in the creation mode. If you omit this option, the system assumes the addition mode.

In the creation mode, all existing data is deleted from the table that is to be loaded, and then the input data is stored in the table.

In the addition mode, the input data is added to the existing data in the table.

Rules

1. In the addition mode, the input data is stored in the pages following the last data in the table. If you use the addition mode to load data on a table with a cluster key defined, the additional data is not stored in the order of cluster key values.
2. In the addition mode, data is always stored in unused pages. If the system cannot allocate unused pages, an error results. Normally, the system will not use available space in used pages that contain data. However, if you specify the `-y` option, the system will use used pages that already contain data.
3. If you execute data loading in units of RDAREAs in the creation mode, the existing table data is deleted only in the specified RDAREA.
4. During data deletion, the system does not allow a `NOWAIT` search, even if `nowait=yes` is specified in the `option` statement.
5. Once data loading with the synchronization point specification terminates abnormally, the creation mode is ignored during the re-execution, if specified.
6. If you execute data loading on a LOB column that is stored in an RDAREA in frozen update status or on a table that contains an abstract data type column with the LOB attribute, specifying the creation mode results in an error because data cannot be deleted. Even when you execute data loading only on a LOB column structure base table, specifying the creation mode results in an error.
7. For a falsification prevented table, data loading in the creation mode is not permitted.

(2) {-a|-b}

～ <<DAT format>>

Specifies the format of the input data file.

Omitted: DAT format

`-a`: Fixed-size data format

`-b`: Binary format

For details about the input data files, see section *5.5 Input data file*.

(3) -i index-creation-method

～ <<`c`>>

Specifies the index creation method. There are four ways to create indexes:

`c`

This indicates the batch index creation mode. If you specify this mode, the system creates indexes in batch mode after creating the table.

When row data is being stored, the system outputs index creation information to an index information file without actually creating an index. The system creates indexes after storing the row data.

Criterion

When a large amount of data is loaded, this method can create indexes at high speed.

Notes

1. For data loading in the addition mode, the system outputs all index information, including both existing data and added data, to the index information file; therefore, all index entries are re-created.

   For additional data loading involving a plug-in index, the system outputs only the index information on the additional data. Therefore, the plug-in index is created only for the index entries that have been added.

2. If you execute data loading with the synchronization point specification in the batch index creation mode, the processing performance decreases for the following reasons:

   • For a B-Tree index, the system stores all data and then searches the data pages again to create an index information file.

   • For a plug-in index, the system starts the index creation process each time the transaction is settled, thereby alternately executing data loading and index creation.

`n`

This indicates the index information output mode. If you specify this mode, the system outputs only the index information to an index information file.

When row data is being stored, the system outputs index creation information to an index information file without actually creating an index.

Criterion

You can use the index information file obtained to achieve high-speed index creation by executing multiple batch index creation processes concurrently with the database reorganization utility. This method is especially effective for a table that is partitioned and stored in the multiple servers that constitute a HiRDB/Parallel Server.

Notes

1. If a table is partitioned into multiple RDAREAs in a server and a part of the partitioned table is created by data loading in units of RDAREAs, the system assumes the index information output mode for the non-partitioning key index that is not row-partitioned in the server.
2. You cannot specify the index information output mode for a table for which a plug-in index is defined.
3. When you specify the index information output mode, you need to use the database reorganization utility to create the index in batch mode after completing the data loading.

`s`

This indicates the index update mode. If you specify this mode, the system updates indexes each time a row of data is stored.

Criterion

Use the index update mode when loading a small amount of data to a table that already contains a large amount of data, or when a unique key index or primary key index has been defined and the key values of the data to be stored may be duplicated.

`x`

This indicates the index information output suppression mode. If you specify this mode, the system does not update indexes nor output index information to an index information file.

Criterion

Use this mode when executing data loading in multiple segments. For example, you should use the index information output suppression mode if

data loading needs to be executed *n* times because there are *n* volumes of MT.

Notes

1. If you are executing data loading using the `-ix` option, specify `-ice` (batch index creation mode) or `-in` (index information output mode) at the final execution. If you specify `-in`, you need to execute batch index creation (`-k imp`) later using the database reorganization utility.

   If you have loaded all data using the `-ix` option, you need to re-create indexes (`-k irk`) later using the database reorganization utility.

2. You cannot specify the index information output suppression mode for a table for which a plug-in index is defined.

3. If you execute data loading using the `-ix` option, the system does not update indexes, leaving indexes in uncreated status.

(a) Notes about the creation of an index

1. If you execute `pdload` in a mode other than the index update mode (`-i s`), the system creates an index in batch mode after storing all row data in a table. If a duplicate key error or an out-of-limit duplicate key error occurs during the index creation process, a mismatch may result between the table and the index.

   To avoid this problem, create a backup copy in either of the following cases, irrespective of which log-acquisition mode is used:

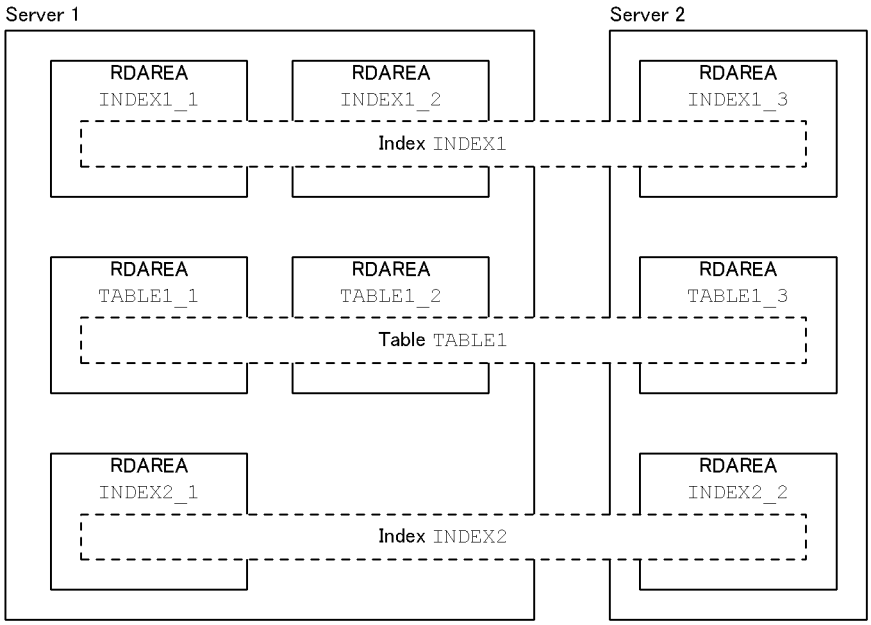
   - Tables with a unique key index or primary key index defined
   - Tables with an index defined on a repetition column

2. Figure 5-8 shows the index creation method for a row-partitioned index and a non-partitioned index. If a table is partitioned and stored in multiple RDAREAs in a server, there are a row-partitioned index that is stored in as many index storage RDAREAs as there are table storage RDAREAs and a non-partitioned index that is stored in a single index storage RDAREA.

   The row-partitioned index is partitioned and stored in the same manner as with the table that is partitioned in the server (RDAREAs `INDEX1_1` and `INDEX1_2` in the figure).

   The non-partitioned index is stored in a single RDAREA in the server, regardless of the number of table storage RDAREAs (RDAREAs `INDEX1_3`, `INDEX2_1`, and `INDEX2_2` in the figure).

Figure 5-8: Index creation method for partitioning key index and non-partitioning key index



Explanation

After completion of data loading, the system creates indexes according to the specified `-i` option. If data loading is executed on `TABLE1_1` in units of RDAREAs, row-partitioned index `INDEX1_1` is created (there is no effect on `INDEX1_2`). To create non-partitioned index `INDEX2_1`, however, information about both `TABLE1_1` and `TABLE1_2` is required. Therefore, in this case, the non-partitioned index is not created (only the index information file for `TABLE1_1` is created). To create the non-partitioned index, you need to execute data loading on `TABLE1_2` also to create the index information file for `TABLE1_2`, then execute `prong`'s batch index creation process using the index information files for `TABLE1_1` and `TABLE1_2`.

3. After table data has been stored, the corresponding indexes are unfinished and unavailable until the batch index creation process is completed.

4. If you specify `-i c` or `-i n`, but there is no data in the input data file, the system outputs index information only for the existing data to the index information file. You can use this file as an input to `pdrorg` to re-create the index. In this case, be sure to execute `pdload` in the addition mode (`-d` option omitted). If you execute it in the creation mode (`-d` option specified), `pdload` deletes the table data at first, thereby losing the index information.

5. If you execute data loading using the -i n option and then use ALTER TABLE to add RDAREAs before executing prong's batch index creation process (-k imp), the index information file output by the -i n option becomes unusable. If you are adding RDAREAs with ALTER TABLE, be sure to complete prong's batch index creation process beforehand. If you have already added RDAREAs using ALTER TABLE, execute prong's index re-creation process (-k ixrc).

(b) Notes about the creation of a plug-in index

1. If a plug-in index is defined, you cannot specify the index information output mode nor the index information output suppression mode. In these cases, specify either the batch index creation mode or the index update mode.
2. For a B-tree structure index, execution of data loading in the batch index creation mode creates all indexes. For a plug-in index, an index is created only on the additional data. Therefore, if you execute data loading involving 0 data records, a B-tree structure index is re-created, but not a plug-in index. However, for a plug-in index, loading 0 data records after re-initializing the index storage RDAREA results in the re-creation of an index. Use this option to repair an index in the event of abnormal termination during an index creation process, or if the index has been damaged due to a disk error.

(c) Notes about specifying -i c or -i n

When -i c or -i n is specified and the index or idxwork statement is omitted, the utility outputs the index information file to the *HiRDB-installation-drive*\tmp directory using the following naming conventions:

HiRDB-installation-drive\tmp\INDEX-*index-name*-*index-storage-RDAREA-name*-*unique-character-string*

If pdload terminates abnormally, this file is not deleted. If you re-execute pdload as is, a new file with a different name is created; this may result in a space shortage in the *HiRDB-installation-drive*\tmp directory. If the idxwork statement is specified, the utility also outputs the index information file to the directory specified in the idxwork statement using the same naming conventions. In this case also, if pdload terminates abnormally, the file is not deleted. You should delete unneeded index information files.

(4) -l log-acquisition-method

～ <<p>>

Specifies the method for acquiring the database updating log when pdload is executed.

If pdload terminates abnormally during execution, the database is not restored to its status before pdload was executed even when the update log has been acquired. For details about what to do in the event of abnormal termination of pdload, see *5.12*

Database status in the event of an error and recovery methods.

`a`

This indicates the log acquisition mode. This method collects database updating log information required for rollback and rollforward.

Criteria

This mode is suitable for loading a small amount of data.

When you execute data loading in the log acquisition mode, there is no need to make backups before and after execution of `pdload`, but performance is lower than in the other modes.

`p`

This indicates the pre-update log acquisition mode. This method collects database updating log information required for rollback, but it does not collect database updating log information required for rollforward.

Criteria

This mode is suitable for loading a large amount of data.

When you execute data loading in the pre-update log acquisition mode, the execution time is shorter than in the log acquisition mode. However, to handle possible media errors, you should back up the RDAREAs that store tables and indexes after executing `pdload`.

In the event of an error during execution of `pdload`, `pdload` restores the database to the synchronization point immediately before the occurrence of the error. This protects the RDAREAs from being placed in no-log shutdown status; however, the target table is not restored to its status before execution of `pdload`.

Notes

If data loading is executed on a table for which an abstract data type provided by a plug-in is defined, whether or not the `p` option (pre-update log acquisition mode) takes effect on the log output by the plug-in depends on the installed plug-in. If the plug-in does not support the `p` option, the `a` option (log acquisition mode) is assumed.

`n`

This indicates the no-log mode. The system does not update the database updating mode information.

Criteria

If only the table subject to data loading and its indexes are defined in the RDAREA, this mode is suitable for loading a large amount of data.

Data loading in the no-log mode is faster than in the other modes. However, to restore the database from its backup in the event of an error during utility execution or from its backup copy and log information in the event of an error on a medium, you must back up the RDAREAs storing the table and indexes both before and after executing the database load utility.

In the event of an error, you can restore the database only up to the point at which the backup was made.

Notes

1. If an error occurs while executing `pdload` in the no-log mode, you must either use a backup copy to restore the RDAREAs that were shut down due to the error or re-initialize them.
2. You cannot use the no-log mode for data loading with the synchronization point specification.

(a) Notes

1. If you specify `n` and the database can be restored from the previously acquired backup copy and log information or from the input data (if the table subject to data loading contains no data and only this table is to be stored in applicable RDAREAs), there is no need to make a backup copy prior to execution of `pdload`.
2. For details about how to operate when `p` or `n` is specified (database updating log information is not collected), see the *HiRDB Version 8 System Operation Guide*.
3. A transaction log is always collected by a transaction (`T`) created by `pdload`, regardless of the `-l` option's specification. The system creates the following amount of transaction log information per server; therefore, the formula for determining the amount of log information during execution of `pdload` is as shown below:

   Amount of log information = (1328 + 176 × 3) × T + A (bytes)

   $T = (x \times 2) + (y \times 2) + (z \times 2)$

   x: Number of tables*

   y: Number of RDAREAs storing LOB columns (LOB attribute)

   z: Number of indexes × number of RDAREAs storing indexes (not required if `-i s` is specified)

   * The value is the number of synchronization points in the case of data loading with synchronization point specification; otherwise, the value is `1`.

   A: Amount of system log information that is output according to the database

manipulation (amount of database updating log information). This value depends on the value of the -l option. For details about determining the amount of system log information, see the *HiRDB Version 8 Installation and Design Guide*.

If p or n is specified, the system collects ENQ log information in lock mode. The following is the amount of ENQ log information collected per server:

$$\textit{ENQ log information} = (p + q + r) \times T$$

p: Number of RDAREAs storing the table

q: Number of RDAREAs storing LOB columns (LOB attribute)

r: Number of RDAREAs storing indexes

Therefore, the amount of system log information the system outputs is the system log file record length × ENQ log information.

(5) -W

Specifies that the file output in binary format by the database reorganization utility with the -W option specified is to be used as the input data file.

Data loading fails if the table definitions, such as the column data types, are different between the unloaded table and the table subject to data loading. However, if you specify a column structure information file, you can load data to a non-FIX table with the column definition sequence and number of columns changed. When you use a UAP or UOC to change the contents of the input data file, you must edit the data in the format in which there are no spaces between the row length, column data offset, and row data.

If you specify this option, make sure that the -b option is also specified.

(6) -k LOB-creation-type

～ <<f or d>>

Specifies the data input method for storing LOB data in a LOB column, if a LOB parameter is used as an argument of the constructor function that generates the values to be stored in an abstract data type column. When this option is omitted, f is assumed, except that d is assumed when the -W option is specified.

f

Specify this option to prepare a file for each unit of LOB data. These files are called LOB input files.

c

Specify this option to prepare one file for all LOB data contained in one LOB column. This file is called a LOB column input file.

A column-unit LOB column input file is created when data is migrated from another database using a program.

You cannot specify this option for LOB data that is specified as an input parameter of the function that creates data for an abstract data type column. Therefore, if a table contains an abstract data type column that uses LOB as an input parameter, you cannot use a LOB column input file even for another LOB column that is not the abstract data type.

`v`

For the data to be stored in a BLOB column, provide a LOB input file for each set of LOB data to be stored in the column. Specify the data (LOB parameter) that serves as an input parameter for the constructor function that generates the values for the abstract data type columns if it is to be stored in an input file.

This specification takes effect only when the input data file is a binary-format file.

`d`

Specify this option to specify LOB data and LOB parameter data for abstract data type columns directly in an input data file.

This specification takes effect only when the input data file is a binary-format file.

When you specify `-k d`, you cannot execute data loading using UOC.

(a) Criteria

Determine the `-k` option according to whether or not there are LOB columns and LOB parameters for abstract data type columns and the format of input data file as follows:

LOB column	Format of input data file	Abstract data type column		
		No	Yes	
			Without LOB parameter	With LOB parameter
No	DAT format	—	—	`-k f`
	Binary format			`-k v` or `-k d`
	Fixed-size data format			`-k f`
	`pdrorg`-generated binary format	—	—	`-k d`

LOB column	Format of input data file	Abstract data type column		
		No	Yes	
			Without LOB parameter	With LOB parameter
Yes	DAT format	`-k f` or `-k c`	`-k f` or `-k c`	`-k f`
	Binary format	`-k f`, `-k c`, or `-k d`	`-k f`, `-k c`, or `-k d`	`-k v` or `-k d`
	Fixed-size data format	`-k f` or`-k c`	`-k f` or `-k c`	`-k f`
	`pdrorg`-generated binary format	`-k d`	`-k d`	`-k d`

—: There is no need to specify the `-k` option.

(b) Rules

1. If you are creating a LOB column structure base table separately from LOB columns, specify the same options when creating them.
2. The contents of an input data file and LOB input files depends on the `-k` option specification as shown as follows:

-k option	Input data file		LOB input file	
	Data stored in LOB column	Abstract data type LOB parameter	Data stored in LOB column	Abstract data type LOB parameter
`f`	Specify the name of the LOB input file	Specify the name of the LOB input file	Provide as many files as there are data items to be stored	Provide as many files as there are data items to be stored
`c`	Specify dummy data	—	Provide as many files as there are columns	—
`v`	Specify the name of the LOB input file	Specify the contents of LOB parameter	Provide as many files as there are data items to be stored	Not required
`d`	Specify the contents of LOB data	Specify the contents of LOB parameter	Not required	Not required

—: Not applicable

3. For data loading with the synchronization point specification, specify `d` in the `-k` option.Neither `f`, nor `c`, nor `v` can be specified.

(7) -c column-structure-information-filename

Specifies the name of a column structure information file.

For details about the column structure information file, see section *5.7 Column structure information file*. Specify a column structure information file in the following cases:

- The order of columns in the input data file does not match the order of columns in the table.
- The number of columns in the input data file does not match the number of columns in the table.
- Data in the input data file is to be converted to the data type of the table columns before being stored.
- Any column data in the input data file that matches the null comparison value is to be stored as the null value.
- A constructor function is used to generate the values for the abstract data type columns.
- The number of repetition column elements in the input data file is less than the maximum number of elements specified during table definition.

(a) Rules

1. If an input data file is in fixed-size data format, be sure to specify a column structure information file. Specification of a column structure information file is optional for an input data file in DAT or `pdrorg`-generated binary format. For an input data file in the `pdrorg`-generated binary format, you can specify a column structure information file to load data to a non-FIX table. An attempt to load data to a FIX table using a column structure information file results in an error. To modify the column structure of a FIX table using a column structure information file, treat the corresponding file as the input data file in the fixed-size data format.
2. If an input data file is in the DAT format and satisfies all the following conditions, there is no need to specify a column structure information file:
   - Columns of input data are in the same order as the order in which the table columns are defined.
   - Input data has the same number of columns as the columns in the table.
   - The constructor function that generates the data to be stored in the abstract data type has the same name as the column data type, and only one constructor function is used.
   - The number of input data elements to be stored in a repetition column is the same as the maximum number of elements specified for the applicable columns during table definition.
3. Provide the column structure information file in the host where the database load utility is executed (the `pdload` command is entered).

4. A column structure information file and a null value/function information file are mutually exclusive.

(8) -v null-value/function-information-filename

Specifies the name of a null value/function information file.

You can use a null value/function information file to convert input data to a value that represents a null value or to specify information about a constructor function that generates values for the abstract data type columns.

For details about the null value/function information file, see section *5.8 Null value/function information file*.

(a) Rules

1. You can specify a null value/function information file if the input data file is in the binary format, not in the DAT or fixed-size data format.
2. A null value/function information file and a column structure information file are mutually exclusive.

(9) -n [batch-output-local-buffer-sectors-count],[div],[local-buffer-sectors-count-for-random-access]

Specifies that a local buffer is to be used for loading data into the table. Specifying this option reduces the number of input/output operations because the system uses the local buffer for batch output to access the database.

When this option is omitted, the system uses the global buffer to output one page at a time.

batch-output-local-buffer-sectors-count ∼ <unsigned integer> ((2-4096))

Specifies the number of local buffer sectors for batch output. The batch output local buffer is used for data pages.

For the number of batch output local buffers, we recommend a value in the range 16-32. A guidelines is 64 kilobytes/page length.

`div`

Specify `div` when all the conditions listed below are applicable. If `div` is not specified when these conditions are all applicable, the number of input/output operations may increase, thereby affecting performance adversely.

- Data is to be loaded by table into a row partitioned table.
- Data is to be loaded into a row partitioned table that uses a hash function (`HASH0-HASH6`), or the key values of the input data that is stored in a table partitioned by key ranges are distributed randomly.

- There are multiple RDAREAs at one server in which the table is stored

When `div` is specified, the required memory size increases because the system allocates as many buffer sectors as there are table partitions in the server.

local-buffer-sectors-count-for-random-access ～ <unsigned integer> ((4-125000))

Specifies the number of local buffer sectors for random access. The random access local buffer is used for index pages.

It is recommended that you change the combination of the number of batch input/output local sectors and the number of random access local buffer sectors according to the table definition. Table 5-1 shows the recommended `-n` option specification.

Table 5-1: Recommended -n option specification (pdload)

Table type	Column definition	Table partitioning in server	
		Yes	No
FIX table	●	`-n` *x*, `div`	`-n` *x*
Non-FIX table	Variable-length character string whose column length exceeds 256 bytes or a BINARY column is defined	`-n`, *y*	
	Abstract data type column is defined	`-n` *x*, `div`, *y*	`-n` *x*, *y*
	Repetition column is defined		
	Other	`-n` *x*, `div`	`-n` *x*

Legend:

x: Number of batch output local buffer sectors

y: Number of random access local buffer sectors

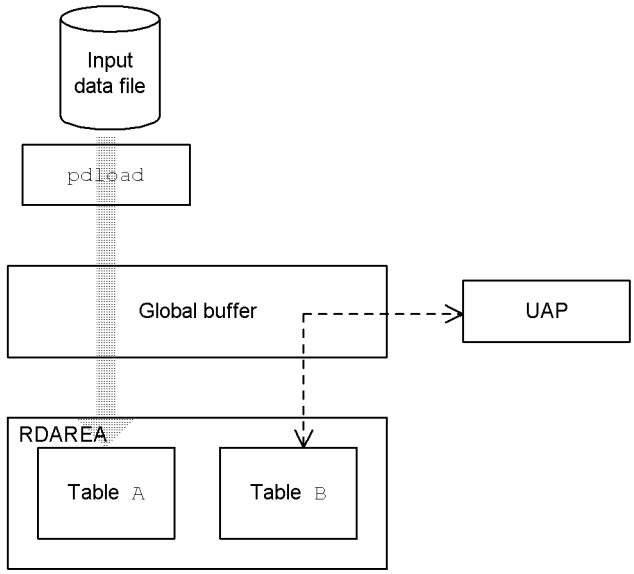
●: Not applicable

(a) Buffer used by pdload

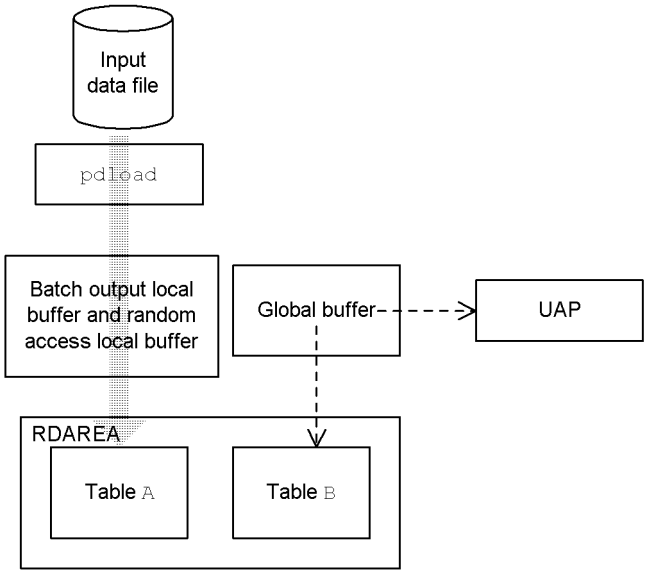
1. When the `-n` option is omitted, the system uses the global buffer. In this case, the transaction performance of a UAP that uses the global buffer drops because a large amount of global buffer space is used during data loading. Specifying the `-n` option eliminates such buffer contention. Figure 5-9 show the relationship between pdload and buffer.

Figure 5-9: Relationship between pdload and buffer

- When only a global buffer is used



- When local buffers and a global buffer are both used



Explanation:

If only the global buffer is used (`-n` option is omitted), buffer contention

occurs between `pdload` and the UAP.

When both local and global buffers are used (`-n` option is specified), buffer contention does not occur between `pdload` and the UAP. However, for data loading to LOB columns, the system uses the global buffer even when the `-n` option is specified.

2. If only the number of batch output local buffer sectors is specified in the `-n` option, the system uses a single batch output local buffer per RDAREA. If there are multiple RDAREAs, buffer contention occurs because only one batch output local buffer is used. If buffer contention occurs, the number of input/output operations increases, thereby affecting performance adversely. In such a case, specify `div`. When `div` is specified, buffer contention will not occur because the system allocates as many batch output local buffers as there are RDAREAs (one batch output local buffer per RDAREA).
3. Even when the batch output local buffer is specified, the system may use the global buffer depending on the conditions. Table 5-2 describes the relationships between the conditions and the buffer that is used.

Table 5-2: Relationship between conditions and the buffer that is used (pdload)

Condition				Specification of random access local buffer				
				Not specified		Specified		
				Global buffer	Batch output local buffer	Global buffer	Batch output local buffer	Random access local buffer
Data page	RDAREA storing LOB column structure base table	FIX table		N	Y	N	Y	N
		Non-FIX table	When variable-length character strings or `BINARY` columns whose column length exceeds 256 bytes are defined	Y	N	Y	N	N
			When abstract data type columns or repetition columns are defined and a row of data cannot fit in one page	Y	N	Y	N	N
			Other	N	Y	N	Y	N
	RDAREA for storing LOB columns			Y	N	Y	N	N
	RDAREA for storing LOB attributes			Y	N	Y	N	N
Index page	When key values are searched from the table (`-i c` or `n` is specified)			N	Y	N	Y	N
	When indexes are created concurrently (`-i s` is specified)			Y	N	N	N	Y
	When batch index creation is executed (`-i c` is specified)			Y	N	N	N	Y
Directory page				Y	N	Y	N	N

Legend:

Y: Used

N: Not used

(b) Rules

1. If the global buffer is not sufficient for a rebalancing table, performance may drop considerably. Therefore, if there is no column whose definition length exceeds 256 bytes, we recommend that you specify the number of batch output pages.

   If the local buffer specification is not effective for a reason such as the existence of a column whose definition length is 256 bytes or greater, allocate at least the following number of global buffer sectors:

   Number of buffer sectors required per RDAREA

   = 1024 ÷ (number of table storage RDAREAs) × 2+ 3

2. For a rebalancing table with FIX hash partitioning, the system allocates a buffer equivalent to the specified number of pages for each hash group, thereby consuming more memory than for a table that is not partitioned or partitioned by other conditions.

(10) -u authorization-identifier

Specifies the authorization identifier of the user executing the database load utility.

For the default value, see *(b) Default value*.

When this option is specified, a message requesting password entry is displayed. If no password is required, enter null in response to the message. The utility checks the authorization identifier entered and password to verify that the user is authorized to establish connection with the HiRDB system and to access the table.

(a) Criterion

Specify this option to use an authorization identifier that is not defined in the `PDUSER` environment variable.

(b) Default value

When this option is omitted, the system assumes the authorization identifier and password as follows:

1. The system assumes the value of the `PDUSER` environment variable during the execution of database load utility. Be sure to specify `PDUSER` if you are executing the utility in an environment in which a password cannot be entered. Following are examples of the `PDUSER` environment variable:

   Specifying a password:

   `PDUSER \"`*authorization-identifier*`\"/\"`*password*`\"`

   Not specifying a password:

   `PDUSER \"`*authorization-identifier*`\"`

2. If the PDUSER environment variable is not set, the system assumes the login window's user name. Enter the password when a message is displayed requesting password entry. If no password is required, enter null in response to the message.

(c) Rules

1. Do not specify this option in an environment in which a password cannot be entered.
2. If you enclose an authorization identifier in \", the system treats it as being case sensitive; otherwise, the system treats it as in all uppercase letters.

(11) -x

When a cluster key has been defined for the table, this option specifies that data loading is to be performed in the order of the input data without checking to see whether the input data is in ascending or descending order of the cluster key values.

When this option is omitted, the utility performs cluster key checking and treats any out-of-order input data as an error.

(a) Criterion

If you know that the input data is sorted by the cluster key, specify this option to reduce the overhead of cluster key checking.

(b) Note

1. For a table for which a UNIQUE cluster key has been defined, checking of duplicate keys is not available; therefore, an attempt to store data with a duplicate key will result in an error (rollback).
2. This option enables input data that is not sorted by the cluster key to be stored in a table for which a cluster key has been defined; however, it does not provide the clustering effect.

(12) -s separator-character

～ <character string>

For an input data file in DAT format, this option specifies the separator character to be used as the delimiter between data items. When this option is omitted, the comma (,) is assumed.

In the binary or fixed-size data format, this option is ignored, if specified.

(a) Criterion

Specify this option to use a character other than the comma (,) as the separator between data items in the input data.

(b) Rules

1. You can specify the tab symbol and the pipe symbol (|) as separator characters if

they do not occur in the input data. However, these special characters should be enclosed in double quotation marks (").

2. None of the following characters can be specified as the separator character:
   - Uppercase letters (A-Z) and lowercase letters (a-z)
   - Numeric characters (0-9)
   - The following characters that are used as reserved characters by the utility:

     Asterisk (*), double quotation mark ("), underline (_)
3. The following characters are not suitable for use as separator characters because they can occur in input data codes. Separator characters are single-byte codes; therefore, you cannot specify a double-byte code as a separator character.
   - Signs for numeric data input (+ or -)
   - Characters that can occur as Japanese input character codes:

     |, \, [,], (,), {, },zueng045.tif
   - The hyphen (-) for date data input
   - The colon (:) for time data input
   - The period (.) for time and date interval data input

(13) -e

Specifies that processing is to be cancelled if an error is detected in the input data.

To cancel processing and ignore the storage processing that had been performed up to that point (to roll back), specify the `dataerr` operand in the `option` statement.

If you omit both the `-e` option and the `dataerr` operand in the `option` statement, the utility continues processing and stores only normal data in the database even if it detects invalid input data.

When the `-e` option is specified, the system outputs the cause of error to the error information file only for the first data item resulting in an error. When the `-e` option is omitted, the system outputs the cause of error for all data resulting in an error.

(a) Criteria

You can cancel processing when the first error is detected in the input data.

Suppose that you are loading data to a repetition column using an input data file in DAT format. Although the input data is created in VV format, it is treated as FF format if the `elmtype` operand is omitted from the `array` statement, in which case all input data results in an error and the system creates an unneeded error information file as well as an error data file. In this case, if you specify the `-e` option, the system cancels the processing when it detects the first error, thereby avoiding the creation of unneeded

files.

(b) Rules

1. If the following conditions are satisfied, the utility cancels processing and rolls back regardless of the specification of this option:

Condition	Reason
In a binary-format input data file, the length field of variable-length data is a negative value.	In a binary-format with no delimitation between rows, row data must be identified because data is divided by the column length. Because data cannot be divided by a negative value, the utility cannot continue processing.
The `dataerr` operand is specified in the `option` statement.	When the `dataerr` operand is specified, the utility rolls back processing. Therefore, continuing with the processing serves no purposes.

2. The name of an error information file is specified in the control information file. If there is no such specification, the database load utility creates the file in the *HiRDB-installation-drive*`\tmp` directory at the host that contains the input data file. In this case, the database load utility assigns a file name and displays it in a message.
3. If the server where an input file was prepared using a HiRDB/Parallel Server (the server specified in the `source` statement) is different from the server containing the table subject to data loading, the system cannot cancel the processing when it detects any of the following errors:
   - Duplicate key error on a unique index key or a primary key index
   - Invalid value for an abstract data type column

   If you specify the `-e` option to execute data loading in units of tables into a table that is divided into multiple servers on a HiRDB/Parallel Server, the system cannot guarantee the extent to which input row data has been stored in a table in the event of an error. Therefore, do not specify the `-e` option if all of the following conditions are met:
   - HiRDB/Parallel Server
   - Data loading in units of tables into a table divided into multiple servers
   - Tables with a unique key index or primary key index defined, or tables containing columns of abstract data type
4. If the `dataerr` operand is specified in the `option` statement and error data is detected, the utility cancels processing even when the `-e` option is omitted.

(14) -r input-begin-line

～ <unsigned integer> ((2-4294967295)) <<1>>

Specifies that data input is to begin at the specified line, not the beginning of the input data file.

(a) Criteria

If the data loading process specifying the `-e` option is cancelled, specify this option to restart the data loading.

(b) Notes

1. When a UOC is used for data loading (to read the input data file), the `-r` option if specified, is ignored.
2. If you re-execute data loading with a synchronization point specified using the `-r` option after the initial data loading rolled back due to abnormal termination, the system determines the input begin line as follows:

Status	Input begin line during re-execution
`-r` *input-begin-line* > Line saved during the data loading with a synchronization point specified	`-r` *input-begin-line*
`-r` *input-begin-line* = Line saved during the data loading with a synchronization point specified	Line saved during the data loading with a synchronization point specified*
`-r` *input-begin-line* < Line saved during the data loading with a synchronization point specified	Line saved during the data loading with a synchronization point specified*

* The `KFPL00810-I` message is displayed, indicating the re-execution of `pdload`.

(15) -z

Specifies that variable-length character string data, variable-length national character string data, and variable-length mixed character string data with a length of 0 is to be stored.

(a) Criteria

Specify this option to avoid storing the null value or single-byte space (one space character) for variable-length character string data, variable-length national character string data, or variable-length mixed character string data.

(b) Notes

1. For data in the DAT format, the system handles data with a length of zero as follows:

   `. . . , " ", . . .` (Treated as data with a length of 0)

. . . , , . . . (Treated as the null value)

2. For data in the binary format, the system sets 0 as the actual data length and treats data without the actual data section as having a length of 0.

(16) -y

Specifies that data is to be stored in unused area in used pages during data loading if all unused pages become completely full. When this option is specified, the system displays the KFPH26010-I message before storing data in the unused area.

(a) Criteria

Normally, when the unused pages are used up during data loading, an error results and the processing is rolled back. To avoid this, specify this option so that data loading can be completed by storing data in unused area.

(b) Rules

1. You cannot specify p (pre-update log acquisition mode) with the -l option.
2. Once an unused page has been completely used, the specified percentage of unused space per page is not applicable to the data storage.
3. The -n option, if specified, has no effect after an unused page has been used up.

(17) -o

Specifies that the index information file specified in the index statement is to be deleted automatically after batch index creation terminates normally. Note that the directory specified in the idxwork statement and the index information file in the *HiRDB-installation-drive*\tmp directory that was created automatically by pdload are deleted after batch index creation terminates normally, regardless of the specification of this option.

(a) Criterion

An index information file created for plug-in indexes tends to be large and requires a large amount of disk space, if kept on the disk. To make sure that this index information file is deleted, you can specify the -o option to automatically delete index information files after the batch index creation process.

(b) Rules

For data loading in units of RDAREAs, it is necessary to execute pdrorg (-k ixmk) using the index information file output after data loading for the following indexes. Therefore, the -o option, if specified, is ignored in this case.

- There are multiple table storage RDAREAs in a single server, but there is only one index storage RDAREA for them.

(18) -m progress-message-output-interval

～ <unsigned integer> ((1-1000)) <<10>>

Specifies, in units of 10,000 lines, an interval at which a message is displayed indicating the progress of the current process.

(a) Criterion

If the default value, which is 10,000 lines, is too many or too few, specify a desired value.

(b) Rule

During batch index creation, the system ignores this option and displays messages only when the index creation begins and ends.

(19) -X response-monitoring-time-for-server-to-server-communication

～ <unsigned integer> ((1-65535)) <<300>>

If an error (such as a communication error) occurs at the server where the command was executed, the command may stop responding and the application may stop. To help you detect errors, `pdload` enables you to monitor the response time for communication during dictionary manipulation performed by the command.

The `-X` option specifies the response monitoring time for dictionary manipulation (in seconds). If the execution time during dictionary manipulation exceeds the value set in the `-X` option, `pdload` assumes a dictionary access error and cancels processing with return code `8`.

Criteria

- If you want to detect an error in less time than 300 seconds in the event of a no-response from the server due to a communication error or unit down, specify a value that is smaller than 300 in the `-X` option.
- If the system switchover facility is used, the command may keep waiting for a response even though system switchover has been completed. In such a case, you can terminate the command immediately by reducing the monitoring time.
- The specified monitoring time may result in a timeout if a response from the dictionary is delayed and the utility's preprocessing is not completed within 300 seconds (which is the default value for the `-X` option). This can happen when many applications and utilities are executing concurrently. In such an environment, specify a value greater than 300 in the `-X` option.

(20) [authorization-identifier.] table-identifier

Specifies the table identifier of the table subject to data loading. When the authorization identifier is omitted, the system assumes the user name used to establish

connection with HiRDB.

(a) Rule

If you enclose an authorization identifier in \", the system treats it as case sensitive; otherwise, the system treats it as in all uppercase letters.

(21) control-information-filename

Specifies the name of the control information file that contains the control statements of the database load utility.

You can specify any of the control statements listed below in the control information file. For details about the control statements, see sections *5.4.3* through *5.4.16*.

- source statement (specification of input data file information)
- index statement (specification of index information file information)
- idxwork statement (specification of index information file directory information)
- sort statement (specification of sort work directory information)
- lobdata statement (specification of LOB input file information)
- lobcolumn statement (specification of LOB column input file information)
- lobmid statement (specification of LOB middle file information)
- srcuoc statement (specification of UOC storage library information)
- array statement (specification of array data format for a table containing repetition columns)
- extdat statement (specification of information about the extended DAT format)
- src_work statement (specification of the output destination of divided-input data files)
- constraint statement (specification of check pending status)
- option statement (specification of data processing information)
- report statement (specification of a file to which pdload tuning information is output)

(a) Number of specifiable control statements

The following table shows the maximum number of control statements permitted in a control information file:

Control statement	Maximum number of control statements
source statement	1

Control statement	Maximum number of control statements
index statement	Number of indexes × number of storage RDAREAs
idxwork statement	Number of index storage servers
sort statement	Number of index storage servers
lobdata statement	1
lobcolumn statement	Number of LOB columns
lobmid statement	1
srcuoc statement	1
array statement	1
extdat statement	1
src_work statement	1
constraint statement	1
option statement	1
report statement	1

(b) Relationship between control statements and options

The following shows the relationship between control statements and options.

- Creating an index

  Specify the following control statements in the -i option

Control statement	Specification of -i option			
	c	n	s	x
source statement	R	R	R	R
index statement	O	O	—	—
idxwork statement	O	O	—	—
sort statement	O	—	—	—

R: Required

O: Optional

—: Not required

- Entering LOB data and parameters

  When you enter LOB data and parameters, the control statements to be specified depend on whether or not the LOB column structure base table and LOB columns are created at the same time.

Control statement	Unit of data loading							
	Loading data to LOB column structure base table and LOB columns				Loading data to LOB column structure base table		Loading data to LOB columns	
	-k f	-k c	-k v	-k d	-k f	-k c	-k f	-k c
`source` statement	R	R	R	R	R	R	—	—
`lobdata` statement	R	R	R	—	—	—	R	O
`lobcolumn` statement	—	R	—	—	—	—	—	R
`lobmid` statement	R*	R	R*	—	R*	R	R*	R

R: Required

O: Optional

—: Not required

* For a LOB parameter to a constructor function that generates data for abstract data type columns, this statement, if specified, is ignored.

(c) Files and directories specified in the control statements

The following rules apply to specifying files and directories in the control statements:

1. Access privileges must be granted to the HiRDB administrator. If some control statements or operands are omitted, the utility may assume that directories or files are to be created in the *HiRDB-installation-drive*\tmp directory; therefore, you must also grant access privileges to the *HiRDB-installation-drive*\tmp directory.
2. Path names are not case sensitive. A duplication error occurs if you specify multiple absolute path names, the only difference among which is uppercase or lowercase letters.
3. A path name that is connected to a network drive cannot be specified.

5.4.3 source statement (specification of input data file information)

The `source` statement specifies information about the input data file.

Criterion

Be sure to specify this statement unless you are loading data only to the LOB columns of a table.

Rule

Specify the `source` statement in single line (with a maximum length of 1023 bytes). You can specify the `source` statement only once in a control information file.

(1) Format

```
source [RDAREA-area] [{server-name|host-name}:]
     {input-data-filename[,input-data-filename]... |(uoc)}
     [error=error-information-filename]
     [errdata=error-data-filename[, output-rows-count]]
     [errwork=work-buffer-size-for-error-data-file-creation]
     [maxreclen=input-data-length]
```

(2) Explanation

(a) RDAREA-name

～ <identifier> ((1-30))

When data loading is to be performed by RDAREA on a row-partitioned table, this option specifies the name of the RDAREA to be subject to data loading.

Rules

1. For a table partitioned by key ranges or FIX hash values, HiRDB checks the data to determine whether or not it can be stored in the RDAREA. If the data falls beyond the specified storage range, an error results. However, for a table partitioned by flexible hash values, HiRDB stores the data as is without checking.
2. The system treats an RDAREA name that is enclosed in double quotation marks (") as case sensitive; otherwise, the system treats it as all uppercase letters. If an RDAREA name contains a space, you must enclose it in double quotation marks.

(b) {{server-name|host-name}

Specifies the name of the server or host containing the input data file(s).

server-name ～ <identifier> ((1-8))

HiRDB/Single Server

Do not specify this option for a HiRDB/Single Server.

HiRDB/Parallel Server

Specify the name of the front-end server or back-end server containing the input data file.

If you specified an RDAREA name, you can omit this information, in which case the system assumes the name of the server where the specified RDAREA is stored. If you omitted the RDAREA name, be sure to specify this option.

host-name ～ <identifier> ((1-32))

HiRDB/Single Server

Specify the name of the host containing the input data file. This is the name of the host where the single server is located.

You can omit this option regardless of whether or not an RDAREA name is specified. When it is omitted, the system assumes the name of the host where the database load utility (`pdload` command) was executed.

If you are using the system switchover facility, specify the primary system's host name.

HiRDB/Parallel Server

Do not specify this option for a HiRDB/Parallel Server.

(c) input-data-filename

～ <pathname>

Specifies the absolute pathname of an input data file containing data to be input.

If there are multiple input data files, separate each file by a comma (,). For details about the input data file, see section *5.5 Input data file*.

The tape device access facility is supported for input data files. For details about the tape device access facility, see *1.4.3 Tape device access facility*.

HiRDB/Single Server

Create the input data file(s) in the server machine where the single server is located.

HiRDB/Parallel Server

Create the input data file(s) in the front-end server or a back-end server.

Rule

1. The system checks the specified input data file for its accessibility before starting data loading. If access is denied, the system does not execute data loading.
2. A file containing a byte order mark (BOM) cannot be used as an input data file. For `pdload`, use files that do not have a BOM.

(d) (uoc)

Specifies that a UOC is to be used to input/output the input data file. For details about UOCs, see section *5.10 Using a UOC to load a table*.

(e) error=error-information-filename

～ <pathname>

Specifies the absolute pathname of the file to which error information is to be output. If an error is detected in the input data, the system outputs the error information to the error information file.

For details about the error information file, see section *5.6 Error information file*.

HiRDB/Single Server

Create the error information file in the server machine that contains the input data file.

HiRDB/Parallel Server

Create the error information file in the server that contains the input data file (front-end server or back-end server).

When this option is omitted, the database load utility creates the file with a unique name in the *HiRDB-installation-drive*`\tmp` directory.

If the `-e` option is specified, the utility creates the error information file only when there is an error in the input data.

(f) errdata=error-data-filename [, output-rows-count]

Specifies that erroneous rows of data are to be output, if detected.

You can correct the rows of data that are output to the error data file and load them again as an input data file.

If the `-e` option is specified, the system ignores the specification of an error data file.

error-data-filename ～ <pathname>

Specifies the absolute pathname of the file to which erroneous row data is to be output. You need to create this file at the same server as the input data file. If omitted, the system does not output erroneous rows of data.

output-rows-count ～ <unsigned integer> ((1-4294967295)) <<100>>

Specifies the maximum number of erroneous rows of data that can be output. If the number of erroneous rows exceeds the specified value, the system continues processing but outputs only the specified number of erroneous rows of data.

Rules

1. The following limitations apply to the output results of the error data file:

   Input data file in the DAT format

   If the actual data is larger than the value specified in the `maxreclen` operand, the system does not output any data. If the `maxreclen` operand is omitted and there is at least 32 KB of data, the system does not output the data.

   Input data file in the binary format

   If the system is unable to edit one line of data due to erroneous length information in the variable-length character string data, the system outputs only the part of the data that was edited successfully. The system does not output fixed-length column data that is less than the defined length.

   LOB column input file

   The system outputs only the erroneous rows of data from the input data file to the error data file. To re-load data after correcting the input data file, you need to correct the LOB column input file according to the output sequence of the input data file.

2. An index key value duplication error is not output to the error data file in the following cases:
   - It is created in the batch index creation mode.
   - For a HiRDB/Parallel Server, the processing is in the index update mode, the input data file is located at a server other than the server containing the table storage RDAREAs, and a buffer shortage occurred during the creation of the error data file.
3. Output of an error data file is not applicable when data loading is executed on LOB columns or the input data file used is in binary format created by `pdrorg`.
4. Even if an error occurs in a variable-length character string in a binary-format input data file, only the columns up to the one immediately preceding the error are output to the error data file. Note these points when checking the error data file.

(g) errwork=work-buffer-size-for-error-data-file-creation

～ ((0-2097152))

When specifying the `errdata` operand, specify the buffer size for creating an error data file in KB.

If the KFPL25222-W message is issued during the data loading specifying the errdata operand, and a part of the error data is not output to the error data file, specify this option to re-execute data loading.

If you specify a value of 0 in the errwork operand, neither key duplication errors nor invalid value errors for abstract data type columns are output, but data loading performance improves.

A KFPL25222-W message is issued when all of the following conditions are met:

1. In a HiRDB/Parallel Server, the server name specified in the source statement is different from the name of the back-end server in which a table storage RDAREA is defined; in a HiRDB/Parallel Server, or, a table is partitioned by row and stored in multiple back-end servers.
2. A unique key index or primary key index is defined for the table subject to data loading, or the table contains columns of abstract data type.
3. One of the following is true:
   - The input data file contains data that is not to be stored in the database (data in a column structure information file for which a skipdata statement is specified; data that is longer than a defined column length; or data that matches a null comparison value and is treated as a null value).
   - Data is to be stored in a column of VARCHAR, NVARCHAR, MVARCHAR, BINARY, or BLOB data type.
   - Data is to be stored in an abstract data type for which the input parameter type of the constructor function is VARCHAR, NVARCHAR, MVARCHAR, BINARY, or BLOB.

Estimation formula

The following shows a formula for estimating the size of the work buffer for the error data file. This formula provides a value in bytes, but round it up to the nearest KB to specify the option.

Buffer size = {zueng010.tif *X*/(average length of database storage row + *Y*) zueng010.tif}
× average length of input data row × 2
× number of servers for which table storage RDAREAs are defined

X: Value of pd_utl_buff_size in the system common definitions × 1024

Y: FIX table: 24

Non-FIX table: (number of columns + 1) × 4 + 24

For details about the average length of database storage row (how to calculate the number of table storage pages), see the *HiRDB Version 8 Installation and Design Guide*.

Note that the previous estimation formula is based on the average row length. Therefore, a buffer shortage may occur depending on the actual arrangement of row data. If there is enough memory, you can ensure output to an error data file by revising the formula as follows:

- For non-FIX tables, set the average length of a database storage row to `0`.
- Set the average length of an input data row to the maximum length of input data.

(h) maxreclen=input-data-length

This option is applicable to an input data file created in DAT format, extended DAT format, binary format, or `pdrorg`-generated binary format. When you are using the input data file in the fixed-length data format or a streaming tape device, this operand is ignored, if specified.

When the input data file is in DAT or extended DAT format ～ <unsigned integer> ((0, 32-524288)) <<32>>

For the input data file in the DAT format, if a row of data exceeds 32 KB, this operand specifies the maximum data length per row in the input data file in KB.

If this operand is omitted and the input data file contains a row of data that is 32 KB or greater, or that is greater than the specified operand value, the system cancels the processing.

Rules

1. If this operand is omitted or a streaming tape device is used, each row of data in the input data file (DAT format) must not exceed 32 kilobytes.
2. If you are using an unload data file in DAT format that was output with the `-W dat` option specified in `pdrorg`, the maximum length is displayed in the `KFPL22222-I` message.
3. If you specify `0` in this operand, `pdload` calculates the row length on the basis of the definition of the table subject to processing. Because the utility uses the table definition, if the input data file contains data that is not to be stored in the table, the value obtained by the utility does not match the input data length, resulting in an error. In this case, specify a non-zero value as the input data length.

Data type or parameter type	Calculation value (bytes)	Formula
`INTEGER`	12	Sign + number of digits + separator character

Data type or parameter type	Calculation value (bytes)	Formula
SMALLINT	7	Sign + number of digits + separator character
DECIMAL(*m, n*)	*m* + 3	Sign + number of digits + decimal point + separator character
FLOAT	24	Sign + mantissa part + decimal point + *e* + sign + exponent part + separator character
SMALLFLT	17	Sign + mantissa part + decimal point + *e* + sign + exponent part + separator character
DATE	11	Specification format + separator character
TIME	9	Specification format + separator character
INTERVAL YEAR TO DAY	11	Sign + number of digits + comma + separator character
INTERVAL HOUR TO SECOND	9	Sign + number of digits + comma + separator character
TIMESTAMP(p)	19 + (p + 1) + 1	Specification format + separator character
CHAR(*n*), MCHAR(*n*), VARCHAR(*n*), or MVARCHAR(*n*)	*n* + 3	Number of characters + double quotation marks + separator character
NCHAR(*n*), or NVARCHAR(*n*)	(*n* × 2) + 3	Number of characters + double quotation marks + separator character
BINARY(n)	n + 3	Number of characters + double quotation marks + separator character
BLOB	1025	Maximum length of path name + separator character

When the input data file is in binary format or the pdrorg-created binary format
～ <unsigned integer> ((0, 32-2097152)) <<0>>

If you are executing data loading on a table with a BINARY column, specify the maximum row length in the input data in kilobytes. If you are executing data loading on a table with no BINARY columns, this operand is ignored, if specified.

Rules

1. If this operand is omitted or 0 is specified, pdload obtains the maximum row length from the table definition and uses that value for processing. Note that if you have specified the input data length, but the value obtained from the table definition by pdload is smaller, the utility still uses the latter for

processing.

2. If the actual maximum data length is less than the maximum row length obtained by `pdload` from the table definition, specify this operand to minimize the area to be allocated.

5.4.4 index statement (specification of index information file information)

When executing data loading in the batch index creation or index information output mode, use the `index` statement to specify information about the index information file to which index information is to be output.

Criteria

Specify the `index` statement, if possible, so as to avoid a space shortage in the HiRDB installation drive.

If there are many indexes or index storage RDAREAs, you should specify the `idxwork` statement.

Rules

1. Specify as many `index` statements as there are (the number of indexes × number of index storage RDAREAs).
2. When the `index` statement is omitted, the system creates an index information file under the directory specified in the `idxwork` statement. If the `idxwork` statement is also omitted, the system creates an index information file in the *HiRDB-installation-drive*`\tmp` directory on the server that contains the index storage RDAREAs.
3. If you specify both `index` and `idxwork` statements, the `index` statement takes effect.
4. If no index has been defined for the target table, do not specify the `index` statement.

(1) Format

```
index index-identifier [RDAREA-name] index-information-filename
```

(2) Explanation

(a) index-identifier

Specifies the identifier of the index.

The system treats an index identifier enclosed in double quotation marks (`"`) as case sensitive; otherwise, the system treats it as all uppercase letters. Enclose an index

identifier in double quotation marks if it contains a space.

(b) RDAREA-name

～ <identifier> ((1-30))

For a row partitioned table, this operand specifies the name of the RDAREA containing the index.

The system treats an RDAREA name enclosed in double quotation marks as case sensitive; otherwise, the system treats it as all uppercase letters. Enclose an RDAREA name in double quotation marks if it contains a space.

(c) index-information-filename

～ <pathname>

Specifies the absolute pathname of the index information file to which index information is to be output.

This index information file must be in the server machine or back-end server where the index storage RDAREA is located.

5.4.5 idxwork statement (specification of index information file directory)

When the `index` statement is omitted, the `idxwork` statement specifies the name of the directory in which an index information file is created automatically.

Criterion

Specify the `idxword` statement, if possible, so as to avoid a space shortage in the HiRDB installation drive.

Rules

1. If the `index` and `idxword` statements are both omitted, the system creates an index information file in the *HiRDB-installation-drive*`\tmp` directory on the server where the index statement RDAREAs are located.
2. You can specify as many `idxwork` statements as follows:

   HiRDB/Single Server:

   Specify only one `idxwork` statement.

   HiRDB/Parallel Server:

   Specify as many `idxwork` statements as there are servers in which the partitioned indexes are stored for a row-partitioned table. For a table that is not partitioned, or when loading data in units of RDAREAs for a partitioned table, specify only one `idxwork` statement.
3. If both `index` and `idxwork` statements are specified, the `index` statement

takes effect, in which case the system ignores the idxwork statement.

4. If no index has been defined for the target table, do not specify the idxwork statement.

(1) Format

```
idxwork [server-name] directory-name
```

(2) Explanation

(a) server-name

～ <identifier> ((1-8))

Specifies the name of the server at which the index information file is to be created.

HiRDB/Single Server

Do not specify this operand for a HiRDB/Single Server. The name of a single server is ignored, if specified.

HiRDB/Parallel Server

Specify the name of the server at which the index information file is to be created.

(b) directory-name

～ <pathname> ((1-255))

Specifies the absolute pathname of the directory in which the index information file is to be created.

(3) Notes

This subsection shows the name of the index information file that is created automatically.

directory-name\INDEX-*index-name*-*index-storage-RDAREA-name*
-*unique-character-string*

Example:

If the directory name is c:\hirdb\pdload, index name is IDX1, and the name of the index storage RDAREA is USER01 in the idxwork statement, the index information file is created under the following name:

```
c:\hirdb\pdload\INDEX-IDX1-USER01-aaaa00001
```

5.4.6 sort statement (specification of sort work directory information)

When loading data in the batch index creation mode for B-tree indexes, you can use the `sort` statement to specify the directory for a sort work file.

Criterion

Specify the `sort` statement, if possible, so as to avoid a space shortage in the *HiRDB-installation-drive*`\tmp` directory.

Rules

1. When the `sort` statement is omitted, the utility assumes the *HiRDB-installation-drive*`\tmp` directory in the server where the index storage RDAREAs are located.
2. If only a plug-in index is defined for the table, the `sort` statement, if specified, is ignored.
3. You can specify as many `sort` statements as follows:

   HiRDB/Single Server:

   Specify only one `sort` statement.

   HiRDB/Parallel Server:

   Specify as many `sort` statements as there are servers in which the partitioned indexes are stored for a row partitioned table. Even with the indexes of a row partitioned table, if data loading is executed in units of RDAREAs, specify only one `sort` statement.
4. If no index has been defined for the target table, do not specify the `sort` statement.

(1) Format

```
sort [server-name] directory-name[,buffer-size-for-sorting]
```

(2) Explanation

(a) server-name

～ <identifier> ((1-8))

Specifies the name of the server in which the sort work file is to be created.

HiRDB/Single Server

Do not specify this operand for a HiRDB/Single Server. A single server name is

ignored, if specified.

HiRDB/Parallel Server

Specify the name of the server in which the sort work file is to be created.

(b) directory-name

～ <pathname>

Specifies the absolute pathname of the directory under which the sort work file is to be created.

(c) buffer-size-for-sorting

～ <unsigned integer> ((128-2097152)) <<1024>>

Specifies in KB the amount of memory that is to be used as the buffer.

For a HiRDB/Single Server, the system reserves this buffer at the single server; for a HiRDB/Parallel Server, the system reserves it at a back-end server.

The sort processing creates a temporary work file in a specified directory. You can use the formula shown below to minimize the size of this file. Note that this is just a guideline. If there is not enough memory or most data is sorted in the order of index key values, do not specify a value that is larger than necessary. This formula provides the value that minimizes the file size; it does not minimize the sorting time. If there is enough memory, specify a buffer whose size is several megabytes to several dozens of megabytes.

• In 32-bit mode HiRDB

$$\text{Buffer size (bytes)} \geqq \frac{R+7}{2} + \sqrt{(B+8) \times n \times A + \frac{(R+7)^2}{4}} + C$$

• In 64-bit mode HiRDB

$$\text{Buffer size (bytes)} \geqq \frac{R+15}{2} + \sqrt{(B+8) \times n \times A + \frac{(R+15)^2}{4}} + C$$

n

Number of data items (sum of existing data items in the table and the data items to be added). For a repetition column, this is the number of elements, not the number of rows.

k

Key length (calculated based on the maximum value). For the formula for determining the key length, see a sample calculation of index storage pages in the *HiRDB Version 8 Installation and Design Guide*.

x

10, if all key structure columns are fixed length; 12, if some of the key structure columns are variable length.

c

Number of columns comprising the index.

z

$c \times 4$ for a variable-length multicolumn index; 0 otherwise.

K

$k + c + 8$ for a variable-length multicolumn index; $k + 12$ otherwise.

N

$(c \times 2) + 2$ for a variable-length multicolumn index; 5 otherwise.

R

$k + x + z$

A

$R + (K + 8) + 28$ for 32-bit mode HiRDB; $R + (K + 8) + 56$ for 64-bit mode HiRDB.

B

$R + (K + 8) + 56$ for 32-bit mode HiRDB; $R + (K + 8) + 104$ for 64-bit mode HiRDB.

C

$2092/(N \times 32) + (K + 8)$ for 32-bit mode HiRDB; $2112/(N \times 32) + (K + 8$ for 64-bit mode HiRDB.

5.4.7 lobdata statement (specification of LOB input file information)

When loading data to a table containing LOB columns or entering LOB data as an input parameter for a constructor function, you can use the `lobdata` statement to specify the LOB information.

Criteria

Specify the `lobdata` statement when loading data to a table with LOB columns (except when loading data only to the LOB column structure base table).

You can also specify the `lobdata` statement when loading data to a table with abstract data type columns using LOB data as an input parameter to a constructor function.

Rules

1. You can specify the `lobdata` statement if you also specify the `-k f`, `-k c`, or `-k v` option.
2. You can load data to both LOB column structure base tables and LOB columns at the same time or to the LOB column structure base tables first and then to the LOB columns.

   If you are loading data only to the LOB column structure base table, do not specify the `lobdata` statement. The `lobdata` statement is required to store data in both LOB column structure base tables and LOB columns at the same time or to the LOB columns later.
3. If you are providing a LOB input file for each LOB column and LOB parameter (LOB data as an input parameter to the constructor function), specify the file name in the data section corresponding to the LOB column in the input data file.

   If you specify the absolute path name of this file, the size of the input data file increases, resulting in complicated coding.

   If all LOB input files are stored in one directory, you should specify the directory name in the `lobdata` statement and the file names in the input data file, in which case the utility combines the specified directory name and file name to obtain the absolute path name. Note that this directory name specification is valid only when the names of LOB input files are specified in the input data file; otherwise, the utility ignores the directory name, if specified.
4. If the target table contains no LOB column or the input parameter of the constructor function for creating values to be stored in the abstract data type is not LOB data, do not specify the `lobdata` statement.

(1) Format

```
lobdata [LOB-input-file-directory-name]
```

(2) Explanation

(a) LOB-input-file-directory-name

～ <path name>

Specifies the name of the directory that contains the LOB input files.

If the group of LOB input files is contained in a single directory, this operand specifies the absolute path name of that directory.

Specify the path name in such a manner that the absolute path name of the LOB input files is obtained when the specified path name is combined with the LOB column file name in the input data file specified in the `source` statement, or with the name of the LOB column input file specified in the `lobcolumn` statement.

Notes

1. The absolute path name created can be no greater than 1,023 bytes.
2. If you specify the absolute path name of the LOB input file in the input data file using the `source` statement or in the LOB column input file using the `lobcolumn` statement, you can omit the directory name of the LOB input files.

(3) Notes

The following table describes the data loading method depending on whether or not there is a LOB column, an abstract data type column with the LOB attribute, or a `lobdata` statement. This example uses SGMLTEXT type as the abstract data type column with the LOB attribute.

Table definition		lobdata statement	
		Specified	Not specified
With LOB column	With `SGMLTEXT` type columns	Data is loaded to the LOB column structure base table, `SGMLTEXT` type columns, and LOB columns at the same time.	Data is loaded to the LOB column structure base table and `SGMLTEXT` type columns at the same time. Data is then loaded to LOB columns.
	Without `SGMLTEXT` type columns	Data is loaded to the LOB column structure base table and LOB columns at the same time.	Data is first loaded to the LOB column structure base table and then to LOB columns.
With no LOB column	With `SGMLTEXT` type columns	Data is loaded to the abstract data type column structure base table and `SGMLTEXT` type columns at the same time.	
	Without `SGMLTEXT` type columns	Results in an error.	Data is loaded to the table without a LOB column or `SGMLTEXT` type column.

5.4.8 lobcolumn statement (specification of LOB column input file information)

When loading data to a table containing LOB columns, if you use LOB column input files, you can use the `lobcolumn` statement to specify the file information.

Criteria

Specify the `lobcolumn` statement to use a LOB column input file to load data to a table with LOB columns.

A LOB column input file is created, for example, when data is migrated from another database using a program.

Rules

1. You can specify the `lobcolumn` statement only when you also specify the `-k c` option.
2. If the table to be loaded contains multiple LOB columns, you need to create the `lobcolumn` statement for each column. However, when loading LOB columns by RDAREA, you need only one `lobcolumn` statement.

   If the input data file of the LOB column structure base table is in DAT format and not all the columns are specified in this file, but only the corresponding columns are specified in the column structure information file, then you can omit the `lobcolumn` statement for those LOB columns that are not specified. In this case, the `lobcolumn` statement, if specified, is ignored, and the `KFPL16325-W` message is issued. The system then creates the LOB specified data as null values or default values, depending on the LOB column specification made during the table definition.
3. There is no need to specify the `lobcolumn` statement when loading data only for the LOB column structure base table.
4. You cannot specify the `lobcolumn` statement for a table that contains columns of abstract data type.
5. If the target table contains no LOB column, do not specify the `lobcolumn` statement.

(1) Format

```
lobcolumn [LOB-column-name]
          LOB-column-input-filename [,LOB-column-input-filename]...
```

(2) Explanation

(a) LOB-column-name

Specifies the name of a LOB column in the table subject to data loading.

You can omit this operand if there is only one LOB column or the LOB columns are to be loaded in units of RDAREAs.

Rules

1. If the table contains multiple LOB columns, be sure to specify this operand.
2. The system treats a LOB column name enclosed in double quotation marks (") as case sensitive; otherwise, the system treats it as all uppercase letters. Enclose a

LOB column name in double quotation marks if it contains a space.

(b) LOB-column-input-filename [,LOB-column-input-filename]...

～ <pathname>

Specifies the names of the LOB column input files.

Specify each file name in such a manner that it results in the absolute pathname when combined with the directory name specified with the `lobdata` statement. If you omit the directory name in the `lobdata` statement, specify the absolute pathnames in this operand.

Rules

1. The maximum length of the file name for a LOB column input file is 1,023 characters.
2. When specifying multiple file names, make sure that each file name corresponds to the right column in the input data file for the LOB column structure base table.
3. Place the LOB column input files at the following location:

   HiRDB/Single Server

   Place the LOB column input files at the host of the single server. To specify the host name, use the `lobmid` statement.

   HiRDB/Parallel Server

   Place the LOB column input files at the front-end server or a back-end server. To specify the server name, use the `lobmid` statement.

5.4.9 lobmid statement (specification of LOB middle file information)

When loading data to a table containing LOB columns, you can use the `lomid` statement to specify information about the LOB middle files.

During the creation of a LOB column structure base table, the system outputs information needed for data loading to LOB columns to LOB middle files.

Criteria

When loading data to a table with LOB columns, if you specify the `-k d` option, be sure to specify the `lobmid` statement. However, the `lobmid` statement, if specified, is ignored for a LOB input parameter to a constructor function that generates values to be stored in the columns of abstract data type.

Rules

1. Specify the `lobmid` statement if you also specify the `-k f`, `-k c`, or `-k v` option.
2. The contents of a LOB middle file depend on the specification of the `-k`

option. If you are creating a LOB column structure base table separately from LOB data, you must specify the same `-k` option.

3. When creating the LOB column structure base table and LOB columns at the same time, the `source` statement has precedence; therefore, there is no need to specify the server name, host name, or error information file name. If specified, they are ignored.
4. If the target table contains no LOB column, do not specify the `lobmid` statement.

(1) Format

```
lobmid  [RDAREA-name][server-name|host-name]:
         LOB-middle-filename[,LOB-middle-filename]...
        [error=error-information-filename]
```

(2) Explanation

(a) RDAREA-name

～ <identifier> ((1-30))

Specifies the name of the user LOB RDAREA.

If you omit this operand, the system loads the entire contents of the specified LOB middle files.

The system treats an RDAREA name in double quotation marks as case sensitive; otherwise, the system treats it as all uppercase letters. Enclose an RDAREA name in double quotation marks if it contains a space.

(b) [server-name|host-name]

Specifies the name of the server or host where the LOB middle files are located.

server-name ～ <identifier> ((1-8))

HiRDB/Single Server

Do not specify this information for a HiRDB/Single Server.

HiRDB/Parallel Server

Specify the name of the front-end server or back-end server where the specified LOB middle files are located.

If you specified an RDAREA name, you can omit this operand (be sure to specify either the RDAREA name or the server name).

If the server name is omitted, the system assumes the name of the server where the specified RDAREA is located.

host-name ～ <identifier> ((1-32))

HiRDB/Single Server

Specify the name of the host where the specified LOB middle files are located.

This must be the host name of the single server.

You can omit this operand whether or not you specified an RDAREA name. If omitted, the system assumes the name of the host where the database load utility (`pdload` command) is executed.

HiRDB/Parallel Server

Do not specify this information for a HiRDB/Parallel Server.

(c) LOB-middle-filename

～ <pathname>

Specifies the absolute pathname of a LOB middle file.

During the creation of a LOB column structure base table, the system outputs the information required for loading LOB column data to this LOB middle file. During the data loading to LOB columns, the system retrieves necessary information from this LOB middle file.

(d) error=error-information-filename

～ <pathname>

Specifies the absolute pathname of the file to which error information is to be output.

This file must be located at the server machine or server where the LOB middle files are located.

If omitted, the database load utility creates a file with a unique name in the *HiRDB-installation-drive*`\tmp` directory. For details about the error information file, see *5.6 Error information file*.

(3) Notes about the LOB middle files

If you are creating a LOB column structure base table separately from LOB columns, note the following when creating the LOB columns:

1. If you created a LOB column structure base table by executing `pdload` more than once using the same input data files, you can specify all the LOB middle files output by `pdload` in the `lobmid` statement when creating the LOB columns. In this case, make sure that each specified LOB middle file corresponds to the right line number in the input data file for the LOB column structure base table.

   If the order of the specified LOB middle files is wrong, an error results because the system cannot establish a correspondence with the LOB column input files.

2. If you created a LOB column structure base table by executing `pdload` more than once using different input data files, you cannot specify more than one LOB middle file output by `pdload` in the `lobmid` statement at any one time when creating the LOB columns.

   If you specify more than one such LOB middle file at the same time, an error results because the system cannot establish a correspondence with the LOB column input files. Therefore, when creating the LOB columns, provide a LOB column input file corresponding to each input data file and execute `pdload` separately.

5.4.10 srcuoc statement (specification of UOC storage library information)

To use UOC to edit data and then store the data in a database, you can use an `srcuoc` statement to specify the UOC information. For details about UOCs, see section *5.10 Using a UOC to load a table*.

Criterion

Specify the `srcuoc` statement to execute data loading using a UOC.

(1) Format

```
srcuoc library-name entry=function-name [param='user-parameters']
```

(2) Explanation

(a) library-name

～ <pathname>

Specifies the absolute pathname of the dynamic link library where the UOC is stored.

(b) entry=function-name

Specifies the name of the function in the library that is to be called.

(c) param='user-parameters'

～ <character string> ((1-1,023 bytes))

Specifies user parameters to be passed to the UOC (because the user cannot use the command line to pass parameters directly to the UOC).

The character string specified in this operand is passed as is to the UOC via the interface area.

Do not include any space or tab in the user parameters character string.

5.4.11 array statement (specification of an array data format for a table containing repetition columns)

For a table containing repetition columns, you can use the `array` statement to specify the handling of the array data format and null values specified in the input data file.

Criteria

Specify the `array` statement to execute data loading on a table containing repetition columns.

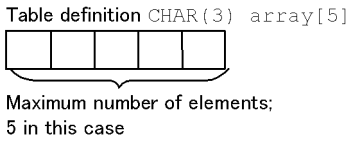
When this statement is omitted, the system assumes the FF format for array data in the DAT or fixed-size data format file or the VV format for array data in the binary format file. For the null value option, the system assumes `c`.

Rules

1. Specify this control statement for all repetition columns contained in the table. If you need to modify a specification on a column-by-column basis, specify this statement in a column structure information file or in a null value/function information file.
2. The following terminology is used in the explanation of data loading into a table containing a repetition column:

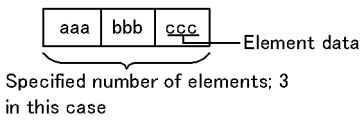
   Maximum number of elements:

   This is the number of elements specified for a given column during table definition.



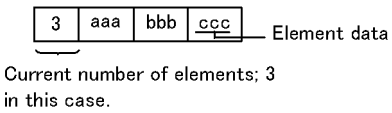
   Specified number of elements:

   This is the number of elements specified in the input data file from the maximum number of elements. You can specify this value in a column structure information file or in a null value/function information file.



   Current number of elements:

   This is the effective number of elements specified in the input data file.



(1) Format

```
array [elmtype=array-data-format] [nullset=null-value-option]
```

(2) Explanation

(a) elmtype=array-data-format

Specify either `ff`, `fv`, or `vv`.

When the `elmtype` operand is omitted, the array data takes the FF format for a DAT or fixed-size data format file; it takes the VV format for a binary-format file.

For details about the array data formats, see section *5.5.4 Arrayed data format for tables containing repetition columns*.

`ff`

This refers to a fixed field fixed element (FF) format.

For this format, always specify a fixed number of element data items without specifying a current number of elements (the effective number of elements specified in the input data file) in the input data.

Criterion

The FF format is useful for storing data in a repetition column without specifying the current number of elements in the input data.

Rules

1. The number of element data items specified in the input data must be no greater than the number of elements specified in the table definition.
2. You can specify the FF format for the following input data formats:
   - DAT format
   - Binary format
   - Fixed-size data format
3. The utility determines whether or not the specified data is to be stored in the database on the basis of whether the following conditions are met:
   - Whether specified data represents null values

- How null values are to be handled

`fv`

This refers to a fixed field variable element (FV) format.

For this format, specify the current number of elements along with the element data in the input data. This input data must contain as many element data items as there are elements specified in the table definition, column structure information file, or null value/function information file.

Criterion

Because the FV format allows the processing of data in fixed length, it facilitates the creation and modification of data by UAPs, but at the expense of an increase in file size.

Rules

1. You can specify the FV format for the following input data formats:
   - Binary format
   - Fixed-size data format
2. The FV format specifies input data in terms of a fixed-length data type. Therefore, for a variable-length data type (`VARCHAR`, `MVARCHAR`, or `NVARCHAR`), the valid data length must be specified at the beginning of each element data item. Following the valid data, the remaining portion of data must be padded with fill characters up to the length defined in the table, so that the input data always has the length defined for the table.

`vv`

This refers to a variable field variable element (VV) format.

For this format, specify the current number of elements along with the element data in the input data. The number of specified element data items must be equal to the current number of elements.

Criteria

Because the VV format requires the specification of a minimum length of data, the file size can be smaller than in the FV format. However, because it supplies data only in the current number of elements, this format makes the task of data modification by a UAP more complicated when compared with the FF or FV formats.

Rules

1. For a variable-length data type, specify data equal in length to the valid data length specified at the beginning of each element data item.

2. You can specify the VV format for the following input data formats:
   - DAT format
   - Binary format

You must specify the array data format described previously in one of the following files according to specification units:

Specification units		File to be specified	Description
Tables		Control information file	Specify with the `elmtype` operand of the `array` statement.
Columns	DAT or fixed-size format	Column structure information file	Specify with the `elmtype` operand.
	Binary format	Null value/function information file	Specify with the `elmtype` operand.

(b) nullset=null-value-option

～ <<c>>

With the FF array data format, this operand specifies how null value storage is to be handled if null values are specified in an element through the end of the specified elements.

When specified with a format other than the FF format, this option is ignored.

Specify one of the following values for the null value option:

`c`: Nothing is to be stored in a specified element.

`e`: Null values are to be stored in a specified element.

5.4.12 extdat statement (specification of information in the extended DAT format)

You can use the extdat statement to use the following extended functions with input data files in the DAT format:

When you specify the extdat statement, then, whether or not the operands are specified, you can change the enclosing character and use all extended functions other than capability to store decimal-point data in INTEGER or SMALLINT columns.

- Entering the null character (0x00), linefeed symbol (0x0a), and carriage-return symbol (0x0d0a) in character, national character string, and mixed character string data types
- Entering data with an enclosing character (") specified in the numeric, date interval, and time interval data types
- Deleting single-byte spaces and tabs from the input data section of columns of the numeric, date, date interval, time, and time interval data types
- Changing the enclosing character
- Entering the enclosing character itself (two consecutive enclosing characters are treated as one data item)
- Storing decimal-point data in INTEGER and SMALLINT columns

The enclosing character is the character specified in the enclose operand of the extdat statement. When this operand is omitted, the double quotation mark (") is used as the enclosing characters.

If the data immediately before or after a separator character is closed in the enclosing characters, the enclosing characters are not input. However, any enclosing character contained in the character string that is enclosed in the enclosing characters is input.

In the DAT format, the double quotation marks are used as the enclosing characters (you cannot change this enclosing character).

Criterion

Specify the extdat statement to use the above extended functions for a DAT-format input data file.

(1) Format

```
extdat [enclose=enclosing-character]
       [decin={yes|no}]
```

(2) Explanation

(a) enclose=enclosing-character

～ <character string> ((1 byte))

Specifies the enclosing character to be used to enclose each column value of the input data.

Rules

1. If data immediately before or after the separator character is enclosed by the enclosing characters, those enclosing characters are not input.
2. To treat an enclosing character as data, specify the enclosing character twice in succession. For example, if the single quotation mark (') is the enclosing character, specify two consecutive single quotation marks ('').
3. There are differences in the handling of enclosing characters between the DAT and extended DAT formats. For details about the entry of input data and handling of enclosing characters, see *5.5.1 DAT format*.
4. None of the following characters can be specified as the enclosing character:

   - Space, tab, asterisk (*)

   - Character used as the separator character
5. In the case of Shift JIS Kanji codes, the following characters are not suitable for the enclosing character because they may duplicate input data codes:

   - Characters used in the data in the input data file

   - Characters that can occur as Japanese input character codes

   ([, *,], ^, _, ', {, |, }, ~)
6. Double-byte codes cannot be specified, because the enclosing character must be a single-byte code.

(b) decin={yes|<u>no</u>}

Specifies whether or not data with a decimal point is to be stored in `INTEGER` and `SMALLINT` columns.

`yes`:

Store data with decimal point.

When `yes` is specified, the `extdat` statement treats all characters up to the first decimal point as data and ignores all subsequent characters; it does not check the format. The value that is actually stored is the integer with all decimal places discarded.

no:

Do not store data with decimal point.

When no is specified, the extdat statement treats any data with decimal point as erroneous data and does not store it in an INTEGER or SMALLINT column.

(3) Description of the extended functions

(a) Entering the null character (0x00), linefeed code (0x0a), and carriage-return code (0x0d0a) in character string data, national character string data, mixed character string data, or BINARY type

If the input data of the character string data, national character string data, mixed character string data, or BINARY type is enclosed in the enclosing character, the utility treats the null character (0x00), linefeed code (0x0a), and carriage-return code (0x0d0a) as data. If the input data is not enclosed in the enclosing character, the utility treats the null character, linefeed code, and carriage-return code as delimiters for data or the end of line.

For example, suppose CHAR(16) data is "California\0Napa". If you specify the extdat statement, California\0Napa is entered; otherwise, you cannot enter this data.

(b) Entering data with the enclosing character (") specified in the numeric, date interval, and time interval data types

You can enter data of the numeric, date interval, and time interval data types even when the data contains an enclosing character ("). In the DAT format, you cannot enter such data.

(c) Deleting single-byte spaces and tabs from the input data section of columns of the numeric, date, date interval, time, and time interval, and time stamp data types

The utility deletes all single-byte spaces and tabs from the input data section of columns of the numeric, date, date interval, time, and time interval, and time stamp data types. For details about the handling when the entire data is single-byte spaces or tabs, see *5.9 Null and default values during data loading*.

(d) Changing the enclosing character

The utility treats the character specified in the enclose operand as the enclosing character.

(e) Entering the enclosing character itself

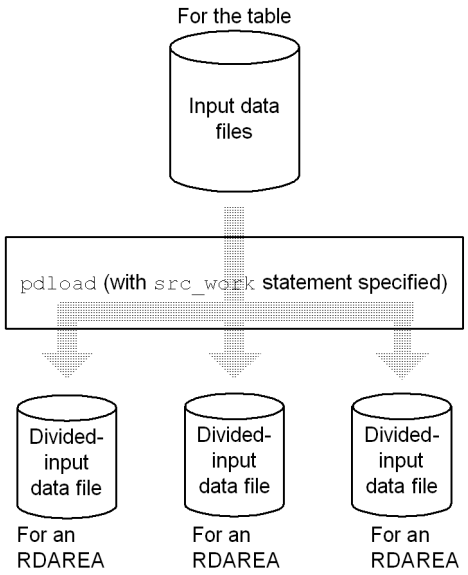
The utility treats any two consecutive enclosing characters as a single data item. This rule also applies to multi-byte codes. If a code assigned to a Gaiji character matches an enclosing character, the code may result in an input data error. In such a case, you must change the enclosing character.

5.4.13 src_work statement (specification of the output destination of divided-input data files)

When you perform data loading in units of RDAREAs, you can use the `src_work` statement to create from the table input data files a separate input data file for each RDAREA (divided-input data files). The `src_work` statement is specified in advance of data loading in order to create divided-input data files; you do not specify it during data loading.

Figure 5-10 shows an overview of creating divided-input data files.

Figure 5-10: Overview of creating divided-input data files



Criteria

Specify the `src_work` statement when you intend to perform data loading in parallel for the RDAREAs of a row-partitioned table, but you have not prepared an input data file for each RDAREA.

The `src_work` statement eliminates the need for the user to create an input data file for each RDAREA.

(1) Prerequisites

1. For the input data files, you can use DAT-format (including extended DAT format) and fixed-length data-format files. The divided-input data files are output in the same format.
2. To create divided-input data files, you must have the `INSERT` privilege for the

table.

3. The target table must be a row-partitioned table (there are no restrictions with respect to the partitioning method or column structure).
4. A divided-input data file must be on a single volume. If the file already exists, it is overwritten; a new file is not created.
5. The divided-input data files are created on the host that contains the input data files.
6. The utility performs some data checking during creation of divided-input data files; however, checking for the following data errors is not performed:
   - Invalid cluster key order
   - Key duplication error
   - Invalid LOB column storage data[1]
   - Invalid abstract-data type storage data[2]

   [1] One of the following:
   - The LOB input file is not accessible.
   - The LOB column's data length is greater than the length defined for the LOB column.

   [2] One of the following:
   - There is invalid data that results in an error during plug-in function data checking.
   - BLOB-type argument error (same as the errors for [1]).
7. Divided-input data files are created under the following names:
   - For a HiRDB/Single Server

     directory-name-specified-in-src_work-statement + *input-data-file-name* + *RDAREA-name*
   - For a HiRDB/Parallel Server

     directory-name-specified-in-src_work-statement + *input-data-file-name* + *server-name* + *RDAREA-name*

   You must ensure that the absolute path name plus file name do not exceed the maximum length supported by the OS.

   When multiple input data files are specified, *input-data-file-name* represents the first file name.

RDAREA-name represents the name of the table storage RDAREA.

8. `pdload` with the `src_work` statement specified does not access the target table (the target table is not locked).

(2) Format

```
src_work divided-input-data-file-output-target-directory
```

(3) Explanation

(a) divided-input-data-file-output-target-directory

Specifies the absolute path name of the directory to which the divided-input data files are to be output.

The specified directory must be located at the server or host that contains the input data files. Access privileges for the specified directory must have been granted to the HiRDB administrator.

(4) Notes

1. Names of the divided-input data files

   If `pdload` is executed using an input data file name that is duplicated for multiple tables defined in the same table storage RDAREA, the names of the divided-input data files will also be duplicated and the file contents will not be reliable. For this reason, you must ensure that *directory-name-specified-in-src_work-statement* or *input-data-file-name* is unique.

2. Table containing LOB columns

   Creation of divided-input data files is simply a matter of subdividing the input data files; it does not involve accessing any LOB input files. In the case of a table with a LOB column, you must place the LOB input file at a location that can be referenced from the host where `pdload` is executed.

3. Handling of `DECIMAL`-type data in fixed-size data format input data files

   When the facility for conversion to a decimal signed normalized number is used, the `DECIMAL`-type values that are output to the divided-input data files are normalized.

4. If there is no data to be stored in a particular RDAREA, the utility creates a divided-input data file whose length is 0 bytes.

(5) Examples

(a) Example of creating divided-input data files and performing data loading in units of RDAREAs

This example performs parallel data loading into a row-partitioned table (T1) in units of RDAREAs. Because an input data file (c:\hirdb\pdload\inputfile) is available for the table, the example creates divided-input data files in order to execute data loading in units of RDAREAs. The example then shows execution of parallel data loading in units of RDAREAs.

1. Defining the table

```
CREATE FIX TABLE T1 (C1 DEC,
                     C2 CHAR(10))
                    IN ((RDAREA1) C1 > 1000,(RDAREA2) C1 < -1000,(RDAREA3));
```

2. Creating divided-input data files

- pdload command

```
pdload T1 c:\hirdb\pdload\control_file
```

Explanation

T1: Name of the target table

c:\hirdb\pdload\control_file: Name of the control information file

- Contents of the control information file (c:\hirdb\pdload\control_file)

```
 source FES:/hirdb/inputfilec:\hirdb\pdload\inputfile error=/hirdb/
errfilec:\hirdb\pdload\errfile    [1]
 src_work /divworkc:\hirdb\divwork    [2]
```

Explanation

1. Specifies the names of the input data file and error information file.
2. Specifies the name of the directory where the divided-input data files are to be created.

3. Executing parallel data loading in units of RDAREAs

As a result of *2. Creating divided-input data files*, the following three divided-input data file are created:

- c:\hirdb\divwork\inputfile_BES1_RDAREA1
- c:\hirdb\divwork\inputfile_BES2_RDAREA2
- c:\hirdb\divwork\inputfile_BES3_RDAREA3

These files are used as the input data files in executing parallel data loading in units of RDAREAs.

(b) Example of using an input data file in the all-column fixed-length binary format

Binary-format input data files cannot be used to create divided-input data files. However, if all columns in the table are of the fixed-length data type,* the table can be handled as if it were in the fixed-size data format.

* Applicable data types are as follows:

`INTEGER`, `SMALLINT`, `DECIMAL`, `FLOAT`, `SMALLFLT`, `DATE`, `TIME`, `INTERVAL YEAR TO DAY`, `INTERVAL HOUR TO SECOND`, `CHAR`, `NCHAR`, `MCHAR`, `TIMESTAMP`

1. Defining the table

```
CREATE FIX TABLE T2 (C1 DEC(7),
                     C2 CHAR(10)) FIX HASH HSAH6 BY C1
                    IN (RDAREA1,RDAREA2);
```

2. Creating divided-input data files

- `pdload` command

```
pdload -a -c c:\hirdb\pdload\column_file T2 c:\hirdb\pdload\control_file
```

Explanation

`-a`: Specifies that the input data file is in the fixed-size data format

`T2`: Specifies the target table

`c:\hirdb\pdload\column_file`: Name of the column structure information file

`c:\hirdb\pdload\control_file`: Name of the control information file

- Contents of the column structure information file (`c:\hirdb\pdload\column_file`)

```
 C1,type=dec(7)
 C2,type=char(10)
```

Explanation

Specifies the table component columns C1 and C2 and their data types.

- Name of the control information file (c:\hirdb\pdload\control_file)

```
 source FES:/hirdb/inputfilec:\hirdb\pdload\inputfile error=/hirdb/
errfilec:\hirdb\pdload\errfile    [1]
 src_work /divworkc:\hirdb\divwork    [2]
```

Explanation

1. Specifies the names of the input data file and error information file.
2. Specifies the name of the directory where the divided-input data files are to be created.

3. Executing parallel data loading in units of RDAREAs

As a result of *2. Creating divided-input data files*, the following two divided-input data file are created:

- c:\hirdb\divwork\inputfile_BES1_RDAREA1
- c:\hirdb\divwork\inputfile_BES2_RDAREA2

These files are used as the input data files in executing parallel data loading in units of RDAREAs.

5.4.14 constraint statement (specification of check pending status)

The `constraint` statement specifies settings for check pending status.

When you perform data loading on a table for which check constraints or referential constraints have been defined, `pdload` may store data whose constraint integrity cannot be guaranteed because the utility does not check data integrity. To restrict data manipulation on a table until its integrity has been verified by `pdconstck`, the `constraint` statement places a table that is subject to data loading and referencing tables related to such a table in check pending status.

You can also specify a control statement that will prevent a table's check pending status from being changed. For details about check pending status, see the manual *HiRDB Version 8 Installation and Design Guide*.

Criteria

If either of the following is applicable, you should specify the `constraint` statement:

- When data loading is performed on a table for which check constraints or referential constraints have been defined, the table's check pending status should not be changed (set).
- When data loading is performed on a referenced table, the check pending status of a referencing table related to that referenced table should not be changed (set).

Rules

1. This control statement is applicable only when the value of the `pd_check_pending` operand in the system definition is `USE`. If any of the following is applicable, the `constraint` statement is ignored and the table's check pending status is not changed:

   • The `pd_check_pending` operand's value in the system definition is `NOUSE`.

   • No constraint has been defined for the table.

   • A divided-input data file is to be created (the `src_work` statement is specified).

(1) Format

```
constraint [pending=no]
          [ref_pending=no]
```

(2) Explanation

(a) pending=no

Specifies that when data loading is to be performed on a referencing table or check constraint table, the table's check pending status is not to be changed. For details about the default value that is assumed when this operand is omitted, see *(4) Range of check pending status settings*.

Criteria

Specify `pending=no` when either of the following is applicable:

- When data loading is performed on a table for which a referencing table or check constraints have been defined, the integrity of the data that is to be stored in columns has been guaranteed by prior user checking.
- Data loading is performed with a data count of 0 in order to re-create the indexes for a table for which a referencing table or check constraints have been defined.

(b) ref_pending=no

Specifies that when data loading is to be performed on a referenced table, the check pending status of a referencing table related to the referenced table is not to be changed. For details about the default value that is assumed when this operand is omitted, see *(4) Range of check pending status settings*.

Criteria

Specify `ref_pending=no` when either of the following is applicable:

- When data loading is performed on a referenced table in the creation mode, the integrity of the data in the referenced table and referencing table has been guaranteed by prior user checking.
- Data loading is performed with a data count of 0 in order to re-create the indexes for a referenced table.

(3) Notes

1. For notes about setting related referencing tables in check pending status when data loading is performed on a referenced table, see *C. RDAREA Status During Command Execution*.
2. When check pending status is set, the related resources are locked. Once setting of the check pending status has been completed, the resources are unlocked. For details about the lock, see *B.2 Lock mode for utilities*.

(4) Range of check pending status settings

Table 5-3 shows the range of check pending status settings for check constraints. Table 5-4 shows the range of check pending status settings for referential constraints.

Table 5-3: Range of check pending status settings for check constraints

Data loading execution condition		Check pending status for check constraint		
pd_check_pending operand value in system definition	Unit of execution	Data dictionary table		Table information in RDAREA
		SQL_TABLES table CHECK_PEND2 column	SQL_CHECKS table CHECK_PEND2 column	Check constraint status
USE	Table	C	C	T
	RDAREA	C	C	D
NOUSE	Table	N	N	N
	RDAREA	N	N	N

Legend:

T: Sets table information (check constraint status) in all RDAREAs that store the table.

C: Sets the check pending status.

D: Sets table information (check constraint status) in the RDAREA subject to data loading.

N: Does not change the check pending status (the current status is maintained).

Table 5-4: Range of check pending status settings for referential constraints

Data loading execution condition			Table subject to data loading					
pd_check_pend operand value in system define	-d option spec	Unit of exe	Referenced table			Referencing table		
			Check pending status of the referencing table related to referenced table			Check pending status of referencing table		
			Data dictionary table		TIRDA	Data dictionary table		TIRDA
			SQLTC	SQLRC	REFCS	SQLTC	SQLRC	REFCS
USE	Spec'd	Table	C	C	T	C	C	T
		RDAREA	C	C	T	C	C	D
	Omitted	Table	N	N	N	C	C	T
		RDAREA	N	N	N	C	C	D
NOUSE	Spec'd	Table	N	N	N	N	N	N
		RDAREA	N	N	N	N	N	N
	Omitted	Table	N	N	N	N	N	N
		RDAREA	N	N	N	N	N	N

Legend:

T: Sets table information (check constraint status) in all RDAREAs that store the table.

C: Sets the check pending status.

D: Sets table information (check constraint status) in the RDAREA subject to data loading.

N: Does not change the check pending status (the current status is maintained).

pd_check_pend operand value in system define: pd_check_pending operand value in system definition

-d option spec: -d option specification

Spec'd: Specified

Unit of exe: Unit of execution

TIRDA: Table information in RDAREA

SQLTC: SQL_TABLES table CHECK_PEND column

SQLRC: SQL_REFERENTIAL_CONSTRAINS table CHECK_PEND column

REFCS: Referential constraint status

(5) Example of whether or not check pending status can be set

This subsection describes whether or not check pending status can be set for tables `T1` through `T5` when the `pending` and `ref_pending` operands are specified in the `constraint` statement.

Description of the tables

`T1`: Table with the primary key (referenced table for `T2`)

`T2`: Table with the primary key and the foreign key that references `T1`'s primary key (referencing table for `T1` and referenced table for `T3`)

`T3`: Table with the foreign key that references `T2`'s primary key (referencing table for `T2`)

`T4`: Table for which check constraints have been defined

`T5`: Table with no constraint definition

Table 5-5 shows the table's check pending status when data loading is performed on tables `T1` through `T5`.

Table 5-5: Table's check pending status when data loading is performed

Table subject to data loading and whether or not there are constraint definitions				Operand specification in the constraint statement		Check pending status for table				
Table name	Referential constraint		Check constraint	pend	ref_ pend	T 1	T 2	T 3	T 4	T 5
	Referenced table	Referencing table								
`T1`	Y	N	N	Omit'd	Omit'd	●	P	●	●	●
					Spec'd	●	N	●	●	●
				Spec'd	Omit'd	●	P	●	●	●
					Spec'd	●	N	●	●	●

Table subject to data loading and whether or not there are constraint definitions				Operand specification in the constraint statement		Check pending status for table				
Table name	Referential constraint		Check constraint	pend	ref_ pend	T1	T2	T3	T4	T5
	Referenced table	Referencing table								
T2	Y	Y	N	Omit'd	Omit'd	●	P	P	●	●
					Spec'd	●	P	N	●	●
				Spec'd	Omit'd	●	N	P	●	●
					Spec'd	●	N	N	●	●
T3	N	Y	N	Omit'd	Omit'd	●	●	P	●	●
					Spec'd	●	●	P	●	●
				Spec'd	Omit'd	●	●	N	●	●
					Spec'd	●	●	N	●	●
T4	N	N	Y	Omit'd	Omit'd	●	●	●	P	●
					Spec'd	●	●	●	P	●
				Spec'd	Omit'd	●	●	●	N	●
					Spec'd	●	●	●	N	●
T5	N	N	N	Omit'd	Omit'd	●	●	●	●	●
					Spec'd	●	●	●	●	●
				Spec'd	Omit'd	●	●	●	●	●
					Spec'd	●	●	●	●	●

Legend:

pend: pending

ref_ pend: ref_pending

Y: Defined

N: Not defined

Omit'd: Omitted

Spec'd: Specified

P: Check pending status is set.

N: Check pending status is not changed (the current status is maintained).

•: Not applicable (`pending` and `ref_pending` operands are ignored).

(6) Timing of setting check pending status

The timing of setting check pending status for a target table subject to data loading and for referencing tables related to that target table depends on the data loading mode. Table 5-6 shows the relationship between the data loading mode and when check pending status is set.

Table 5-6: Check pending status setting timing

Data loading mode	When check pending status is set
Creation mode	Before data deletion processing begins
Addition mode	Before data loading begins

If `pdload` rolls back due to an error after it has set the table's check pending status, the table is still placed in check pending status because setting of the check pending status has been completed (commit is completed).

5.4.15 option statement (specification of data processing information)

The `option` statement specifies the optional data processing functions for data loading.

Criteria

Specify the `option` statement to use any of the following optional functions:

- Specification of the space conversion level (`spacelvl` operand)
- Change to the percentage of free space during data loading (`tblfree` operand)
- Change to the percentage of free space during index creation (`idxfree` operand)
- Data loading with the synchronization point specification (`job` operand)
- Output of warning messages in the event of data truncation (`cutdtmsg` operand)
- Execution of `NOWAIT` search on a table subject to data loading (`nowait` operand)
- Specification of LOB columns or LOB parameters in the binary-format input data file generated by `pdrorg` (`bloblimit` operand)

- Monitoring of pdload execution time (exectime operand)
- Method for storing data when data is loaded into a table with the DEFAULT clause specified and the input data is the null value (null_string operand)
- Specification for ignoring the storage processing up to the point where an input data error is detected (dataerr operand)
- Specification for detecting an input data error when the input data length in the DAT-format input data file is greater than the defined column length (lengover operand)
- Suppression of error data information (divermsg operand)

(1) Format

```
option [spacelvl = {0|1|3}]
       [tblfree = {percentage-of-unused-space
       |([percentage-of-unused-space]
          ,percentage-of-free-pages-in-segment)}]
       [idxfree = percentage-of-unused-area]
       [job = [job-name], [rows-count-between-synchronization-points]
              [,CLR]]
       [cutdtmsg = {on|off}]
       [nowait = {yes|no}]
       [bloblimit = area-allocation-size]
       [exectime = pdload-execution-monitoring-interval]
       [null_string = {null | default}]
       [dataerr=rollback]
       [lengover=err]
       [divermsg=off]
```

(2) Explanation

(a) spacelvl = {0|1|3}

Specifies whether or not to execute space conversion on the input data.

Available space conversion levels are 0, 1, and 3.

When the spacelvl operand is omitted, the system executes space conversion according to the value of the pd_space_level operand in the system common definitions.

0

The system does not execute space conversion on the input data.

1 or 3

The system executes space conversion on the input data. Values of 1 and 3 have the same effect.

If the table columns subject to data loading are of the national or mixed character string type, the system converts the input data and stores the data in the database as follows:

- National character string type

  The system converts two consecutive single-byte spaces in the input data to one double-byte space, in units of two bytes from the top.

- Mixed character string type

  The system converts each double-byte space in the input data to two consecutive single-byte spaces.

  When the character codes are utf-8, the system converts one double-byte space (3 bytes) to two single-byte spaces. For `MCHAR`, the system adds trailing single-byte spaces up to the definition length. For `MVARCHAR`, the data length remains shortened.

If the data is to be stored via a constructor function for the columns of abstract data type and the data for the corresponding constructor function's argument has the national or mixed character string type, the system executes space conversion before passing the input data as an argument to the constructor function.

Criteria

Specify this operand to make all spaces in the table data either single- or double-byte characters.

For details about how to make all spaces in table data either single- or double-byte characters, see the *HiRDB Version 8 System Operation Guide*.

Note

When executing space conversion using the `spacelvl` operand or the system common definitions, note the following:

1. The system checks the order of cluster key values after the data conversion; therefore, the cluster key sequence checking may result in an error even if the data was sorted by the cluster key values before conversion. In this case, specify the `-x` option to skip the cluster key sequence checking. To store data in the order of cluster key values, reorganize the data after data loading is completed.
2. If the column used as the unique key index or primary key index is subject to space conversion, duplicate key values may result in the index. In this case, take one of the following actions:

   • Unique key index: Either remove the unique index specification beforehand or modify the corresponding data in the input data.

   • Primary key index: Modify the corresponding data in the input data.

3. A comparison value for the null value specified in the column structure information file or null value/function information file is not subject to space conversion. In this case, the comparison value for the null value is compared to the input data before space conversion.
4. The length of the input data before space conversion is used for checking the input data length.

(b) tblfree = {percentage-of-unused-space|([percentage-of-unused-space], percentage-of-free-pages-in-segment)}

Specify this operand to change the percentage of free space specified with `CREATE TABLE` (value of `PCTFREE`) for storing data during data loading.

percentage-of-unused-space

You can specify a value in the range from 0 to 99.

percentage-of-free-pages-in-segment

You can specify a value in the range of 0 to 50.

Criteria

If you have specified a non-zero value as the percentage of free space during table definition, a shortage of RDAREA may occur while data is stored because the defined percentage of free space takes effect during data loading. If you specify the `tblfree` operand in this case, you can complete the data loading without having to extend the RDAREA temporarily.

Note

It is more advantageous to specify this operand than the `-y` option to store data in free space for the following reasons:

- Batch input/output operations take effect even when data is stored in free space.
- You can specify `p` in the `-l` option.

(c) idxfree = percentage-of-unused-area

If you are creating indexes in batch index creation mode, specify this operand to change the percentage of unused area specified with `CREATE INDEX` (value of `PCTFREE`).

The permitted value range is from 0 to 99.

Criteria

If you have specified a non-zero value as the percentage of unused space during index definition, a shortage of RDAREA may occur while indexes are stored because the defined percentage of free area takes effect during batch index

creation processing. If you specify the `idxfree` operand in this case, you can complete the batch index creation processing without having to extend the RDAREA temporarily.

(d) job = [job-name], [rows-count-between-synchronization-points] [, CLR]

Specify this operand to execute data loading with the synchronization point specification.

Data loading with the synchronization point specification is a method of data loading in which a transaction is settled each time the specified number of data items are stored. If an error occurs while data is stored, this method enables you to restore the database in a short period of time without having to store data from the beginning.

job-name ～ <alphanumerics> ((1-3))

Specify the name of the job that executes data loading with the synchronization point specification.

This job name is used during re-execution in the event the utility terminates abnormally during data loading. If this job name is the same as another `pdrorg` job name, re-execution results in a malfunction. Therefore, be sure to specify a unique job name.

rows-count-between-synchronization-points ～ ((1-1000)) <<100>>

Specify the number of data items as an interval at which a transaction is to be settled. For example, a value of 100 sets a synchronization point at every one million data items.

A small value settles a transaction after a small number of data items; therefore, rollback processing would take a short time in the event of abnormal termination. However, transaction generation and settlement and process restart occur frequently, resulting in a large overhead. On the other hand, a large value has a little effect on the performance, but the error recovery time would be long.

An error results if you specify only the number of rows between synchronization points without specifying a job name.

`CLR`

Specify this operand if there is no need to re-execute `pdload` or if `pdload` is to be re-executed after clearing the current synchronization point information in the data base.

If the utility terminates abnormally, the synchronization point information remains in the database. If this information is retained during the next execution, the utility may not function correctly.

Note that if you specify `CLR`, the system clears all synchronization point information, whether it was specified by another user or utility.

To determine if the synchronization point information is present in the database, use the database condition analysis utility (`pddbst`) to execute condition analysis in units of tables.

Criterion

This operand is useful for loading a large amount of data that will take a considerable amount of time to store the data. However, this has adverse effects on performance due to synchronization point processing. Additionally, more data pages are required than in normal data loading because data storage begins on a new page at each synchronization point.

Example of job operand specification

- Loading data with a synchronization point specified for every one million data items

  `job=JOB,100` or `job=JOB`

- Clearing the current synchronization point information and specifying new synchronization points at every one million data items to execute data loading

  `job=JOB,100,CLR` or `job=JOB,,CLR`

- Clearing the current synchronization point information and executing normal data loading

  `job=,,CLR`

Notes

1. Synchronization point information is managed by the job name. Suppose that `pdload(A)` that was executing data loading with the synchronization point specification terminates abnormally. If you execute `pdrorg(B)` specifying the same job name for the same table, `pdrorg(B)` uses the synchronization point information that was left by `pdload(A)`. Therefore, the job names must be controlled so that they are not duplicated.
2. If data loading with the synchronization point specification terminates abnormally, you cannot execute the normal data loading (without the synchronization point specification) on the same table until you re-execute the data loading with the synchronization point specification having the same job name.
3. The synchronization point information is managed in units of RDAREAs. For a row-partitioned table, inconsistency occurs on the synchronization point information for the entire table in the following cases:
   - Only some of the RDAREAs were re-initialized.
   - Error occurred during data loading with the synchronization point

specification in units of RDAREAs.

If the job name is the same, but the number of lines does not match, you cannot use the synchronization point information managed by each RDAREA for re-execution. If some of the RDAREAs have synchronization point information while the others do not, you can re-execute data loading using the retained synchronization point information. The following shows how the system handles data loading when the job name for synchronization point information is the same but the number of lines does not match:

Retained synchronization point information (number of lines between synchronization points)			Input begin line for data loading with synchronization point specification in units of tables
RDAREA 1	**RDAREA 2**	**RDAREA 3**	
0	0	0	Item 0
0	*n*	0	Item *n*
n	0	*m*	Cannot be executed.
n	*m*	*p*	Cannot be executed.

0: No synchronization point information retained

n, *m*, *p*: Retained number of lines between synchronization points:

4. If data loading with the synchronization point specification terminates abnormally, you cannot modify the index creation method during re-execution (`-i` option).
5. If data loading with the synchronization point specification terminates abnormally, and if you re-initialize RDAREAs or execute the `PURGE TABLE` statement before re-execution, the existing synchronization point information is deleted.
6. If data loading with the synchronization point specification terminates abnormally during the batch creation of a plug-in index, and the corresponding plug-in index supports the batch plug-in index creation partial recovery facility, execute `pdrorg` to create the plug-in index that resulted in an error, then re-execute `pdload`. If you re-execute `pdload` without executing `pdrorg`, the entire plug-in index is re-created.
7. The maximum number of data items that can be controlled during data loading with the synchronization point specification is 4,294,960,000. To load more data items than this, use multiple input data files and execute `pdload` more than once.

(e) cutdtmsg = {on|off}

If you are storing character string data from the DAT-format input data file into the columns with the following data types during data loading, and the input data is longer than the defined column length, the excess part of the data is discarded:

- Character string data type (CHAR, VARCHAR)
- National character string data type (NCHAR, NVARCHAR)
- Mixed character string data type (MCHAR, MVARCHAR)
- BINARY data type

This operand specifies whether or not to output a warning message (KFPL31090-W) to the error information file when this data truncation occurs.

on: Output warning messages.

off: Do not output warning messages.

Rules

1. If the -e option is specified, the system assumes cutdtmsg=off.
2. For an input data file other than in the DAT format, this operand, if specified, is ignored.
3. If cutdtmsg=on is specified and data truncation occurs, pdload returns code 4, even when all data has been stored.
4. A warning message is output for each data item resulting in truncation (each column and element), not for each line. Therefore, if you are specifying cutdtmsg=on, check the disk space, because many warning messages may be output.
5. The system outputs warning messages to the error information file, but it does not output the corresponding input data to the error data file.

(f) nowait = {yes|no}

Specifies whether or not a NOWAIT search is to be conducted on the table subject to data loading. The NOWAIT search is an SQL function for executing a search process with the NOWAIT lock option specified.

yes: Execute NOWAIT search.

no: Do not execute NOWAIT search.

Rules for nowait=yes

1. You cannot specify the batch index creation mode (-i c). Specify the index update mode (-i s).
2. You cannot use the local buffer (-n option).

3. The system does not support data loading in units of RDAREAs.
4. In the creation mode (-d option specified), you cannot execute a NOWAIT search on the table while data is being deleted (if executed, processing is placed in lock-release wait status). You can execute a NOWAIT search while data is being stored.
5. For a table with columns of abstract data type provided by a plug-in, you can execute a NOWAIT search if the plug-in provides the no-lock search facility. To see if the plug-in provides the no-lock search facility, see the applicable plug-in manual.
6. When loading data only to the LOB column of a table, you can reference the LOB column structure base table but not the LOB columns, regardless of this operand specification. To reference the LOB columns, execute data loading on both the LOB column structure base table and LOB columns at the same time.
7. In the event of a pdload rollback due to abnormal termination, segments will not be released once they are allocated during data loading. These segments are reused by pdload during re-execution, but not by a UAP's INSERT or UPDATE statement. Therefore, if you execute a UAP's INSERT or UPDATE statement prior to re-execution of pdload, the segments that are not reusable by pdload or UAP will be retained until the corresponding RDAREA runs out of all unused segments. To release such segments, you need to reorganize the table with pdrorg. To avoid this, place the RDAREA in reference-possible shutdown status, then execute pdload. (No UAP will be able to access the table even if pdload terminates abnormally.)
8. If specified, yes is ignored for a shared table (no is assumed).

(g) bloblimit = area-allocation-size

～ <unsigned integer> ((1-2,097,152)) <<1024>>

If you are executing the following type of data conversion using a pdrorg-generated binary-format input data file, you may need to retain data in memory:

- Converting data for BLOB parameter with abstract data type to a BLOB-type column
- Converting BLOB data to an abstract data type column with BLOB parameter

In this case, specify the size of the memory area to retain data in KB. The system compares this operand value with the BLOB parameter or the defined length for BLOB type and allocates the area based on the smaller size.

Criteria

Specify this operand if either one of the following conditions is true:

- When converting data for a BLOB parameter with abstract data type to a BLOB-type column, the data for the BLOB parameter is followed by the data to be stored in the abstract data type column with BLOB parameter.
- When converting BLOB data to an abstract data type column with BLOB parameter, the BLOB data is preceded by the data to be stored in the BLOB-type column.

Rules

1. This operand takes effect only when the -W option is specified.
2. If there is a data item that exceeds the specified size, an error results.

(h) exectime=pdload-execution-monitoring-interval

～ <unsigned integer> ((1-35791394)) <<0>>

Specifies an interval in minutes for monitoring the pdload execution time. If this operand is omitted, pdload does not monitor the execution time. If pdload does not terminate within the specified interval, the system terminates the pdload process forcibly and acquires error information in order to determine the cause of the no-response.

This operand's value takes precedence over the value of the pd_utl_exec_time operand in the system definition.

Criterion

If you specify a monitoring interval for processing such as nighttime batch processing, the processing will terminate abnormally if an error such as a communication error (including a transient error) or a disk error occurs during execution of pdload and pdload is placed in no-response status. This enables you to detect the abnormality and start recovery processing at an early stage.

Guidelines for the specification value

The purpose of this operand is to detect a no-response error, not to monitor the execution time of a long-running transaction. Therefore, the operand value must be large enough for the applicable table processing. For example, to monitor the execution time of a pdload that should terminate within 7- 8 minutes, specify exectime=20, not exectime=10. When you are executing data loading in the addition mode, specify a larger value because the execution time will vary depending on the number of base data items even when there is a predefined number of data items in the input data file (in the case of a table whose data increases monotonously, you should check and, if necessary, revise the specified value).

(i) null_string={null|default}

Specifies whether the default value set in the DEFAULT clause or the null value is to be

stored when the input data is the null value ("*" or omitted) during data loading on a table with the DEFAULT clause specified.

This operand is applicable to a DAT-format input data file. An error results if this operand is specified for an input data file in the binary, fixed-length data, or pdrorg-generated binary format.

null: Store the null value.

default: Store the default value set in the DEFAULT clause.

(j) dataerr=rollback

Specifies that data storage processing (rollback) is to be ignored when an input data error (logical error) is detected.

Table 5-7 shows the relationship between the -e and dataerr operands. Table 5-8 provides notes about specifying the dataerr operand together with other options and control statements.

Table 5-7: Relationship between the -e and dataerr operands

-e option	dataerr operand in the option statement	pdload operation when input data error is detected	Transaction settlement method	pdload's return code
Omitted	Omitted	Skips database storage processing for the erroneous data and resumes processing.	Commit	4
	Specified	Cancels processing when the erroneous data is detected.	Rollback	8
Specified	Omitted	Cancels processing when the erroneous data is detected.	Commit	4
	Specified	Cancels processing when the erroneous data is detected.	Rollback	8

Table 5-8: Notes about specifying the dataerr operand together with other operands and control statements

Option and control statement	Note about specifying together with the dataerr operand
-d	A control statement error occurs.
-l	Specify either a or p.

Option and control statement	Note about specifying together with the dataerr operand
-i	If all the following conditions are satisfied, index unfinished status results after rollback; therefore, specifying this option together with the dataerr operand results in a control statement error: • Loading of data is in units of RDAREAs • There is a non-partitioning key index • The value of the -i option is not s Consider changing the pdload options, control statements, or index definition so that not all the above conditions are satisfied.
	If a key duplication error may occur during data loading, specify s. If any other value is specified, data stored in the table will not be rolled back even if a key duplication error is detected.
-k	If f, v, or c is specified, the unneeded LOB middle file remains after rollback. Before re-executing pdload, delete this LOB middle file.
	If f, v, or c is specified and an error is detected during data loading on a LOB column, the LOB column structure base table is committed and the LOB column is rolled back, regardless of the dataerr operand specification. In this case, correct the LOB data and re-execute data loading.
-e	Regardless of the option specification, processing is cancelled and then rolled back when erroneous data is detected.
Other option	None
job operand in the option statement	A control statement error occurs.
RDAREA name in the source statement	If all the following conditions are satisfied, the index unfinished status results after rollback; therefore, specifying this option together with the dataerr operand results in a control statement error: • Loading of data is in units of RDAREAs • There is a non-partitioning key index • The value of the -i option is not s Consider changing the pdload options, control statements, or index definition so that not all the above conditions are satisfied.
error operand in the source statement	Information about the detected error is output and then processing is rolled back.
errdata operand in the source statement	A control statement error occurs.
Other control statements	None

(k) lengover=err

Specifies that an input data error is to be detected when the input data in a DAT-format

(including extended DAT format) input data file that is to be stored in any of the following data-type columns is longer than the defined column length:

- CHAR
- VARCHAR
- NCHAR
- NVARCHAR
- MCHAR
- MVARCHAR
- BINARY

Table 5-9 shows the relationships among the input data length, defined column length, and the lengover operand. Table 5-10 provides notes about specifying the lengover operand together with other options and control statements.

Table 5-9: Relationships among input data length, defined column length, and lengover operand

Relationship between input data length and defined column length	lengover omitted	lengover specified
m=*n*	Stores the input data as is.	
m<*n*	Discards the trailing end of the input data beginning at location *m* + 1 and then stores the remaining part of the input data.	Sets an input data error.
m>*n*	CHAR, NCHAR, and MCHAR: Pads the trailing end of the input data with spaces beginning at location *n* + 1 and then stores the input data. VARCHAR, NVARCHAR, MVARCHAR, and BINARY: Stores the input data as is.	

Table 5-10: Notes about specifying the lengover operand together with other operands and control statements

Option and control statement	Note about specifying together with the lengover operand
-a, -b	A control statement error occurs because the lengover operand is applicable to the DAT format.
Other option	None

Option and control statement	Note about specifying together with the lengover operand
cutdtmsg operand in the option statement	The KFPL31090-E message is displayed even when the cutdtmsg operand is not specified.
dataerr operand in the option statement	An input data error subject to rollback is detected.
extdat statement	The extended DAT-format input data files are also supported.
error operand in the source statement	The number of times the KFPL31090-W/E message is displayed changes as follows: When the lengover operand is specified: The message is displayed only for the first column in which an invalid data length is detected. When the lengover operand is omitted: The message is displayed for all columns in which an invalid data length is detected.
errdata operand in the source statement	The applicable data is output to the error data file.
Other control statements	None

(l) divermsg=off

Specifies suppression of a type of error data information. When data loading is performed in units of RDAREAs on a table whose partition storage condition have been changed, and the input data contains row data that does not satisfy the RDAREA storage conditions, the utility displays error data information. This operand is used to suppress output of this error data information.

This operand is applicable if the table subject to data loading is a row-partitioned table and data loading is performed in units of RDAREAs. If this operand is specified under any other conditions, it is ignored.

Table 5-11 shows the relationship between the divermsg operand and each item when the input data contains row data that does not satisfy the partition storage conditions.

Table 5-11: Relationship between the divermsg operand and each item when input data contains row data that does not satisfy the partition storage conditions

Item	divermsg operand	
	Specified	Omitted
pdload's return code	0*	4
Error information file	KFPL31009-E message is not output.	KFPL31009-E message is output.

Item	divermsg operand	
	Specified	Omitted
Error data file	Error data is not output.	Error data is output.
`-e` option	Disabled	Enabled

* In the event of a data format error, the return code is 4.

5.4.16 report statement (specification of a file to which pdload tuning information is output)

The `report` statement specifies a file to which the tuning information is to be output during the execution of `pdload`.

Criterion

Specify the `report` statement if you want to output the tuning information.

Rules

1. You can specify only one `report` statement.
2. The system does not output the tuning information unless the `report` statement is specified.

(1) Format

```
report file=process-results-filename
```

(2) Explanation

(a) process-results-filename

～ <pathname>

Specifies the absolute pathname of the file to which the tuning information is to be output.

This file must be located at the host where `pdload` is executed.

(3) Output format

The following shows the tuning information to be output:

```
pdload VV-RR(Object Option) *** DB LOAD *** 2003-03-31 11:11:27 ..................1

*** statistics report ***

execute time :   0: 1:52 ..........................................................2
prepare time :   0: 0: 0 ..........................................................3

*** shell/server information ***

name       cpu svrup(first) reciv  wait  sort  send delet dtlod ixlod dtins ......4
-------------------------------------------------------------------------------
pdload       0     0(    0) ***** ***** ***** ***** ***** ***** ***** *****
pdloadm      5     4(    4)    28 ***** ***** 5.76k ***** ***** ***** *****
bes1        21 *****(*****)     7     0     2 2.88k     1    26    67    27
bes2        22 *****(*****)     4     0     1 2.88k     1    28    67    29
-------------------------------------------------------------------------------
total       48     4(    4)    39     0     3 11.5k     2    54   134    56 ......5


*** buffer information ***

server   maxio minio sumio brreq bwreq bfhit hitrt  read write lrreq lbrht lbrrt
lwreq lbwht lbwrt lread lwrit flush bfupd bfred bfwrt cinsm cafls cafwr cfmax
cfavg ldirc ldiuc ldihc ldird ldiwt lbfshc .....................................6
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------
bes1         0     0    71 64.5k 44.3k 52.4k 81.1%     3 12.2k     0     0 *****
    0     0 *****     0     0 10.6k     0     0     0     0     0     0     0
1.00k 11.4k 11.4k     0     0 11.4k 10.6k
bes2         0     0    71 64.5k 44.3k 52.4k 81.1%     3 12.2k     0     0 *****
    0     0 *****     0     0 10.6k     0     0     0     0     0     0     0
1.00k 11.4k 11.4k     0     0 11.4k 10.6k
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------
total        0     0   142  129k 88.6k  105k 81.1%     6 24.3k     0     0 *****
    0     0 *****     0     0 21.3k     0     0     0     0     0     0     0
 1.00k 22.8k 22.8k     0     0 22.8k 21.3k

pdload terminated, return code=0 ..........................................7
```

Explanation

1. Header information (version number and `pdload` start time)
2. `pdload` processing time
3. `pdload` preprocessing time
4. Information about each server:

   `name`: Command name or server name

   `cpu`: CPU processing time (seconds)

`svrup(first)`: Server startup time (initial server access time) (seconds)

`reciv`: Total wait time for communication from other processes (seconds)

`wait`: Total time required for allocating locked resources (seconds)

`sort`: Sorting time (seconds)

`send`: Number of communications to other processes

`delet`: Data deletion time (seconds)

`dtlod`: Data loading time (seconds)

`ixlod`: Index creation time (seconds)

`dtins`: Communication message processing time (seconds)

5. Total value for information about each server
6. Information about each buffer:

   `server`: Name of back-end server

   `maxio`: Maximum input/output time (seconds)

   `minio`: Minimum input/output time (seconds)

   `sumio`: Total input/output time (seconds)

   `brreq`: Number of `READ` requests for data and index buffer

   `bwreq`: Number of `WRITE` requests for data and index buffer

   `bfhit`: Data and index buffer hit count

   `hitrt`: Data and index buffer hit ratio (%)

   `read`: Data and index buffer `READ` count

   `write`: Data and index buffer `WRITE` count

   `lrreq`: Number of `READ` requests for LOB buffer

   `lbrht`: LOB buffer `READ` hit count

   `lbrrt`: LOB buffer `READ` hit ratio (%)

   `lwreq`: Number of `WRITE` requests for LOB buffer

   `lbwht`: LOB buffer `WRITE` hit count

   `lbwrt`: LOB buffer `WRITE` hit ratio (%)

   `lread`: LOB buffer `READ` count

   `lwrit`: LOB buffer `WRITE` count

   `flush`: Buffer flush count

`bfupd`: `READ` wait count during buffer updating

`bfred`: `READ` wait count during buffer `READ`

`bfwrt`: `WRITE` wait count during buffer `WRITE`

`cinsm`: Internal information used by the system

`cafls`: Internal information used by the system

`cafwr`: Internal information used by the system

`cfmax`: Internal information used by the system

`cfavg`: Internal information used by the system

`ldirc`: Number of times index pages were referenced using the random access local buffer

`ldiuc`: Number of times index pages were updated using the random access local buffer

`ldihc`: In `bfhit`, the buffer hit count for index pages in the random access local buffer

`ldird`: In `read`, the number of real READs for index pages in the random access local buffer

`ldiwt`: In `write`, the number of real WRITEs for index pages in the random access local buffer

`ldfshc`: In `flush`, the buffer flush count in the random access local buffer

7. `pdload`'s return code

Notes

1. Each item is displayed with a maximum length of three digits. If a value exceeds three digits, the system rounds up the fourth digit and displays the value with a decimal point and one of the units shown as follows:

   K: Kilo ($\times 10^3$ for time and count)

   M: Mega ($\times 10^6$ for time and count)

   G: Giga ($\times 10^9$ for time and count)

2. If there is no corresponding output information, the system displays `*****`.
3. The processing time less than one second is rounded off. This means that a value less than one second is displayed as 0 second.

5.5 Input data file

This section describes the input data files.

- DAT format

  In this format, column data is specified in characters.

  For details, see *5.5.1 DAT format*.

- Binary format

  This format is based on the internal HiRDB format.

  For details, see *5.5.2 Binary format*.

- Fixed-size data format

  This format allows data specification in characters, as in the DAT format, or in internal HiRDB format, as in the binary format.

  For details, see *5.5.3 Fixed-size data format*.

- Arrayed data format for a table containing repetition columns

  If a table subject to data loading contains repetition columns, the input data file must contain data in the arrayed data format according to the repetition columns. The data can be specified in either DAT, binary, or fixed-size data format.

  For details, see *5.5.4 Arrayed data format for tables containing repetition columns*.

- LOB column input file

  In the LOB column input file, as many LOB data items are specified as there input lines.

  For details, see *5.5.5 LOB column input file*.

5.5.1 DAT format

In the DAT format, both input data and the input parameters for a constructor function are specified as character string data.

(1) Specification format

Each column data item must be separated by a separator character.

Table 5-12 shows the specification format of column data and parameters for a construction function by the data type (DAT format).

Table 5-12: Specification format of column data and parameters by data type (DAT format)

Data type		Specification format of column data and parameter	
		General data	Null value or default value[1]
Numeric data	INTEGER	• Specify a numeric value in characters. • For a negative value, add a minus sign (-). • Specify a decimal point as *integer fraction*. • Specify a floating point as *mantissa*-e-*exponent*. Integer (Example: -1234...0...1234) Decimal point (Example: -1.56...0...1.56) Floating point (Example: -2.4e+9...0e0...2.4e+9)	* or not specified
	SMALLINT		
	DECIMAL		
	FLOAT		
	SMALLFLT		
Character string data	CHARACTER	Specify as characters or a character string enclosed in double quotation marks ("). (Example: abcdef, ABCDEF) (Example: "abcd", "ABCD")	Not specified
	VARCHAR		
National character string data	NCHAR		
	NVARCHAR		
Mixed character string data	MCHAR		
	MVARCHAR		

Data type		Specification format of column data and parameter	
		General data	Null value or default value[1]
Date data	DATE	Specify in the numeric format *yyyy-mm-dd*. *yyyy*: Year. *mm*: Month. *dd*: Day. (Example: January 1, 1995 → 1995-01-01)	* or not specified
Date interval data	INTERVAL YEAR TO DAY	Specify in the numeric format [-]*yyyymmdd*. For a negative value, add a minus sign (-). (Example: 1111 years, one month, and one day → 11110101)	
Time data	TIME	Specified in the numeric format *hh*:*mm*:*ss*. *hh*: Hour. *mm*: Minute. *ss*: Second. (Example: 12:01:01 → 12:01:01)	
Time interval data	INTERVAL HOUR TO SECOND	Specified in the numeric format [-]*hhmmss*. For a negative value, add a minus sign (-). (Example: 1 hour, 1 minute, and 1 second → 010101)	
Time stamp data	TIMESTAMP	Specified in the numeric format *YYYY-MM-DD hh*:*mm*:*ss*[.*nnnnnn*]. For the fraction part of the second, specify a value with a length of 0 to 6 digits. If the fraction part is less than the defined length, the system assumes 0; if it is longer than the defined length, the system discards the excess portion. If the fraction part consists of more than 6 digits, an error (KFPL31002-E) results.[2]	
Binary data	BINARY	Specifies the binary data or binary data enclosed in the enclosing characters. (Examples: abc 123 or "abc 123")	Not specified
Large object data	BLOB	-k f specified: Specify the name of the LOB input file. (Example: lobfile01) -k c specified: Specify a character string with a length of 1-1,023 bytes without any separator characters. This character string is ignored when LOB data is loaded. (Example: @)	

Notes

The system handles the null values for each data type as follows:

- For the fixed-length character string, national character string, mixed character string, binary, and large object data types, the system treats the following specification as the null value or default value:

  ```
  ..., , ...
  ..., " ", ...
  ```

- For the variable-length character string, national character string, mixed character string, and binary data types with the `-z` option specified, the system treats the following specification as the null value or default value:

  ```
  ..., , ...
  ```

  `..., " ", ...` is treated as data with a length of 0.

- For the other data types, the system treats the following specification as the null value or default value:

  ```
  ..., , ...
  ..., " ", ...
  ..., *, ...
  ```

If the `enclose` operand is specified in the `extdat` statement, the double quotation mark (`"`) becomes the enclosing character specified in the `enclose` operand.

[1] The value to be stored depends on the `null_string` operand value specified in the option statement.

[2] The following table describes the storage method when the length of the fraction part for seconds is specified for the `TIMESTAMP` type:

<table>
<tr><th rowspan="2">Defined digits</th><th colspan="8">Length of the fraction part of the second in input data</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7 or more</th></tr>
<tr><td>0</td><td>Stored as is.[1]</td><td colspan="6">Discards excess digits.</td><td rowspan="4">Error (KFPL31002-E)</td></tr>
<tr><td>2</td><td colspan="2">Adds zeros up to the defined length.[2]</td><td>Stored as is.[1]</td><td colspan="4">Discards excess digits.</td></tr>
<tr><td>4</td><td colspan="4">Adds zeros up to the defined length.[2]</td><td>Stored as is.[1]</td><td colspan="2">Discards excess digits.</td></tr>
<tr><td>6</td><td colspan="6">Adds zeros up to the defined length.[2]</td><td>Stored as is.[1]</td></tr>
</table>

[1] The input data is converted to the TIMESTAMP type and then stored as is.

[2] The real data is in bytes. If the defined length consists of an odd number of bytes, as many zeros as equals the defined length + 1 are added.

(2) Rules

- One line in the file corresponds to one line of table data (each line must end with the linefeed code).
- When coding input parameters that are passed to a constructor function, delimit each parameter with a separator character.
- Neither the null character (0x00), linefeed code (0x0a), nor carriage return at the end of a line (0x0d0a) is treated as data. To treat them as data, specify the extdat statement and enclose the input data in the enclosing character.
- The maximum length of one line is 32 KB. However, if you specify the source statement with the maxreclen operand, the system accepts more than 32 KB of data per line during data loading.
- If an input value begins with the enclosing character, consecutive enclosing and separator characters are treated as the end of the column. To specify an input value consisting of the consecutive enclosing and separator characters, use the -s option to change the separator character.
- If character string data immediately before or after a separator character is enclosed in the enclosing character ("), the enclosing characters are not subject to the input operation. If the separator character is encountered in a character string that is enclosed in the enclosing character, the separator character is treated as data. Table 5-13 shows how character string data and enclosing characters are handled.

Table 5-13: Handling of character string data and enclosing characters

Character string data in input data file	Data stored in table	
	DAT format	Extended DAT format
...,ABCD,...	ABCD	ABCD
...,AB"CD,...	AB"CD	Input data error
...,AB""CD,...	AB""CD	AB"CD
...,"ABCD",...	ABCD	ABCD
...,"AB"CD",...	AB"CD	Input data error
...,"AB""CD",...	AB""CD	AB"CD
...,""ABCD"",...	"ABCD"	Input data error
...,"""ABCD""",...	""ABCD""	"ABCD"
...,,...	Null value or default value*	Null value or default value*
...,"",...	Null value or default value*	Null value or default value*
...,"""",...	""	"
...,"AB,CD",...	AB,CD	AB,CD
...,AB\0CD,...	Input data error	Input data error
...,AB\nCD,...	Input data error	Input data error
...,"AB\0CD",...	Input data error	AB\0CD
...,"AB\nCD",...	Input data error	AB\nCD

Legend:

,: Separator character

": Enclosing character

\0: Null character (0x00)

\n: Linefeed code

* The value to be stored depends on the `null_string` operand value specified in the `option` statement.

- Table 5-14 shows the format of character string data that is converted to numeric data. For the limitations to the specification of character string data for a numeric data type in input data, see the limitations on the use of numeric literals in the

manual *HiRDB Version 8 SQL Reference*.

Table 5-14: Format of character string data converted to numeric data

Numeric value expressed as character string data	Notation	
	Specification format	Example (specification → stored value)
Integer	Δ [{+ | -}] *natural-number* A natural number is a string of numeric characters 0-9.	+10Δ → 10 -009 → -9 ΔΔΔ0Δ → 0
Decimal number	Δ [*integer*] . [*fraction*] Δ A fraction is an unsigned integer. Either the integer or the fraction must be specified.	Δ.05Δ → 0.05 -0.8Δ → -0.8 1.0ΔΔ → 1.0 +1.Δ → 1.0
Floating point number	Δ *mantissa*-e[*exponent*] Δ Mantissa is expressed as an integer or decimal number. E may be specified for e. Exponent is expressed as a one-digit or two-digit integer.	.3e → 0.3×10 2.4e-02 → 2.4 × 10-2 67e-5Δ → 67 × 10-5

Δ: Indicates an optional space.

- Table 5-15 shows the handling of storage data when the length of input data does not match the data length in table.

Table 5-15: Handling of storage data when the length of input data doesn't match the data length in table

Data length in table (m)	Data length of input data (n)	
	Fixed-length character string data	Variable-length character string data or binary data
$m = n$	Input data is stored as is.	
$m < n$	When the lengover operand is omitted in the option statement, the portion of the input data following (m + 1) is discarded.* When the lengover operand is specified in the option statement, the input data is not stored because it is regarded as an input data error.	

Data length in table (m)	Data length of input data (n)	
	Fixed-length character string data	Variable-length character string data or binary data
m > *n*	Input data is padded with spaces after (*n* + 1).	Input data is stored as is.

* If `cutdtmsg=on` is specified in the `option` statement and data truncation occurs, the system outputs warning messages to the error information file.

- If a data section in the input data contains separator characters (or code that is the same as the separator characters), the system treats the data as separator characters unless the entire data item is enclosed in double quotation marks (`"`).
- Table 5-16 describes the data to be stored and how to specify the null value when `enclose="` is specified in the `extdat` statement.

Table 5-16: Data to be stored and how to specify the null value when enclose=" is specified in the extdat statement

Data type		Input data	Data to be stored in table	How to specify null value or default value*
Numeric data	INTEGER	"1234"	1234	*, "", "*", or not specified
	SMALLINT	"1234"	1234	
	DECIMAL(6,1)	"000001"	00001.0	
	FLOAT	"2.225075e-308"	+2.225075000000000E-308	
	SMALLFLT	"1.175495e-38"	+1.175495051471520E-38	
Character string data	CHAR	"ABCDE"	ABCDE	"" or not specified
	VARCHAR	"ABCDE"	ABCDE	
National character string data	NCHAR	"あいうえお"	あいうえお	
	NVARCHAR	"あいうえお"	あいうえお	
Mixed character string data	MCHAR	"ABCDE"	ABCDE	
	MVARCHAR	"ABCDE"	ABCDE	

Data type		Input data	Data to be stored in table	How to specify null value or default value*
Date data	DATE	"2000-05-31"	2000-05-31	*, "", "*", or not specified
Date interval data	INTERVAL YEAR TO DAY	"+00000000."	00000000	
Time data	TIME	"09:00:00"	09:00:00	
Time interval data	INTERVAL HOUR TO SECOND	"+000000."	000000	
Time stamp data	TIMESTAMP(6)	"2002-08-08△20:25:43.39"	2002-08-08 20:25:43.390000	
Binary data	BINARY	"ABCDE"	ABCDE	"" or not specified
Large object data	BLOB	"*LOB-input-file-name*"	Contents of the LOB input file	

Legend: △: Single-byte space

* The value to be stored depends on the `null_string` operand value in the `option` statement.

- Table 5-17 shows examples of space deletion when `enclose="` is specified in the `extdat` statement.

  Table 5-17: Examples of space deletion when enclose=" is specified in the extdat statement

Data type		Input data	Data to be stored in table
Numeric data	INTEGER	"△1234△"	1234
	SMALLINT	"△"	Null value
	DECIMAL(6,1)	"△000001"	00001.0
	FLOAT	"△2.225075e-308"	+2.225075000000000E-308
	SMALLFLT	"1.175495e-38△"	+1.175495051471520E-38
Date data	DATE	"△2000-05-31"	2000-05-31

Data type		Input data	Data to be stored in table
Date interval data	INTERVAL YEAR TO DAY	"+Δ00000000."	00000000
Time data	TIME	"Δ09:00:00"	09:00:00
Time interval data	INTERVAL HOUR TO SECOND	"Δ+Δ000000."	000000
Time stamp data	TIMESTAMP	"Δ2002-08-08Δ Δ20:25:43.39Δ"	2002-08-08 20:25:43.39

Legend: Δ: Single-byte space

(3) Example

Figure 5-11 shows a specification example of DAT-format data.

Figure 5-11: Specification example of DAT-format data

Input data file

```
Jones,36,1958-10-15,Chicago
Baker,25,1968-01-01,Dallas
Robinson,41,1953-12-31,Seattle
Smith,36,1963-05-30,Boston
Young,46,1948-04-01,Detroit
```

Note

The data types are, from left to right, NCHAR, INTEGER, DATE, and VARCHAR.

5.5.2 Binary format

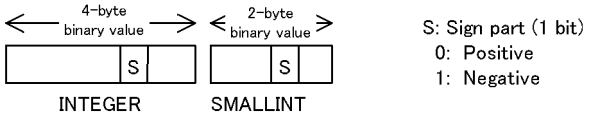
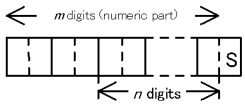
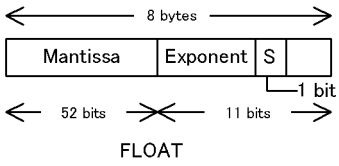
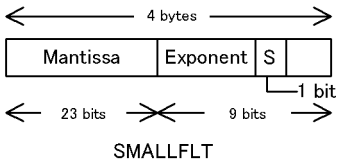
In the binary format, the input data is specified in the format stored in the database.

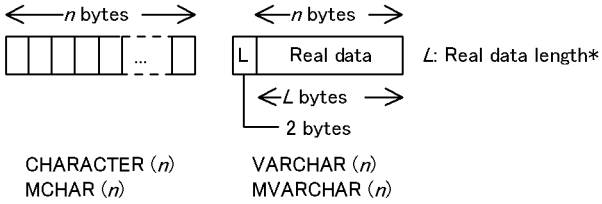
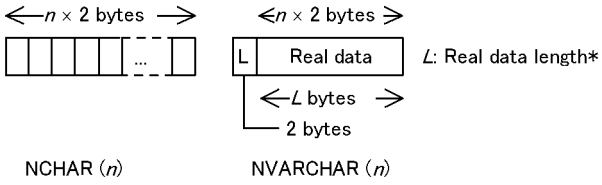
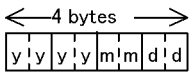
(1) Specification format

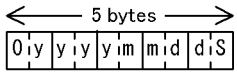
(a) Specification format of column data

Table 5-18 shows the specification format of column data and parameters by the data type (binary format).

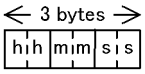
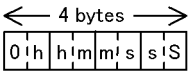
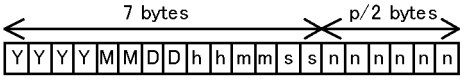
Table 5-18: Specification format of column data and parameters (binary format)

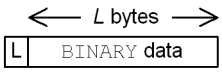
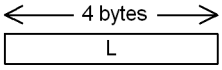
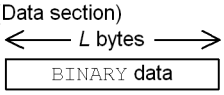
Data and parameter types		Column data specification format
Numeric data	INTEGER, SMALLINT	 S: Sign part (1 bit) 0: Positive 1: Negative
	DECIMAL(m, n) $1 \leq m \leq 29$ $0 \leq n \leq 29$ $m \geq n$	  1 byte S: Sign part (4 bits) Decimal point – In packed format, 1 byte represents 2 numeric digits. – The sign part is not normalized.
	FLOAT, SMALLFLT	 S: Sign part (1 bit) 0: Positive 1: Negative  – Same format as IEEE standard.

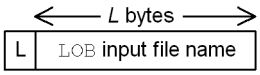
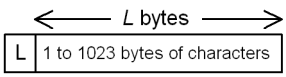
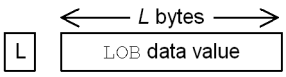
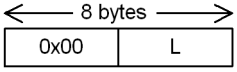
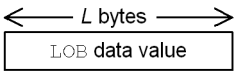
<table>
<tr><th colspan="2">Data and parameter types</th><th>Column data specification format</th></tr>
<tr><td rowspan="2">Character string, national character string, and mixed character string data</td><td>CHARACTER(n),
VARCHAR(n),
MCHAR(n),
MVARCHAR(n)</td><td> 
</td></tr>
<tr><td>NCHAR(n),
NVARCHAR(n)</td><td> 
</td></tr>
<tr><td>Date data</td><td>DATE</td><td> 

yyyy: Year
mm: Month
dd: Date
4-byte decimal number in packed format without the sign part
– In the packed format, 1 byte represents 2 numeric digits.</td></tr>
<tr><td>Date interval data</td><td>INTERVAL
YEAR TO DAY</td><td> 

5-byte decimal number in packed format
yyyy: Year
mm: Month
dd: Date
S: Sign part (4 bits)
– In the packed format, 1 byte represents 2 numeric digits.
– The sign part is not normalized.
– If 12 or a greater value is specified for mm, the year is incremented.</td></tr>
</table>

Data and parameter types		Column data specification format
Time data	TIME	 *hh*: Hour *mm*: Minutes *ss*: Seconds 3-byte decimal number in the packed format without the sign part – In the packed format, 1 byte represents 2 numeric digits.
Time interval data	INTERVAL HOUR TO SECOND	 *hh*: Hour *mm*: Minutes *ss*: Seconds *S*: Sign part (4 bits) 4-byte decimal number in the packed format – In the packed format, 1 byte represents 2 numeric digits. – The sign part is not normalized. – If 60 or a greater value is specified for *mm* or *ss*, the hour or minute value is incremented, respectively.
Time stamp data	TIMESTAMP(p)	 *YYYY*: Year *MM*: Month *DD*: Date *hh*: Hour *mm*: Minute *ss*: Second *nnnnnn*: Fraction part of seconds Unsigned packed decimal number with a maximum length of 10 bytes – The fraction part for seconds is 0, 2, 4, or 6 according to the definition.

Data and parameter types		Column data specification format
BINARY data, BINARY attribute data	BINARY	<Input data file for the database load utility>  The first 4 bytes store the data length, as a binary number. <Input data file for the database reorganization utility with -W specified> (Length section)  4 bytes of the column data section store the data length, as a binary number. (Data section)  Data is output after all column data (LOB column structure base table data). Binary data and binary attribute data are output after all column data (LOB column structure base table data), but before the LOB attribute data or LOB data in the order of the binary attribute data and the binary data based on the order of column definitions.

Data and parameter types		Column data specification format
LOB data, LOB attribute data	BLOB	<When -k f is specified>  The first 4 bytes store the name of the LOB input file, as a binary number. <When -k c is specified>  The first 4 bytes store the length of the specified character string, as a binary number. This character string is ignored when LOB data is created. <When -k v or -k d is specified>  See *(4)(a) -k* or *(4)(b) -k*. <Input data file for the database reorganization utility with -W specified> (Length section)  The first 4 bytes of the 8-byte column data section store 0x00, and the remaining 4 bytes store the data length as a binary value. (Data section)  Data is output after all column data (LOB column structure base table data). The LOB attribute data is output after the LOB column structure base table data, but before the LOB data.

* Only when the -z option is specified, does the system set the actual data length to 0 and treats the part without the actual data as 0-length data.

(b) Input parameter specification format

The following describes the input parameter specification format:

- Code the input parameters for a constructor function for generating values that are stored in an abstract data type using the same structure as the function's parameter type.
- Specify the input parameters in the same order as the arguments for the constructor function.
- If the target storage column is an abstract data type column, the constructor function must have a name identical to the abstract data type. If there are multiple constructor functions with the same name or if you are using a constructor

function with a name that is different from the name of abstract data type, specify the name of the function in the null value/function information file.

- If there are multiple input parameters, specify parameter values one after another, without placing any space between them.

(2) Rules

- If you are using a magnetic tape unit to create input data in the variable-length block mode, set the block length to 32 KB or less.
- In input data, columns must have the same sequence as the columns in the table definition.
- There must be no space between column data items. For a variable-length character string, if the actual data is shorter than the defined length, specify the data with its actual length and move up the subsequent data item.
- If the input data contains any of the following data, you should not use data loading with the synchronization point specification. If the input data contains the following data, the system stores all up to the synchronization point in the event of an error:
  - Repetition column
  - Variable-length data (`VARCHAR`, `NVARCHAR`, `MVARCHAR`, `BINARY`, `BLOB`)
  - Variable-length data for an argument of abstract data type
- In the binary format, the default value cannot be stored because input data is written directly. To store the default value, directly write the default value when you create the input data file.

(3) Example

Figure 5-12 shows a specification example of binary-format data.

Figure 5-12: Specification example of binary-format data

Input data file

```
928691ba8140814081400000000243f8000000008796f6b6f68616d61
 Jones                36      1     Chicago
8db293a18140814081400000001440000000000773686962757961
 Smith                20      2     Boston
97e996d88140814081400000029c059999a0005746f6b796f
 Baker                 41   -3.4     Dallas
```

Notes

1. The upper row of the input data indicates the data, and the lower row indicates the values to be stored.
2. The data types are, from left to right, `NCHAR(5)`, `INTEGER`, `SMALLFLT`, and `VARCHAR`.
3. National character data is in shift JIS codes.

(4) Coding LOB data and LOB parameters when v or d is specified in the -k option

(a) -k v

When specifying `v` in the `-k` option, directly code the input file name for LOB data and the data for the LOB parameter in the data input file.

When compared with specifying `f` in the `-k` option, this method can reduce input/output time for the LOB input file.

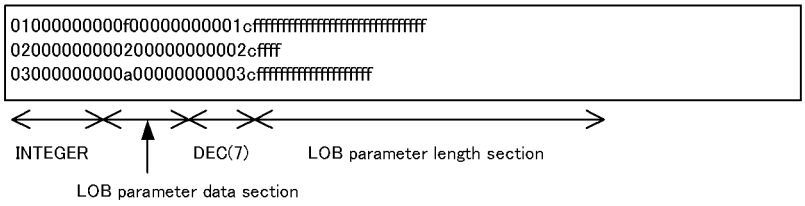
Rules

1. Specify the LOB data and the LOB parameter in a format in which a given column in row data has a 4-byte length section, such that the data section is coded following the row data. Specify these items in the column order of the row data.
2. If the LOB data or the LOB parameter is the null value, specify a value of `-1` in the length section. The system ignores any null comparison values produced by a null value/function information file or a function information file. Do not specify the data section.
3. If the LOB data or the LOB parameter has 0 bytes, specify a value of `0` in the length section without specifying a data section.

Example

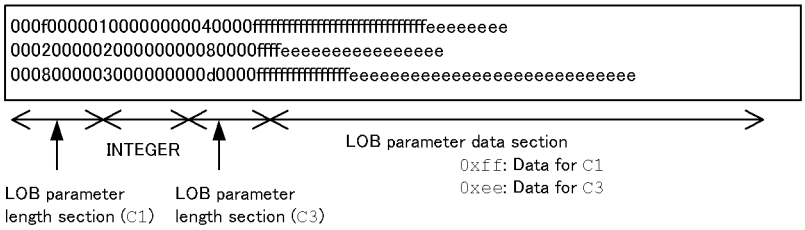
- Table with a column of abstract data type with BLOB input:

  Table definition: `CREATE TABLE T1 (C1 INTEGER, C2 SGMLTEXT, C3 DEC(7,0))`



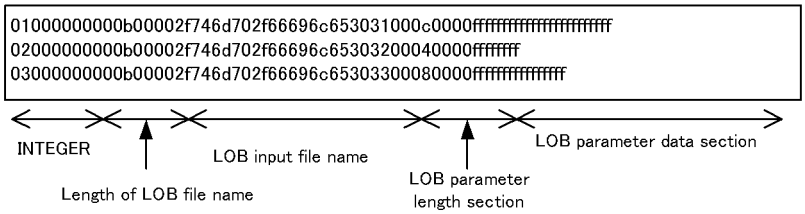
- Table with two columns of abstract data type with BLOB input:

  Table definition: `CREATE TABLE T2 (C1 SGMLTEXT, C2 INTEGER, C3 SGMLTEXT)`



- Table having a column of abstract data type with input of a LOB column and BLOB:

  Table definition: `CREATE TABLE T1 (C1 INTEGER, C2 BLOB (100), C3 SGMLTEXT)`



(b) -k d

When specifying `d` in the `-k` option, specify the data itself, not the name of the LOB data storage file, in the input data file.

Rules

1. Specify the LOB data and the LOB parameter data in a format in which a given column in row data has a 4-byte length section, such that the data section is coded following the row data.
2. Specify the input data file in the following order:
   [*row data*] [*LOB parameter data*] [*LOB data*]
   1 2 3

   Explanation:

   1. Specify the length section in the row data in the order of column definition.
   2. Specify the LOB parameter data immediately after the row data in the order of column definition.
   3. Specify the LOB data immediately following item B in the order of column definition.
3. If the LOB data or the LOB parameter data is the null value, specify a value of -1

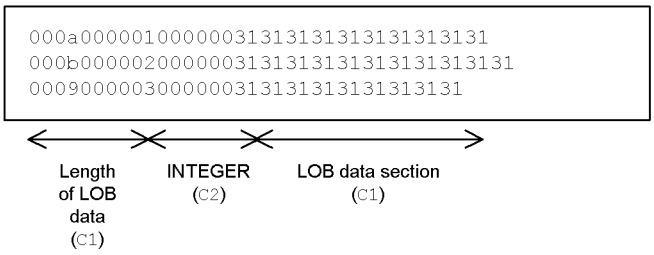
in the 4-byte length section. The system ignores any null comparison values produced by a null value/function information file or a function information file. Do not specify the data section. There is no data section.

4. If the LOB data or the LOB parameter has 0 bytes, specify a value of 0 in the length section without specifying a data section.

Example

- Table with one LOB column

  Table definition: `CREATE TABLE T1 (C1 BLOB(100),C2 INTEGER);`



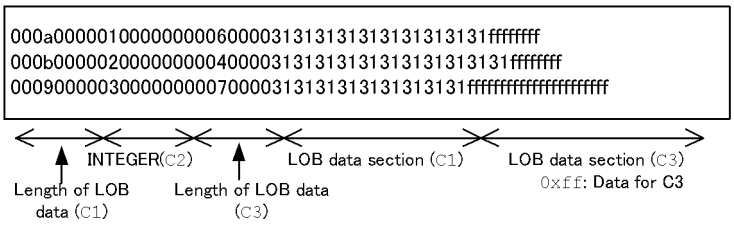
Specification method:
1. Specify the LOB data's length section (4 bytes) before the row data.
2. Specify the LOB data following item 1.

Specification method:

1. Specify the LOB data's length section (4 bytes) before the row data.
2. Specify the LOB data following item 1.

- Table with two LOB columns

  Table definition: `CREATE TABLE T2 (C1 BLOB(100),C2 INTEGER,C3 BLOB(100));`



Specification method:

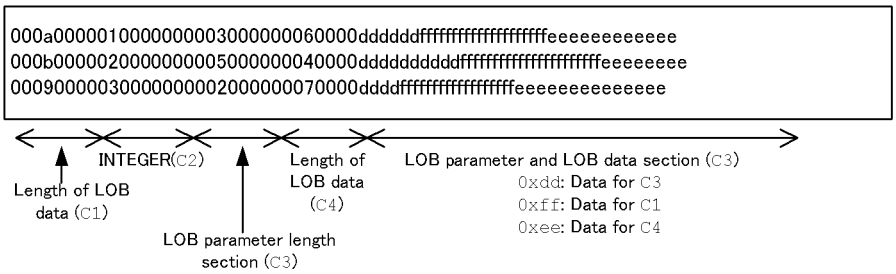
1. Specify the LOB data's length section (4 bytes) in the row data in the order of column definition (which is LOB data length (`C1`) and LOB data length

(C3)).

2. Specify the LOB data following item 1 in the order of column definition (which is LOB data (C1) and LOB data (C3)).

- Table with two LOB columns and an abstract data type column with LOB input

  Table definition: `CREATE TABLE T3(C1 BLOB(100),C2 INTEGER,C3 SGMLTEXT,C4 BLOB(100));`



Specification method:

1. Specify the LOB data and LOB parameter length section (4 bytes) in the row data in the order of column definition (which is LOB data length (C1), LOB parameter length section (C3), and LOB data length (C4)).
2. Specify the LOB parameter data immediately after the row data in the order of column definition (which is LOB parameter data (C4) only).
3. Specify the LOB data following item 2 in the order of column definition (which is LOB data (C1) and LOB data (C4)).

5.5.3 Fixed-size data format

For an input data file in fixed-size data format, specify the input data in characters or in the format stored in the database. When using a constructor function to generate the data to be stored in an abstract data type, code the input parameter either in character string data or in the data type of the function argument.

(1) Specification format

For details about how to specify input data in characters, see *5.5.1 DAT format*. For details about how to specify input data in the format stored in the database, see *5.5.2 Binary format*.

(2) Rules

- Each line has a fixed length.
- All the values in a column must begin at the same location. Character strings to be converted to numeric values may be left justified or right justified, but

character strings to be stored as characters must be left justified (variable-length character strings must be padded with blanks so that the next column begins at the correct location).

- The null value depends on the specification in the column structure information file.
- If you input fixed-length character string data for CHAR, VARCHAR, NCHAR, NVARCHAR, MCHAR, or MVARCHAR (by specifying type=char(*n*)), the system deletes any consecutive single-byte spaces, tab characters, or \0s at the end of a column value except for the first byte. The system them compares the null comparison value and stores the data in the database. Therefore, you cannot store tab characters or \0s in a CHAR or MCHAR column. You cannot store single-byte spaces, tabs, or \0s in a VARCHAR, NCHAR, NVARCHAR, or MVARCHAR column. If you input data containing single-byte spaces, tabs, or \0s to a data type that does not accept single-byte spaces (NCHAR and NVARCHAR), the system handles the input as follows:

  (a) Data consisting of only single-byte spaces, tabs, or \0s

  Only the first byte remains after deleting all single-byte spaces, tabs, and \0s. For the NCHAR or NVARCHAR type, this results in an invalid data error because it requires an even number of bytes for the data length.

  (b) Data consisting of single-byte spaces, tabs, or \0s, and other data

  If the length of data without any consecutive single-byte spaces, tabs, or \0s at the end of the column value is an odd number of bytes, an invalid data error results in the same manner as with (a) previously. If the data length obtained is an even number of bytes, the system stores the data in the database.

  Therefore, you should use double-byte spaces instead of single-byte spaces, tabs, or \0s as the characters to pad the trailing end of fixed-length character string data that is stored in the NCHAR or NVARCHAR type. For a table with the NCHAR data type, if the input data is shorter than the length defined for the table due to the presence of data that is excluded from input, the system assumes double-byte spaces as the trailing part of the data.
- Be sure to use the column structure information file to specify the column data in the input data file.
- If the length of the input character string data does not match the length defined for the storage table, the system handles the input data as fixed-length data in a DAT-format file. However, the data is not stored for variable-length character string data with a length greater than the length specified in the column structure information file. If the length of variable-length character string data is a negative value, the system continues processing.
- To specify input data in the format in which it is stored in the database, use the

binary format. For details about the binary format, see *5.5.2 Binary format*.

For specifying input data using the character string data format, see Table 5-19.

Table 5-19: Data storage when the input data is specified in the character string data format (fixed-size data format)

Data type of table column or parameter data type of constructor function	Specification of column name statement	Data specified in the input data file	Storage data
INTEGER, SMALLINT, DECIMAL, FLOAT, and SMALLFLT	type=char(3)	1△ △	1 of each data type
		△ △1	
		△1△	
CHAR(7) and MCHAR(7)	type=char(7) or type=char(7),mode=text	ABC△ △ △ △	ABC△ △ △ △
		"ABC"△ △	
		"ABC△ △"	
		""ABC""	"ABC"△ △
		△ △ △ △ △ △ △	△ △ △ △ △ △ △
		""△ △ △ △ △	
	type=char(7),mode=bin	ABC△ △ △ △	ABC△ △ △ △
		"ABC"△ △	"ABC"△ △
		"ABC△ △"	"ABC△ △"
		""ABC""	""ABC""
		△ △ △ △ △ △ △	△ △ △ △ △ △ △
		""△ △ △ △ △	""△ △ △ △ △

Data type of table column or parameter data type of constructor function	Specification of column name statement	Data specified in the input data file	Storage data
NCHAR(3)	type=char(6)	あ ▲ ▲	あ ▲ ▲
		"あ"▲	"あ"▲
		"あ ▲"	あ ▲ ▲
		""あ""	"あ"▲
		▲ ▲ ▲	▲ ▲ ▲
		""▲ ▲	""▲ ▲
VARCHAR(7) and MVARCHAR(7)	type=char(7) or type=char(7),enclose_del=yes	ABC△ △ △ △	ABC (3 bytes)
		"ABC"△ △	
		"ABC△ △"	ABC△ △ (5 bytes)
		""ABC""	"ABC" (5 bytes)
		△ △ △ △ △ △ △	△ (1 byte)
		""△ △ △ △ △	(0 byte)
	type=char(7),enclose_del=no	ABC△ △ △ △	ABC (3 bytes)
		"ABC"△ △	"ABC" (5 bytes)
		"ABC△ △"	"ABC△ △" (7 bytes)
		""ABC""	""ABC"" (7 bytes)
		△ △ △ △ △ △ △	△ (1 byte)
		""△ △ △ △ △	"" (2 bytes)

Data type of table column or parameter data type of constructor function	Specification of column name statement	Data specified in the input data file	Storage data
NVARCHAR(3)	type=char(6)	あ ▲ ▲	あ ▲ ▲ (6 bytes)
		" あ " ▲	" あ " ▲ (6 bytes)
		" あ ▲ "	あ ▲ (4 bytes)
		" " あ " "	" あ " (4 bytes)
		▲ ▲ ▲	▲ ▲ ▲ (6 bytes)
		" " ▲ ▲	" " ▲ ▲ (6 bytes)
DATE	type=char(12)	2003-12-31△ △	2003-12-31
		△ △2003-12-31	
		△2003-12-31△	
INTERVAL YEAR TO DAY	type=char(11)	00010101.△ △	+00010101.
		△ △00010101.	
		△00010101.△	
TIME	type=char(10)	08:45:00△ △	08:45:00
		△ △08:45:00	
		△08:45:00△	
INTERVAL HOUR TO SECOND	type=char(9)	010101.△ △	+010101.
		△ △010101.	
		△010101.△	
TIMESTAMP	type=char(21)	1991-01-01_11:22:33△ △	1991-01-01_11:22:33
		△ △1991-01-01_11:22:33	
		△1991-01-01_11:22:33△	

Data type of table column or parameter data type of constructor function	Specification of column name statement	Data specified in the input data file	Storage data
BLOB	type=char(13)	c:\pdload\file01△△	Data in c:\pdload\file01
		△△c:\pdload\file01	Error
		△c:\pdload\file01△	
BINARY(7)	type=char(7)	△△△△△△△	△△△△△△△

Legend:

△: Single-byte space (0x20)

▲: Double-byte space

(3) Example

Figure 5-13 shows an example of coding data in the fixed-size data format.

Figure 5-13: Example of data in fixed-size data format

– Text file

```
Nelson       20     Ohio           00:00:00
Carr          2     California     13:01:27
Oliver       -1     Texas          06:33:14
Thomas       36     Florida        20:00:01
Marsh         0     Michigan       17:15:58
```

CHARACTER INTEGER NCHAR TIME

– Binary file

```
6e616b616d75726120200000036c0008796f6b6f68616d612020
Nelson 36       Boston
7361746f752020202020000020c0007736869627579612020 20
Samuel 20       Chicago
73757a756b69202020200000041c0005746f6b796f2020202020
Samson 41       New York
```

CHARACTER DECIMAL VARCHAR

5.5.4 Arrayed data format for tables containing repetition columns

This section describes the method for specifying arrayed data in an input data file for tables containing repetition columns.

The format that the input data file takes depends on the input data format (DAT format, binary format, or fixed-size data format), on the arrayed data format (FF, FV, or VV), and whether the data consists of a variable-length data type.

(1) Specification format

Table 5-20 shows the methods for specifying arrayed data in an input data file for tables containing repetition columns.

Table 5-20: Methods for specifying arrayed data in an input data file for tables containing repetition columns

Description	File format	Arrayed data format		
		FF	FV	VV
Specifying current number of elements in input data	—	Not specified	Specified	
Method for specifying current number of elements	—	—	For DAT format, specify in characters. For fixed-size data or binary format, specify in 2-byte numeric values (hexadecimal).	
Number of element data items specified in input data	—	A number less than or equal to the number of elements specified in table definition. See (3) *Relationship between the number of elements in input data and the number of elements specified in table definition*.		Current number of elements
Number of elements stored in database	—	Depends on how null values are treated. See (2) *Handling of null values for columns and elements*.	Current number of elements	
Specifiable input data formats	DAT format	Specifiable	Not specifiable	Specifiable
	Binary format	Variable length not specifiable		Specifiable
	Fixed-size data format	Specifiable		Not specifiable (because all row data are fixed length)

<table>
<tr><th rowspan="2">Description</th><th rowspan="2">File format</th><th colspan="3">Arrayed data format</th></tr>
<tr><th>FF</th><th>FV</th><th>VV</th></tr>
<tr><td rowspan="3">Null value specification method (for elements)</td><td>DAT format</td><td>Specify by using consecutive delimiters as in normal columns. See section 5.5.1 DAT format.</td><td colspan="2">—</td></tr>
<tr><td>Binary format</td><td colspan="3">Specify null-equivalent values in null-value/information file as in normal columns.</td></tr>
<tr><td>Fixed-size data format</td><td colspan="2">Specify null-equivalent values in column structure information file as in normal columns.</td><td>—</td></tr>
<tr><td>Null-value specification method (for columns)</td><td>—</td><td>Specify a null value in all elements. Either a null value can be assigned to all elements or a column can be specified as a null value. See (2) Handling of null values for columns and elements.</td><td colspan="2">Specify 0 as the current number of elements.</td></tr>
<tr><td>Number of elements stored in database</td><td>—</td><td>Depends on how null values are treated. See (2) Handling of null values for columns and elements.</td><td colspan="2">Stores as many data items as there are specified current elements.</td></tr>
<tr><td rowspan="3">Variable-length data type specification method</td><td>DAT format</td><td>Same as normal column</td><td>—</td><td>Same as normal column</td></tr>
<tr><td>Binary format</td><td colspan="2">Not specifiable</td><td>Specify valid data length (no fill characters).</td></tr>
<tr><td>Fixed-size data format</td><td colspan="2">Specify data equal to table definition length and fill the remainder.</td><td>—</td></tr>
<tr><td>Element data delimiters for DAT-format files</td><td>—</td><td>Same as column delimiters</td><td>—</td><td>Same as column delimiters</td></tr>
<tr><td>Element data error checking</td><td>—</td><td colspan="3">As with normal columns, perform data-type specific checking; for FF and VV formats, check the current number of elements. For FF and VV formats, specify data for the part other than the effective elements, but do not check the data following the effective element data.</td></tr>
</table>

—: Not applicable

(a) DAT format

In a DAT-format file, you can specify arrayed data in FF or VV format.

FF format

Specify as many data items as there are specified elements by separating each data item by a delimiter. If you omit a column structure information file or the number of elements in the column structure information file, the system assumes the maximum number of elements as the specified number of elements.

VV format

Specify the current number of elements in the range from 0 to the maximum number of elements at the beginning of the element data, then specify as many data items as there are current elements by separating each data item by a delimiter.

(b) Binary format

In the binary format, you can specify arrayed data in FF, FV, or VV format.

FF format

Specify as many data items as there are specified elements in the format shown in Table 5-18. If you omit the null value/function information file itself or the number of elements in the null value/function information file, the system assumes the maximum number of elements as the specified number of elements.

FV format

Specify the current number of elements, expressed as a 2-byte binary value, at the beginning of the element data, then specify as many data items as there are current elements, as shown in Table 5-18. If you omit the null value/function information file itself or the number of elements in the null value/function information file, the system assumes the maximum number of elements as the specified number of elements.

VV format

Specify the current number of elements, expressed as a 2-byte binary value, at the beginning of the element data, then specify as many data items as there are current elements, as shown in Table 5-18.

(c) Fixed-size data format

In the fixed-size data format, you can specify arrayed data in FF or FV format.

If you omit the column structure information file or null value/function information file or the number of elements in these files, the system assumes the maximum number of elements as the specified number of elements.

If you specify data in binary format, and elements specified in `VARCHAR`, `NVARCHAR`,

or MVARCHAR are less than the defined length, you need to pad the data with fill characters up to its defined length.

FF format

Specify as many data items as there are specified elements in the format shown in Table 5-18.

FV format

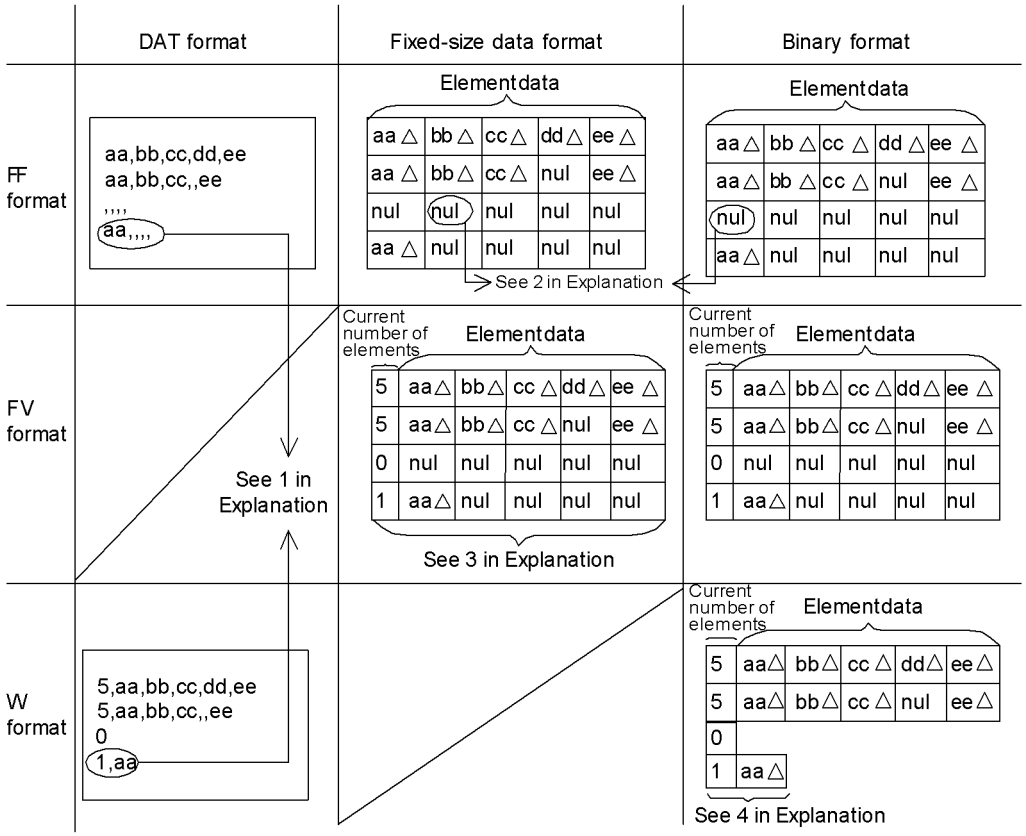
Specify the current number of elements, expressed as a 2-byte binary value, at the beginning of the element data, then specify as many data items as there are specified elements, as shown in Table 5-18.

(d) Specification examples of arrayed data in fixed-size data format

Figure 5-14 shows specification examples of arrayed data in fixed-size data format.

These examples are based on columns of CHAR(3) ARRAY[5]. The 3-byte null is specified as the null value.

Figure 5-14: Specification examples of arrayed data in fixed-size data format



△ : One space.

Note:
For a DAT file, each row of input data is followed by a linefeed; for a fixed-size data or binary format file, row data is specified successively.

Explanation:
1. For a DAT format file, always specify a fixed number of delimiters in the FF format; in the VV format, specify delimiters equal to the current number of elements.
2. For a null value, specify the null value specified in the column structure information file or in the null value information file.
3. In the FV format, specify the current number of elements. Note that you can specify a fixed number of data items.
4. In the VV format, specify the current number of elements and the specified number of element data items.

(e) Specification examples of arrayed data in variable-length data type

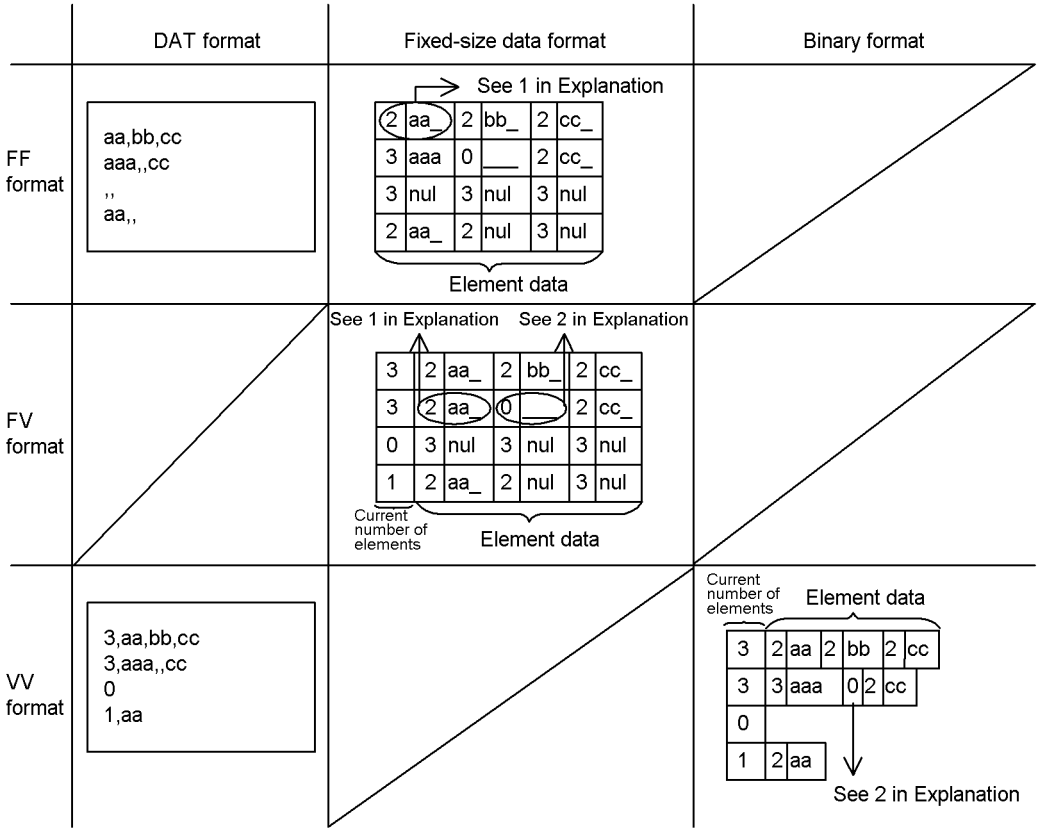
When specifying a variable-length data type (VARCHAR, MVCHAR, or NVARCHAR) in the fixed-size data or binary format, always specify the defined length of data in the FF or

FV format.

Pad the valid data with any fill characters at the end. In the VV format, data is not padded with fill characters.

Figure 5-15 shows specification examples of arrayed data in the variable-length data format. These examples are based on columns of `VARCHAR(3) ARRAY[3]`. The 3-byte character `null` is specified as the null value.

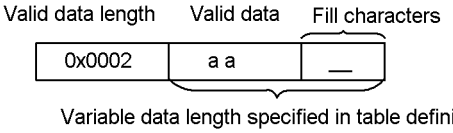
Figure 5-15: Specification examples of arrayed data in variable-length data format



_: One fill character.

Note:
For a DAT format file, each row of input data is followed by a line feed; for a fixed-size data or binary format file, row data is specified successively.

Explanation:
1. For an FF or FV format, specify variable-length element data in the following format:



2. Specify a 0-byte VARCHAR. Fill characters are not used in the VV format.

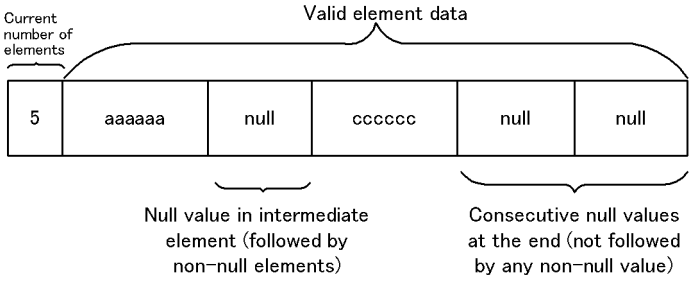
(2) Handling of null values for columns and elements

A repetition column can have a null value as a column or a null value as an element.

The database load utility can store either type of data in a database.

When data is loaded into a repetition column, element null values in the input data file can occur in the following pattern. Figure 5-16 shows input data for a repetition column and available types of null value.

Figure 5-16: Input data for a repetition column and available types of null value



In the example shown in Figure 5-16, you can specify how to handle null values in the null value option (with the null value option in the `array` statement or with the `nullset` operand in a column structure information file and a null value/function information file) as follows: treat the entire column as a null value if all element data are null values; or, do not treat the consecutive null values at the trailing end of the element data as valid data, so that they are not stored in the database (not included in the number of elements).

Table 5-21 shows the handling of null values by the database load utility (`pdload`) when data is loaded into a repetition column.

Table 5-21: Handling of null values by the database load utility (pdload) during data loading into a repetition column

Item		Handling of null values	
		FF format	FV and VV format
Null value testing method	DAT format	Tested as with a regular column (such as by reading consecutive delimiters); the elements passing the test are treated as null values.	
	Binary format	Tested as with a regular column (by specifying null-equivalent values in a column structure information file); the elements passing the test are treated as null values.	
	Fixed-size data format	Tested as with a regular column (by specifying null-equivalent values in a null value/function information file); the elements passing the test are treated as null values.	

Item		Handling of null values	
		FF format	FV and VV format
Storing null values in database	Specifying a null value as a column	Intermediate null values are stored in the database. For the treatment of consecutive trailing null values or when all specified elements are null values, see (a) *Handling of null values in FF format*.	If the specified number of elements is 0, the system stores the column null values. Even if all element data is null values, the system assigns null values equal to the current number of elements without treating them as column null values.
	Specifying a null value as an element	Intermediate null values are stored in the database. For the treatment of consecutive trailing null values or when all specified elements are null values, see (a) *Handling of null values in FF format*.	If both intermediate null values and consecutive trailing null values are within the current number of elements specified in input data, the system stores the null values in the database.

(a) Handling of null values in FF format

With the FF format, the current number of elements is not specified in input data. Therefore, if there are consecutive trailing null values, you can use the following operands to treat the null values as invalid element data so that they are not stored in the database:

- `nullset` operand in the `array` statement in the control information file
- `nullset` operand in the column structure information file
- `nullset` operand in the null value/function information file

In a column structure information file or a null value/function information file, specifications are made for each column; in the `array` statement in a control information file, specifications are made for each table. If you specify all three options, the specifications in the column structure information file or the null value/function information file take precedence.

If the `nullset` operand is omitted, the system assumes `nullset=c`.

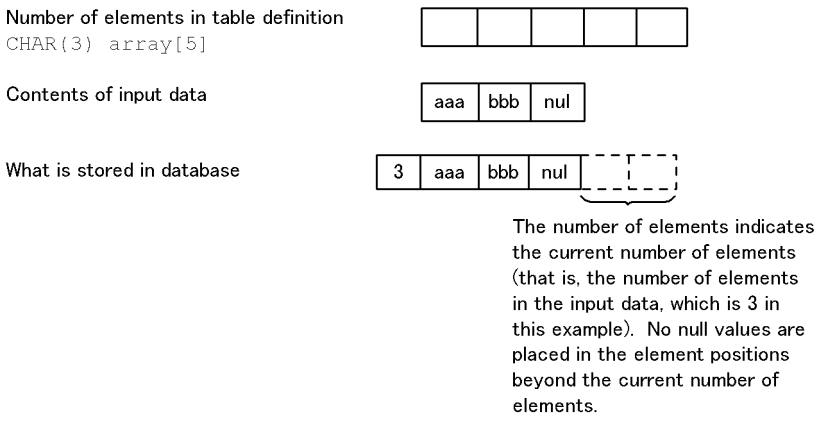
Table 5-22 shows null value option specification values and methods for storing null values in a database.

Table 5-22: Null value option specification values and methods for storing null values in a database using the FF format

Null value type	Null value option specification value	
	c (default)	e
Intermediate null value	The system stores a null value in applicable elements.	The system stores a null value in applicable elements.
Consecutive trailing null values (preceded by at least one valid element data item)	The system treats applicable elements (consecutive trailing null values) as invalid data and does not store them in database.	The system stores a null value in applicable elements (consecutive trailing null values).*
Consecutive trailing null values (all element data null)	The system treats the column as null.	The system stores a null value in all elements.*

Note

When using the FF format, the number of element data items in the input data file may be less than the number of elements specified in the table definition, in some cases. In this case, no data is specified beyond the current number of elements in the input data file. This means that the element data is not present, not that it has null values; therefore, the system does not store null values in the database, even when `nullset=e` is specified.



* Specifying `nullset=e` causes null values equal to the number of element data items in the input file to be stored in the database. Therefore, you should specify `nullset=c` for database storage efficiency. You can use the `nullset=e` operand to control the number of null values.

(3) Relationship between the number of elements in input data and the number of elements specified in table definition

For the FF or FV format, you can specify a desired number of element data items in the input data up to the maximum number of elements specified in the table definition. You can use the following operand to specify the number of element data items (current number of elements) in the input data:

DAT format and fixed-size data format

`element` operand in the column structure information file

Binary format

`element` operand in the null value/function information file

If the number of elements in the input data is equal to the number of elements specified in the table definition, there is no need to specify the previous operand, in which case the system assumes the number of elements specified in the table definition.

Table 5-23 shows what is stored and what is not stored in the database if the number of elements specified for a repetition column in the input data is different from the number of elements specified in the table definition.

Table 5-23: What is stored and what is not stored in the database if the number of elements specified for a repetition column in input data differs from the number of elements specified in table definition

Number of repetition element data items in input data: Number of repetition elements specified in table definition	Arrayed data format for repetition column		
	FF format	FV format (current number of elements specified in data)	VV format (current number of elements specified in data)
Number of elements in input data > number of elements in table definition	Error	Error	Error
Number of elements in input data = number of elements in table definition	Data is stored normally	Data stored normally (element data equal to the number of elements specified in input data is stored in database)	Data is stored normally

Number of repetition element data items in input data: Number of repetition elements specified in table definition	Arrayed data format for repetition column		
	FF format	FV format (current number of elements specified in data)	VV format (current number of elements specified in data)
Number of elements in input data > number of elements in table definition	Data is stored normally (data equal to the number of elements specified in `element` operand is stored in database; any subsequent elements are not stored)	Data is stored normally (element data equal to the number of elements specified in input data is stored in database)	Data is stored normally

(4) Specification example

(a) DAT format

Figure 5-17 shows a specification example of DAT-format data.

Table definition:

```
CREATE TABLE T1 (C1 INTEGER, C2 CHAR(1) ARRAY[10])
```

Figure 5-17: Specification example of DAT-format data (repetition column)

Input data file
(FF format, 7 specified as the number of elements, null value option omitted)

```
1, a, b, c, d, e, f, g                                                    1
2, a, b, c, d, e, ,                                                       2
3, a, b, , d, e, , g                                                      3
4, , , , , , ,                                                            4
```

In each row, the first element is data C1. Elements 2 through 8 are data in repetition column C2. If the number of elements is not specified, data items equal to the maximum number of elements (10) must be specified.

Explanation:
1. Stores specified values in elements 1–7.
2. Stores specified values in elements 1–5. Because the null value option is omitted, elements 6 and 7 are not stored in the database.
3. Stores specified values in elements 1–7, storing nothing in subsequent elements. The system stores a null value in elements 3 and 6.
4. A null value is specified for all specified elements. Because the null value option is omitted, the system stores null values in the column.

Input data file (VV format)

```
1, 7, a, b, c, d, e, f, g                                                 1
2, 5, a, b, c, d, e, ,                                                    2
3, 7, a, b, , d, e, , g                                                   3
4, 7, , , , , , ,                                                         4
5, 0                                                                      5
```

In each row, the first element is data C1. Elements 2 through 8 are data in repetition column C2. Because this is a VV format, the current number of elements is specified at the beginning of data C2. If the number of elements is not specified, data items equal to the maximum number of elements (10) must be specified.

Explanation
1. Stores specified values in elements 1–7 because the current number of elements is 7.
2. Stores specified values in elements 1–5 because the current number of elements is 5.
3. Stores specified values in elements 1–7 because the current number of elements is 7. The system stores a null value in elements 3 and 6.
4. Stores null values in elements 1–7 because the current number of elements is 7.
5. Stores null values in the column because the current number of elements is 0.

(b) Binary format

Figure 5-18 shows a specification example of binary-format data.

Table definition:

```
CREATE TABLE T1 (C1 CHAR(4), C2 CHAR(10), C3 CHAR(5) ARRAY[10])
```

Figure 5-18: Specification example of binary-format data (repetition column)

Input data file (concept diagram for VV format)

C1(CHAR(4))	C1(CHAR(10))	C3 (CHAR(5) ARRAY[10])					
Data	Data	Number of elements	Element data 1	Element data 2	Element data 3	Element data 4	Element data 5
0001	WWWWWWWWW	2	AAAAA	BBBBB			
0002	XXXXXXXXXX	5	AAAAA	BBBBB	CCCCC	DDDDD	EEEEE
0003	YYYYYYYYY	0					
0004	ZZZZZZZZZ	4	AAAAA	BBBBB	CCCCC	DDDDD	

Input data file (VV format with 'EEEEE' specified as the null comparison value)

```
3030303157575757575757575702004141414141424242424
 0 0 0 1 W W W W W W W W W W  2  A A A A A B B B B B
3030303258585858585858585805004141414142424242434343434444444444454545454
 0 0 0 2 X X X X X X X X X X  5  A A A A A  B B B B B C C C C C D D D D D E E E E E
3030303359595959595959590000
 0 0 0 3 Y Y Y Y Y Y Y Y Y Y  0
303030345a5a5a5a5a5a5a5a5a040041414141424242424343434344444444444
 0 0 0 4 Z Z Z Z Z Z Z Z Z Z  4  A A A A A B B B B B C C C C C D D D D D
```

1 1 2 2 3 3 4 4

CHAR(4) CHAR(10) CHAR(5) ARRAY[10]

Note 1: The upper row in the input data indicates actual data; the lower row indicates the value to be stored.

Note 2: The first 4 bytes in each row represent C1 data; bytes 5-14 represent C2; and data in repetition column C3 represents data that is stored beginning in byte 15. Because this is a VV format, data can be specified up to the maximum number of elements (10).

Explanation:

1. The system stores the specified values in elements 1 and 2.
2. The system stores the specified values in elements 1 through 5. Because the null comparison value is 'EEEEE', a null value is stored in element 5.
3. Because the value 0 is specified as the real number of elements, the system stores a null value in the column.
4. The system stores the specified values in elements 1 through 4.

(c) Fixed-size data format

Figure 5-19 shows a specification example of data in fixed-size data format.

Table definition:

```
CREATE TABLE T1 (C1 CHAR(4), C2 CHAR(10),
                 C3 VARCHAR(5) ARRAY[10])
```

Figure 5-19: Specification example of data in fixed-size data format (repetition column)

Input data file (concept diagram for FV format)

C1(CHAR(4))	C1(CHAR(10))	C3 (VARCHAR(5) ARRAY[10])						
Data	Data	Number of elements	Element data 1		Element data 2		Element data 3	
0001	WWWWWWWWW	2	3	AAA	5	BBBBB		
0002	XXXXXXXXX	3	2	AA	2	BB	4	CCCC
0003	YYYYYYYYY	1	4	AAAA				
0004	ZZZZZZZZZ	0						

Input data file (FV format with 3 specified as the number of elements and 'CCCC' specified as the null comparison value)

```
303030315757575757575757570200030041414100000500424242424200000000000000   1
 0 0 0 1 W W W W W W W W W  2   3  A A A     5  B B B B B                    1
3030303258585858585858585803000200414100000002004242000000040043434343 00  2
 0 0 0 2 X X X X X X X X X  3   2  A A        2  B B        4  C C C C      2
303030335959595959595959010004004141414100000000000000000000000000000000   3
 0 0 0 3 Y Y Y Y Y Y Y Y Y  1   4  A A A A                                   3
303030345a5a5a5a5a5a5a5a5a000000000000000000000000000000000000000000000000   4
 0 0 0 4 Z Z Z Z Z Z Z Z Z  0                                               4
```

CHAR(4) CHAR(10) VARCHAR(5) ARRAY[10]

Note 1: The upper row in the input data indicates actual data; the lower row indicates the value to be stored.

Note 2: The first 4 bytes in each row represent C1 data; bytes 5-14 represent C2; and data in repetition column C3 represents data that is stored beginning in byte 15. Because this is an FV format, data must be specified up to the specified number of elements (3 elements). If the specified number of elements is omitted, data must be specified up to the maximum number of elements (10 elements).

Explanation:

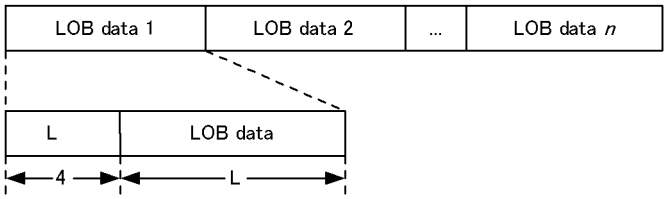
1. Because the real number of elements is 2, the system stores the specified values in elements 1 and 2. Element 1 is padded with 0s because its data length is less than the defined length. Element 3 is also padded with 0s because the real number of elements is 2.
2. Because the real number of elements is 3, the system stores specified values in elements 1 through 3. Because the null comparison value is 'CCCC', a null value is stored in element 3. Elements 1, 2, and 3 are padded with 0s because their data length is less than the defined length.
3. Because the real number of elements is 1, the system stores a specified value in element 1. Element 1 is padded with 0s because its data length is less than the defined length.
4. Because a value of 0 is specified as the real number of elements, a null value is stored in the column.

5.5.5 LOB column input file

(1) Creating a LOB column input file

Figure 5-20 shows the format of a LOB column input file.

Figure 5-20: Format of LOB column input file



L: LOB data length (0 ≤ L ≤ defined length)

Create as many LOB data items as there are input lines in the order of input data in the LOB column structure base table. With some LOB data, you need to specify the following information:

- To store 0 or more bytes of LOB data

  Specify the LOB data according to the format shown in Figure 5-20. If $L = 0$, specify only the 4-byte LOB data length.

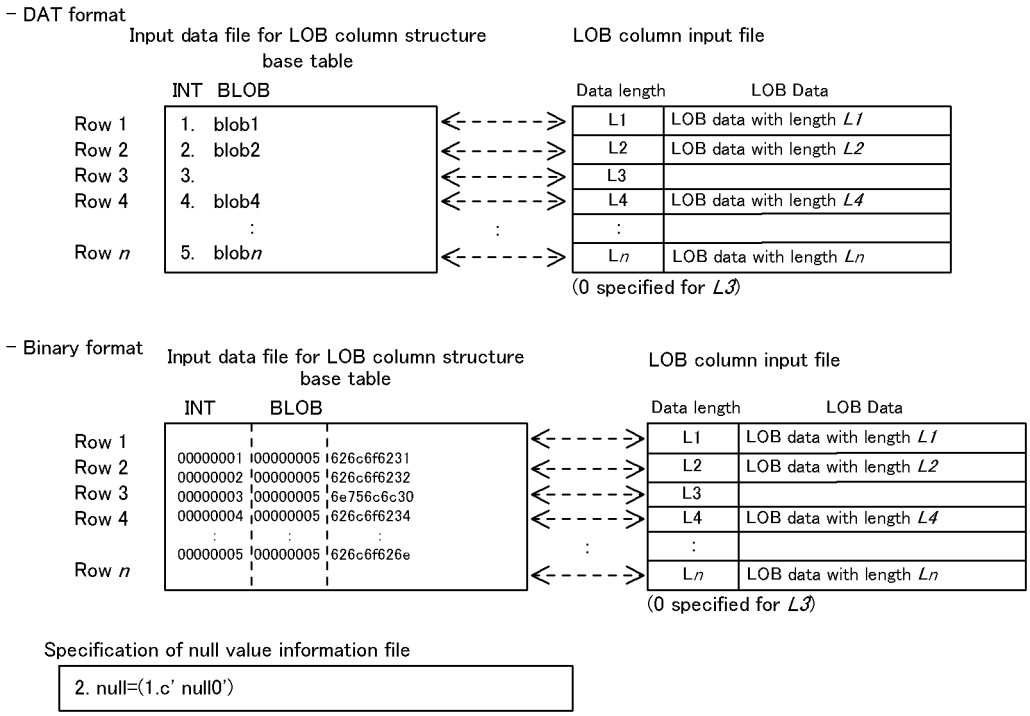
- To store the default or null value

  Specify only the 4-byte LOB data length based on $L = 0$ (whether or not the LOB data is the null value is determined by the specification of the row data of the corresponding LOB column structure base table).

Establish correspondence between the row data in the LOB column structure base table and the LOB data in the LOB column input file based on the specification sequence of the LOB data; i.e., the data in row *n* of the LOB column structure base table corresponds to data item *n* in the LOB column input file.

Figure 5-21 shows an example of correspondence between the row data in the LOB column structure base table and the LOB data in the LOB column input file. In this example, the LOB value in row 3 is treated as the null value.

Figure 5-21: Example of correspondence between row data in LOB column structure base table and LOB data in LOB column input file



(2) Notes about the LOB column input file

(a) Correspondence between specification of the input data file for the LOB column structure base table and LOB data in the LOB column input file

An error results if the specifications of the input data file for the LOB column structure base table do not agree with the specifications of the corresponding LOB data in the LOB column input file.

This type of error occurs in the following cases:

- **DAT-format input data file for LOB column structure base table**

  An error results if at least one byte of LOB data is specified in the LOB column input file, when the corresponding specification in the input data file for the LOB column structure base table is the null value or the default value.

- **Binary-format input data file for LOB column structure base table**

  An error results if at least one byte of LOB data is specified in the LOB column input file, when the corresponding specifications in the input data file for the LOB column structure base table and the null value/function information file match and

are the null value.

(b) Notes about the number of data items in the input data file for the LOB column structure base table and the number of data items in the LOB column input file

When loading LOB data using a LOB column input file, the database load utility checks the number of data items in the LOB column input file. However, the utility may skip this checking due to the specification of an option or due to the existence of erroneous data in the input data file. When this happens, the utility issues the `KFPL16328-W` message.

Table 5-24 shows the relationship between the specification of the option and the existence of erroneous data in an input data file.

Table 5-24: Relationship between specification of option and existence of erroneous data in input data file

Option		Existence of erroneous data in input data file for LOB column structure base table	
		No	Yes
`-e` option	Not specified	Y	Y[1]
	Specified	Y	—[2]

Y: The utility checks the number of data items in LOB column input file.

—: The utility does not check the number of data items in LOB column input file.

[1] The utility outputs the LOB column information from the input data file to a LOB middle file for the rows that did not result in an error, then checks the number of data items in the LOB column input file.

To use an error data file later to load the row data for the LOB column structure base table that resulted in an error, you need to prepare another LOB column input file that corresponds to the input file for the LOB column structure base table that is subject to subsequent data loading. You cannot use the initial LOB column input file as is.

[2] The utility outputs the corresponding LOB column information to a LOB middle file up to the data immediately preceding the erroneous row in the input data file for the LOB column structure base table.

Consequently, the utility outputs the LOB column information to a LOB middle file as is, even if the LOB data is shifted (due to invalid or missing data) in the LOB column input file that corresponds to the input data file for the LOB column structure base table up to the data immediately preceding the erroneous row.

If you load the remaining row data in the addition mode specifying the `-r` option after

correcting the erroneous data in the input data file for the LOB column structure base table, the utility loads data through the last row in the input data file for the LOB column structure base table, then checks the number of data items in the LOB column input file for the first time during the LOB column creation process.

If this data item count checking results in an error, you need to take one of the following actions according to the location of the error in the LOB column input file:

(a) Error within in the range of data in the LOB column input file being loaded

You can load the LOB column by eliminating the cause of the error in the LOB column input file.

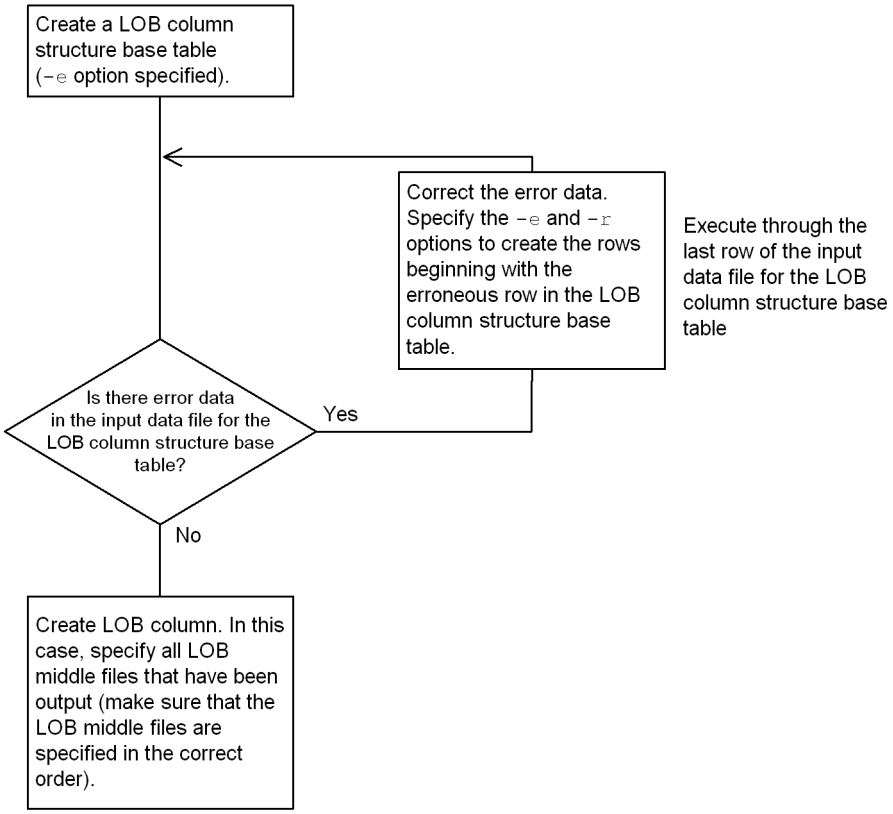
(b) Error beyond the range of data in the LOB column input file being loaded

You need to eliminate the cause of the error in the LOB column input file and load the LOB data in the LOB column input file that was previously being loaded as well as the LOB data that has already been loaded.

In this case, check to see if the LOB column structure base table to be loaded is a partitioned table or a non-partitioned table, and also check whether the loading units for the LOB column structure base table are in tables or in RDAREAs. Then, re-initialize the erroneous user LOB RDAREAs or all the user LOB RDAREAs corresponding to the erroneous LOB column, restore them from their backup copy, then load the LOB column.

Figure 5-22 shows a simple procedure for handling a data item count error in the LOB column input file.

Figure 5-22: Simple procedure for handling a data item count error in the LOB column input file



5.6 Error information file

In the event of an input data error, the error information is output to the error information file.

5.6.1 Contents of error information file

The system outputs a logical error number indicating the cause of an error to the error information file. This logical error number (*XX*) corresponds to the KFPL310*XX* message.

Table 5-25 shows lists and explains the logical error numbers for logical errors.

Table 5-25: Logical error numbers and descriptions

Logical error number	Description of logical error
1	Input data for NCHAR or NVARCHAR is invalid: • Length of the input data is an odd value
2	• The numeric or sign part of packed-format decimal numeric data is invalid. • The numeric part of date, time, or time stamp data is invalid, or the length of the fraction part of time stamp data is invalid. • The numeric or sign part of date interval or time interval data is invalid.
3	The length of variable-length data (VARCHAR, NVARCHAR, MVARCHAR, BLOB, or BINARY) is invalid.
4	Data conversion error occurred.
5	• Input data does not match the table in terms of the number of columns. • The column, not enclosed in double quotation marks ("), cannot be recognized. • A null character (0x00) is found as part of data in a text (DAT format) file.
6	Null value was specified for a table or column for which null value is prohibited: • Null value was specified for a table with FIX attribute. • Null value was specified for a cluster key component column. • Null value was specified for a column with NOT NULL attribute. • Null value was specified for the primary key component column.
7	Input records are not sorted by the cluster key.
8	For a table for which a cluster key or index has been defined, the following error occurred: • Duplicate key is prohibited, but a specified key is duplicated.
9	A column value was found that did not satisfy the partitioning conditions for the table created (being processed).
10	The row length of binary-format input data is invalid.

Logical error number	Description of logical error
11	The LOB data is invalid: • The length of the file name of the LOB input file is invalid that was specified in the input data file for the LOB column structure base table. • Invalid LOB data length was specified for a LOB column input file or a binary-format file. • An invalid EOF was detected while acquiring the 4-byte LOB data length in the LOB column input file or while acquiring the LOB data with the specified length.
12	The specified LOB input file was not found.
13	The length of the LOB data exceeds the defined length.
14	There is a contradiction between the specifications of the LOB column structure base table and the specifications in the LOB input file: • One or more bytes of data are specified for a null value specification. • One or more bytes of data are specified for a default value.
15	The numbers of data items do not match between the LOB column structure base table and the LOB column input file.
16	Invalid parameter for a constructor function
17	No access privilege for LOB input file
18	Invalid data format for storage in abstract data type
19	Invalid number of data elements for storage in repetition column
20	There is an enclosing character at an invalid location.
90	The input data is longer than the defined column length. When the `lengover` operand is omitted: The system discards the excess portion of the data and then stores the data. When the `lengover` operand is specified: The system does not store any data because it assumes an input data error.

Notes

1. If the `-e` option is not specified and the length of specified variable-length data is invalid, the system treats the specified data length as being invalid and skips the corresponding row data.
2. If there is an error in a binary-format input data file (excluding an input data file output by the database reorganization utility), the system outputs a dump image listing of the first erroneous input data in hexadecimal and text format (2-byte codes are ignored). The output format depends on the data that is contained, as shown below:

• For data in a `BINARY`-type column or abstract data-type column with the

BINARY attribute

If the data length exceeds 1,024 bytes, the system outputs the length section with a length of 4 bytes and the data section with a maximum length of 1,024 bytes from the beginning. After a linefeed, the system outputs ： at the beginning and then outputs the remaining column data. The following shows an output example:

```
 **** First error data [CODE:4] ****
41424344 00000001 61620001 0000001c    ABCD....ab......
ffffffff 00002000 00010002 00030004    ................
00050006 00070008 0009000a 000b000c    ................

                    ⋮

03fd03fe 03ff0400                      ................
   :
46364                                   ....cd
```

- For data in a BLOB-type column

  The data is not output.

5.6.2 Output example of error information file

This section presents an output example of error information file.

This dump image listing shows the row data in line 10 that resulted in a data conversion error.

```
pdload (03-00) **** DB LOAD **** 1996-05-07 12:34:56   single   ........1
KFPL31004-E Data conversion error occurred,
            line=10, column=4   ...........................................2
KFPL31002-E Invalid pack format value,
            line=77, column=5   ...........................................2
KFPL30000-I Server name=SDS   .............................................3
KFPL30001-I Rows successfully loaded=98   .................................4

**** First error data [CODE:4] ****   .....................................5
41424344 00000001 61620001 0000001c  ABCD ... ab ..
ffffffff 6364                        ... cd   ............................6
```

Explanation:

1. Header
2. Input data error message
3. Server name

4. Number of stored rows
5. Sub-header (code number is the logical error number)
6. Dump image listing of the first error data (hexadecimal and ASCII characters)

Following are the contents of the error information file, depending on whether or not the `-e` option is specified:

-e option	Input data error	Error information file contents
Specified	No	—
	Yes	1 - 6
Not specified	No	1, 3, 4
	Yes	1 - 6

—: No information is output.

Note

The numbers shown as the contents of the error information file correspond to the numbers used in the error information file output example.

5.6.3 Notes about referencing error information

- For a binary-format input file for a non-FIX table that is output by the database reorganization utility, no error data is subject to dump image listing output or error data file output to an error information file.
- For a binary-format file, the following types of input data error prevent columns beginning with the affected column from being output. Consider this when referencing an error data file:
  - An error in the variable-length character string or BINARY column
  - Input data column length less than a defined length for that column
- For a DAT-format file, data with a row length greater than the value of `maxreclen` operand of the `source` statement (if the `maxreclen` operand is omitted in the `source`, a default value of 32 KB) is not output to an error data file.
- Information is output to an error data file not in the order in which data is stored in the input file, but in the order in which error messages are output to an error information file.
- For a binary-format file, the dump image listing from an error information file is not output for a table containing LOB columns, columns of an abstract data type, or repetition columns. If the length section of a variable-length character or

BINARY column is invalid, data is output only up to the character string preceding the erroneous variable-length character or BINARY column.

- If data that has not been output remains in the error data file, only one warning message is issued, irrespective of the number of data items remaining in that file.
- If an input data error is detected when the input data file is in the user-created binary format or fixed-size data format, a dump image listing of the input data in which the first error was detected is output to the error information file. If the work area (memory) required for writing the dump image listing cannot be acquired, the KFPL25222-W message is displayed. In such a case, the system resumes data loading without output of the dump image listing.

  Table 5-26 shows whether or not a dump image listing is output when an input data error is detected.

  Table 5-26: Whether or not a dump image listing is output when an input data error is detected

Input data file format	Output of dump image listing	Work memory allocation	
		Allocatable	Unallocatable
DAT format	N	●	●
Extended DAT format	N	●	●
Fixed-size data format	Y	Outputs a dump image listing and then resumes data loading.	Resumes data loading without outputting a dump image listing.
Binary format	Y		
pdrorg-generated binary format	N	●	●

Legend:

Y: Output

N: Not output

●: Not applicable

5.6.4 Rules for output when the errdata operand is specified in a source statement

This section describes the rules of error data file output and dump image listing output to the error information file for the corresponding data when the errdata operand is specified in the source statement.

- The following table shows whether or not the files are output for a HiRDB/ Parallel Server with a server name specified in the source statement different

from a back-end server containing a table storage RDAREA, and for a HiRDB/Parallel Server with a table storage RDAREA that extends over multiple back-end servers:

Relationship with error data file creation work buffer	Format of input data file	Input data error type	
		Duplicate key error or invalid value for storage in abstract data type	Other errors
Insufficient buffer or `errwork=0`	DAT format	—	N
	Fixed-size data format	—	Y
	Binary format	—	Y
	Binary-format input file for a non-FIX table that was output by the database reorganization utility	—	—
Sufficient buffer	DAT format	N	N
	Fixed-length data format	N	Y
	Binary format	N	Y
	Binary-format input file for a non-FIX table that was output by the database reorganization utility	—	—

Y: Outputs an error data file and a dump image listing for an error information file.

N: Outputs an error data file but not a dump image listing for an error information file.

— : Outputs neither an error data file nor a dump image listing for an error information file.

Note

If some of the error data fails to be output to the error data file due to a buffer shortage, a message `No output for error data` is attached to the error message before it is output to an error information file. In this case, specify a large enough buffer size in the `errwork` operand to ensure the output of error data.

- The following table shows whether or not the files are output for a HiRDB/Parallel Server with a server name specified in the `source` statement that matches a back-end server containing a table storage RDAREA:

Format of input data file	Input data error type	
	Duplicate key error or invalid value for storage in abstract data type	Other errors
DAT format	N	N
Fixed-size data format	Y	Y
Binary format	Y	Y
Binary-format input file for non-FIX table that was output by database reorganization utility	—	—

Y: Outputs an error data file and a dump image listing for an error information file.

N: Outputs an error data file but not a dump image listing for an error information file.

—: Outputs neither an error data file nor a dump image listing for an error information file.

5.7 Column structure information file

A column structure information file is specified in the following cases:

Item	Format of input data file			Explanation
	DAT format	Fixed-size data format	pdrorg-generated binary format	
Storing data with a column sequence different from input data file and table	Y	Y	Y	1
Storing data with a column count different from input data file and table	Y	Y	Y	2
Storing data in the input data file after converting it to the data type of the column in the table	—	Y	NS	3
Storing a column value in the input data file that matches the null comparison value as the null value	—	Y	Y	4
Specifying a constructor function that generates values to be stored in abstract data type	Y	Y	NS	5
The number of elements for repetition columns in the input data file is less than the maximum number of elements in table definition	Y	Y	—*	6

Y: Specifiable.

NS: Not specifiable.

—: Not applicable

* If the arrayed data format is VV, you cannot specify the number of elements.

Explanation:

1. Specify column name statements in the order of corresponding input data.
2. If some of the input data is not to be stored in the table, specify a `skipdata` statement for such data. If there is no input data to be stored in the table, omit the `column name` statement.
3. Specify the data type of the table column in the corresponding column name statement.

4. Specify the comparison value (input data) to be treated as the null value in the column name statement.
5. Specify the constructor function name and parameter type in the column name statement. If the name of the constructor function that generates the data to be stored in the abstract data type is different from the name of the column data type, or if there are multiple functions with the same name, be sure to specify the column structure information file. For the fixed-size data format, also specify the data type of the input parameter and the null comparison value.
6. Specify the number of input data elements in the column name statement. For details about the column name statement, see *5.7.1 Column name statement*; for details about the skipdata statement, see *5.7.2 skipdata statement*.

Rules

1. If an input data file is in the fixed-size data format, be sure to specify a column structure information file. For an input data file in the DAT format or the `pdrorg`-generated binary format, you can specify a column structure information file as required.

   For an input data file in the `pdrorg`-generated binary format, you can specify a column structure information file only for a non-FIX table. Specifying a column structure information file for a FIX table results in an error. To reorganize the column structure of a FIX table using a column structure information file, handle the input data file as being in the fixed-size data format.
2. Specify each column name statement and skipdata statement on a single line. If you need more than one line and the last character in the line is a comma (`,`) for separating operands, specify at least one byte of space or tab between the comma and the continuation character (`$`), and specify a continuation character at the end of the line. In this manner, you can specify information over more than one line.
3. When specifying a column structure information file, also specify the `-c` option.
4. If you are creating a LOB column structure base table separately from LOB columns and specify a column structure information file during the creation of the LOB column structure base table, you need to specify a column structure information file containing the same information also during the creation of the LOB columns. Otherwise, the LOB column creation process may result in an error.

5.7.1 Column name statement

(1) Format

```
column-name[{[,type=data-type
              [,null={'numeric-comparison-value'|(begin-position,{c|x}'comparison-value')}]]
              [,element=elements-count][,elmtype=arrayed-data-format]
              [,nullset=null-value-option][,mode={text|bin}]
              [,enclose_del={yes|no}][,filldata=filler-data]
            |[,func=([authorization-identifier.]function-name
              ,param=argument-type[,type=data-type
              [,null={'numeric-comparison-value'|(begin-position,{c|x}'comparison-value')}]]
              [,mode={text|bin}][,enclose_del={yes|no}]
              [,filldata=filler-data]
              [,param=argument-type[,type=data-type
                [,null={'numeric-comparison-value'|(begin-position,{c|x}'comparison-value')}]]
                [,mode={text|bin}][,enclose_del={yes|no}]
                [,filldata=filler-data]]...)]
          }]
```

Some of the column name statement's operands are not specifiable depending on the format of input data file. The following table shows whether each operand is specifiable for each format.

Type of specified column	Operand	Input data format		
		DAT format	Fixed-size data format	pdrorg-generated binary format
Predefined type (column specification)	column-name	R	R	R
	type	N	R	O[3]
	null	N	O	O
	element	O	O	N
	elmtype	O	O	N
	nullset	O	O	N
	mode	N	O[4]	N
	enclose_del	N	O[5]	N
	filldata	N	O	N
Abstract data type (function specification)	column-name	R	R	R
	func	O[1]	O[1]	O[1]
	param	O[2]	O[2]	O[2]
	type	N	O[2]	O[3]
	null	N	O	O
	mode	N	O[4]	N
	enclose_del	N	O[5]	N
	filldata	N	O	N

R: Required

O: Optional

N: Not specifiable

[1] Required if a function is specified.

[2] Required if the `func` operand is specified with the function.

[3] Specifiable if the data type of the column or the data specified in `param` with function specification is `BLOB` or `BINARY`. If any one of the following conditions is applicable,

make sure that the operand is specified:

- Loading unload data from a BLOB column to a column of an abstract data type for which the BINARY or BLOB parameter has been specified
- Loading unload data from the BLOB parameter of an abstract data type to a column of the BLOB or BINARY type
- Loading unload data from a BLOB column to a BINARY column
- Loading unload data from the BLOB parameter of an abstract data type to a column of the abstract data type for which the BINARY parameter has been specified

[4] Specifiable only when the type operand value is char and the table definition is CHAR or MCHAR.

[5] Specifiable only when the type operand value is char and the table definition is VARCHAR or MVARCHAR.

(2) Operands

(a) column-name

Specifies the name of the table column in which input data is to be stored.

If the *column-name* is enclosed in double quotation marks ("), it is treated as being case sensitive; otherwise, it is treated as all uppercase letters. If the *column-name* contains a space, enclose the entire *column-name* in double quotation marks.

(b) type=data-type

Specifies the data type of the column in the input data. For the data types for columns in input data, see (3) *Specification of data types and argument types*. To determine whether or not the input data is stored in the database when the data type specified in the column structure information file does not match the data type of the table column, see (4) *Data storage in the database when the data type specified in the column structure information file does not match the data type in the table*.

(c) null={'numeric-comparison-value'|(begin-position,{c|x}'comparison-value')}

Specifies that the null value is to be stored in specified columns of the table.

Because the column data is compared with the input data, specify a comparison value that corresponds to the data type specified in the type operand.

The available comparison method depends on the type of input data. The following table shows the comparison method for each type of input data:

Input data type	Comparison method		
	Numeric comparison	Character comparison	Hexadecimal comparison
DECIMAL	Y	N	Y
DATE	Y	N	Y
INTERVAL YEAR TO DAY	Y	N	Y
TIME	Y	N	Y
INTERVAL HOUR TO SECOND	Y	N	Y
TIMESTAMP	Y	N	Y
CHAR	N	Y	Y
VARCHAR	N	Y	Y
NCHAR	N	Y	Y
NVARCHAR	N	Y	Y
MCHAR	N	Y	Y
MVARCHAR	N	Y	Y
BINARY	N	Y	Y
BLOB	N	Y	Y
INTEGER	Y	N	Y
SMALLINT	Y	N	Y
FLOAT	Y	N	Y
SMALLFLT	Y	N	Y
yymmdd format	Y	N	Y
yy-mm-dd format	N	Y	Y

Legend:

Y: Can be specified

N: Cannot be specified

Rules

1. The null value is stored in the columns containing the value specified in this

operand.

2. This operand is not applicable to a column with the NOT NULL attribute, a cluster key component column, or a primary key component column.
3. For an input data file in pdrorg-generated BINARY format, the null value is specified by setting the offset to 0 for LOB and BINARY column data and the LOB and BINARY parameters of a abstract data type; therefore, you cannot specify this operand for these columns. When these data types have real data, to treat them as the null value by setting the offset to 0, you must first have deleted the real data part from the corresponding input data (simply editing the corresponding offset to 0 does not result in correct processing).
4. For an input data file in fixed-size data format, the system performs comparison after dividing the data into columns; thus, all trailing spaces are deleted before data comparison.

numeric-comparison-value ∼ ((up to 255 characters))

For numeric input data, this operand specifies the numeric value to be compared.

Rules

1. The specification method is the same as for numeric data in the DAT format, but do not include spaces.
2. The numeric attribute applies to the INTEGER, SMALLINT, FLOAT, SMALLFLT, DEC, DATE, TIME, INTERVAL YEAR TO DAY, INTERVAL HOUR TO SECOND and TIMESTAMP data types, and to the *yymmdd* format.
3. If a numeric value of TIMESTAMP type is specified and the fraction part is shorter than the defined value, the system assumes 0.

Example

- DATE: *yyyy-mm-dd*

  Example: January 1, 1995 → 1995-01-01
- TIME: *hh:mm:ss*

  Example: 12:01:01 → 12:01:01
- INTERVAL YEAR TO DAY: *yyyymmdd*.

  Example: 1900 years, 1 month, and 1 day → 19000101.
- INTERVAL HOUR TO SECOND: *hhmmss*.

  Example: 12 hours, 1 minute, and 1 second → 120101.

- TIMESTAMP: *yyyy-mm-dd hh:mm:ss.nnnn*

  Example: 1900-01-01 at 01:10:30:45

  → 1900-01-01 01:10:30.4500 or 1900-01-01 01:10:30.45

(*begin-position*,{c|x}'*comparison-value*')

For a character comparison value, specify this information when the input data has the character attribute or is binary or LOB data. The character attribute applies to the CHAR, NCHAR, MCHAR, VARCHAR, NVARCHAR, and MVARCHAR data types and to the *yy-mm-dd* format.

For a hexadecimal comparison value, this information can be specified even if the input data has the numeric attribute (comparison is available with all data types).

begin-position ～ <unsigned integer> ((1-32000))

Specifies in bytes the beginning position of the storage data for the target column that is to be compared with the comparison value in order to determine whether or not the input data is treated as the null value.

The system compares the portion of the input data beginning at the specified position with the specified comparison value. If they match, the system treats the input data as the null value. The following shows examples:

Input data of CHAR(5)	Result of comparison with null=(2,'XYZ')	Handling of input data
XYZ	Perfectly matches the comparison value.	Treated as the null value.
XYZ**	Matches the comparison value, but the beginning position does not match.	Treated as the input data.
*XY**	Does not match the comparison value.	

c:

Specifies that the comparison value is a character string.

If the number of characters specified for *begin-position* + *comparison-value* is longer than the defined length of the corresponding column, a control statement error results.

x:

Specifies that the comparison value is a hexadecimal value.

Two hexadecimal characters (0 to f) are treated as 1 byte.

The specified characters are not case sensitive.

If the specified character string does not consist of hexadecimal characters or if the number of characters specified for *begin-position* + *comparison-value* is longer than the defined length of the corresponding column, a control statement error results.

For a numeric-type column, the comparison value may be the same as the input data. In such a case, specify the comparison value as hexadecimal characters. The following shows examples:

Data type of column	Value treated as the null value	Specification following the part null=
DECIMAL	Character string NULL	(1,x'4e554c4c')
DATE	0000-00-00	(1,x'00000000')
INTERVAL YEAR TO DAY	Byte 1 is 0xff.	(1,x'ff')
TIME	99:99:99	(1,x'999999')
INTERVAL HOUR TO SECOND	Byte 2 is 0xff.	(2,x'ff')
TIMESTAMP	Bytes 2 and 3 are 0xff.	(2,x'ffff')
INTEGER*	Character string NULL	(1,x'4e554c4c')
SMALLINT*	Character string NL	(1,x'4e4c')
FLOAT*	Character string **NULL**	(1,x'2a2a4e554c4c2a2a')
SMALLFLT*	Character string NULL	(1,x'4e554c4c')

* A specified hexadecimal comparison value may match some value (in the example of SMALLINT, this comparison value matches the big-endian 20044). To treat the value as the null value, you must specify a value that is not stored.

comparison-value ～ ((up to 255 characters))

Specifies the characters to be compared with the input data. If you are specifying x, be sure to specify an even number of characters (with a maximum of 254 characters).

(d) element=elements-count

～ <1 to number of elements specified for the corresponding column in table definition>

For a repetition column, specify the number of elements to be specified in the input data file.

Specify the `element` operand if the number of elements specified in the input data file is less than the number of elements specified for the corresponding column during table definition.

If the `element` operand is omitted, the system assumes the maximum number of elements specified for the corresponding column during table definition.

Rules

1. If the arrayed data format is the VV format, the system always uses the number of elements specified in the input data file. Therefore, this operand is ignored whether or not it is specified.
2. An error results if you specify this operand for an input data file in the `pdrorg`-generated binary format.

(e) elmtype=arrayed-data-format

For a repetition column, specify its array data format.

arrayed-data-format

Specify one of the following formats:

- `ff`: FF format
- `fv`: FV format
- `vv`: VV format

For details about the arrayed data format, see *5.5.4 Arrayed data format for tables containing repetition columns*.

Rules

1. If you omit the `elmtype` operand and also omit the `array` statement in the control information file, the system assumes the following format for the arrayed data format:
   - DAT format file: FF format
   - Binary format file: VV format
   - Fixed-size data format file: FF format
2. An error results if you specify this operand for an input data file in the `pdrorg`-generated binary format.

(f) nullset=null-value-option

This specifies how null value storage is to be handled if a null value is specified from an element through the end of specified elements for an FF arrayed data format.

null-value-option

c: Nothing is stored in specified element.

e: Null value is stored in specified element.

Rules

1. When the nullset operand is omitted, the system assumes the following value:
   - If an array statement is not specified in the control information file: c
   - If an array statement is specified in the control information file: Value specified in the nullset operand of the array statement
2. An error results for the binary-format input data file generated by pdrorg.

(g) mode={text|bin}

Specifies the input data storage method when the input data is a fixed-length character string (type=char(*n*)) and the column data type of the storage target is CHAR or MCHAR. This operand is applicable only to the fixed-size data format.

Criteria

To store the following characters as part of the input data, specify bin:

- Trailing single-byte spaces (0x20), tab characters (\t), and null characters (\0) in the input data
- Quotation marks (") at both ends of the input data

To delete the above characters from the input data, specify text (default value).

text:

Specifies that the input data is to be edited as follows and then stored:

- Delete the trailing single-byte spaces (0x20), tab characters (\t), and null characters (\0) from the input data.
- Delete the quotation marks (") at both ends of the input data.
- If the input data is shorter than the defined length of the storage column, pad the column with single-byte spaces.

bin:

Specifies that the input data is to be edited as follows and then stored:

- If the input data is shorter than the defined length of the storage column, pad the column with single-byte spaces.

When bin is specified, space conversion cannot be performed (the spacelvl operand cannot be specified in the option statement).

(h) enclose_del={yes|no}

Specifies the input data storage method when the input data is a fixed-length character string (`type=char(n)`) and the column data type of the storage target is `VARCHAR` or `MVARCHAR`. This operand is applicable only to the fixed-size data format.

Criteria

To store the quotation marks (`"`) at both ends of the input data as part of the input data, specify `no`.

To delete the quotation marks at the ends of the input data, specify `yes` (default value).

`yes`:

Specifies that the input data is to be edited as follows and then stored:

- Delete the trailing single-byte spaces (`0x20`), tab characters (`\t`), and null characters (`\0`) from the input data.
- Delete the quotation marks (`"`) at both ends of the input data.

`no`:

Specifies that the input data is to be edited as follows and then stored:

- Delete the trailing single-byte spaces (`0x20`), tab characters (`\t`), and null characters (`\0`) from the input data.

To store single-byte spaces, tab characters, and null characters as part of the input data, enclose them in quotation marks.

(i) filldata=filler-data

Specifies the trailing filler data that is to be used when `BINARY` data is specified as `CHAR` type in the fixed-size data format. The filler data is 1-byte data in the range 0-255 expressed in hexadecimal (`X'00"` to `X'ff'`). When this operand is omitted, `0` is assumed.

Rules

1. When converting data from `CHAR` to `BINARY`, the system treats the data without the consecutive trailing fillers as the data section of the `BINARY` data.
2. A control statement error results if this operand is specified for an input data file that is not in the fixed-size data format or for a column that is not the `BINARY` type.
3. Specifying this operand for a parameter other than a `BINARY`-type parameter results in a control statement error.

(j) func=([authorization-identifier.]function-name,param=argument-type[,type=data-type[,null={'numeric-comparison-value'|(begin-position,{c|x}'com

parison-value')}]][,null={'numeric-comparison-value'|(begin-position,{c|x}'comparison-value')}]][,mode={text|bin}][,enclose_del={yes|no}][,filldata=filler-data][,...])

If a specified column name has an abstract data type, this operand specifies information about the constructor function that generates the values to be stored.

For data storage in the database when the input parameter specified in the column structure information file has a different data type than the argument of the constructor function to be called, see (4) *Data storage in the database when the data type specified in the column structure information file does not match the data type in the table*.

`[`*authorization-identifier*`.]`*function-name*

When generating the data to be stored by calling a constructor function supplied by a plug-in, specify the name of the function.

You cannot specify a function that produces a return value different from the abstract data type of the column to be stored, or that has an abstract data type argument.

If the authorization identifier is omitted, the system assumes the user who defined the abstract data type.

`param=`*argument-type*

Specify the data type of the parameter that is to be passed to the constructor function.

If there are multiple parameters, specify them in the order based on the function's input format. For details about how to specify the argument types, see (3) *Specification of data types and argument types*. Note that you cannot specify either *yy-mm-dd* or *yymmdd*. `blobprm` is permitted only for an input data file in the `pdrorg`-generated binary format.

`type=`*data-type*

If the input data file is a fixed-size data format file, specify the data type of the parameter to be input. If there are multiple parameters, specify them in the order based on the function's input format. For details about how to specify the argument types, see (3) *Specification of data types and argument types*. Note that you cannot specify either *yy-mm-dd* or *yymmdd*. `blobprm` is permitted only for an input data file in the `pdrorg`-generated binary format.

`null={'`*numeric-comparison-value*`'|(`*begin-position*`,{c|x}'`*comparison-value*`')}`

For an input data file in the fixed-size data format, specify this operand to handle the input parameter as the null value.

The input parameter's data type is subject to comparison. For the specification method, see *(c)*

null={'numeric-comparison-value'|(begin-position,{c|x}'comparison-value')}, previously.

`mode={text|bin}`

For details about this specification, see *(g) mode={text|bin}*.

`enclose_del={yes|no}`

For details about this specification, see *(h) enclose_del={yes|no}*.

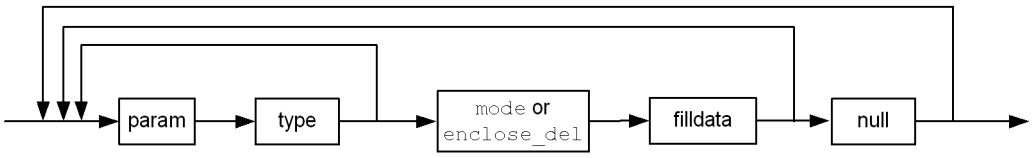
`filldata=`*filler-data*

For details about this specification, see *(i) filldata=filler-data*.

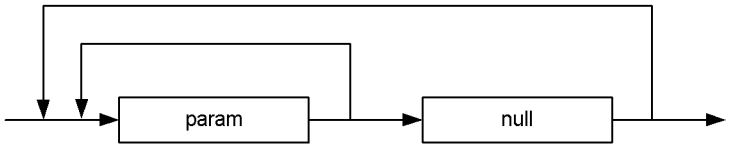
Specification order in func operand:

The following shows the order in which `param`, `type`, `null`, `filldata`, `mode`, and `enclose_del` are specified in the func operand:

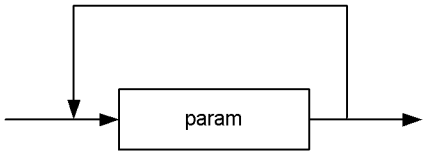
- In the fixed-size data format



- In the binary format



- In the DAT format



(3) Specification of data types and argument types

Table 5-27 shows the method for specifying data types and argument types.

Table 5-27: Specification of data types and argument types

Specification	Remarks
`integer`	—
`smallint`	
`char`(*n*)[1, 2]	*n* indicates the `CHAR` or `MCHAR` length in bytes; *m* indicates the `NCHAR` length in digits. These values may not match the column length defined for the target table. If converting characters to numeric values, do not include the null character (`\0`) in the character string. For the `param` operand, omit (*n*) and (*m*).
`nchar`(*m*)[1]	
`mchar`(*n*)[1]	
`varchar`(*n*)[1]	*n* indicates the maximum length of `VARCHAR` or `MVARCHAR` in bytes; *m* indicates the maximum length of `NVARCHAR` in digits. These values may not match the column length defined for the target table. For the `param` operand, omit (*n*) and (*m*).
`nvarchar`(*m*)[1]	
`mvarchar`(*n*)[1]	
`float`	—
`smallflt`	
`dec` (*precision* [, *decimal-places*])	For the `param` operand, omit the precision and scaling factor.
`date`	—
interval-year-to-day	
`time`	
interval-hour-to-second	
yy-mm-dd[1, 3]	You can use this format only when storing character data indicating the last two digits of the year in a `DATE`-type column. Any one-byte character can be specified as the separator between the year, month, and date (e.g., *yy-mm-dd* or *yy/mm/dd*).
yymmdd[3]	You can use this format only when storing the unsigned packed-format data indicating the last two digits of the year in a `DATE`-type column.
`timestamp`(*p*)	*p* indicates the length of the fraction part for seconds (0, 2, 4, or 6). For the `param` operand, omit `(`*p*`)`.
`binary`(*n*)	*n* indicates the maximum length of the `BINARY` type in bytes. For the `param` operand, omit `(`*n*`)`.

Specification	Remarks
binaryprm	For an input data file in pdrorg-generated binary format, you can use this type in the type operand. When skipping abstract data-type data, if the abstract data type is the BINARY-type parameter, use this type in the type operand of the skipdata statement. A control statement error results if this type is specified in an operand other than the type operand of the skipdata statement for an input data file in pdrorg-generated binary format.
blob(*n*)	• When specifying in the type operand for the fixed-size data format *n* indicates the maximum length of the LOB input file name and may not match the length defined for the LOB column. Use this format for binary-format data containing its length in the first four bytes. • When specifying in the type operand for the pdrorg-generated binary format Omit (*n*). • When specifying in the param operand Omit (*n*). • When specifying in the type operand of a skipdata statement Omit (*n*).
blobprm	For an input data file in the pdrorg-generated binary format, you can use this format in the type operand. When unloading an abstract data type, use this format for the BLOB parameter data that has been unloaded.

[1] Specify a left-justified column value.

[2] If the characters are to be converted to a numeric value, there is no need to left-justify them.

[3] The last two digits of the year are expanded to the format 19*yy* and then stored.

(4) Data storage in the database when the data type specified in the column structure information file does not match the data type in the table

Tables 5-28 and 5-29 show whether or not input data is stored in database if the data type specified in the column structure information file does not match the data type in the table.

Table 5-28: Data storage in the database when the data type specified in the column structure information file does not match the data type in the table (1/2)

Data type specified in column structure information file	Data type in table									
	IN	SI	CH	NC	MC	VC	NV	MV	FL	SM
integer	Y	—	—	—	—	—	—	—	—	—

Data type specified in column structure information file	Data type in table									
	IN	SI	CH	NC	MC	VC	NV	MV	FL	SM
smallint	—	Y	—	—	—	—	—	—	—	—
char	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
nchar	—	—	—	Y	—	—	—	—	—	—
mchar	—	—	—	—	Y	—	—	—	—	—
varchar	—	—	—	—	—	Y	—	—	—	—
nvarchar	—	—	—	—	—	—	Y	—	—	—
mvarchar	—	—	—	—	—	—	—	Y	—	—
float	—	—	—	—	—	—	—	—	Y	—
smallflt	—	—	—	—	—	—	—	—	—	Y
dec	—	—	—	—	—	—	—	—	—	—
date	—	—	—	—	—	—	—	—	—	—
interval-year-to-day	—	—	—	—	—	—	—	—	—	—
time	—	—	—	—	—	—	—	—	—	—
interval-hour-to-second	—	—	—	—	—	—	—	—	—	—
timestamp	—	—	—	—	—	—	—	—	—	—
binary	—	—	—	—	—	—	—	—	—	—
yy-mm-dd	—	—	—	—	—	—	—	—	—	—
yymmdd	—	—	—	—	—	—	—	—	—	—
blob	—	—	—	—	—	—	—	—	—	—

Legend:

IN: INTEGER

SI: SMALLINT

CH: CHAR

NC: NCHAR

MC: MCHAR

VC: VARCHAR

NV: NVARCHAR

MV: MVARCHAR

FL: FLOAT

SF: SMALLFLT

Y: Can be stored.

—: Cannot be stored.

Table 5-29: Data storage in the database when the data type specified in the column structure information file does not match the data type in the table (2/2)

Data type specified in column structure information file	Data type in table							
	DEC	DATE	IYTD	TIME	IHTS	TMST	BIN	BLOB
integer	—	—	—	—	—	—	—	—
smallint	—	—	—	—	—	—	—	—
char	Y	Y	Y	Y	Y	Y	Y	Y
nchar	—	—	—	—	—	—	—	—
mchar	—	—	—	—	—	—	—	—
varchar	—	—	—	—	—	—	—	—
nvarchar	—	—	—	—	—	—	—	—
mvarchar	—	—	—	—	—	—	—	—
float	—	—	—	—	—	—	—	—
smallflt	—	—	—	—	—	—	—	—
dec	Y	—	—	—	—	—	—	—
date	—	Y	—	—	—	—	—	—
interval-year-to-day	—	—	Y	—	—	—	—	—
time	—	—	—	Y	—	—	—	—
interval-hour-to-second	—	—	—	—	Y	—	—	—
timestamp	—	—	—	—	—	Y	—	—
binary	—	—	—	—	—	—	Y	—

Data type specified in column structure information file	Data type in table							
	DEC	DATE	IYTD	TIME	IHTS	TMST	BIN	BLOB
yy-mm-dd	—	Y	—	—	—	—	—	—
yymmdd	—	Y	—	—	—	—	—	—
blob	—	—	—	—	—	—	Y*	Y

Legend:

DEC: DECIMAL

DATE: DATE

IYTD: INTERVAL YEAR TO DAY

TIME: TIME

IHTS: INTERVAL HOUR TO SECOND

TMST: TIMESTAMP

BIN: BINARY

BLOB: BLOB

Y: Can be stored.

—: Cannot be stored.

* Data can be stored for an input data file in the pdrorg-created binary format.

(5) Examples of type operand specification in the column name statement for an input data file in pdrorg-generated binary format

The following table provides examples of type operand specification in the column name statement for an input data file in pdrorg-generated binary format:

Input data type	Data type after conversion			
	BLOB type	BLOB parameter	BINARY type	BINARY parameter
BLOB type	—	func=(F_BLOB, param=blob)	type=blob	func=(F_BIN, param=binary, type=blob)
BLOB parameter	type=blobprm	func=(F_BLOB, param=blob)	type=blobprm	func=(F_BIN, param=binary, type=blobprm)
BINARY type	N	N	—	N

Input data type	Data type after conversion			
	BLOB type	BLOB parameter	BINARY type	BINARY parameter
BINARY parameter	N	N	N	func=(F_BIN, param=binary)

Legend:

—: Specification is not necessary.

N: The corresponding input data type cannot be converted to the indicated data type.

F_BIN: This is a plug-in-provided constructor function of an abstract data type that takes BINARY type as a parameter.

F_BLOB: This is a plug-in-provided constructor function that takes BLOB type as a parameter.

5.7.2 skipdata statement

(1) Format

```
*skipdata* [{,length=bytes-count|,type=data-type[,element=elements-count]}]
```

(2) Operands

(a) length=bytes-count

～ <unsigned integer> ((1-32000))

For an input data file in the fixed-size data format, this operand specifies in bytes the length of the column that is to be skipped.

Rules

1. There is no need to specify this operand for a DAT-format file.
2. To skip a linefeed code, if the linefeed code consists of linefeed code (\n) and carriage return code (\r), specify as length=2.

(b) type=data-type

For an input data file in the pdrorg-generated binary format, this operand specifies the data type of the column to be skipped.

For details about how to specify the data types, see (3) in section *5.7.1 Column name statement*. Note that you cannot specify *yy-mm-dd* or *yymmdd*. blobprm is applicable only to an input data file in the pdrorg-generated binary format.

If a table with an abstract data type column is output to a binary-format input data file by pdrorg's unload processing, data in the abstract data type column in the file is converted to the data type of the constructor function's argument by the constructor parameter reverse creation function that was specified during unloading. Therefore, specify in the type operand the data type that is created by the constructor parameter reverse creation function. For details about the data type that is created by the constructor parameter reverse creation function, see the documentation for the applicable plug-in.

(c) element=elements-count

For an input data file in the pdrorg-generated binary format, if a repetition column is to be skipped, this operand specifies the maximum number of elements for the repetition column.

This operand must follow the type operand.

(3) Notes

1. If an input data file in the fixed-size data format is a text file, specify the skipdata statement to avoid storing the linefeed code at the end of a line.
2. The skipdata statement is specifiable for a column, but not for a function argument.
3. The following table shows whether or not each operand is specifiable by the type of input data file:

Operand	Format of input data file		
	DAT format	Fixed-size data format	pdrorg-generated binary format
length	—	Y	—
type	—	—	Y

Y: Mandatory

—: Not specifiable

5.7.3 Specification examples

This section presents specification examples of the column structure information file.

(1) DAT format

Table definition:

```
CREATE TABLE T1 (COL1 INTEGER,COL2 DATE,COL3 CHAR(10));
```

Input data file and column structure information file:

Input data file (containing data: COL2, COL1, unneeded data, COL3)

```
1997-06-26,-1,Y,abc
1964-09-13,0,Y,ABC
1999-12-31,999,N,abcdefg
```

Column structure information file

```
COL2
COL1
*skipdata*
COL3
```

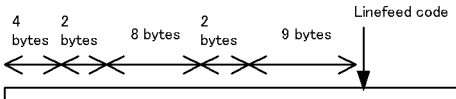
(2) Fixed-size data format

Table definition:

```
CREATE TABLE T1 (COL1 INTEGER,COL2 DATE,COL3 CHAR(10));
```

Input data file and column structure information file:

Input file (containing data: COL1, COL2, COL3)




```
  -1   97/06/26   abc
   0   64/09/13   ABC
999   99/12/31   abcdefg
```

Column structure information file

```
COL1, type=char(6)                    1
COL2, type=yy-mm-dd
*skipdata*, length=2                  2
COL3, type=char(9)                    3
*skipdata*, length=1                  4
```

Explanation:

1. Data consisting of 4 bytes of characters and 2 bytes of spaces can be specified on one line as shown in this example, instead of using two lines `COL1,type=char(4)` and `*skipdata*,length=2`.
2. The skipdata statement is specified for left-justification purposes because the data for `COL3` is characters.

3. The length of one line equals the sum of the column lengths specified in the column structure information file. Therefore, the length of the input data is 9 bytes, although `CHAR(10)` is defined for the table.
4. For a text file, the skipdata statement must be specified because the file contains a linefeed code. The length of one line (one record) in this fixed-size-data-format file is determined to be 26 bytes (25 bytes of data section + linefeed code).

(3) pdrorg-generated binary format

This example loads table `T1`'s unload data to table `T2`, where `T1` has `SGMLTEXT` columns including `BLOB` parameter data and `T2` has `BLOB` columns. The table definitions show only the row name and data type specification sections; the other sections are omitted.

(a) Order of column definitions do not match between T1 and T2

- Table definition during unloading:

```
T1(C1 INT, C2 SGMLTEXT, C3 BLOB)
```

- Table definition during data loading:

```
T2(C4 BLOB ,C5 INT, C6 SGMLTEXT)
```

- Contents of column structure information file:

```
C5
C6, func = (SGMLTEXT, param = blob)
C4
```

(b) T2 has more columns than T1

- Table definition during unloading:

```
T1(C1 INT, C2 SGMLTEXT, C3 BLOB)
```

- Table definition during data loading:

```
T2(C4 INT ,C5 BLOB)
```

- Contents of column structure information file:

```
C4
*skipdata*, type = blobprm
```

```
C5
```

(c) T2 has more columns than T1

- Table definition during unloading:

```
T1(C1 INT, C2 BLOB)
```

- Table definition during data loading:

```
T2(C4 INT, C5 BLOB, C6 SGMLTEXT)
```

- Contents of column structure information file:

```
C4
C5
```

(d) Replacing column data

This example replaces LOB attribute data (C2) in T1's SGMLTEXT type column to T2's LOB column (C6) and T1's LOB column data (C3) to the LOB attribute (C5) of the SGMLTEXT type column in T2.

- Table definition during unloading:

```
T1(C1 INT, C2 SGMLTEXT, C3 BLOB)
```

- Table definition during data loading:

```
T2(C4 INT ,C5 SGMLTEXT, C6 BLOB)
```

- Contents of column structure information file:

```
C4
C6, type=blobprm
C5, func=(sgmltext, param=blob)
```

(e) Changing the data type of columns

This example loads the LOB attribute data (C2) from an SGMLTEXT type column to the BINARY attribute (C5) of an SGMLBIN type column and loads data (C3) from a LOB column to a BINARY type column (C6).

- Table definition during unloading:

```
T1(C1 INT, C2 SGMLTEXT, C3 BLOB)
```

- Table definition during data loading:

```
T2(C4 INT ,C5 SGMLBIN, C6 BINARY)
```

- Contents of column structure information file:

```
C4   ..............................................1
C5, func=(sgmlbin, param=binary, type=blobprm)   ....2
C6, type=blob   ....................................3
```

Explanation

1. Stores `T1`'s `C1` data to `T2`'s `C4`. Specification of data type is omitted because `C1` and `C4` have the same data type.
2. Stores `T1`'s `C2` data to `T2`'s `C5`. To store data in the abstract data type column `SGMLBIN` provided by the plug-in, this example uses the `func` option to specify the constructor function `sgmlbin` and the attribute data type `binary`. Because the input data has the LOB attribute of the abstract data type column `SGMLTEXT` provided by the plug-in, the example uses the `type` operand to specify the LOB attribute (`blobprm`).
3. Stores `T1`'s `C3` data in `T2`'s `C6`. Because `C3` and `C6` have different data types, this example specifies the data type of the input data (`blob`) in the `type` operand.

(f) Area required for data conversion

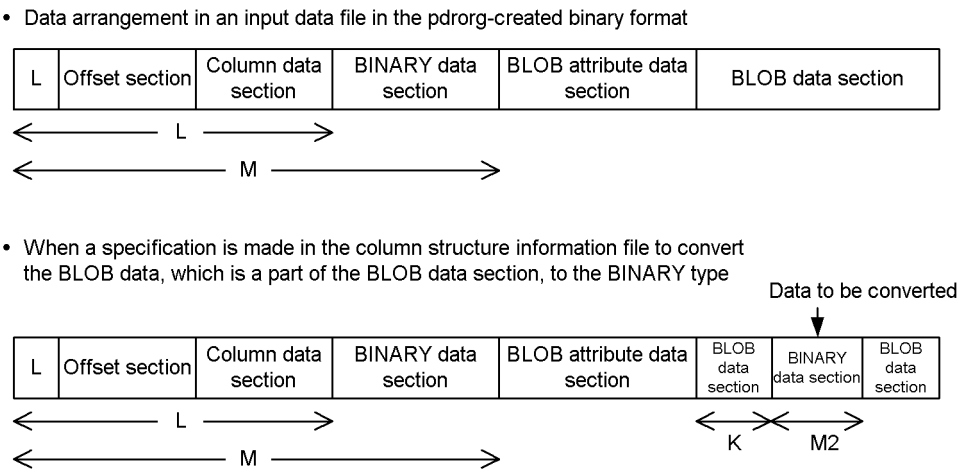
If data conversion results in BLOB data placed between the BINARY type data, BINARY parameter data of abstract data type, and BLOB parameter data of an abstract data type, you need to allocate memory to retain that BLOB data. If the utility is unable to allocate this memory, it terminates with a memory allocation error. Memory allocation is required in the following cases:

- When BLOB data is loaded to an abstract data type column with BLOB attribute, the data may be loaded to another BLOB type column before that BLOB data.
- When BLOB parameter data of an abstract data type is loaded to a BLOB type column, the data may be loaded to an abstract data type column with BLOB attribute after that BLOB parameter data of the abstract data type.
- When BLOB data or BLOB parameter data of an abstract data type is loaded to a BINARY type column or an abstract data type column with BINARY attribute, the data may be loaded to another BLOB type column before the target BLOB

data or BLOB parameter data of the abstract data type.

Figure 5-23 shows the data arrangement in the input data file in the `pdrorg`-created binary format that is output from a table for which BINARY type columns, abstract data type columns with BLOB attribute, and BLOB type columns are defined.

Figure 5-23: Data arrangement in an input data file in the pdrorg-created binary format



Explanation

`pdload` allocates an input data area of at least `M`. The maximum size of `M` is determined from the target table definition (you can also specify a desired size in the `macreclen` operand of the `source` statement).

If data is loaded in this arrangement, `pdload` processes the data sequentially and does not allocate any more area for input data.

If data conversion is specified in the column structure information file, `pdload` allocates an area of `M` + `M2` + `K`. As the input data area, `pdload` allocates an area of `M` + `M2` and to retain the BLOB data section preceding the data converted to the `BINARY` type, `pdload` allocates an area of `K`. You can specify the size of the `K` area using the `bloblimit` operand in the `option` statement. `pdload` allocates the area whose size is obtained from the target table definition or the `option` statement's `bloblimit` operand, whichever is smaller.

5.8 Null value/function information file

You can use a null value/function information file to store the null value in a table when a specified comparison value matches the input data. Specify this operand if the name of the constructor function that generates the values to be stored in a column of abstract data type differs from the name of the data type of the column, or if you want to specify information about multiple functions.

A statement specified in this null value/function information file is called a *column number statement*.

A null value/function information file is applicable only to an input data file in the binary format.

Be sure to specify a column number statement in one line. To specify information for multiple columns, specify as many column number statements as there are columns.

5.8.1 Column number statement

(1) Format

```
column-number {,null={(start-position,[{c|x}]'comparison-value')
              |'numeric-comparison-value'}
              |{,element=elements-count}{,elmtype=arrayed-data-type}
                {,nullset=null-value-option}
              |,func=(function-name,param=argument-type
                [,null={start-position,[{c|x}]'comparison-value')
                |'numeric-comparison-value'}][,...])}
```

(2) Operands

(a) column-number

～ <unsigned integer> ((1-30000))

Specifies in order of the definitions (ascending order) the table column number in which the input data is to be stored.

(b) Other operands

For details about the other operands, see section *5.7.1 Column name statement*.

(3) Specification example

The following presents a specification example of null value/function information file. This example uses the binary format.

- Table definition:

```
CREATE TABLE T1 (C1 INT,C2 DATE,C3 VARCHAR(10),C4 BLOB(100k));
```

- Null value/function information file:

```
1,null = '-1'
2,null = '0000-00-00'
3,null = (5,x'20')
4,null = (1,'@')
```

Explanation:

The null value is stored if the data corresponding to each column is the following value:

```
C1: 0xffffffff

C2: 0x00000000

C3: 0x000a61626364202020202020

C4: 0x0000000140
```

(4) Notes

If an input file subject to data loading does not contain data for some columns, the utility stores in those table columns the default value specified in `WITH DEFAULT` or the null value. For details, see *5.9 Null and default values during data loading*.

5.9 Null and default values during data loading

The null and default values that are stored during data loading depend on the specification of these values during table definition and the control statement specification for execution of `pdload`. Table 5-30 describes the null and default values that are stored during data loading.

Table 5-30: Null and default values that are stored during data loading

Format of input data file	Specification value during table definition			Value of null_string in option statement during execution of pdload	Value stored during data loading		
	NOT NULL constraint	WITH DEFAULT	DEFAULT clause		When input data is null value[1]	When column name statement is omitted[2]	non-null value
DAT format	`NULL`	N/A	Specified	default	Default value in `DEFAULT` clause	Default value in `DEFAULT` clause	Value of input data
				null	Null value		
			Not specified	default		Null value	
				null			
	`NOT NULL`	Specified	N/A	default	Default value in `WITH DEFAULT` clause	Default value in `WITH DEFAULT` clause	
				null			
		Not specified	Specified	default	Default value in `DEFAULT` clause	Default value in `DEFAULT` clause	
				null	Null value (results in an error)		
			Not specified	default		D	
				null			

Format of input data file	Specification value during table definition			Value of null_string in option statement during execution of pdload	Value stored during data loading		
	NOT NULL constraint	WITH DEFAULT	DEFAULT clause		When input data is null value[1]	When column name statement is omitted[2]	non-null value
Extended DAT format	NULL	N/A	Specified	default	Default value in DEFAULT clause	Default value in DEFAULT clause	Value of input data
				null	Null value		
			Not specified	default		Null value	
				null			
	NOT NULL	Specified	N/A	default	Default value in WITH DEFAULT clause	Default value in WITH DEFAULT clause	
				null			
		Not specified	Specified[3]	default	Default value in DEFAULT clause	Default value in DEFAULT clause	
				null	Null value (results in an error)		
			Not specified[3]	default		D	
				null			
			Specified[4]	default	Default value in DEFAULT clause	Default value in DEFAULT clause	
				null	Default value in WITH DEFAULT clause		
			Not specified[4]	default		D	
				null			

Format of input data file	Specification value during table definition			Value of null_string in option statement during execution of pdload	Value stored during data loading		
	NOT NULL constraint	WITH DEFAULT	DEFAULT clause		When input data is null value[1]	When column name statement is omitted[2]	non-null value
Binary format	NULL	N/A	Specified	N/A	Null value	D	Value of input data
			Not specified				
	NOT NULL	Specified	N/A		D		
		Not specified	Specified				
			Not specified				
Fixed-size data format	NULL	N/A	Specified	N/A	Null value	Default value in DEFAULT clause	Value of input data
			Not specified			Null value	
	NOT NULL	Specified	N/A		D	Default value in WITH DEFAULT clause	
		Not specified	Specified			Default value in DEFAULT clause	
			Not specified			D	

Format of input data file	Specification value during table definition			Value of null_string in option statement during execution of pdload	Value stored during data loading		
	NOT NULL constraint	WITH DEFAULT	DEFAULT clause		When input data is null value[1]	When column name statement is omitted[2]	non-null value
pdrorg-created binary format	NULL	N/A	Specified	N/A	Null value	Default value in DEFAULT clause	Value of input data
			Not specified			Null value	
	NOT NULL	Specified	N/A		Null value (results in an error)	Default value in WITH DEFAULT clause	
		Not specified	Specified			Default value in DEFAULT clause	
			Not specified			D	

Legend:

Default value in DEFAULT clause: For details, see Table 5-31.

Default value in WITH DEFAULT clause: For details, see Table 5-32.

N/A: Not applicable

D: The column name statement cannot be specified or omitted depending on specifications when the corresponding table was defined or on the combination of values specified for execution of pdload.

Note

If a primary key or cluster key with UNIQUE specified has been defined for the column for which the WITH DEFAULT or DEFAULT clause is specified, storing the default value results in a key duplication error. In such a case, make sure that you specify the input data.

[1] The following input data is treated as the null value:

In DAT format:

- "*"
- Data with no specification

In extended DATA format:

- "*"
- Data with no specification
- Input data for a column of numeric, date, date interval, time, time interval, or time stamp data type that consists of all single-byte spaces or tabs

In binary format:

Data that matches the comparison value specified in the `null` operand of the column number statement in the null value/function information file

In fixed-size data format:

Data that matches the comparison value specified in the `null` operand of the column name statement in the column structure information file

In `pdrorg`-created binary format:

Data whose column data offset is 0

[2] This applies when the column name statement for the corresponding column is omitted in the column structure information file.

[3] This applies to the following columns:

- Index component column with `UNIQUE` specified
- Cluster key column with `UNIQUE` specified
- Cluster key column with `PRIMARY` specified
- Primary key column

[4] Columns other than those in [3].

Table 5-31: Default value in the DEFAULT clause that is stored during data loading

<table>
<tr><th colspan="2" rowspan="2">Data type</th><th colspan="7">Specification of DEFAULT clause for the column to be stored</th></tr>
<tr><th>Literal</th><th>USER</th><th>C_T or CT</th><th>C_D or CD</th><th>C_TS or CTS</th><th>NULL</th><th>Omitted</th></tr>
<tr><td rowspan="2">Numeric data</td><td rowspan="2">INTEGER
SMALLINT
DECIMAL
FLOAT
SMALLFLT</td><td rowspan="13">Default value</td><td rowspan="2">N/A</td><td rowspan="2">N/A</td><td rowspan="2">N/A</td><td rowspan="2">N/A</td><td rowspan="13">Null value</td><td rowspan="13">Same as the default value in the WITH DEFAULT clause</td></tr>
<tr><td rowspan="2">Character string data</td><td>CHARACTER</td><td rowspan="5">Authorization identifier of the user who inserted rows</td><td>pdload execution time[1]</td><td>pdload execution date[2]</td><td>pdload execution date and time[3]</td></tr>
<tr><td>VARCHAR</td><td rowspan="6">N/A</td><td rowspan="5">N/A</td><td rowspan="6">N/A</td></tr>
<tr><td rowspan="2">Mixed character string data</td><td>MCHAR</td></tr>
<tr><td>MVARCHAR</td></tr>
<tr><td rowspan="2">National character string</td><td>NCHAR</td><td rowspan="10">N/A</td></tr>
<tr><td>NVARCHAR</td></tr>
<tr><td>Date data</td><td>DATE</td><td>pdload execution date</td></tr>
<tr><td>Date interval data</td><td>INTERVAL YEAR TO DAY</td><td rowspan="3">N/A</td></tr>
<tr><td>Time data</td><td>TIME</td><td>pdload execution time</td></tr>
<tr><td>Time interval data</td><td>INTERVAL HOUR TO SECOND</td><td>N/A</td></tr>
<tr><td>Time stamp data</td><td>TIMESTAMP</td><td>pdload execution date and time[4]</td></tr>
</table>

Data type		Specification of DEFAULT clause for the column to be stored						
		Literal	USER	C_T or CT	C_D or CD	C_TS or CTS	NULL	Omitted
Binary data	BINARY					N/A		
Large object data	BLOB	N/A						

Legend:

C_T or CT: CURENT_TIME or CURRENT TIME

N/A: CURRENT_DATE or CURRENT DATE

N/A: CURRENT_TIMESTAMP or CURRENT TIMESTAMP

N/A: Cannot be set.

[1] Stored in the format *hh*:*mm*:*ss*.

[2] Stored in the format *yyyy*-*mm*-*dd*.

[3] Stored in the format *yyyy*-*mm*-*dd hh*:*mm*:*ss*[.000000].

[4] Zeros are stored for the fraction part of seconds.

Table 5-32: Default value of WITH DEFAULT that is stored during data loading

Data type		Value to be stored
Numeric data	INTEGER SMALLINT DECIMAL FLOAT SMALLFLT	0
Character string data	CHARACTER	Space
	VARCHAR	Space equivalent to 1 byte
National character string	NCHAR	Space
	NVARCHAR	Space equivalent to 1 character

Data type		Value to be stored
Mixed character string data	MCHAR	Space
	MVARCHAR	Space equivalent to 1 byte
Date data	DATE	Current date
Date interval data	INTERVAL YEAR TO DAY	0 year, 0 month, and 0 day
Time data	TIME	Current time
Time interval data	INTERVAL HOUR TO SECOND	00:00:00
Time stamp data	TIMESTAMP	Current date and time
Binary data	BINARY	Data with a length of 0
Large object data	BLOB	Data with a length of 0

Note: The current date, current time, and current date and time are based on the date and time the database load utility is executed.

5.10 Using a UOC to load a table

5.10.1 Overview

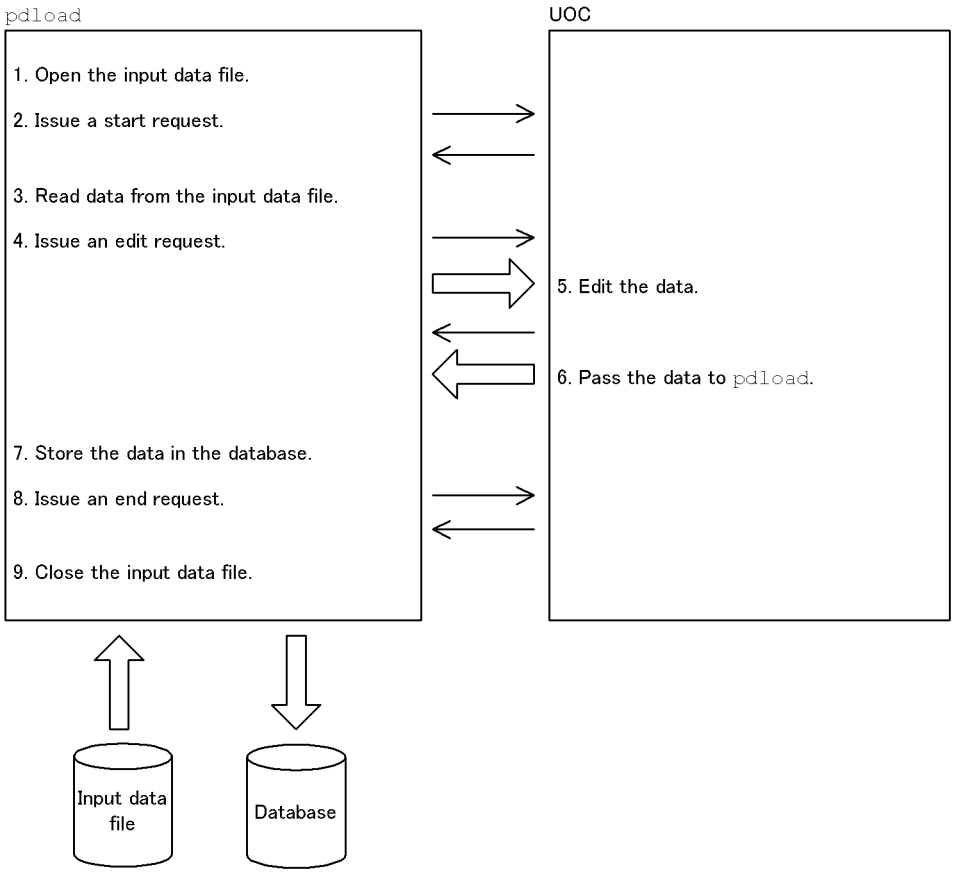
You may want to store the input data file after editing it with a user-created program, instead of storing it in the database as is. Such a user program is called a UOC (user's own coding). You can create UOCs in C language.

There are two ways to load a table using a UOC:

- Using the utility to input the input data file (by specifying the file name in the `source` statement)
- Using a UOC to input the input data file (by specifying (`uoc`) in the `source` statement)

Figures 5-24 and 5-25 show the concept of these methods:

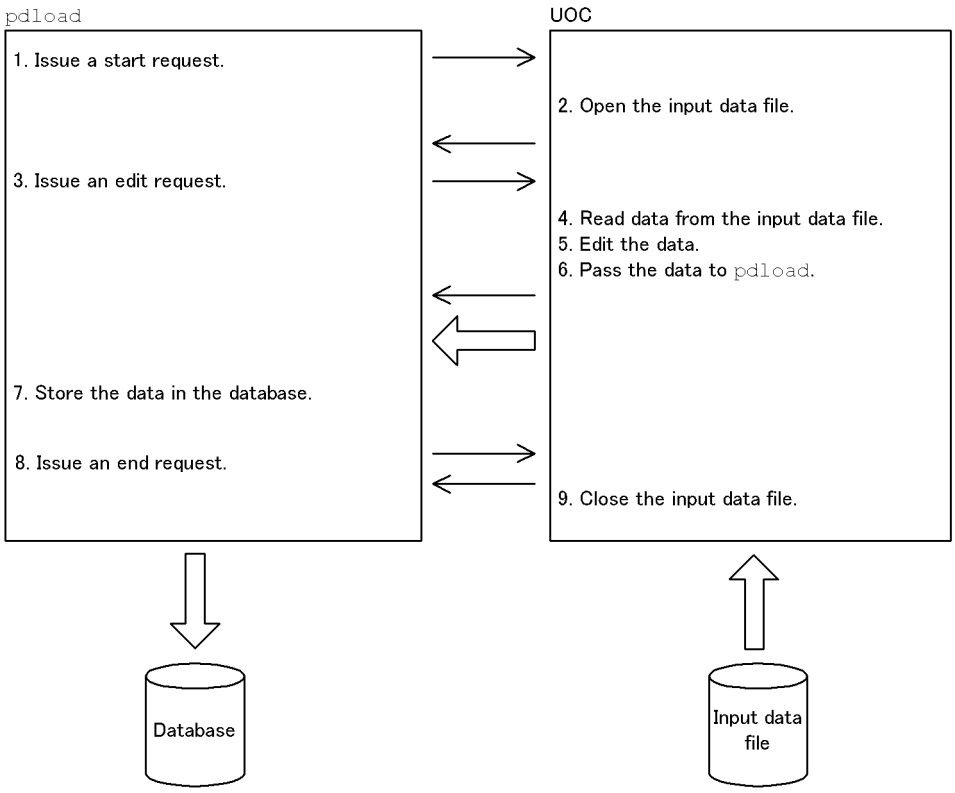
Figure 5-24: Table loading using a UOC (using the utility to input the input data file)



Explanation:

Repeat steps 3 through 7 until all data is loaded from the input data file.

Figure 5-25: Table loading using a UOC (using a UOC to input the input data file)



Explanation:

Repeat steps 3 through 7 until all data is loaded from the input data file.

5.10.2 UOC interface

Information is input and output between the database load utility and the UOC via the interface area. The database load utility allocates this area. The UOC receives the address of this area in the first argument of the function, and references and updates the area.

Figures 5-26 and 5-27 show the structure of the UOC interface area. Table 5-33 describes the contents of the UOC interface area.

Figure 5-26: Structure of UOC interface area (32-bit mode)

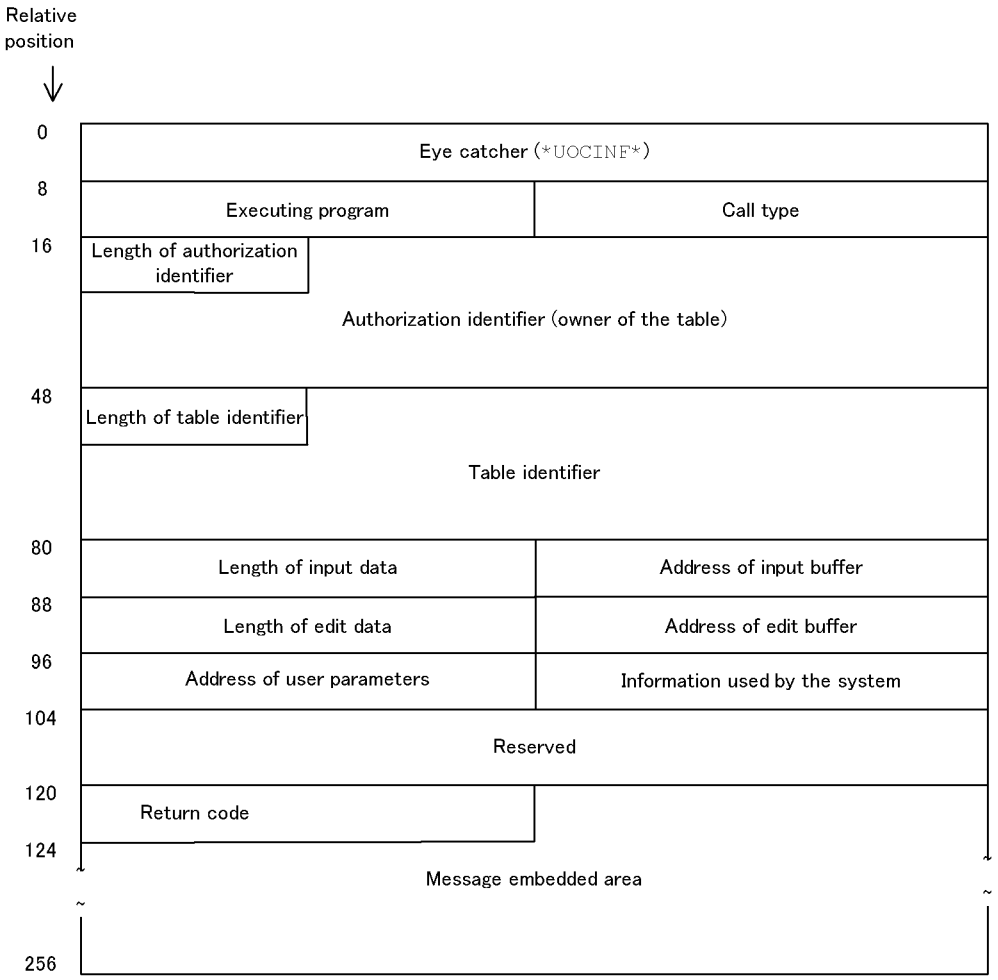


Figure 5-27: Structure of UOC interface area (64-bit mode)

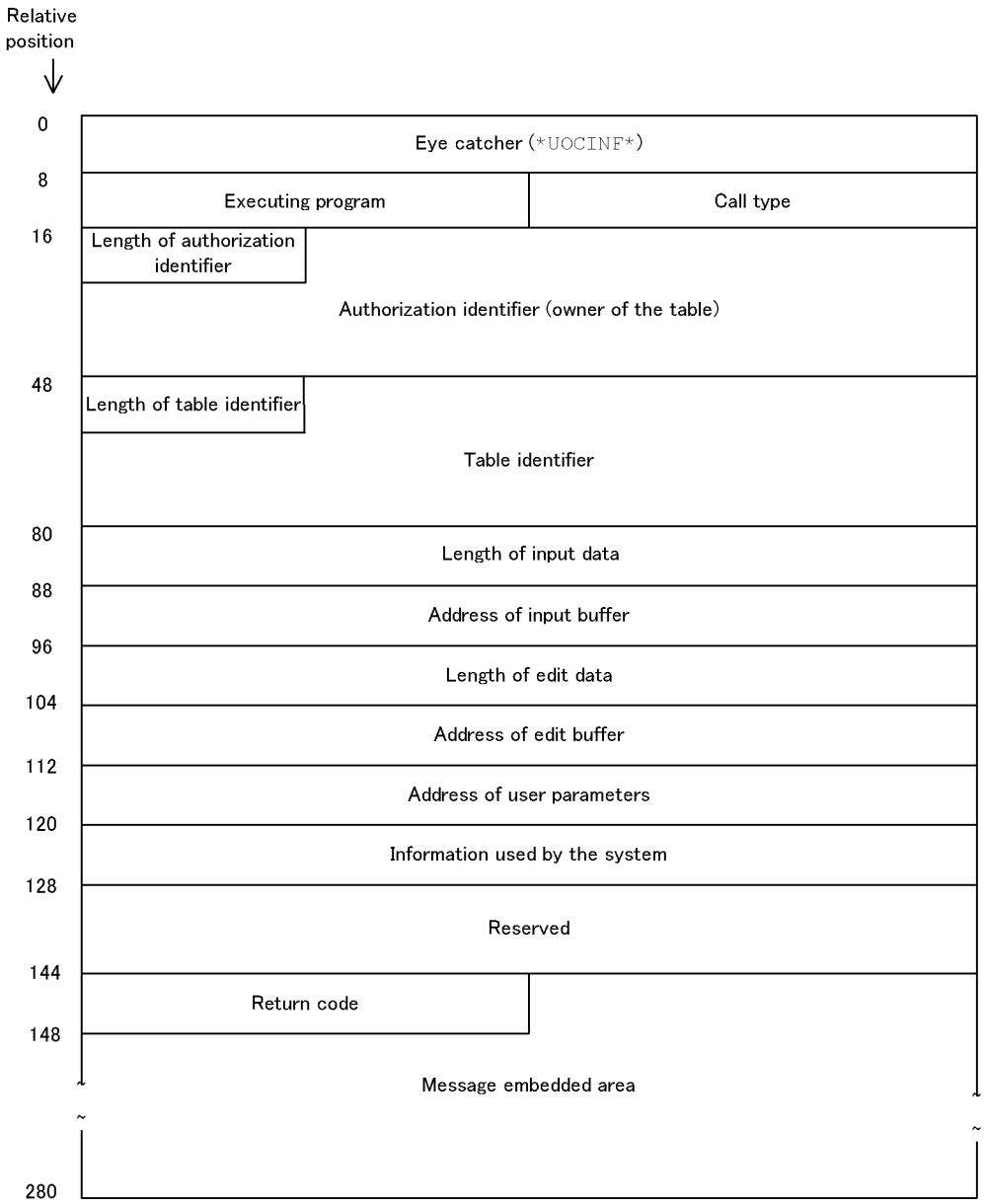


Table 5-33: Contents of UOC interface area

Relative position		Field name	Length (Byte)		Attribute	Value setter	Contents
32	64		32	64			
0	0	Eye catcher	8	8	char	pdload	Interface area (*UOCINF*).
8	8	Executing program	4	4	long	pdload	Program receiving control: 0: pdload 1: UOC
12	12	Call type	4	4	long	pdload	Type of processing request to UOC: o: Start request Requests processing startup preparations (prepares to start processing, such as by opening the input data file). e: Edit request[1] Requests editing of row data. For method (1), one row of data input by pdload is set in the input buffer pointed to by the input buffer address. For method (2), the UOC reads the input data file. In both methods, the results of editing the data are then set in the edit buffer pointed to by the edit buffer address. This call is repeated until all data has been processed or the return code from the UOC is a value other than 0 or 4. c: End request Requests termination of processing. Such termination processing as closing the input data file is performed. For method (1), the request is issued when there is no more data to be input by pdload. For method (2), the request is issued when return code 16 is received from the UOC.

Relative position		Field name	Length (Byte)		Attribute	Value setter	Contents
32	64		32	64			
							`t`: Termination request Requests forcible termination of processing. This call is used instead of an end request when an error occurs on the `pdload` side or when a return code other than `0`, `4`, or `16` is received from the UOC. Once a start request is issued, an end request is issued when processing terminates normally, and a termination request is issued when processing terminates abnormally. The UOC is not called once an end request or termination request is issued.
16	16	Length of authorization identifier	2	2	`short`	`pdload`	Length of name of table owner.
18	18	Authorization identifier	30	30	`char`	`pdload`	Name of table owner.
48	48	Length of table identifier	2	2	`short`	`pdload`	Length of table name.
50	50	Table identifier	30	30	`char`	`pdload`	Table name.
80	80	Length of input data[2]	4	8	`long`	`pdload`	1. For a binary-format or extended-DAT-format input data file, this is set to the length of the data (full length) set in the input buffer.[3] 2. This item is ignored.
84	88	Address of input buffer	4	8	`void*`	`pdload`	1. Set to the address of the buffer that stores the data input by `pdload`. 2. This item is ignored.
88	96	Length of edit data[2, 5]	4	8	`long`	`UOC`	For a binary-format or extended-DAT-format input data file, this is set to the length of the data (full length) set in the edit buffer.[4] 1. Set to -1 if no data is stored.

Relative position		Field name	Length (Byte)		Attribute	Value setter	Contents
32	64		32	64			
92	104	Address of edit buffer	4	8	`void*`	`UOC`	Address of buffer that stores the data edited by the UOC.
96	112	Address of user parameters	4	8	`void*`	`pdload`	Set to the address of the character string specified in the `param` operand of the `srcuoc` statement. If the `param` operand is omitted, it is set to `NULL`.
100	120	Information used by system	4	8	`void*`	`pdload`	Information used by the system (must not be used by the UOC).
104	128	Reserved	16	16	`char`	`pdload`	Reserved area (must not be used by the UOC).
120	144	Return code	4	4	`long`	`UOC`	Return code that notifies the calling program of the results of UOC processing. The UOC that received control from `pdload` must return one of the codes explained as follows. Because this area is cleared (set to `-1`) each time `pdload` calls the UOC, the UOC must set this area after every call. `0`: Start or end request executed normally, or one row of data has been made available in the edit buffer for an edit request. `4`: Message output processing is to be performed (even if processing terminated normally); the details are the same as for the `0` return code. When the return code is `4`, the message is output up to three times. Once the message has been output three times, the return code is ignored and the message is no longer output, even if the return code remains as `4`.

Relative position		Field name	Length (Byte)		Attribute	Value setter	Contents
32	64		32	64			
							`8`: Input processing is to be committed up to that point but further processing is to be discontinued because an error occurred during the UOC processing. However, if an internal error occurred on the HiRDB side, input processing is rolled back. The message output to the message log should be checked to determine whether input processing was committed or rolled back. `16`: The UOC has read the input data file, and `pdload` is to be notified that there is no more data to be processed. When the input data file is read by `pdload`, this return code can also be set when no more data is to be received after this time. When `pdload` receives this return code, it performs end processing. `20`: An error occurred during UOC processing. This return code is set when input processing is rolled back up to the time of the error and processing is discontinued. If a return code other than `0` or `16` is set, the contents of the message embedded area are output to the standard output and the message log. The message should be a character string of up to 131 bytes terminating with `\0`.
124	148	Message embedded area	132	132	`char`	`UOC`	Storage area for message output to standard output and log.

32: Relative position and length for HiRDB in 32-bit mode

64: Relative position and length for HiRDB in 64-bit mode

Note

1. and 2. in the *Contents* column indicate the following:

1. Using `pdload` to input the input data file
2. Using a UOC to input the input data file

[1] When the `-e` option is specified, UOC processing is also discontinued if erroneous data is detected. If `pdload` was used to input the input data file, the following results:

- If the `-r` option is specified, data up to the specified row data is not passed.
- If data contains a carriage return code (`\r`) before a linefeed code (`\n`) such as those in a DAT-format file created by a personal computer, the data is passed with the carriage return deleted.
- For an input data file in `pdrorg`-generated binary format with the `-W` option specified, neither `LOB` nor `BINARY` columns can be edited.
- When processing a binary-format input data file, data is not passed if it has an invalid length for a variable-length character string.

[2] The value is set to -1 when the corresponding data is not to be input In the case of a HiRDB in the 64-bit mode, the maximum length of data valid for processing is 2 gigabytes minus 1 byte.

[3] The area that is pointed to by the input buffer address field is called the input buffer. The input buffer is allocated by `pdload` when `pdload` reads the input data file. The input buffer contains each row of data input by `pdload`. The format of the input buffer is as follows:

- DAT-format input data file

  Text data read up to the linefeed code is stored as is. The end of the data consists of a linefeed code (`\n`) plus the NULL character (`\0`). Do not reference or update data beyond the last NULL character.

- Extended-DAT-format input data file

  Text data up to the null character or linefeed code that is not enclosed in the enclosing character is stored as is. The length of the stored data is set as the input data length. Do not reference or update the area that is larger than this input data length.

- Binary-format input data file

  The row length of the data is stored, as calculated from the table definition. The length of the stored data is set in the input data length. Do not reference or update the area beyond the length set in the input data length.

- Input data file output by pdrorg

  For a FIX table, the input data length is set to the length of the data section only. For a non-FIX table, the input data length is set to the sum of the offset section plus the data section. Do not reference or update the area beyond the length set in the input data length. Because UOCs cannot handle LOB or BINARY columns, use pdrorg's UOC to edit tables that contain LOB or BINARY columns.

- Fixed-size-data-format input data file

  The row length of the data is stored, as calculated from the column information in the column structure information file. For a text file, the data still contains linefeed (\n) and carriage return (\r) codes.

[4] The area pointed to by the edit buffer address field is called the edit buffer. The UOC allocates the edit buffer. However, when pdload reads the input data file, the input buffer can be used as is as the edit buffer if the input buffer is not to be updated beyond the specified data length. For binary-format data, set the length of data in the edit data length field to indicate the actual data part in the edit buffer. For DAT-format data, there is no need to set the length, but the end of the edit buffer data must be a linefeed code (\n) followed by the NULL character (\0).

[5] For an input data file in the binary or extended DAT format, specify the length of data for the edit buffer. For an input data file in the extended DAT format, make sure that the length of edited data does not exceed the maxreclen operand value specified in the source statement. For an input data file in any other format, specify the initial value of 0.

5.10.3 Examples of UOCs

This section describes examples of data loading using UOCs.

These UOC examples are provided as sample databases. The storage directory is %PDDIR%\sample\sampleUOC.

(1) Example of data loading using UOC

This example changes an input data file with a 2-digit year designation to a 4-digit designation to store the data in a Y2K-supported database that can handle dates in the year 2000 and beyond. The input data file is in DAT format, and pdload is used to input the input data file.

(a) Database table definition

```
CREATE TABLE DIRECTORY (EMPLOYEE_NUMBER INTEGER,
                        BIRTH_DATE DATE,
                        HOMETOWN CHAR(10),
                        NAME CHAR(16));
```

(b) Format of input data file

```
10001, 68/04/30, BOSTON,DAVID IVERSON
10002, 64/09/13, TEXAS,DONALD YOUNG
20001, 70/11/02, CHICAGO,MATT CARR
```

(c) Command format

```
pdload DIRECTORY control-information-file-name
```

(d) Control information file

```
source input-data-filename
srcuoc library-name entry = date_change_func
```

(e) UOC example

```
(File name: sample1.c)


/*****************************************************************************
**                                                                          **
**  HiRDB sample Data input User Own Coding (for CSV file)                  **
**                            ~    ~   ~                                    **
**  name    :   date_change_func                                            **
**                                                                          **
**  func    :   YY/MM/DD  ===>  YYYY-MM-DD                                  **
**                                                                          **
**  i/o     :   none                                                        **
**                                                                          **
**  return  :   none                                                        **
**                                                                          **
*****************************************************************************/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <pdutluoc.h>                                                    1
```

```
#define DATELEN         10                  /* DATE length                  */

    static  void    sub_date_change(char *,char *);

void date_change_func(
    UTL_UOC_INF    *uocinf                  /* A(UOC interface area)        */    2
){
    long            wk_leng;                /* length                       */
    char            date_data[10];          /* DATE (HiRDB)                 */
    char           *date_ptr;               /* A(birth date)                */
    char           *address_p;              /* A(native)                    */
    static  char    buff[512];              /* buffer                       */

    switch(uocinf->req_cd){                                                  3
    case UTL_UOC_START:

/*--------------------------------------------------------------------------*/
/*  START                                                                   */
/*--------------------------------------------------------------------------*/

        uocinf->edit_adr = buff;
        uocinf->rtn_code = UTL_UOC_NML;
        break;
    case UTL_UOC_EDIT:

/*--------------------------------------------------------------------------*/
/*  EDIT                                                                    */
/*--------------------------------------------------------------------------*/

        date_ptr = strchr(uocinf->data_adr,',');
        if (date_ptr == NULL){
            strcpy(uocinf->err_msg,"Invalid data");
            goto OWARI;
        }
        sub_date_change(++date_ptr,date_data);
        wk_leng = (long)date_ptr - (long)uocinf->data_adr;
        strncpy(buff,uocinf->data_adr,wk_leng);
        strncpy((char*)((long)buff + wk_leng),date_data,DATELEN);
        wk_leng += DATELEN;
        address_p = strchr(date_ptr,',');
        strcpy((char*)((long)buff + wk_leng),address_p);
        uocinf->rtn_code = UTL_UOC_NML;
        break;
    case UTL_UOC_END:
    case UTL_UOC_TERM:

/*--------------------------------------------------------------------------*/
/*  END                                                                     */
/*--------------------------------------------------------------------------*/
```

```
        uocinf->rtn_code = UTL_UOC_NML;
        break;
    default:
        strcpy(uocinf->err_msg,"Invalid request code");
        goto OWARI;
    }
    return;
OWARI:
    uocinf->rtn_code = UTL_UOC_ERR;
    return;
}


static void sub_date_change(
    char    *year_two,                      /* YY/MM/DD                     */
    char    *year_four                      /* YYYY-MM-DD                   */
){
    strcpy(year_four,"19");
    strncat(year_four,year_two,8);
    year_four[4] = '-';
    year_four[7] = '-';
    return;
}
```

Explanation:

1. Acquire the UOC creation header provided by HiRDB.
2. Receive the address of the UOC interface area as an argument.
3. Determine the call type and execute the appropriate processing.

5.10.4 Creating a dynamic link library

Once you have created a UOC, create a dynamic link library. For details about creating a dynamic link library, see *E.1 Creating a dynamic link library*.

5.11 Rules and notes

(1) Rules

(a) Execution of the utility

- You can execute pdload only while HiRDB is active.
- Execute pdload at the server machine containing the single server or at the server machine where the system manager is located.
- If you execute pdload, specify the LANG environment variable. For details about LANG, see the *HiRDB Version 8 UAP Development Guide*.

(b) Whether or not the utility can be executed

Whether or nor pdload can be executed depends on the open attribute of the RDAREAs containing the tables, indexes, and LOB columns subject to data loading, as well as the status of the RDAREAs. For details, see Appendix *C. RDAREA Status During Command Execution*.

(c) Maximum number of concurrent executions of the utility

The maximum number of times pdload can be executed concurrently depends on the value of the pd_utl_exec_mode operand.

pd_utl_exec_mode=0 specified:

A maximum of 32 concurrent executions of pdload are permitted.

pd_utl_exec_mode=1 specified:

The maximum number of concurrent executions of pdload is the value of the pd_max_users operand.

(d) Using tables and indexes during utility processing

- While pdload is creating a table or an index, no other UAP or utility can access the table or the index. However, if you specify nowait=yes in the option statement, you can conduct a NOWAIT search. A NOWAIT search is an SQL search function with NOWAIT specified in the lock option.

  If you shut down the RDAREAs containing the table subject to data loading with the pdhold command, an UAP results in a shutdown error. This prevents the utility's execution from being placed in lock-release wait status by UAP execution or the UAP execution from being placed in lock-release wait status by utility execution.
- Make sure that you do not execute a definition SQL statement for a table currently undergoing data loading. If a definition SQL statement is executed, pdload terminates abnormally. To prevent execution of a definition SQL on a table

currently subject to data loading, you need to issue the pdhold command to shut down the RDAREA that contains the table and index to be created. To reference another table in the same RDAREA during data loading, use the pdhold command to place the RDAREA in reference-possible shutdown status.

- If you execute pdload on a table or index with an RDAREA that is being processed by a definition SQL statement, the table or index is placed in lock-release wait status.
- For an input data file in DAT format or fixed-size data format, if the WITH DEFAULT clause is not specified for the table columns, and no input data is specified for a column with the NOT NULL attribute, using the utility results in an error.

(e) Relationship between CREATE TABLE or CREATE INDEX and PCTFREE

- Percentage of free space in page and in segment during data storage

  During data loading, the data is stored according to the percentage of free space in the pages or segments specified in the table definition (PCTFREE operand of CREATE TABLE).

  For an index, if the batch index creation mode is specified or new data is created in the creation mode, the data is stored according to the percentage of free space in the pages specified in the index definition (PCTFREE operand of CREATE INDEX).

  However, if the -y option is specified, the utility stores data in available pages regardless of the percentage of free space specified during table and index definitions. Additionally, you can change the percentage of free space in the table or index definition by using the option statement with the tblfree operand or the idxfree operand, respectively, specified.

- Searching for free space

  During execution, pdload always stores data in unused pages (an error results if unused pages cannot be allocated). When loading data in the addition mode (-d specified), the utility does not use any free space in used pages that contain existing data, even if such space is available, because it takes time to search for the free space in used pages. (This could result in a considerable increase in execution time because a large amount of data is added during data loading.) However, if the -y option is specified, no error results even when the utility is unable to allocate unused pages (in which case the utility stores data in available space in used pages). During data loading on a table, the specification of SEGMENT REUSE is ignored, if specified for the table.

(f) Synchronization points during data loading

Normally, synchronization point dumps are not collected during data loading. If other UAPs are executing while data loading is underway and a system failure occurs, the

time required for restart increases. Therefore, you should not execute any UAPs during data loading.

Data loading with the synchronization point specification enables you to collect synchronization point dumps at intervals specified as the number of lines. In the event of abnormal termination, this method requires less start time than the normal data loading method that does not collect synchronization point dumps.

(g) Checking national character string data

Values stored in national character string data are not subject to checking as to whether or not the data is in multi-byte codes.

(h) Facility for conversion to a DECIMAL signed normalized number

The facility for conversion to a DECIMAL signed normalized number handles the sign part of DECIMAL type, shown as follows. For details about the facility for conversion to a DECIMAL signed normalized number, see the *HiRDB Version 8 System Operation Guide*.

- Specifications for the sign part of DECIMAL type

  The following table shows the HiRDB specifications for the sign part of DECIMAL type:

Sign part	Description
X'C'	Indicates a positive number.
X'D'	Indicates a negative value.
X'F'	Indicates a positive value.

- Conversion rules for the sign part of DECIMAL type

When data is inserted, the facility for conversion to a DECIMAL signed normalized number converts the sign part of DECIMAL type according to the following rules.

- Conversion rules for the sign part of DECIMAL type (other than 0 data)

Sign part of embedded variable data	Not normalized	Normalized
X'A'	Error	Converted to X'C'
X'B'	Error	Converted to X'D'
X'C'	Not converted	Not converted
X'D'	Not converted	Not converted
X'E'	Error	Converted to X'C'
X'F'	Not converted	Converted to X'C'

Sign part of embedded variable data	Not normalized	Normalized
X'0' to X'9'	Error	Error

- Conversion rules for the sign part of DECIMAL type (0 data)

Sign part of 0 data	Not normalized	Normalized
X'A'	Error	Converted to X'C'
X'B'	Error	
X'C'	Not converted	
X'D'	Not converted	
X'E'	Error	
X'F'	Not converted	

Thus, if the 0 data is normalized, -0 is converted to +0.

For an input data file in the DAT format that is subject to conversion of character string data, the facility stores the normalized values (sign part is X'C' for positive values and 0 data and X'D' for negative values) regardless of the system definitions.

- Referencing error information files and error data files

When the facility for conversion to a decimal signed normalized number is used, the DECIMAL columns are output in the error data dump-image listing that is output to the error information file and in the input data that is output to the error data file as normalized values up to the column where an error was detected by pdload.

(i) Loading data to a table containing an abstract data type

For a table containing an abstract data type provided by a plug-in, you can call a constructor function to generate the values for the corresponding columns to load the data. However, this data loading is not possible if the constructor function is not created in the shared library.

For a table containing a user-defined abstract data type, data loading is not possible because the system cannot generate values for the corresponding columns.

(j) Loading data to a rebalancing table

If you have added an RDAREA to a rebalancing table, you cannot load data to the added RDAREA (during data loading in units of RDAREAs) unless you execute pdrbal on the rebalancing table (return code = 0).

For a table partitioned by flexible hash partitioning, the utility stores data and ignores the hash group; therefore, the data rearranged by pdrbal does not link to the hash key

values.

(k) Data loading on external tables

You cannot execute data loading on external tables.

(l) Data loading on falsification prevented tables

You should note the following about executing data loading on falsification prevented tables:

- The `-d` option cannot be specified.
- Data loading is disabled if the RDAREA subject to data loading is in reload-not-completed data status. An RDAREA is in reload-not-completed data status when reorganization of a falsification prevented table has not been completed due to an error during reorganization processing.
- The date and time that execution of `pdload` started (the date and time at the host that executed `pdload`) is stored in the insert history maintenance column (column with `SYSTEM GENERATED` specified).
- Input data for an insert history maintenance column is ignored, if specified.

(m) Data loading on a table with a trigger defined

Execution of `pdload` does not activate a trigger. Before executing `pdload`, check the contents of the table's trigger definition.

(2) Notes

(a) Result of utility execution

You can check the result of `pdload` with `pddbst` (or by running the UAP or using `pdrorg` to perform unload processing). When `pdload` processing is finished, a return code is set. Table 5-34 shows the `pdload` return codes.

Table 5-34: pdload return codes

Return code	Description	Action
`0`	Normal termination. All data in the input data file was loaded.	None
	Normal termination. All data in the input data file was output to the divided-input data files.	Use the divided-input data files to perform data loading in units of RDAREAs.

Return code	Description	Action
4	Input data error. Data loading was skipped because some data in the input data file was invalid.	See the error information file, correct the erroneous data in the input data file, and then re-execute data loading.
	Input data error. Normal data in the input data file was output to the divided-input data files, but erroneous data was not (when the `-e` option was specified, data was output to the divided-input data files until erroneous data was detected).	See the error information file, correct the erroneous data in the input data file, and then re-create the divided-input data files.
8	Input data error. When the `dataerr` operand was specified in the `option` statement, data loading was rolled back due to an error in the input data file.	Correct the erroneous input data and then re-execute data loading.
	Key duplication error. A key duplication error was detected during batch creation of index.	Restore the database from its backup, correct the erroneous input data, then re-execute data loading.
	Abnormal termination. Data loading was cancelled due to an error.	Eliminate the cause of the error and then re-execute data loading.
	Abnormal termination. Data output from input data file to divided-input data files failed.	See the error message, eliminate the cause of the error, and then re-create the divided-input data files.

(b) Cancelling processing during utility execution

To cancel processing during the execution of `pdload`, use the `pdcancel` command. To terminate `pdload` forcibly because of a no-response error (such as when a routine job does not finish data loading within the normal amount of time), redirect the display result of the `pdls` command (with `-d rpc -a` specified) to a file and then execute the `pdcancel -d` command.

In this case, if `pdload` was executing in the creation mode (`-d` option specified), all the data stored in the table is deleted. If `pdload` was executing in the addition mode (`-d` option omitted), processing is rolled back.

If the facility for predicting reorganization time is used and `pdload` is terminated forcibly by a signal interrupt, the database management table cannot be updated. To terminate `pdload` while using the facility for predicting reorganization time, make sure that you use the `pdcancel` command.

(c) LOB column structure base tables and resources such as the RDAREA and the buffer for LOB data

To load data to a table with LOB columns, you need to prepare all the required

resources, such as the RDAREAs and buffers of the LOB column structure base table and LOB data, even if you are loading data only to the LOB column structure base table.

(d) File media that can be used during execution of a utility

The table below shows the file media that are supported during execution of `pdload`.

File	File	Fixed-length blocked tape	Variable-length blocked tape
Input data file	Y	Y*	Y
Control information file	Y	—	—
Column structure information file	Y	—	—
Null value/function information file	Y	—	—
Error information file	Y	—	—
Error data file	Y	—	—
LOB input file	Y	—	Y
LOB middle file	Y	—	—
Index information file	Y	—	—
Work file for sorting	Y	—	—
Process results file	Y	—	—

Y: Can be used

—: Cannot be used

* Cannot be used for binary format or fixed-size data format.

(e) Executing a utility on an extraction database that is subject to data linkage

You should note the following about executing `pdload` on a database to be extracted that is subject to data linkage:

- Do not execute the utility in the creation mode for an existing table (do not specify the `-d` option).
- To add data to an existing table in the addition mode, execute the utility in the log acquisition mode (specify `a` in the `-l` option).
- To add a new table subject to HiRDB Datareplicator linkage, execute the utility in the log acquisition mode (specify `a` in the `-l` option).

(f) Loading data using a DAT-format input file containing a comma (,) at the end of each row

When you load data using a DAT-format input file that contains a comma (,) at the end of each row, the data loading may fail due to a column count mismatch. In this case, specify the names of all table columns in the column structure information file and specify the `skipdata` statement at the end. There is no need to revise the input data.

If coding of the column structure information file is not feasible because of a large number of table columns involved, you can prepare the file with the following method:

1. Use the SQL Executer to output table column names to a file.

   When directing the output of column names to a file, you need to output the column names in the order in which the columns are defined. Therefore, specify the SQL statement as follows:
   `PUTFILE TO` *filename* `SELECT COLUMN_NAME FROM MASTER.SQL_COLUMNS WHERE TABLE_NAME ='`*table-name*`' ORDER BY COLUMN_ID`

2. Append the `skipdata` statement to the last row for the name of the file to which the results of editing in Step 1 are output.

(g) Loading data to a table with a plug-in index defined

- When loading data in the no-log mode to a table for which a plug-in index is defined, or when loading additional data in the batch index creation mode, be sure to back up both the table storage RDAREAs to be processed (including LOBs) and the index storage RDAREAs, to protect against abnormal termination of the utility. Otherwise, the RDAREAs cannot be recovered in the event of an error, in which case you need to re-create them from the initial state.
- When you create a plug-in index using the batch index creation facility (with the `-i c` option specified), if the plug-in provides the batch plug-in index creation partial recovery facility, you can restore the index by rolling back the process using the log acquired during the abnormal termination of the utility. If the plug-in does not provide this facility, recovery by rollback process is not possible, in which case you need to re-create the entire index.
- When you use the plug-in index delayed batch creation facility and you have executed a UAP on a table with `YES` specified for `PDPLGIXMK` in the client environment definition, you cannot execute `pdload` in the addition mode until you use the database reorganization utility to create the index in the batch mode. However, you can execute `pdload` in the creation mode. Do not use the index information file output directory used by this facility (directory with `pd_plugin_ixmk_dir` specified) as a file output area with `pdload`.

(h) Using the differential index function of the HiRDB Text Search Plug-in

When you use the differential index function of the HiRDB Text Search Plug-in,

pdload updates the following indexes according to the specification of PDPLUGINNSUB in the client environmental definition:

Presence of existing data	PDPLUGINNSUB specified		
	Y	N	Not set
No existing data (data loading in creation mode)	M	M	M
Existing data present (data loading in addition mode)	S	M	S

M: Updates the MASTER index

S: Updates the differential index

(i) Loading data to a table containing a unique key index or primary key index

When loading data to a table containing a unique key index or primary key index, note the following:

- If an input data file contains duplicate key values, do not execute data loading in any mode other than the index update mode.

  If you execute data loading in the batch index creation mode, index information output mode, or index information output suppression mode, the utility stores the table and outputs the index key information to an index information file.

  Duplicate key checking is not executed at this point. It is executed when the index data is stored.

  If a duplicate key value is detected, index creation processing rolls back, but the data has already been stored (committed and cannot go back). In this case, you need to restore the corresponding RDAREAs using their backup copies.

  Therefore, when you load data using an input data file that contains data with duplicate key values, be sure to specify the index update mode. In this mode, the utility updates the index each time it stores data; therefore, the utility can detect a duplicate key value immediately and skips the data without storing it in the database.

  Note that the default is the batch index creation mode.

(j) List created on the basis of a table subject to data loading

If you have created a list on the basis of a table subject to data loading, and execute a search process on the list after data loading, the following events may occur:

- Retrieval of a wrong row
- Specified row not found

In this case, you need to re-create the list before executing the search.

(k) Locked resources required per server during execution of pdload

While executing `pdload`, each server requires the following amount of locked resources:

- Resources subject to lock control beyond the transactions

  $X = 2 \times (a + b + c + 1) + (b \times d)$

  Note

  The value of the `pd_lck_until_disconnect_cnt` operand in the system definitions must be at least the value of *X*.

- Resources subject to lock control within the transactions

  $Y = e + f + g$

  Note

  The value of the `pd_lck_pool_size` operand in the system definition must be at least $\uparrow Y/x \uparrow$ kilobytes. Because utility preprocessing requires (209 + *A*) resources, the value of *Y* must be at least (209 + *A*). The resources allocated during preprocessing are released before data loading begins; therefore, if the obtained value of *Y* is (209 + *A*) or greater, use that value.

Legend:

a: Number of table storage RDAREAs

b: Number of index storage RDAREAs

c: Number of LOB column storage RDAREAs

d: Number of indexes

e: Number of segments used by table storage RDAREAs*

f: Number of segments used by index storage RDAREAs*

g: $\uparrow$ Number of segments used by LOB column storage RDAREAs/64000 $\uparrow$*

x: 6 for HiRDB in the 32-bit mode; 4 for HiRDB in the 64-bit mode

A: MAX (number of table columns, number of table indexes, number of table storage RDAREAs)

* When you have specified the `-d` option and the number of segments used by the table storage RDAREAs, index storage RDAREAs, or `LOB` column storage RDAREAs is greater, you should use the number of segments actually in use; if the number of segments used by the data to be stored is greater, use the number of segments scheduled to be used for storing data.

(l) Messages displayed in the command execution window

pdload outputs progress messages to the standard output during processing. In the event of an error, error messages are output to standard error. If pdload is executed in an environment in which output to standard output or standard error is suppressed, pdload may stop responding due to a message output wait, or it may output the KFPL20003-E message to the message log file and terminate abnormally. For this reason, you should not execute pdload in an environment in which messages cannot be output to the standard output or standard error output. Note that the sequence and number of messages output to the standard output and to the standard error output may not match the sequence and number in the message log file and the event log. To obtain the accurate messages, view the message log file or the event log.

(m) Data loading to a shared table

To perform data loading on a shared table, the system places the RDAREAs containing the shared table and shared indexes defined for the target table in the EX lock mode. If the corresponding RDAREAs contain other tables and indexes, these tables and indexes cannot be referenced or updated. For details about the lock mode used for data loading on shared tables, see Appendix *B.2 Lock mode for utilities*.

(n) Data loading on a table for which referential constraints or check constraints have been defined

When data is loaded into a table for which referential constraints or check constraints have been defined, pdload does not check for data integrity. For this reason, you should use pdconstck to check data integrity during data loading. For details about how to check data integrity, see the *HiRDB Version 8 Installation and Design Guide*.

(o) Relationship with the facility for predicting reorganization time

When data loading is performed in the creation mode (-d specified), the history of data deletions from tables and indexes is applied to the results of the facility for predicting reorganization time.

If pdload has terminated abnormally, executing the pddbst condition analysis result accumulation facility results in an invalid prediction result* because the reorganization timing cannot be predicted in the data loading completed status. Therefore, if pdload terminates abnormally, re-execute pdload to terminate it normally, then execute the pddbst condition analysis result accumulation facility.

* For example, if rollback occurs due to abnormal termination of data loading, empty database information may be accumulated and used for prediction.

(3) Whether or not options and control statements can be specified when pdload functions are used

Whether or not options and control statements can be specified depends on the pdload functions being used. The applicable functions are as follows:

- Registering data in audit trail table
- Creating divided-input data files

Tables 5-35 and 5-36 show whether or not options and control statements can be specified when these functions are used. For details about registering data in audit trail tables, see the *HiRDB Version 8 System Operation Guide*.

Table 5-35: Whether or not options can be specified when pdload functions are used

Option	Whether or not specifiable	
	Registering data in audit trail table	Creating divided-input data files
-d	O	N
{-a|-b}	M (specify -b)	O (specify -a if the input data file is in the fixed-size data format)
-i	O	N
-l	O	N
-W	M	N
-k	N	N
{-c|-v}	O (specify -v)	O (specify -c if the input data file is in the fixed-size data format)
-n	O	N
-u	O	O
-x	O	N
-s	N	O
-e	N	O
-r	N	N
-z	O (assumed even when omitted)	O
-y	O	N
-o	O	N
-m	O	O
-X	O	O

Legend:

M: Mandatory

O: Optional

N: Cannot be specified

Table 5-36: Whether or not control statements can be specified when pdload functions are used

Control statement	Operand	Whether or not specifiable	
		Registering data in audit trail table	Creating divided-input data files
source	—	M (specify (uoc))	M (specify input data file)
index	—	O	N
idxwork	—	O	N
sort	—	O	N
lobdata	—	N	N
lobcolumn	—	N	N
lobmid	—	N	N
srcuoc	—	M	N
array	—	O	O
extdat	—	N	O
src_work	—	N	M

Control statement	Operand	Whether or not specifiable	
		Registering data in audit trail table	Creating divided-input data files
option	spacelvl	O	O
	tblfree	O	N
	idxfree	O	N
	job	N	N
	cutdtmsg	N	N
	nowait	O	N
	bloblimit	N	N
	exectime	O	O
	null_string	N	N
	divermsg	N	N
	dataerr	N	N
	lengover	N	O
report	—	O	N

Legend:

M: Mandatory

O: Optional

N: Cannot be specified

—: Not applicable

(4) Using files with a BOM

If you selected utf-8 as the character encoding in the pdntenv command, you may be able to use a file with a BOM as the input file for pdload. Table 5-37 shows whether or not files with a BOM can be used with pdload. Note that even when a file with a BOM is used as the input file for pdload, the BOM is skipped. No BOM is included in the file that is output by pdload.

Table 5-37: Whether or not files with a BOM can be used in pdload (applicable to UTF-8)

Option or control statement	Input file		Use of file with a BOM
`-c`	Column structure information file		Y
`-v`	Null value information file		Y
●	Control information file		Y
`source`	Input data file	DAT	Y
		Extended DAT	Y
		Binary	N
		Fixed-size	N
		Created with `pdrorg -W`	N
`index`	Index information file		N
`lobdata`	LOB input file		N
`lobcolumn`	LOB column input file		N
`lobmid`	LOB middle file		N

Legend:

Y: Can be used

N: Cannot be used

●: Not applicable

5.12 Database status in the event of an error and recovery methods

Tables 5-38 through 5-40 describe the database statuses and recovery methods in the event of an error during data loading; Table 5-41 describes the database statuses and recovery methods in the event of an error during data loading on LOB columns, and Table 5-42 describes the database statuses and recovery methods in the event of an input data error.

In the *Classification* column of the tables, Status means the status of the database when the error occurred and Recovery means the recovery method.

Table 5-38: Database status and recovery method (in the event of an error during data loading) (1/3)

Option		Classification	Message output immediately before error		
-l	-i		1.	2.	3.
			None (excluding control statement errors)	KFPL00721-I Row data deleted	KFPL00702-I Data loading started
a or p	s	Status	Status before execution	Null	Null
		Recovery	Re-execute	Re-execute	Re-execute
	c	Status	Status before execution	Null	Null
		Recovery	Re-execute	Re-execute	Re-execute
	n	Status	Status before execution	Null	Null
		Recovery	Re-execute	Re-execute	Re-execute
	x	Status	Status before execution	Null	Null
		Recovery	Re-execute	Re-execute	Re-execute
n	s	Status	Not guaranteed	Not guaranteed	Not guaranteed
		Recovery	Re-execute after recovering database	Re-execute after recovering database	Re-execute after recovering database
	c	Status	Not guaranteed	Not guaranteed	Not guaranteed
		Recovery	Re-execute after recovering database	Re-execute after recovering database	Re-execute after recovering database

Option		Classification	Message output immediately before error		
-l	-i		1.	2.	3.
			None (excluding control statement errors)	KFPL00721-I Row data deleted	KFPL00702-I Data loading started
	n	Status	Not guaranteed	Not guaranteed	Not guaranteed
		Recovery	Re-execute after recovering database	Re-execute after recovering database	Re-execute after recovering database
	x	Status	Not guaranteed	Not guaranteed	Not guaranteed
		Recovery	Re-execute after recovering database	Re-execute after recovering database	Re-execute after recovering database

Table 5-39: Database status and recovery method (in the event of an error during data loading) (2/3)

Option		Classification	Message output immediately before error		
-l	-i		4.	5.	6.
			KFPL00800-I Notification of synchronization point	KFPL15225-E Rollback	KFPL24510-E Detection of invalid data
a or p	s	Status	Data was stored through the row displayed in KFPL00800-I	Status of 3 or 4	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.
		Recovery	Re-execute	Re-execute	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.
	c	Status	Data was stored through the row displayed in KFPL00800-I	Status of 3 or 4	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.

Option		Classification	Message output immediately before error		
-l	-i		4.	5.	6.
			KFPL00800-I Notification of synchronization point	KFPL15225-E Rollback	KFPL24510-E Detection of invalid data
		Recovery	Re-execute	Re-execute	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*
	n	Status	Data was stored through the row displayed in KFPL00800-I	Status of 3 or 4	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.
		Recovery	Re-execute	Re-execute	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*
	x	Status	Data was stored through the row displayed in KFPL00800-I	Status of 3 or 4	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.
		Recovery	Re-execute	Re-execute	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*

Option		Classification	Message output immediately before error		
-l	-i		4.	5.	6.
			KFPL00800-I Notification of synchronization point	KFPL15225-E Rollback	KFPL24510-E Detection of invalid data
n	s	Status	N/A	Not guaranteed	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.
		Recovery	N/A	Re-execute after recovering database	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*
	c	Status	N/A	Not guaranteed	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.
		Recovery	N/A	Re-execute after recovering database	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*
	n	Status	N/A	Not guaranteed	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.

Option		Classification	Message output immediately before error		
-l	-i		4.	5.	6.
			KFPL00800-I Notification of synchronization point	KFPL15225-E Rollback	KFPL24510-E Detection of invalid data
		Recovery	N/A	Re-execute after recovering database	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*
	x	Status	N/A	Not guaranteed	Status of 5 if KFPL15225-E has been displayed. If not, data was stored through the row displayed in KFPL24509-E.
		Recovery	N/A	Re-execute after recovering database	If it is in the status of 5, re-execute. If some of the rows have been stored, specify the -r option and then re-execute.*

Legend:

N/A: If the applicable option is specified, the corresponding message is not displayed.

* If you are re-executing the utility with the -r option specified, do not specify the -d option.

Table 5-40: Database status and recovery method (in the event of an error during data loading) (3/3)

Option		Classification	Message output immediately before error		
-l	-i		7.	8.	9.
			KFPL00703-I Data loading completed	KFPL00715-I Index creation started	KFPL00716-I Index creation completed
a or p	s	Status	Creation of table and index completed	N/A	N/A
		Recovery	Unnecessary[1]	N/A	N/A
	c	Status	Table creation completed	Table creation completed	Creation of table and index completed
		Recovery	Batch creation of index was executed by `pdrorg -k ixmk`	Batch creation of index was executed by `pdrorg -k ixmk`[2]	Unnecessary[1, 2]
	n	Status	Table creation completed	N/A	N/A
		Recovery	Batch creation of index was executed by `pdrorg -k ixmk`[1]	N/A	N/A
	x	Status	Table creation completed	N/A	N/A
		Recovery	Batch creation of index was executed by `pdrorg -k ixmk`[1]	N/A	N/A
n	s	Status	Table creation completed	N/A	N/A
		Recovery	Batch creation of index was executed by `pdrorg -k ixmk`[1]	N/A	N/A
	c	Status	Table creation completed	Table creation completed	Creation of table and index completed
		Recovery	Re-execute after recovering database[3]	Re-execute after recovering database[3]	Unnecessary[1]

Option		Classification	Message output immediately before error		
-l	-i		7.	8.	9.
			KFPL00703-I Data loading completed	KFPL00715-I Index creation started	KFPL00716-I Index creation completed
	n	Status	Table creation completed	N/A	N/A
		Recovery	Batch creation of index was executed by pdrorg -k ixmk[1]	N/A	N/A
	x	Status	Table creation completed	N/A	N/A
		Recovery	Batch creation of index was executed by pdrorg -k ixmk[1]	N/A	N/A

Legend:

N/A: If the applicable option is specified, the corresponding message is not displayed.

[1] If you are executing data loading with an option other than -k d, also see Table 5-41.

[2] If there are multiple index storage RDAREAs, the indexes for which the KFPL00716-I message was issued have been created, but no other indexes have been created. Execute batch creation of indexes for the indexes that have not been created.

[3] If the table storage RDAREA differs from the index storage RDAREA, you can release the table storage RDAREA from the no-log shutdown status, re-initialize the index storage RDAREA, and then execute pdrorg -k ixrc.

Table 5-41: Database status and recovery method (in the event of an error during data loading on LOB columns)

Control statement	-l option	Classification		Message output immediately before error	
				KFPL00707-I Data loading on LOB column started	KFPL00708-I Data loading on LOB column completed
When executing data loading on both LOB column structure base table and LOB columns: `source` statement `lobmid` statement `lobdata` statement	`a` or `p`	Status	`USER`	Creation of LOB column structure base table and index completed	Table creation completed
			`ULOB`	Null if the `-d` option is specified. Status before execution if the option is not specified.	Creation of LOB column created
		Recovery	`USER`	Unnecessary	Unnecessary
			`ULOB`	Data loaded only on LOB columns[2]	Unnecessary[2]
	`n`	Status	`USER`	No-log shutdown	Table creation completed
			`ULOB`	Not guaranteed	Creation of LOB column created
		Recovery	`USER`	Re-execute after recovering database[1]	Unnecessary
			`ULOB`	Re-execute after recovering database[1]	Unnecessary
When executing data loading only on LOB columns: `lobmid` statement `lobdata` statement	`a` or `p`	Status	`USER`	N/A	N/A
			`ULOB`	Status before execution (`-d` option is ignored)	Creation of LOB column created
		Recovery	`USER`	N/A	N/A
			`ULOB`	Re-execute	Unnecessary

Control statement	-l option	Classification		Message output immediately before error	
				KFPL00707-I Data loading on LOB column started	KFPL00708-I Data loading on LOB column completed
	n	Status	USER	N/A	N/A
			ULOB	Not guaranteed	Creation of LOB column created
		Recovery	USER	N/A	N/A
			ULOB	Re-execute after recovering database	Unnecessary

Legend:

N/A: If the applicable option is specified, the corresponding message is not displayed.

USER: User RDAREA

ULOB: User LOB RDAREA

[1] In the no-log mode, you must re-execute data loading on the LOB column structure base table and LOB columns after recovering the table (LOB column structure base table, LOB columns, abstract data type columns with the LOB attribute, and indexes) even when data loading on LOB columns terminated abnormally.

[2] If there are multiple index storage RDAREAs, the indexes for which the KFPL00708-I message was issued have been created, but no other LOB columns have been created. Execute data loading on the LOB columns that have not been created.

Table 5-42: Database status and recovery method (in the event of an input data error)

-e option	errdata operand in source statement	Classification	Error that does not require rollback (termination with return code 4)	Error that requires rollback (termination with return code 8)
Not specified	Not specified	Status	All rows have been stored in the database except those that were output to the error information file.	If `-l n` was not specified, the status before `pdload` was executed. If `-l n` was specified, there is no guarantee.
		Recovery	Create a file containing the applicable data in the input data file, correct them, then re-execute data loading.	If `-l n` was not specified, correct the input data and then re-execute data loading. If `-l n` was specified, recover the database, correct the input data, and then re-execute data loading.
	Specified	Status	All rows have been stored in the database except those that were output to the error information file.	If `-l n` was not specified, the status before `pdload` was executed. If `-l n` was specified, there is no guarantee.
		Recovery	Correct the data that was output to the error data file and then re-execute data loading.	If `-l n` was not specified, correct the input data and then re-execute data loading. If `-l n` was specified, recover the database, correct the input data, and then re-execute data loading.
Specified	Not applicable	Status	Stored in the database up to the row displayed in `KFPL24509-E`.	If `-l n` was not specified, the status before `pdload` was executed. If `-l n` was specified, there is no guarantee.
		Recovery	Correct the row displayed in `KFPL24509-E` and then re-execute data loading with the `-r` option specified.	If `-l n` was not specified, correct the input data and then re-execute data loading. If `-l n` was specified, recover the database, correct the input data, and then re-execute data loading.

Note

Errors that require rollback include LOB column storage data errors and negative-value errors on the length of variable-length character string and on the number of elements for repetition columns.

A key value duplication error in a mode other than the index update mode (`-i s`) results in a database corruption error where the table and indexes do not match. To recover the error, restore the database to its status before `pdload` was executed, correct the input data, and then re-execute data loading.

5.13 Examples

Table 5-43 provides a list of the examples of using the database load utility.

Table 5-43: Examples of using the database load utility

Example	Description	Classification
Example 1	Loading data to a row-partitioned table • Explanation is based on the input data file in the DAT format and in the fixed-size data format. • The example uses a column structure information file.	S
Example 2	Loading data to a row-partitioned table in units of tables	S
Example 3	Loading data to a row-partitioned table in units of RDAREAs	S
Example 4	Loading data to a table with LOB columns • The example loads data to both a LOB column structure base table and LOB columns at the same time.	S
Example 5	Loading data to a table with LOB columns • The example loads data only to the LOB column structure base table.	S
Example 6	Loading data to a table with LOB columns • The example loads data only to the LOB columns.	S
Example 7	Loading data into an audit trail table • The example loads data from audit trail files waiting for data loading into an audit trail table.	S
Example 8	Loading data to a non-row-partitioned table • The input data file is located at the front-end server.	P
Example 9	Loading data to a row-partitioned table • The input data file is located at the front-end server.	P
Example 10	Loading data to a row-partitioned table • The input data file is located at each back-end server.	P
Example 11	Loading data to a table with LOB columns • The example loads data to both LOB column structure base tables and LOB columns at the same time. • The table is row partitioned and the example loads data in units of RDAREAs. • The input data file is located at the back-end server that contains the RDAREAs subject to data loading.	P

Example	Description	Classification
Example 12	Loading data to a table with LOB columns • The example uses a LOB column input file. • The table is row partitioned and the example loads data in units of tables. • The input data file is located at the front-end server.	P
Example 13	Loading data to a table with repetition columns • The table is row partitioned and the example loads data in units of tables. • The table has a cluster key index and the input data is sorted by the cluster key values; therefore, the example specifies an option to not check the sequence of cluster key values.	P
Example 14	Loading data to a table using a plug-in • The table contains columns of SGMLTEXT type. • The input data file is in the DAT format. • The example uses a column structure information file to specify information about a constructor function.	P
Example 15	Loading data to a table using a plug-in • The table contains columns of GEOMETRY type. • The input data file is in the fixed-size data format. • The example uses a column structure information file to specify information about a constructor function.	P
Example 16	Loading data to a table using a plug-in • The table contains columns of SGMLTEXT type and LOB columns. • The example directly specifies the LOB parameters and the name of the LOB input file in the input data file, and provides a separate LOB input file.	P
Example 17	Loading data to a table using a plug-in • The table contains columns of SGMLTEXT type and LOB columns. • The example loads data only to the LOB column structure base table at first, then loads data to the LOB columns concurrently at each back-end server. • The example directly specifies the LOB parameters and the name of the LOB input file in the input data file, and provides a separate LOB input file	P

S: HiRDB/Single Server

P: HiRDB/Parallel Server

5.13.1 HiRDB/Single Server

(1) Loading data to a table

Example 1

This example loads data to table TABLE1. The explanation is based on the input data file in the DAT format and in the fixed-size data format.

The example assumes that the following table and index have been defined:

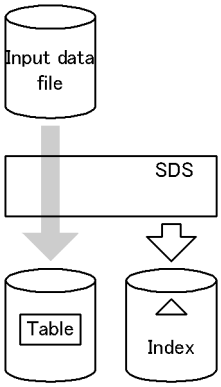
- Table definition:

```
CREATE TABLE TABLE1(C1 INT,C2 CHAR(8),C3 INT) IN PDBUSER01
```

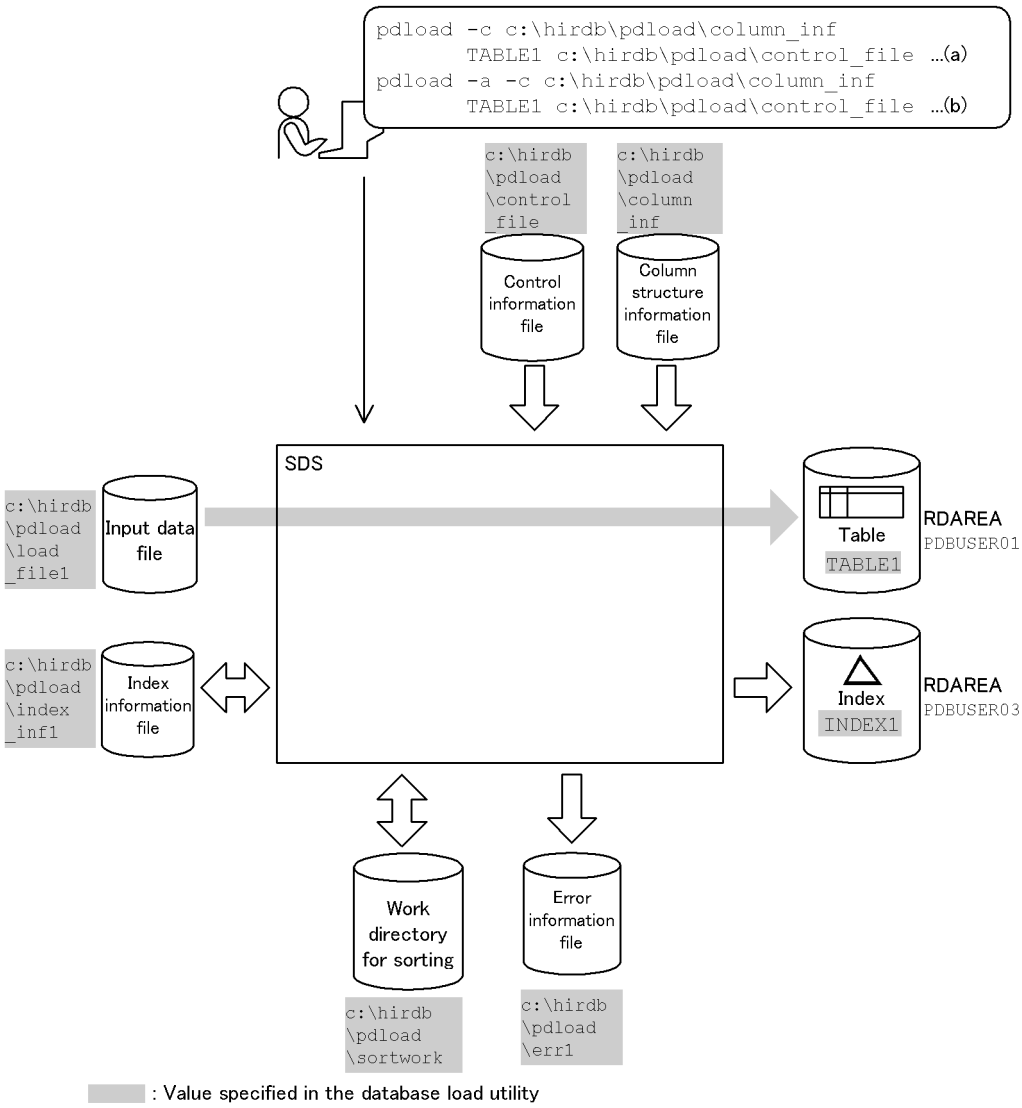
- Index definition:

```
CREATE INDEX INDEX1 ON TABLE1(C1) IN PDBUSER03
```

Overview



Relationship between input/output files and RDAREAs



(a) Input data file in the DAT format

Contents of the input file

```
1,ABCDEFG,99999,
2,a,-1,
3,XYZ,100,
    :
```

Explanation of the command

The example loads data to the table (TABLE1).

c:\hirdb\pdload\column_inf: Name of the column structure information file

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the column structure information file (c:\hirdb\pdload\column_inf)

```
C1                  1
C2                  1
C3                  1
*skipdata*          2
```

Explanation:

1. Stores input data in TABLE1 in the order of the columns (C1, C2, and C3).
2. Specifies a skipdata statement to ignore the separator character at the end of a line.

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1   1
index INDEX1 c:\hirdb\pdload\index_inf1                        2
sort c:\hirdb\pdload\sortwork,8192                             3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   c:\hirdb\pdload\load_file1: Name of the input data file

   c:\hirdb\pdload\err1: Name of the error information file

2. Specifies the index information file to which index information is to be output:

   INDEX1: Index identifier

   c:\hirdb\pdload\index_inf1: Name of the index information file

3. Specifies the work directory for sorting:

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

(b) Input data file in the fixed-size data format

Contents of the input file

```
1  ABCDEFG  99999
2  a        -1
3  XYZ      100
     :
```

Note: there is a linefeed code at the end of the line.

Explanation of the command

The example loads data to the table (`TABLE1`).

`-a`: Specifies that the input data file is in the fixed-size data format.

`c:\hirdb\pdload\column_inf`: Name of the column structure information file

`c:\hirdb\pdload\control_file`: Name of the control information file

Contents of the column structure information file (c:\hirdb\pdload\column_inf)

```
C1,type=char(3)              1
C2,type=char(8)              1
C3,type=char(6)              1
*skipdata*,length=1          2
```

Explanation:

1. Specifies the data type of the columns in the input data.
2. Specifies a skipdata statement to ignore the linefeed code.

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1   1
index INDEX1 c:\hirdb\pdload\index_inf1                        2
sort c:\hirdb\pdload\sortwork,8192                             3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   `c:\hirdb\pdload\load_file1`: Name of the input data file

   `c:\hirdb\pdload\err1`: Name of the error information file

2. Specifies the index information file to which index information is to be output:

   `INDEX1`: Index identifier

   `c:\hirdb\pdload\index_inf1`: Name of the index information file

3. Specifies the work directory for sorting:

   `c:\hirdb\pdload\sortwork`: Name of the directory in which the sort work file is created

   `8192`: Size of buffer for sorting (in KB)

(2) Loading data to a row-partitioned table in units of tables

Example 2

This example loads data to table TABLE1 in units of tables.

The example assumes that the following table and indexes have been defined:

- Table definition:

```
CREATE TABLE TABLE1(C1 INT NOT NULL,C2 CHAR(8),C3 INT)
             IN ((PDBUSER01) C1 > 10,(PDBUSER02))
```

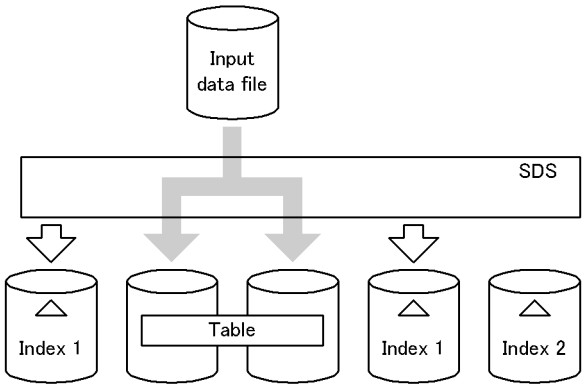
- Index definition (partitioning key index):

```
CREATE INDEX INDEX1 ON TABLE1(C1) IN
((PDBUSER03),(PDBUSER04))
```

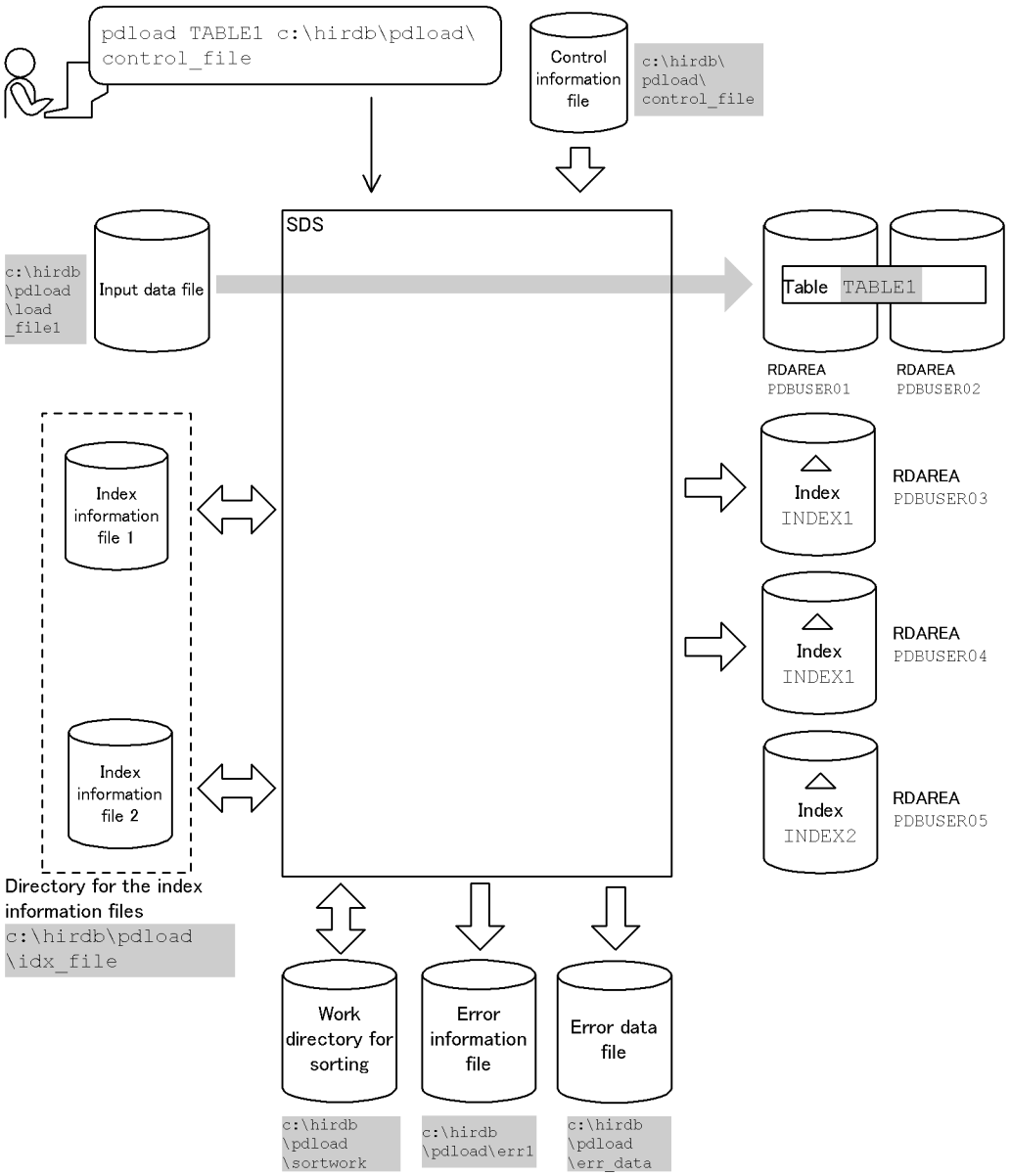
- Index definition (non-partitioning key index):

```
CREATE INDEX INDEX2 ON TABLE1(C2,C1) IN (PDBUSER05)
```

Overview



Relationship between input/output files and RDAREAs


pdload TABLE1 c:\hirdb\pdload\
control_file
Control information file
c:\hirdb\
pdload\
control_file
SDS
c:\hirdb
\pdload
\load
_file1
Input data file
Table TABLE1
RDAREA
PDBUSER01
RDAREA
PDBUSER02
Index
information
file 1
Index
INDEX1
RDAREA
PDBUSER03
Index
INDEX1
RDAREA
PDBUSER04
Index
information
file 2
Index
INDEX2
RDAREA
PDBUSER05
Directory for the index
information files
c:\hirdb\pdload
\idx_file
Work
directory for
sorting
Error
information
file
Error data
file
c:\hirdb
\pdload
\sortwork
c:\hirdb
\pdload\err1
c:\hirdb
\pdload
\err_data
: Value specified in the database load utility

(a) Loading data to TABLE1

Explanation of the command

The example loads data to a row-partitioned table (TABLE1) in units of tables.

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1
       errdata=c:\hirdb\pdload\err_data,1000                    1
idxwork c:\hirdb\pdload\idx_file                                2
sort c:\hirdb\pdload\sortwork,8192                              3
```

Explanation:

1. Specifies the input data file and the error information file to which error information (up to 1000 entries) is to be output in the event of an error in the input data:

   c:\hirdb\pdload\load_file1: Name of the input data file

   c:\hirdb\pdload\err1: Name of the error information file

   c:\hirdb\pdload\err_data: Name of the error data file

2. Specifies the directory where index information files are to be created:

   c:\hirdb\pdload\idx_file: Name of the directory where the index information files are to be created

3. Specifies the work directory for sorting:

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

(3) Loading data to a row-partitioned table in units of RDAREAs

Example 3

This example loads data to a row-partitioned table (TABLE1) in units of RDAREAs. It executes data loading twice on PDBUSER01 and PDBUSER02.

The example assumes that the following table and indexes have been defined:

- Table definition:

```
CREATE TABLE TABLE1(C1 INT NOT NULL,C2 CHAR(8),C3 INT)
             IN ((PDBUSER01) C1 > 10,(PDBUSER02))
```

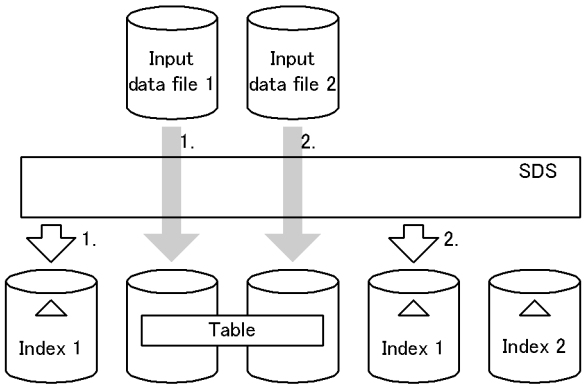
- Index definition (partitioning key index):

```
CREATE INDEX INDEX1 ON TABLE1(C1) IN
((PDBUSER03),(PDBUSER04))
```

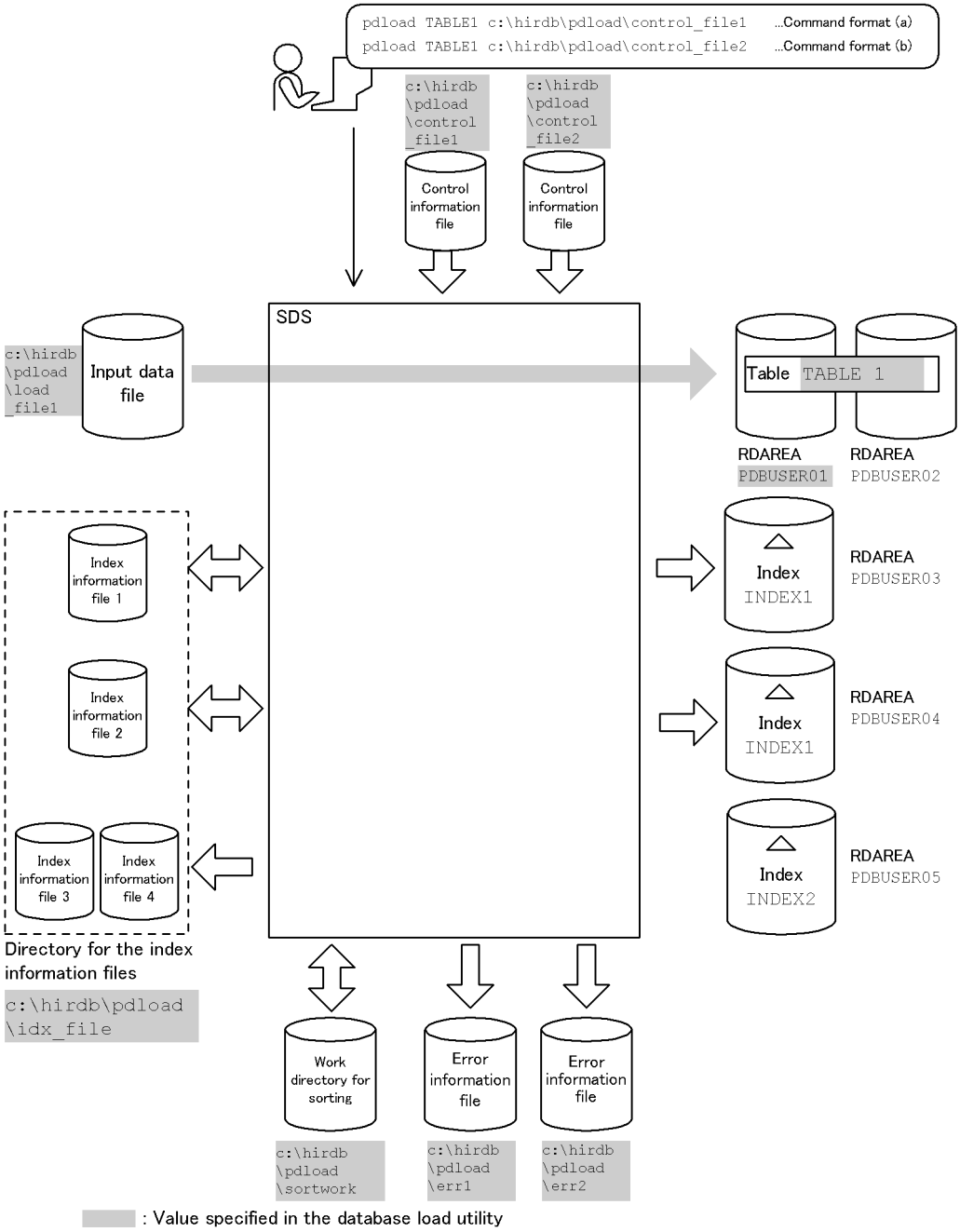
- Index definition (non-partitioning key index):

```
CREATE INDEX INDEX2 ON TABLE1(C2,C1) IN (PDBUSER05)
```

Overview



Relationship between input/output files and RDAREAs



(a) Loading data to PDBUSER01

Explanation of the command

The example loads data to a row-partitioned table (TABLE1) in units of RDAREAs (PDBUSER01).

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source PDBUSER01 c:\hirdb\pdload\load_file1
       error=c:\hirdb\pdload\err1                    1
idxwork c:\hirdb\pdload\idx_file                     2
sort c:\hirdb\pdload\sortwork,8192                   3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   PDBUSER01: Name of the RDAREA that contains the table subject to data loading

   c:\hirdb\pdload\load_file1: Name of the input data file

   c:\hirdb\pdload\err1: Name of the error information file

2. Specifies the directory where index information files are to be created:

   c:\hirdb\pdload\idx_file: Name of the directory where the index information files are to be created

3. Specifies the work directory for sorting:

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

(b) Loading data to PDBUSER02

Explanation of the command

The example loads data to a row-partitioned table (TABLE1) in units of RDAREAs (PDBUSER02).

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source PDBUSER02 c:\hirdb\pdload\load_file2
       error=c:\hirdb\pdload\err2                    1
idxwork c:\hirdb\pdload\idx_file                     2
sort c:\hirdb\pdload\sortwork,8192                   3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   `PDBUSER02`: Name of the RDAREA that contains the table subject to data loading

   `c:\hirdb\pdload\load_file2`: Name of the input data file

   `c:\hirdb\pdload\err2`: Name of the error information file

2. Specifies the directory where index information files are to be created:

   `c:\hirdb\pdload\idx_file`: Name of the directory where the index information files are to be created

3. Specifies the work directory for sorting:

   `c:\hirdb\pdload\sortwork`: Name of the directory in which the sort work file is created

   `8192`: Size of buffer for sorting (in KB)

(c) Creating INDEX2

You cannot use `INDEX2` as is.

Execute batch index creation with `pdrorg` (specifying `-k ixmk`) using index information files 3 and 4 to which index information has been output.

(4) Loading data to LOB column structure base tables and LOB columns

Example 4

For a table with LOB columns (`TABLE2`), this example loads data to both the LOB column structure base table and LOB columns at the same time.

The example assumes that the following table and index have been defined:

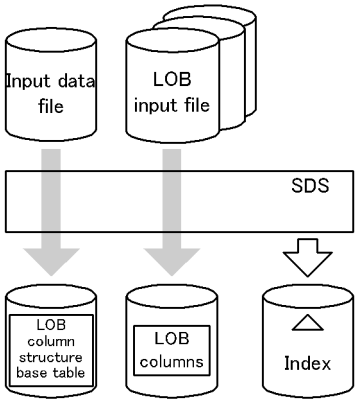
- Table definition:

```
CREATE TABLE TABLE2(C1 INT,C2 BLOB IN LOBUSER01) IN PDBUSER01
```

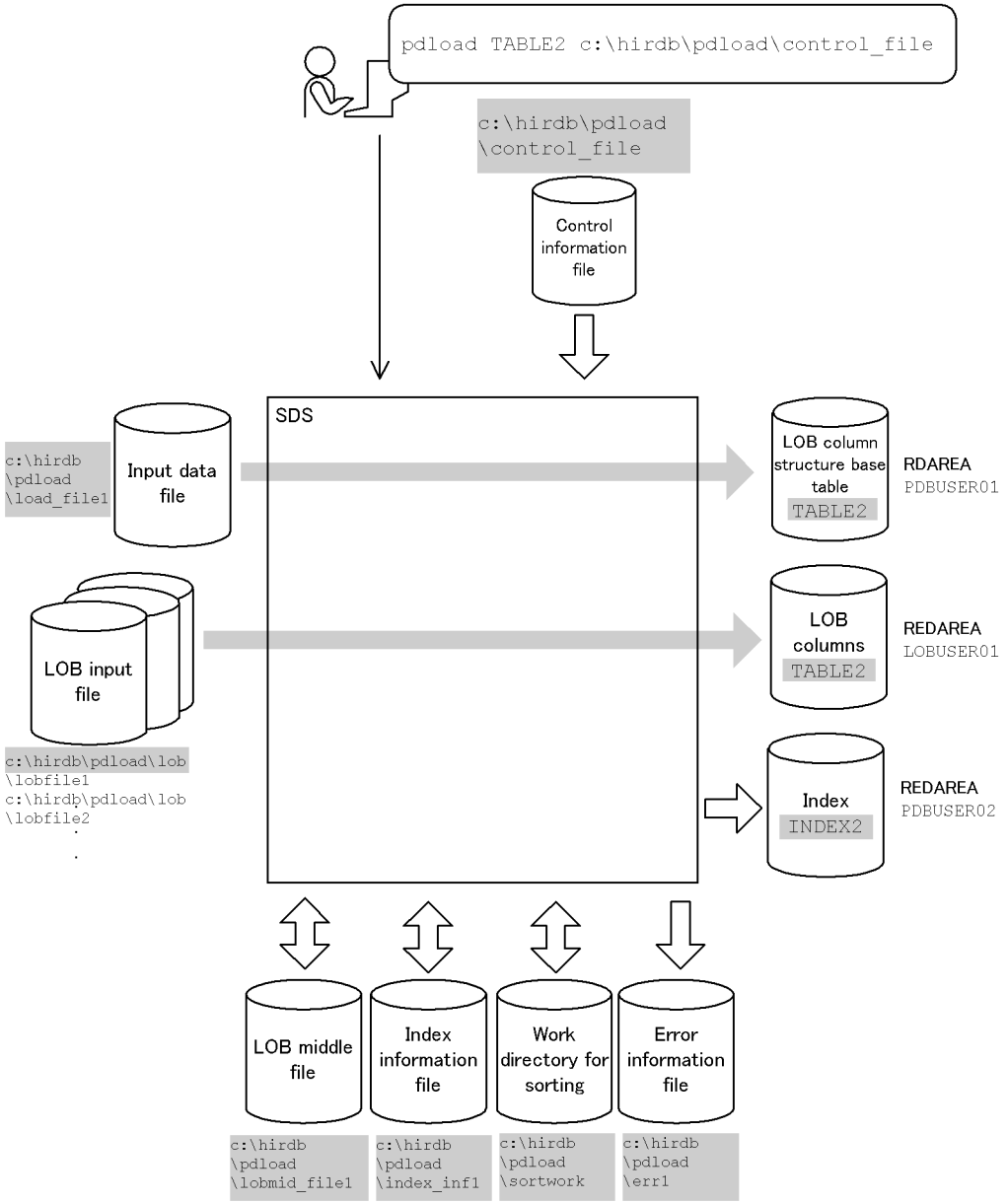
- Index definition:

```
CREATE INDEX INDEX2 ON TABLE2(C1) IN PDBUSER02
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to a table with LOB columns (TABLE2) in units of tables.

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1   1
index INDEX2 c:\hirdb\pdload\index_inf1                       2
sort c:\hirdb\pdload\sortwork,8192                            3
lobdata c:\hirdb\pdload\lob                                   4
lobmid c:\hirdb\pdload\lobmid_file1                           5
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   c:\hirdb\pdload\load_file1: Name of the input data file

   Contents of the input data file (DAT format):

   ```
   1,lobfile1
   2,lobfile2
   3,lobfile3
   ```

   c:\hirdb\pdload\err1: Name of the error information file

2. Specifies the index information file to which index information is to be output.

   INDEX2: Index identifier

   c:\hirdb\pdload\index_inf1: Names of the index information file

3. Specifies the work directory for sorting:

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

4. Specifies the directory containing the LOB input files:

   c:\hirdb\pdload\lob: pathname of the LOB input files

5. Specifies the file to which information required for loading LOB column data is to be output when loading data to the LOB column structure base table (this file is used as is when loading data to the LOB columns):

   c:\hirdb\pdload\lobmid_file1: Name of the LOB middle file

(5) Loading data to a LOB column structure base table only

Example 5

For a table with LOB columns (TABLE2), this example loads data to the LOB column structure base table only.

The example assumes that the following table and indexes have been defined:

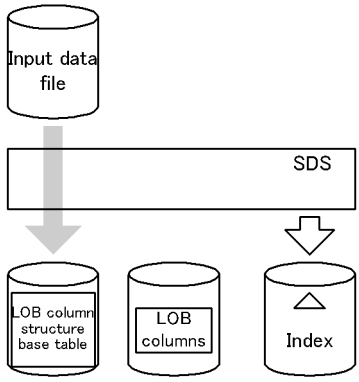
- Table definition:

```
CREATE TABLE TABLE2(C1 INT,C2 BLOB IN LOBUSER01) IN PDBUSER01
```

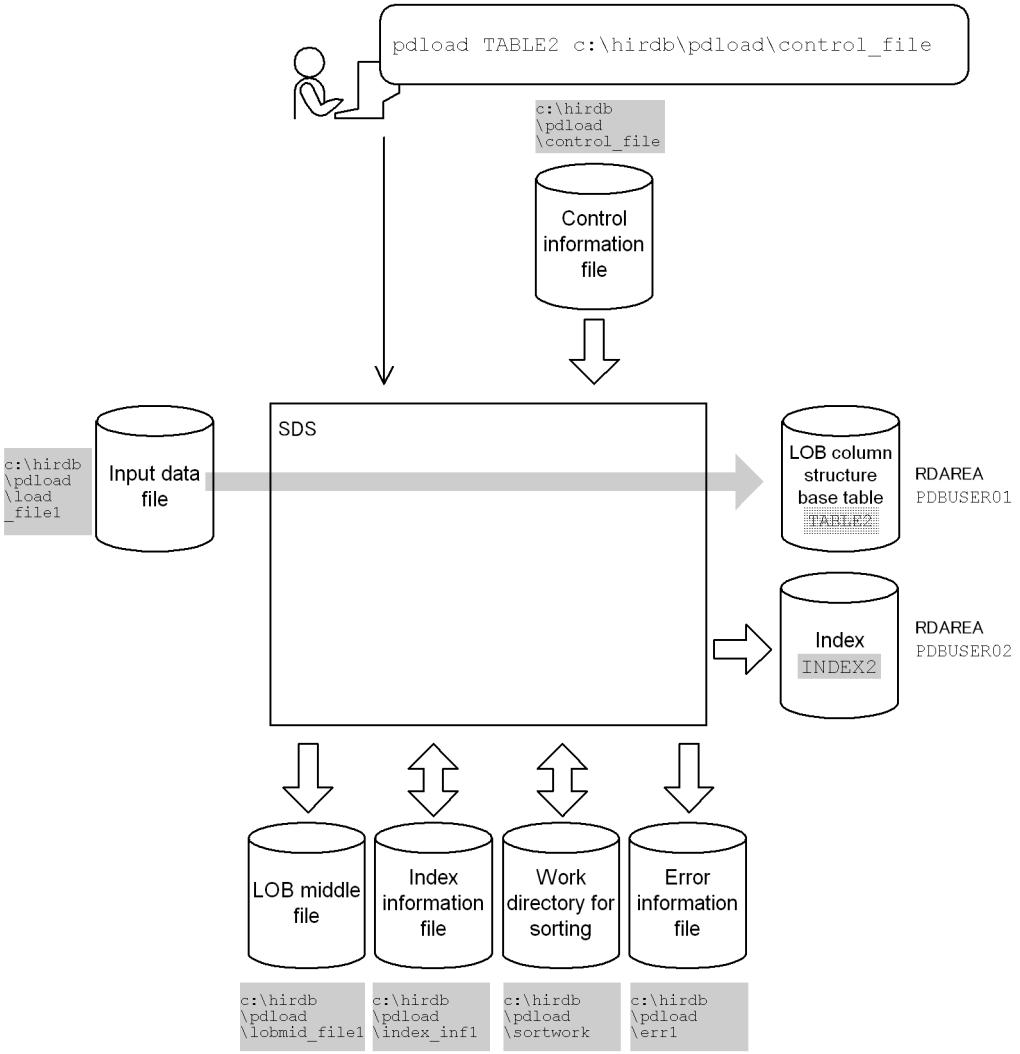
- Index definition:

```
CREATE INDEX INDEX2 ON TABLE2(C1) IN PDBUSER02
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to a table with LOB columns (`TABLE2`) in units of tables.

`c:\hirdb\pdload\control_file`: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1   1
index INDEX2 c:\hirdb\pdload\index_inf1                       2
sort c:\hirdb\pdload\sortwork,8192                            3
lobmid c:\hirdb\pdload\lobmid_file1                           4
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   `c:\hirdb\pdload\load_file1`: Name of the input data file

   Contents of the input data file (DAT format):

   ```
   1,lobfile1

   2,lobfile2

   3,lobfile3
   ```

   `c:\hirdb\pdload\err1`: Name of the error information file

2. Specifies the index information file to which index information is to be output.

   `INDEX2`: Index identifier

   `c:\hirdb\pdload\index_inf1`: Name of the index information file

3. Specifies the work directory for sorting:

   `c:\hirdb\pdload\sortwork`: Name of the directory in which the sort work file is created

   `8192`: Size of buffer for sorting (in KB)

4. Specifies the file to which information required for loading LOB column data is to be output when loading data to the LOB column structure base table:

   `c:\hirdb\pdload\lobmid_file1`: Name of the LOB middle file

(6) Loading data to LOB columns only

Example 6

For a table with LOB columns (`TABLE2`), this example loads data to the LOB columns only.

The example assumes that the following table and index have been defined:

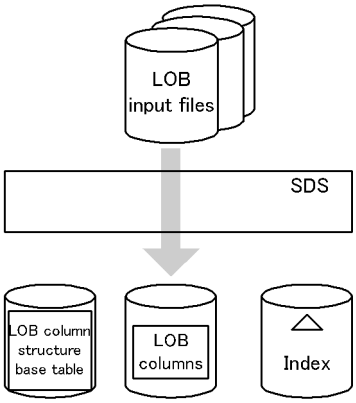
- Table definition:

```
CREATE TABLE TABLE2(C1 INT,C2 BLOB IN LOBUSER01) IN PDBUSER01
```

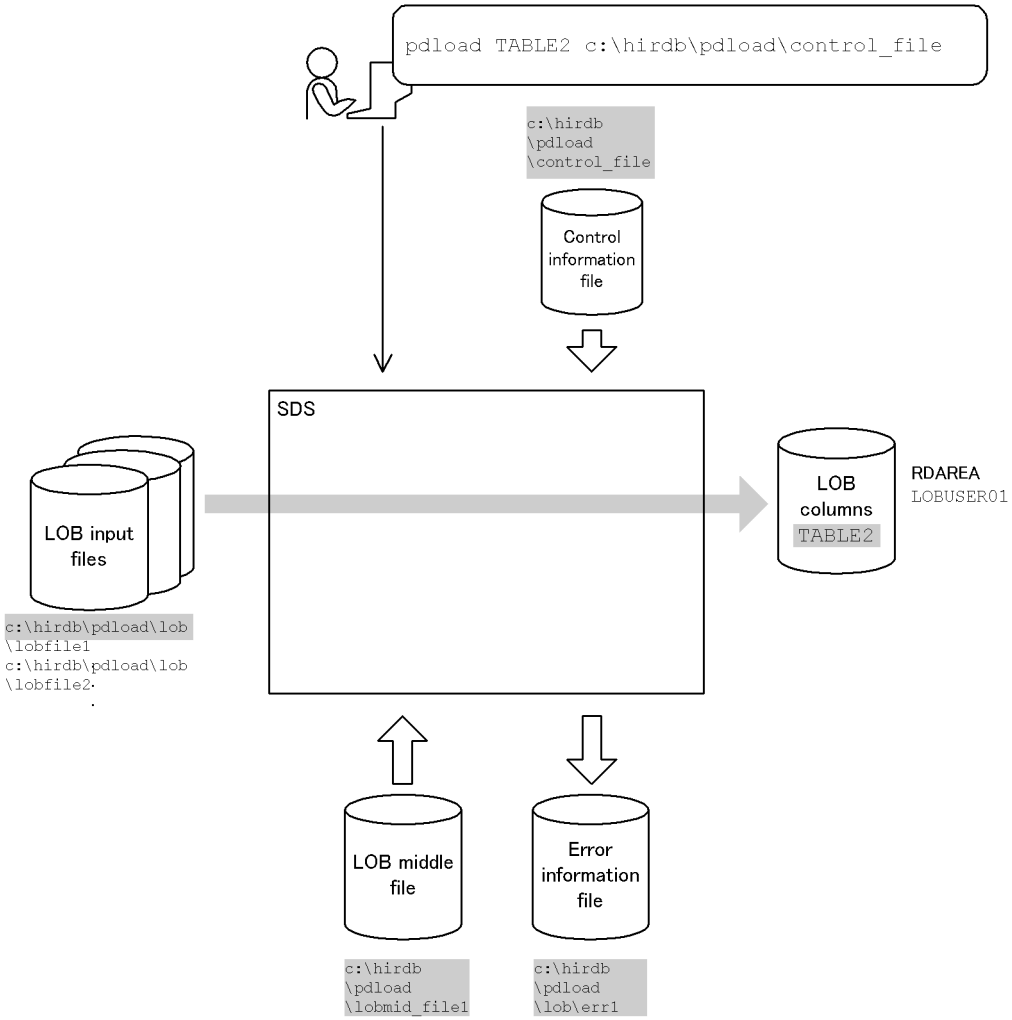
- Index definition:

```
CREATE INDEX INDEX2 ON TABLE2(C1) IN PDBUSER02
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to a table with LOB columns (`TABLE2`) in units of tables.

`c:\hirdb\pdload\control_file`: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
lobdata c:\hirdb\pdload\lob                                          1
lobmid c:\hirdb\pdload\lobmid_file1 error=c:\hirdb\pdload\err1   2
```

Explanation:

1. Specifies the directory containing the input data file.

   `c:\hirdb\pdload\lob`: Pathname of the LOB input files

   The example uses the following input data file while loading data to the LOB column structure base table:

   Contents of the input data file (DAT format):

   ```
   1,lobfile1
   2,lobfile2
   3,lobfile3
   ```

2. Specifies the LOB middle file containing the information required for loading data to a LOB column that was created during data loading to the LOB column structure base table, and the error information file to which error information is to be output in the event of a LOB input error:

   `c:\hirdb\pdload\lobmid_file1`: Name of the LOB middle file

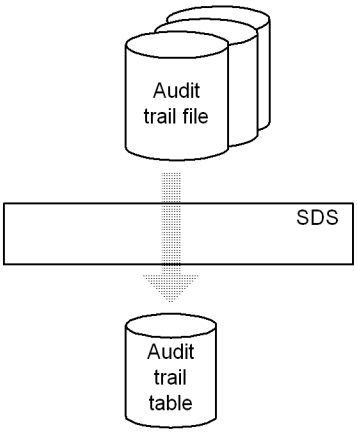
   `c:\hirdb\pdload\err1`: Name of the error information file

(7) Loading data into an audit trail table

Example 7

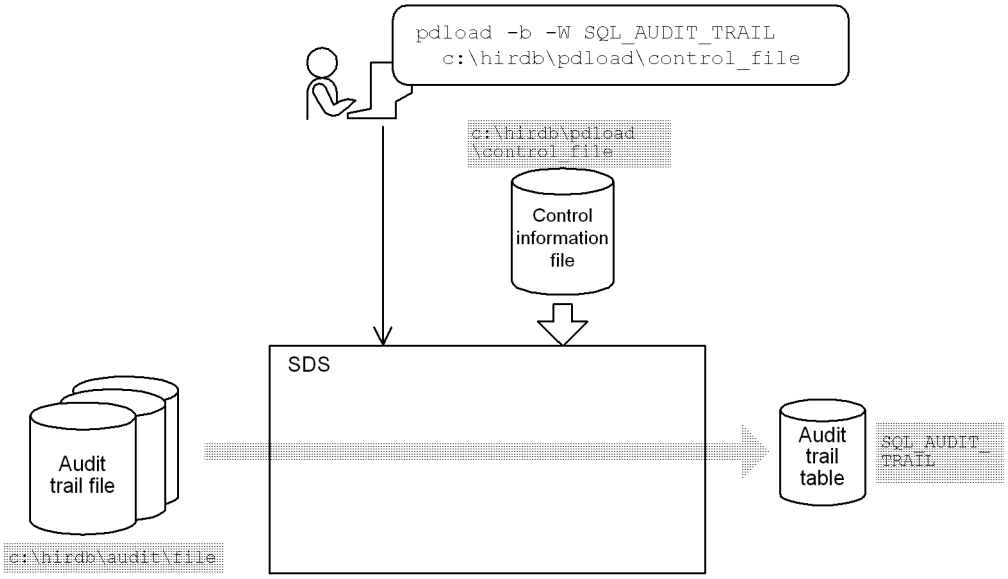
This example loads data from audit trail files waiting for data loading into the audit trail table `SQL_AUDIT_TRAIL`. The example assumes that no index has been defined for the audit trail table.

Overview



SDS: Single server

Relationship between input/output files and RDAREAs



: Value specified in the database load utility
SDS: Single server

Explanation of the command

This example loads data from audit trail files waiting for data loading (`c:\hirdb\audit\file`) into the audit trail table `SQL_AUDIT_TRAIL`.

`-b` and `-W`: Mandatory options for loading data into audit trail table

`c:\hirdb\pdload\control_file`: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source (uoc)                                          1
srcuoc param='dir=c:\hirdb\audit\file,file=all'       2
```

Explanation:

1. Specifies that an audit trail data registration UOC is to be used.
2. Specifies the parameters that are to be passed to the audit trail data registration UOC.

   `dir=c:\hirdb\audit\file`: HiRDB file name of the audit trail file

`file=all`: Specifies that all audit trail files waiting for data loading are to be used

5.13.2 HiRDB/Parallel Server

(1) Loading data to a table

Example 8

This example loads data to table TABLE1.

The example assumes that the following table and index have been defined:

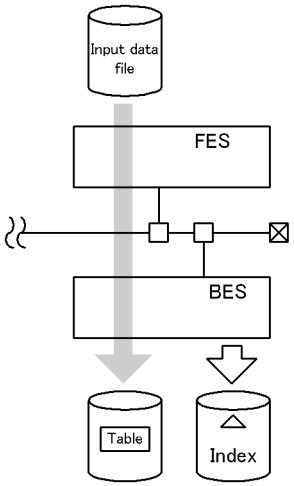
- Table definition:

```
CREATE TABLE TABLE1(C1 INT NOT NULL,C2 CHAR(8),C3 INT) IN
PDBUSER01
```

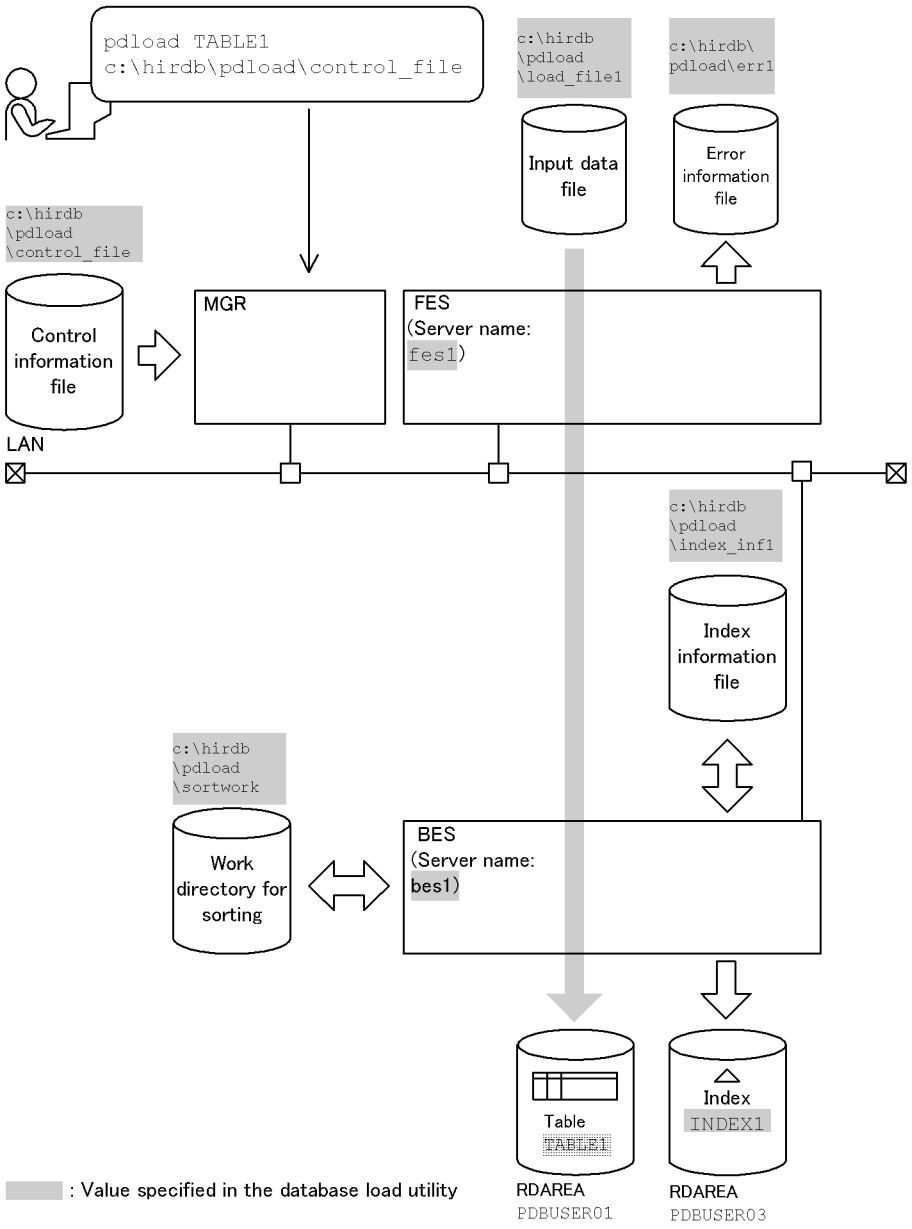
- Index definition:

```
CREATE INDEX INDEX1 ON TABLE1(C1) IN PDBUSER03
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to the table (TABLE1).

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source fes1:c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1   1
index INDEX1 c:\hirdb\pdload\index_inf1                            2
sort bes1 c:\hirdb\pdload\sortwork,8192                            3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   `fes1`: Name of the server containing the input data file

   `c:\hirdb\pdload\load_file1`: Name of the input data file

   `c:\hirdb\pdload\err1`: Name of the error information file

2. Specifies the index information file to which index information is to be output:

   `INDEX1`: Index identifier

   `c:\hirdb\pdload\index_inf1`: Name of the index information file

3. Specifies the work directory for sorting:

   `bes1`: Name of the server used to create the sort work file

   `c:\hirdb\pdload\sortwork`: Name of the directory in which the sort work file is created

   `8192`: Size of buffer for sorting (in KB)

(2) Loading data to a row-partitioned table in units of tables

Example 9

This example loads data to table TABLE1 in units of tables.

The example assumes that the following table and indexes have been defined:

- Table definition:

```
CREATE TABLE TABLE1(C1 INT NOT NULL,C2 CHAR(8),C3 INT)
             IN ((PDBUSER01) C1 > 10,(PDBUSER02))
```

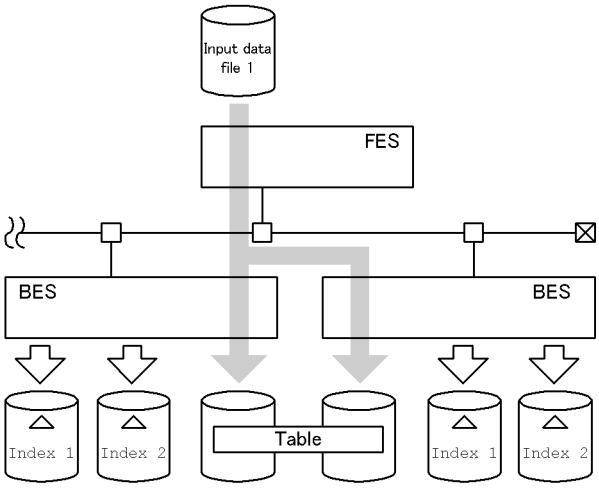
- Index definition (partitioning key index):

```
CREATE INDEX INDEX1 ON TABLE1(C1) IN
((PDBUSER03),(PDBUSER05))
```

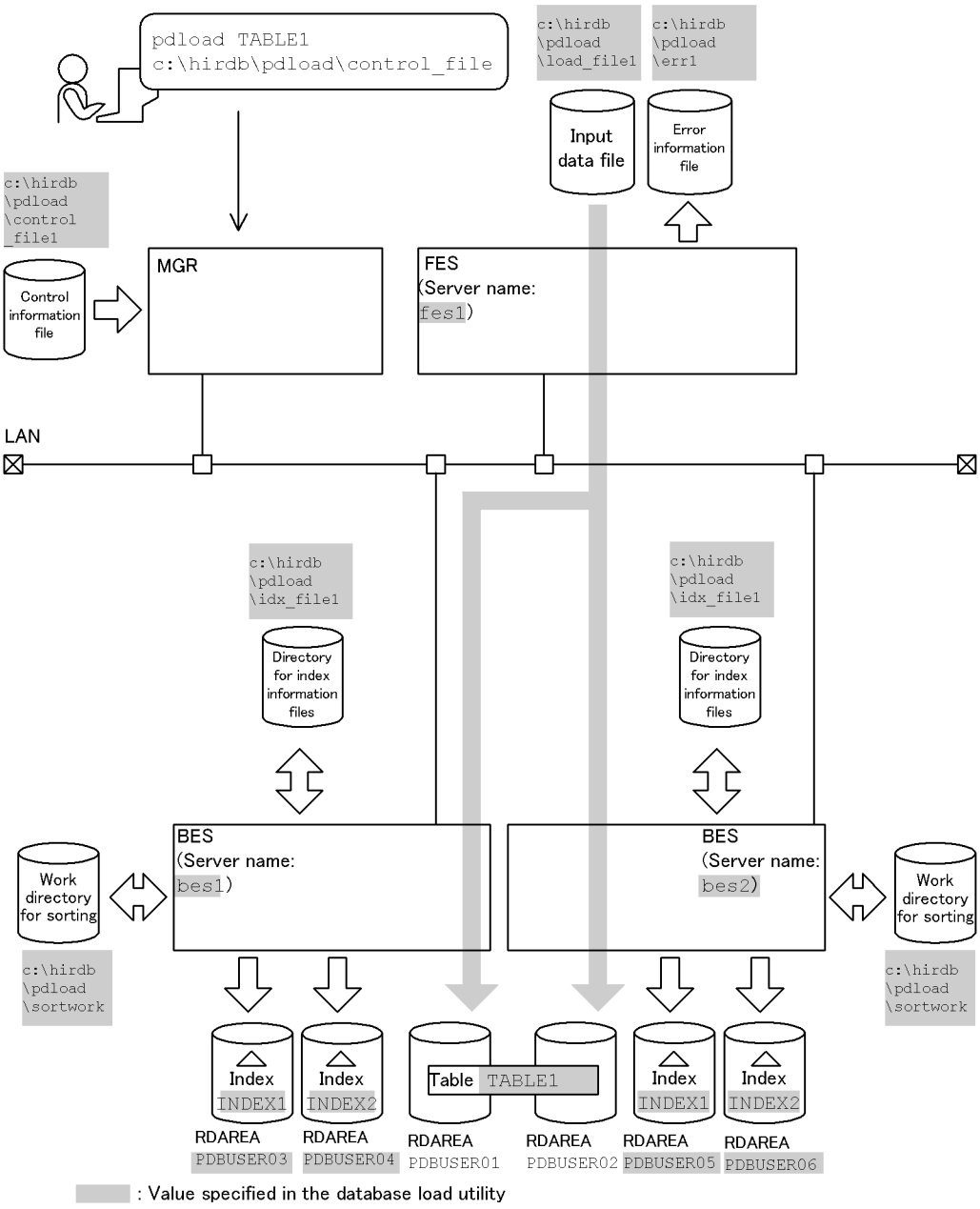
- Index definition (non-partitioning key index):

```
CREATE INDEX INDEX2 ON TABLE1(C2,C1) IN
((PDBUSER04),(PDBUSER06))
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to a row-partitioned table (TABLE1) in units of tables.

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source fes1:c:\hirdb\pdload\load_file1 error=c:\hirdb\pdload\err1   1
idxwork bes1 c:\hirdb\pdload\idx_file                              2
idxwork bes2 c:\hirdb\pdload\idx_file                              2
sort bes1 c:\hirdb\pdload\sortwork,8192                            3
sort bes2 c:\hirdb\pdload\sortwork,8192                            3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   fes1: Name of the server containing the input data file

   c:\hirdb\pdload\load_file1: Name of the input data file

   c:\hirdb\pdload\err1: Name of the error information file

2. Specifies the directory where index information files are to be created:

   bes1, bes2: Names of the servers where index information files are to be created

   c:\hirdb\pdload\idx_file: Name of the directory where the index information files are to be created

3. Specifies the work directory for sorting:

   bes1, bes2: Names of the servers where the sort work file is to be created

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

(3) Loading data to a row-partitioned table in units of RDAREAs

Example 10

This example loads data to a row-partitioned table (TABLE1) in units of RDAREAs.

The example assumes that the following table and indexes have been defined:

- Table definition:

```
CREATE TABLE TABLE1(C1 INT NOT NULL,C2 CHAR(8),C3 INT)
             IN ((PDBUSER01) C1 > 10,(PDBUSER02))
```

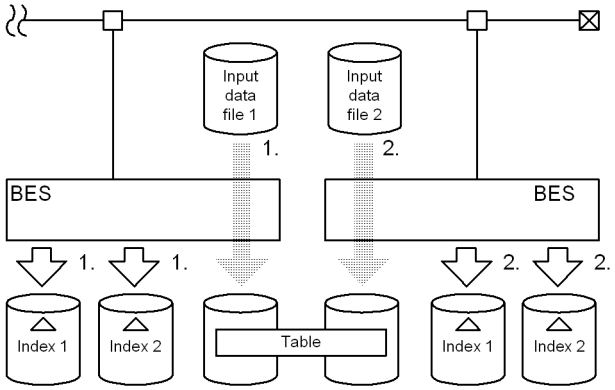
- Index definition (partitioning key index):

```
CREATE INDEX INDEX1 ON TABLE1(C1) IN
((PDBUSER03),(PDBUSER05))
```

- Index definition (non-partitioning key index):

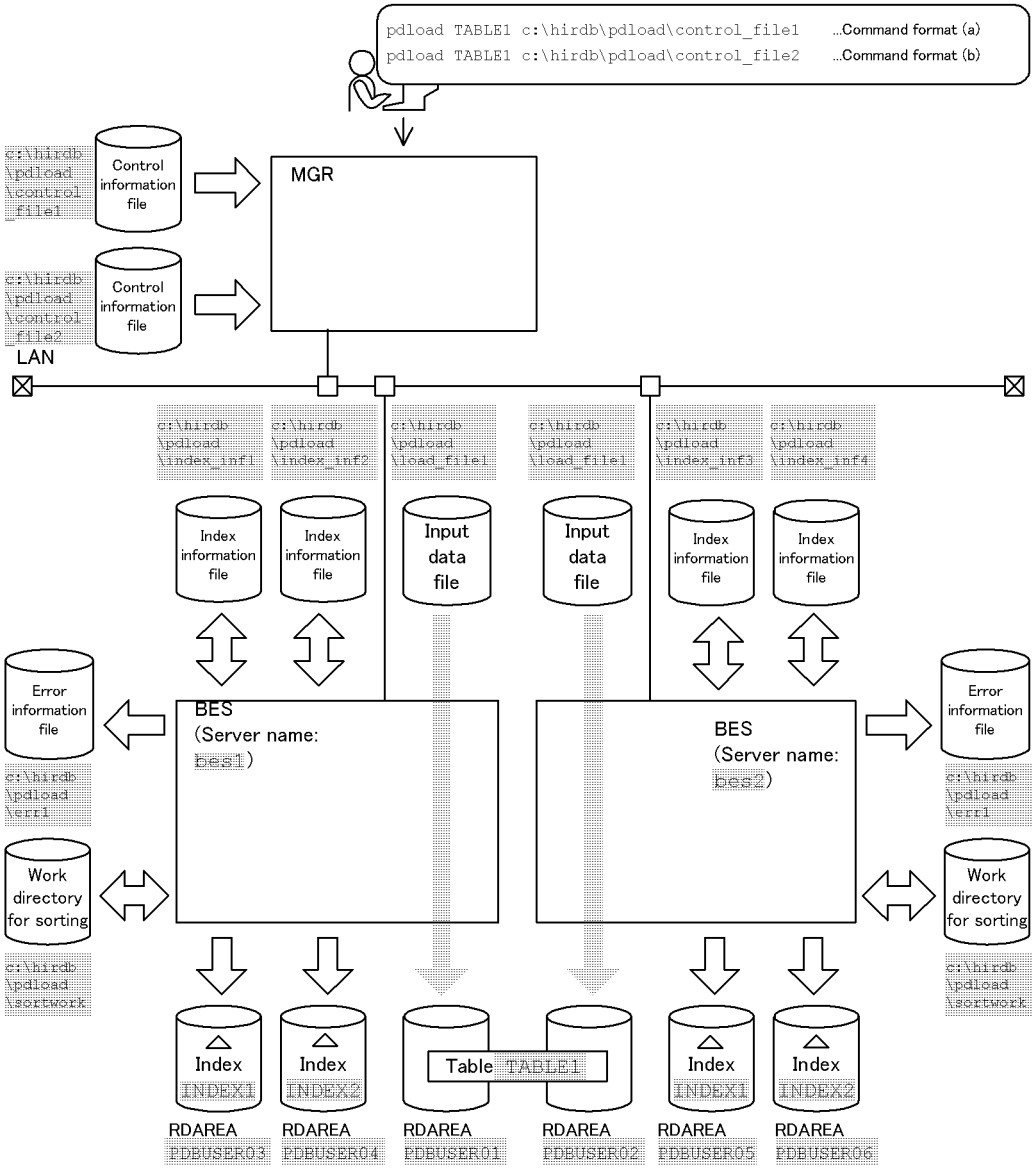
```
CREATE INDEX INDEX2 ON TABLE1(C2,C1) IN
((PDBUSER04),(PDBUSER06))
```

Overview



FES: Front-end server
BES: Back-end server

Relationship between input/output files and RDAREAs



(a) Loading data to PDBUSER01

Explanation of the command

The example loads data to a row-partitioned table (TABLE1) in units of RDAREAs (PDBUSER01).

c:\hirdb\pdload\control_file1: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file1)

```
source PDBUSER01 bes1:c:\hirdb\pdload\load_file1
       error=c:\hirdb\pdload\err1                          1
index INDEX1 PDBUSER03 c:\hirdb\pdload\index_inf1          2
index INDEX2 PDBUSER04 c:\hirdb\pdload\index_inf2          2
sort bes1 c:\hirdb\pdload\sortwork,8192                    3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   PDBUSER01: Name of the RDAREA that contains the table subject to data loading

   bes1: Name of the server containing the input data file

   c:\hirdb\pdload\load_file1: Name of the input data file

   c:\hirdb\pdload\err1: Name of the error information file

2. Specifies the index information files to which index information is to be output:

   INDEX1, INDEX2: Index identifiers

   PDBUSER03, PDBUSER04: Names of the index storage RDAREAs

   c:\hirdb\pdload\index_inf1, c:\hirdb\pdload\index_inf2: Names of the index information files

3. Specifies the work directory for sorting:

   bes1: Name of the server used to create the sort work file

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

(b) Loading data to PDBUSER02

Explanation of the command

The example loads data to a row-partitioned table (TABLE1) in units of RDAREAs (PDBUSER02).

`c:\hirdb\pdload\control_file2`: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file2)

```
source PDBUSER02 bes2:c:\hirdb\pdload\load_file2
       error=c:\hirdb\pdload\err2                          1
index INDEX1 PDBUSER05 c:\hirdb\pdload\index_inf3          2
index INDEX2 PDBUSER06 c:\hirdb\pdload\index_inf4          2
sort bes2 c:\hirdb\pdload\sortwork,8192                    3
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   `PDBUSER02`: Name of the RDAREA that contains the table subject to data loading

   `bes2`: Name of the server containing the input data file

   `c:\hirdb\pdload\load_file2`: Name of the input data file

   `c:\hirdb\pdload\err2`: Name of the error information file

2. Specifies the index information files to which index information is to be output:

   `INDEX1`, `INDEX2`: Index identifiers

   `PDBUSER05`, `PDBUSER06`: Names of the index storage RDAREAs

   `c:\hirdb\pdload\index_inf3`, `c:\hirdb\pdload\index_inf4`: Names of the index information files

3. Specifies the work directory for sorting:

   `bes2`: Name of the server used to create the sort work file

   `c:\hirdb\pdload\sortwork`: Name of the directory in which the sort work file is created

   `8192`: Size of buffer for sorting (in KB)

(4) Loading data to a table with LOB columns (loading data to LOB RDAREA and LOB columns at the same time)

Example 11

For a table with LOB columns (TABLE2), this example loads data to both the LOB column structure base table and LOB columns at the same time.

The example assumes that the following table and index have been defined:

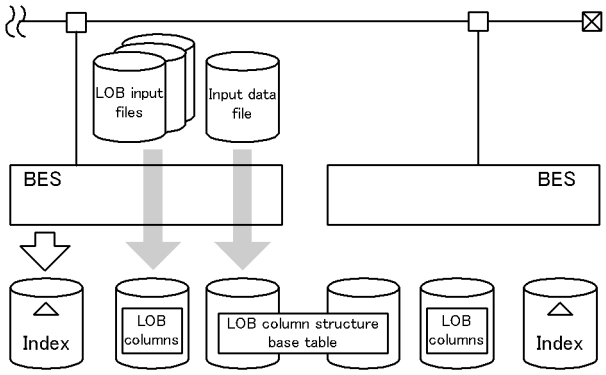
- Table definition:

```
CREATE TABLE TABLE2(C1 INT,C2 BLOB
             IN ((LOBUSER01),(LOBUSER02)))
             IN ((PDBUSER01) C1 > 10,(PDBUSER02))
```

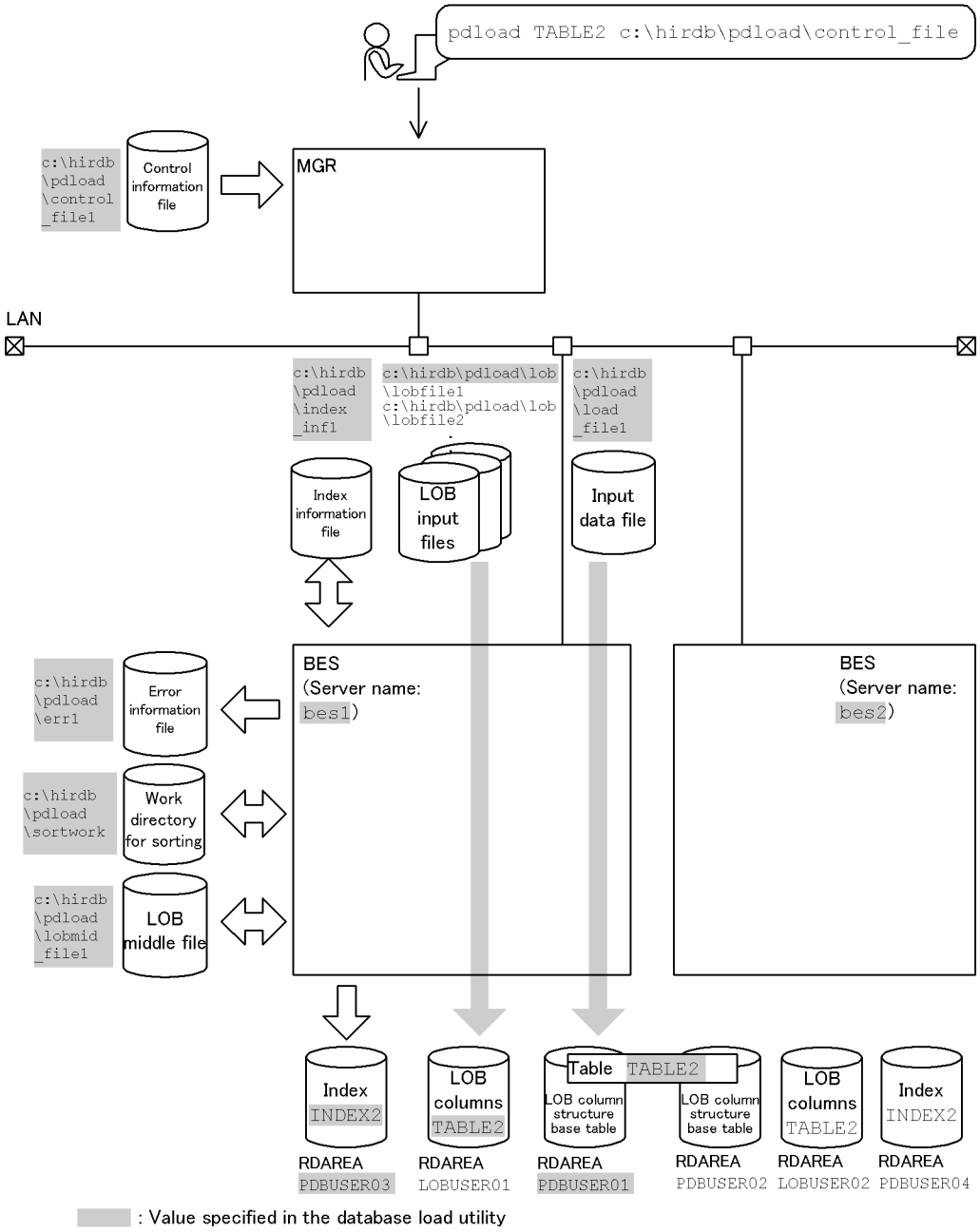
- Index definition (partitioning key index):

```
CREATE INDEX INDEX2 ON TABLE2(C1)
             IN((PDBUSER03),(PDBUSER04))
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to a table with LOB columns (TABLE2) in units of RDAREAs (PDBUSER01, LOBUSER01).

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source PDBUSER01 bes1:c:\hirdb\pdload\load_file1
       error=c:\hirdb\pdload\err1                          1
index INDEX2 PDBUSER03 c:\hirdb\pdload\index_inf1          2
sort bes1 c:\hirdb\pdload\sortwork,8192                    3
lobdata c:\hirdb\pdload\lob                                4
lobmid c:\hirdb\pdload\lobmid_file1                        5
```

Explanation:

1. Specifies the input data file and the error information file to which error information is to be output in the event of an error in the input data:

   PDBUSER01: Name of the RDAREA that contains the table subject to data loading

   bes1: Name of the server containing the input data file

   c:\hirdb\pdload\load_file1: Name of the input data file

   Contents of the input data file (DAT format):

   ```
   1,lobfile1
   2,lobfile2
   3,lobfile3
   ```

   c:\hirdb\pdload\err1: Name of the error information file

2. Specifies the index information file to which index information is to be output.

   INDEX2: Index identifier

   PDBUSER03: Name of the index storage RDAREA

   c:\hirdb\pdload\index_inf1: Name of the index information file

3. Specifies the work directory for sorting:

   bes1: Name of the server used to create the sort work file

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

`8192`: Size of buffer for sorting (in KB)

4. Specifies the directory containing the LOB input files:

   `c:\hirdb\pdload\lob`: pathname of the LOB Input files

5. Specifies the file to which information required for loading LOB column data is to be output when loading data to the LOB column structure base table (this file is used as is when loading data to the LOB columns):

   `c:\hirdb\pdload\lobmid_file1`: Name of the LOB middle file

(5) Loading data to a table with LOB columns (using a LOB column input file)

Example 12

For a table with LOB columns (`TABLE2`), this example loads data to both the LOB column structure base table and LOB columns at the same time using a LOB column input file to store data to the LOB columns.

The example assumes that the following table and index have been defined:

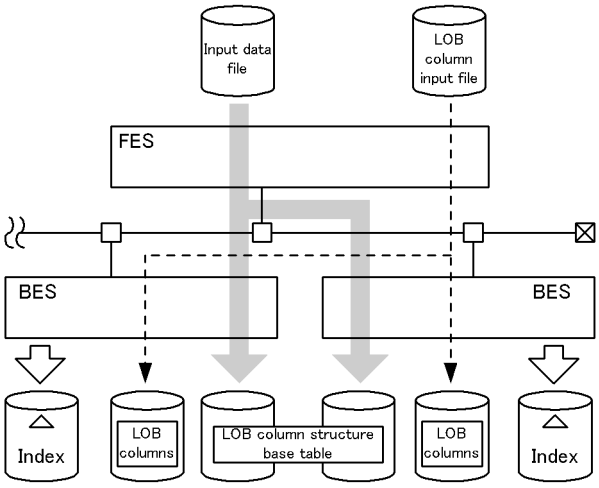
- Table definition:

```
CREATE TABLE TABLE2(C1 INT,C2 BLOB
             IN ((LOBUSER01),(LOBUSER02)))
             IN ((PDBUSER01) C1 > 10,(PDBUSER02))
```

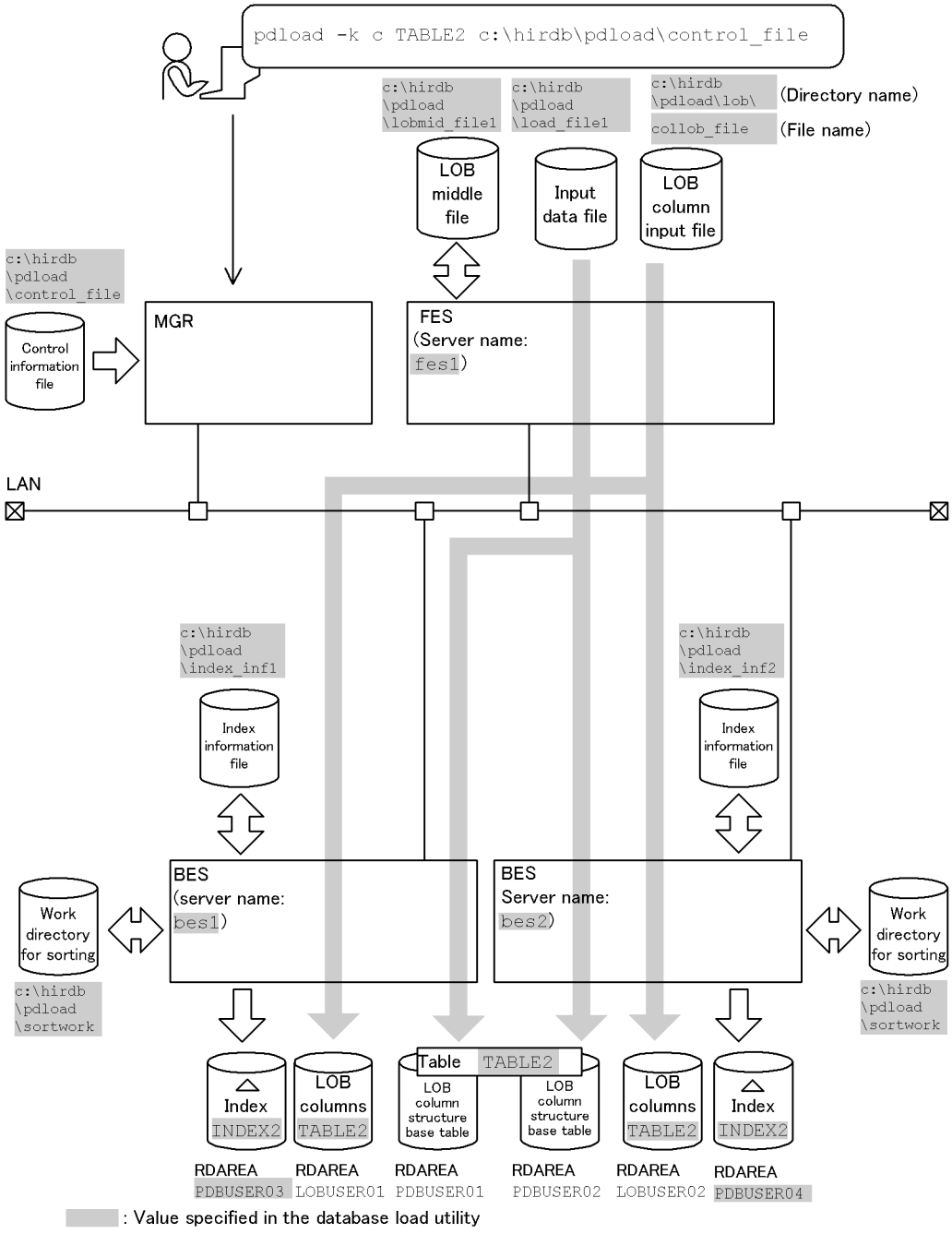
- Index definition (partitioning key index):

```
CREATE INDEX INDEX2 ON TABLE2(C1)
IN((PDBUSER03),(PDBUSER04))
```

Overview



Relationship between input/output files and RDAREAs



Explanation of the command

The example loads data to a table with LOB columns (TABLE2).

-k c: Specification for using a LOB column input file

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source fes1:c:\hirdb\pdload\load_file1              1
index INDEX2 PDUSER03 c:\hirdb\pdload\index_inf1    2
index INDEX2 PDUSER04 c:\hirdb\pdload\index_inf2    2
sort bes1 c:\hirdb\pdload\sortwork,8192             3
sort bes2 c:\hirdb\pdload\sortwork,8192             3
lobdata c:\hirdb\pdload\collob                      4
lobcolumn collob_file                               5
lobmid c:\hirdb\pdload\lobmid_file                  6
```

Explanation:

1. Specifies the input data file:

   fes1: Name of the server containing the input data file

   c:\hirdb\pdload\load_file1: Name of the input data file

2. Specifies the index information files to which index information is to be output:

   INDEX2: Index identifier

   PDBUSER03, PDBUSER04: Names of the index storage RDAREAs

   c:\hirdb\pdload\index_inf1, c:\hirdb\pdload\index_inf2: Names of the index information files

3. Specifies the work directory for sorting:

   bes1, bes2: Names of the servers used to create the sort work file

   c:\hirdb\pdload\sortwork: Name of the directory in which the sort work file is created

   8192: Size of buffer for sorting (in KB)

4. Specifies the directory containing the LOB column input file:

   c:\hirdb\pdload\collob: Name of the directory containing the LOB column input file

5. Specifies the LOB column input file:

   collob_file: Name of the LOB column input file

6. Specifies the LOB middle file:

   `c:\hirdb\pdload\lobmid_file`: Name of the LOB middle file

(6) Loading data to a table with repetition columns

Example 13

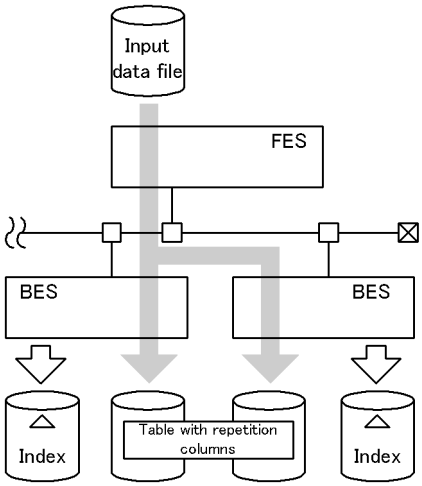
This example loads data to a table with repetition columns (SCORE_TABLE) without checking the order of cluster key values.

The example assumes that the following table has been defined:

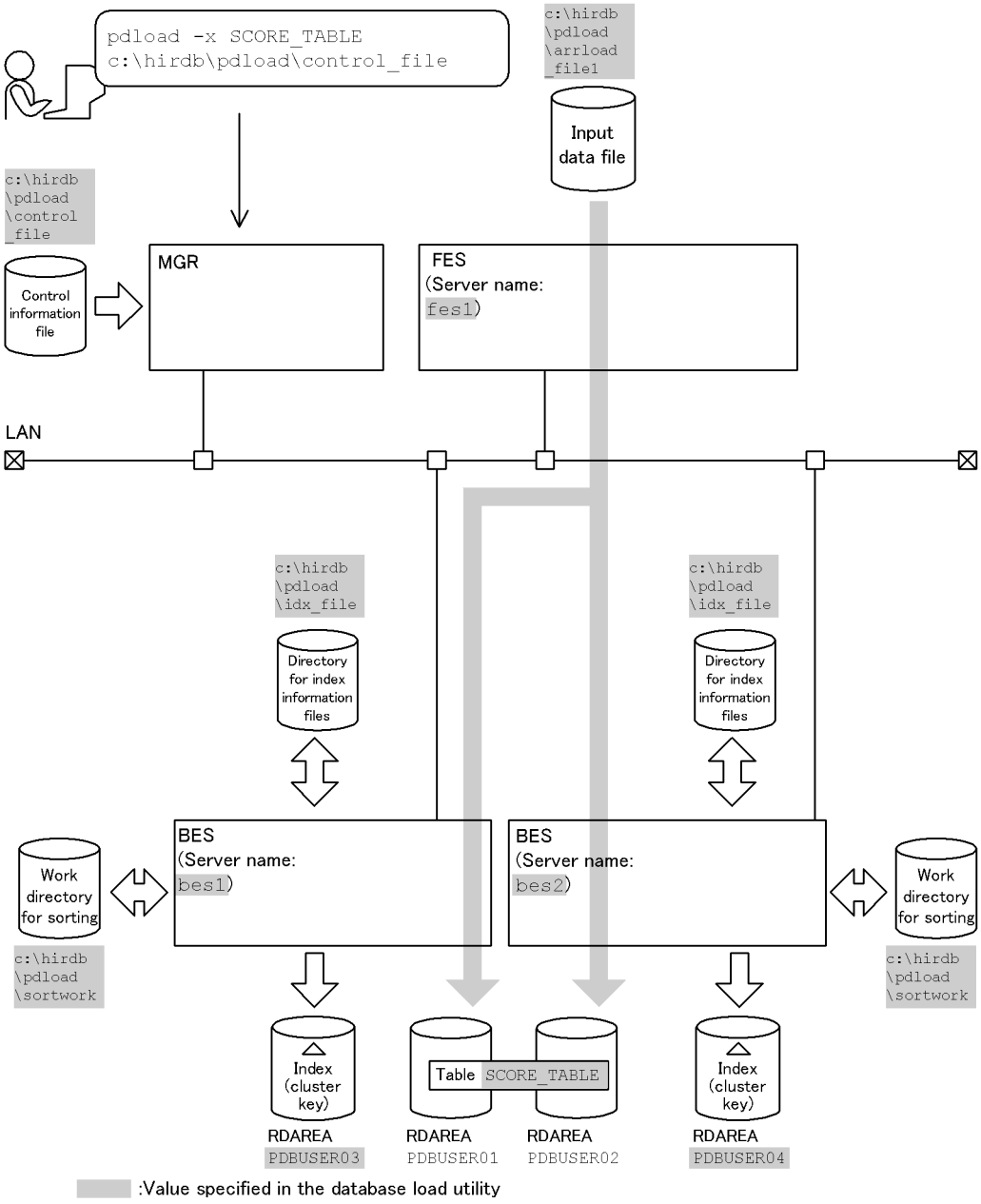
- Table definition:

```
CREATE TABLE SCORE_TABLE
    (NUMBER CHAR(6) UNIQUE CLUSTER KEY IN
     ((PDUSER03),(PDUSER04))
     NAME NCHAR(10),
     SUBJECT_NAME NCHAR(8) ARRAY[5],
     SCORE INT ARRAY[5])
  IN ((PDUSER01) < =  1Z9999,(PDUSER02))
```

Overview



Relationship between input/output files and RDAREAs



Contents of the input data file (DAT format)

```
1A0001, ANDERSON, JAPANESE, MATHEMATICS, ENGLISH, SOCIOLOGY, SCIENCE, 90, 85, 75, 100, 80
1A002,  HART,     JAPANESE, MATHEMATICS, ENGLISH, SOCIOLOGY, SCIENCE, 55, 80, 45,  70, 60
                                  :

NUMBER   NAME                     SUBJECT_NAME                         SCORE
```

Explanation of the command

The example loads data to a table with repetition columns (`SCORE_TABLE`) in units of tables.

`-x`: Specification for not checking the order of cluster key values

`c:\hirdb\pdload\control_file`: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source fes1:c:\hirdb\pdload\arrload_file1      1
idxwork bes1 c:\hirdb\pdload\idx_file          2
idxwork bes2 c:\hirdb\pdload\idx_file          2
sort bes1 c:\hirdb\pdload\sortwork,8192        3
sort bes2 c:\hirdb\pdload\sortwork,8192        3
array elmtype=ff nullset=c                     4
```

Explanation:

1. Specifies the input data file:

   `fes1`: Name of the server containing the input data file

   `c:\hirdb\pdload\arrload_file1`: Name of the input data file

2. Specifies the directory where index information files are to be created:

   `bes1`, `bes2`: Names of the servers used to create index information files

   `c:\hirdb\pdload\idx_file`: Name of the directory in which index information files are created

3. Specifies the work directory for sorting:

   `bes1`, `bes2`: Names of the servers used to create the sort work files

   `c:\hirdb\pdload\sortwork`: Name of the directory in which the sort work file is created

   `8192`: Size of buffer for sorting (in KB)

4. Specifies that data is to be loaded to a table with repetition columns:

ff: Arrayed data format

c: Specification for not storing any data for the corresponding element

(7) Loading data to a table using a plug-in

Example 14

This example loads data to a table (TBL01) with an abstract data type (SGMLTEXT) using the HiRDB Text Search Plug-in.

The example assumes that the following table and index have been defined:

- Table definition:

```
CREATE TABLE TBL01(C1 INTEGER,C2 VARCHAR(10),C3 SGMLTEXT
ALLOCATE
                        (SGMLTEXT IN LOBUSER01) PLUGIN ...)
                        IN PDBUSER01
```

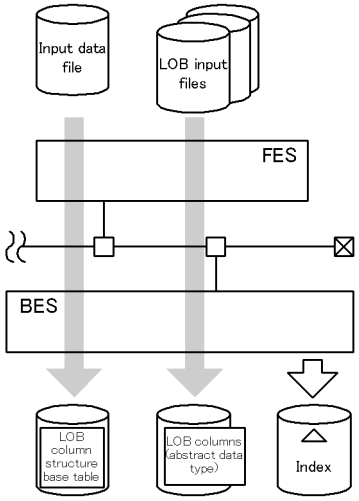
- Plug-in index definition:

```
CREATE INDEX INDEX1 USING TYPE NGRAM ON TBL01(C3) IN
PDBUSER02
```

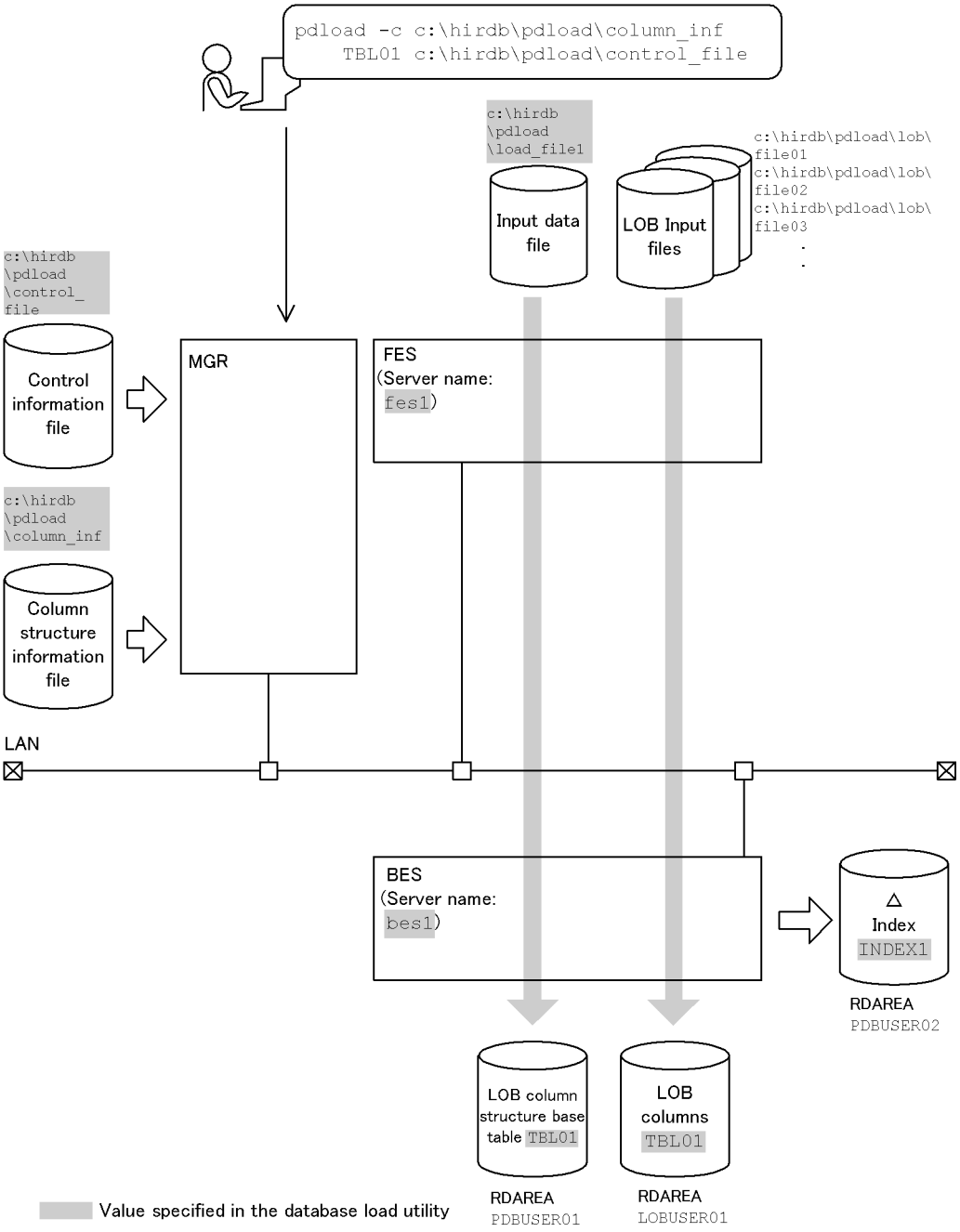
Following is the information about the constructor function of SGMLTEXT type:

- Name of the function: SGMLTEXT, input: BLOB, output: the SGMLTEXT type

Overview



Relationship between input/output files and RDAREAs



Contents of the input data file (DAT format)

```
1,"HiRDB",c:\hirdb\pdload\lob\file01
-1,"",c:\hirdb\pdload\lob\file02
999,"HITACHI",c:\hirdb\pdload\lob\file03
       :
```

Explanation of the command

The example loads data to a table with an abstract data type (TBL01).

c:\hirdb\pdload\column_inf: Name of the column structure information file

c:\hirdb\pdload\control_file: Name of the control information file

Contents of the column structure information file (c:\hirdb\pdload\column_inf)

```
C1                                    1
C2                                    1
C3,func=(SGMLTEXT,param=blob)         1,2
```

Explanation:

1. Stores the input data in TBL01's columns (C1, C2, and C3).
2. Calls the constructor function and stores data in the column of abstract data type (C3):

   SGMLTEXT: Name of the constructor function

   blob: Type of the argument that is passed to the constructor function

Contents of the control information file (c:\hirdb\pdload\control_file)

```
source fes1:c:\hirdb\pdload\load_file1    1
lobdata                                   2
```

Explanation:

1. Specifies the input data file:

   fes1: Name of the server containing the input data file

   c:\hirdb\pdload\load_file1: Name of the input data file
2. Specifies the LOB input files.

Example 15

This example loads data to a table (TBL02) with an abstract data type (GEOMETRY).

The example assumes that the following table has been defined:

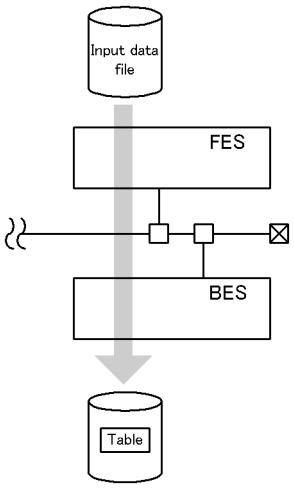
- Table definition:

```
CREATE TABLE TBL02(C1 INTEGER,C2 DECIMAL(7,0),C3 GEOMETRY)
                       IN PDBUSER02
```

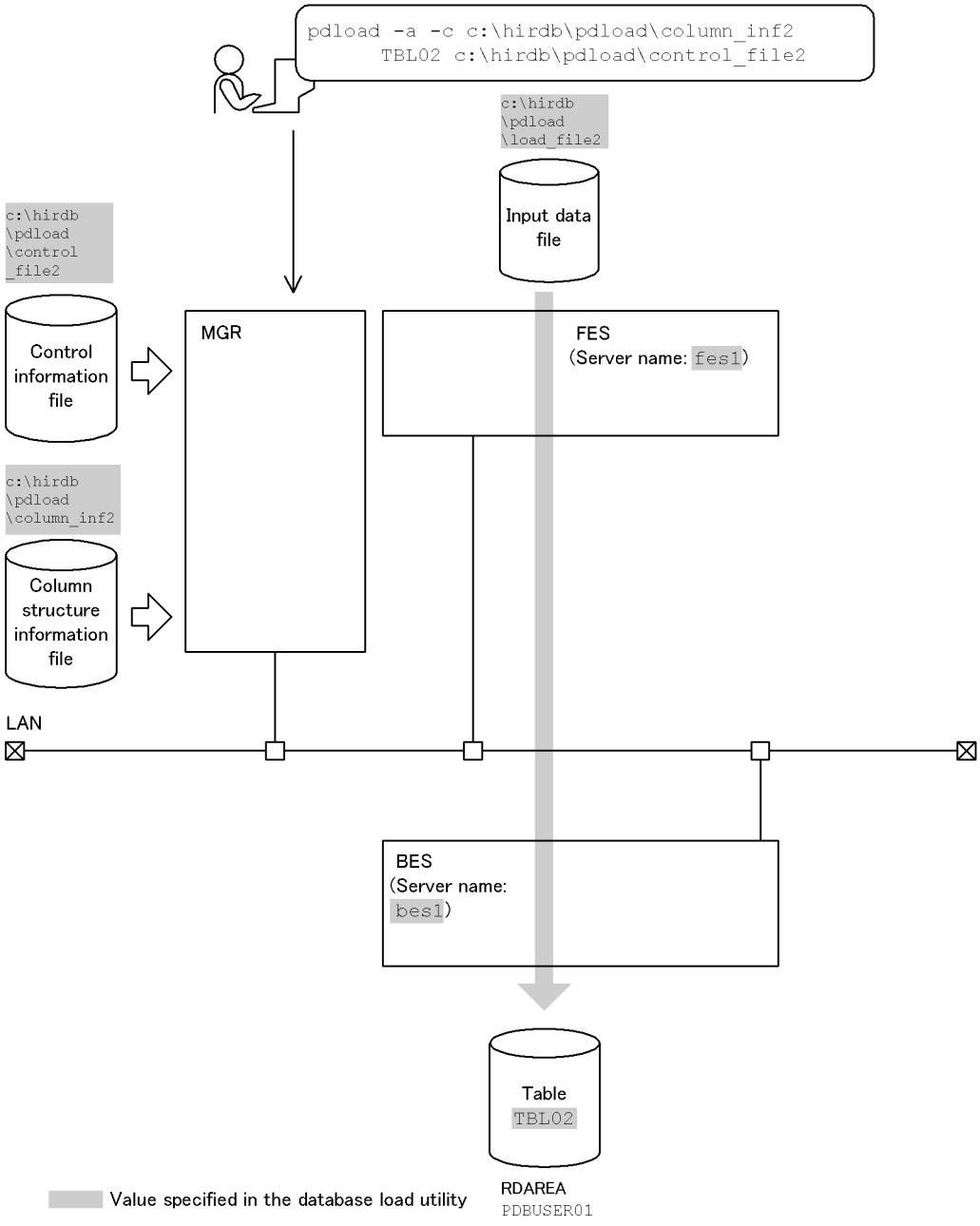
Following is the information about the constructor function of the GEOMETRY type:

- Name of the function: GEOMETRY, input: VARCHAR, output: GEOMETRY type

Overview



Relationship between input/output files and RDAREAs



Contents of the input data file (fixed-size data format)

```
                                                  Description
000000010000001c000affffffffffffffffff     1, 1, 10-byte data
ffffffff0000001d0001ff000000000000000000     -1, -1, 1-byte data
000003e70000999c0005fffffffff0000000000     999, 999, 5-byte data

<-int -><-dec -><-------varchar(10)------>
```

Explanation of the command

The example loads data to a table with an abstract data type (`TBL02`).

`c:\hirdb\pdload\column_inf2`: Name of the column structure information file

`c:\hirdb\pdload\control_file2`: Name of the control information file

Contents of the column structure information file (c:\hirdb\pdload\column_inf2)

```
C1,type=integer                                          1
C2,type=dec(7,0)                                         2
C3,func=(GEOMETRY,param=varchar,type=varchar(10))        3
```

Explanation:

1. Stores `INTEGER` data in column `C1`.
2. Stores `DECIMAL` (7,0) data in column `C2`.
3. Stores the value that was generated by passing data `varchar (10)` to the `GEOMETRY` constructor function, the input parameter of which is `varchar` in the abstract data type column `C3`.

   `GEOMETRY`: Name of the constructor function

   `varchar`: Type of argument that is passed to the constructor function

Contents of the control information file (c:\hirdb\pdload\control_file2)

```
source fes1:c:\hirdb\pdload\load_file2          1
```

Explanation:

1. Specifies the input data file:

   `fes1`: Name of the server containing the input data file

   `c:\hirdb\pdload\load_file2`: Name of the input data file

Example 16

This example loads data to a table (TBL03) with an abstract data type (SGMLTEXT) and LOB columns.

The example assumes that the following table and index have been defined:

- Table definition:

```
CREATE TABLE TBL03(C1 INTEGER,C2 BLOB(4096) IN LOBUSER01),
                    C3 SGMLTEXT ALLOCATE
                      (SGMLTEXT IN LOBUSER02) PLUGIN ...)
                      IN PDBUSER03
```

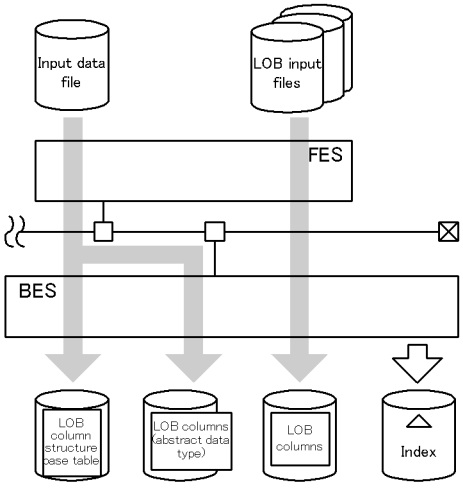
- Plug-in index definition:

```
CREATE INDEX INDEX1 USING TYPE NGRAM ON TBL03(C3) IN
PDBUSER04
```

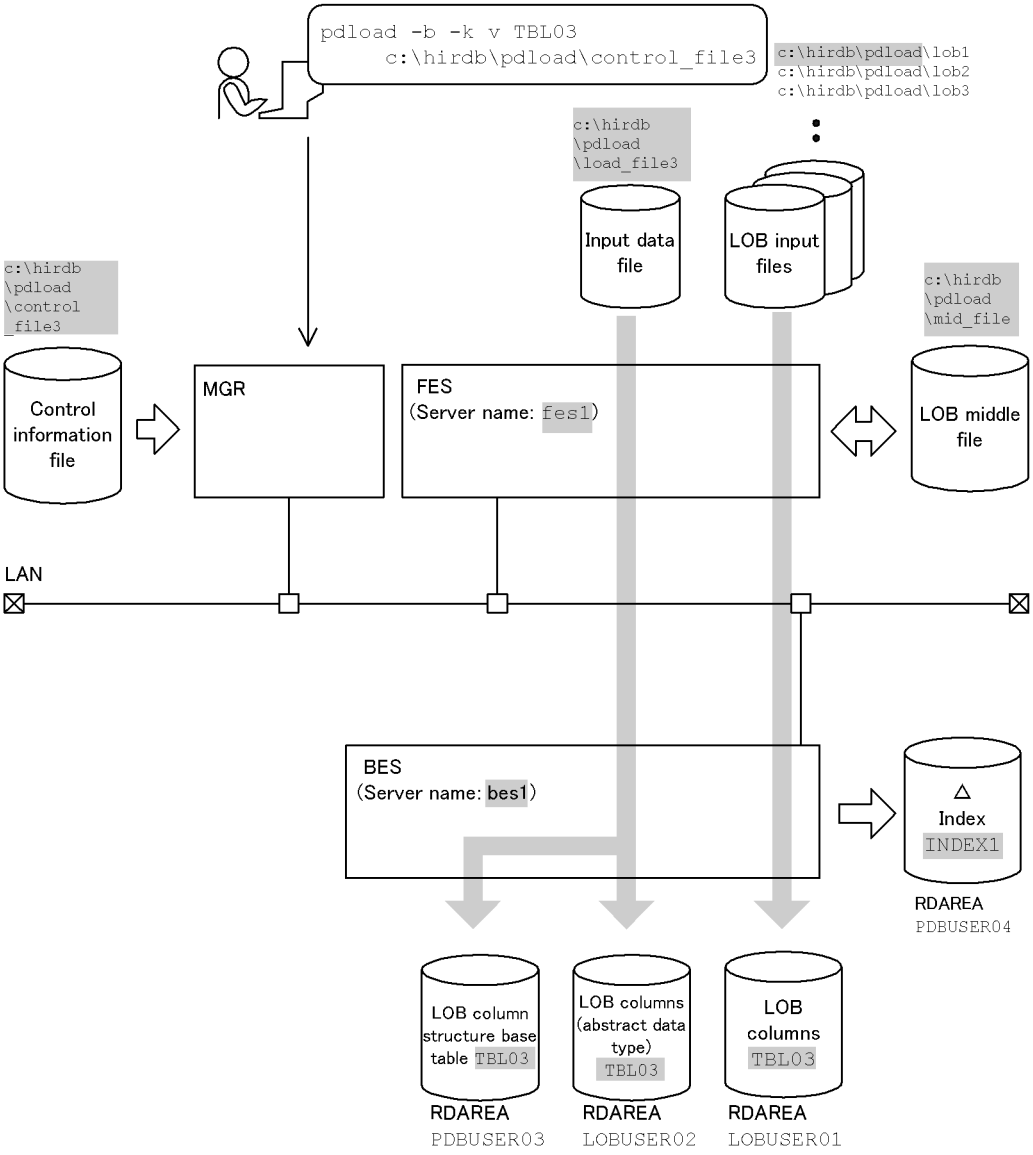
Following is the information about the construction function of SGMLTEXT type:

- Name of the function: SGMLTEXT, input: BLOB, output: SGMLTEXT type

Overview



Relationship between input/output files and RDAREAs



: Value specified in the database load utility

Contents of the input data file (binary format)

```
                                                    Description
00000001000000046c6f62310000000affffffffffffffffffff 1, lob1, 10-byte data
ffffffff000000046c6f623200000001ff                  -1, lob2, 1-byte data
000003e7000000046c6f6233000000005fffffffff          999, lob3, 5-byte data
```

Explanation of the command

The example loads data to a table with an abstract data type (`TBL03`).

`-k v`: Specifies that the input data file is to contain the LOB parameters and the names of the LOB input files for the data to be stored in the LOB columns.

`c:\hirdb\pdload\control_file3`: Name of the control information file

Contents of the control information file (c:\hirdb\pdload\control_file3)

```
source fes1:c:\hirdb\pdload\load_file3       1
lobmid c:\hirdb\pdload\mid_file              2
lobdata c:\hirdb\pdload                      3
```

Explanation:

1. Specifies the input data file:

   `fes1`: Name of the server containing the input data file

   `c:\hirdb\pdload\load_file3`: Name of the input data file

2. Specifies the LOB middle file:

   `c:\hirdb\pdload\mid_file`: Name of the LOB middle file

3. Specifies the LOB input files:

   `c:\hirdb\pdload`: Name of the directory that contains LOB input files

Example 17

This example loads data to a table (TBL04) with an abstract data type (SGMLTEXT) and LOB columns. First, it loads data to the LOB column structure base table and abstract data type, then loads data to the LOB columns concurrently at each back-end server.

The example assumes that the following table and index have been defined:

- Table definition:

```
CREATE TABLE TBL04(C1 INTEGER,C2 BLOB(100) IN
                   (LOB11),(LOB12)),
                   C3 SGMLTEXT ALLOCATE
                      (SGMLTEXT IN ((LOB21),(LOB22))
                       PLUGIN...)
                       IN ((PDBUSER01) C1 >
                       1000,(PDBUSER02))
```

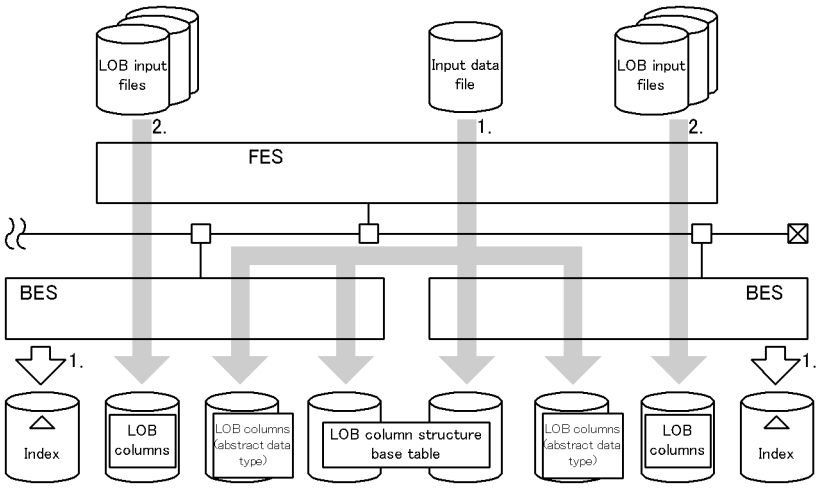
- Plug-in index definition:

```
CREATE INDEX INDEX1 USING TYPE NGRAM ON TBL04(C3) IN
((PDBUSER03),(PDBUSER04))
```

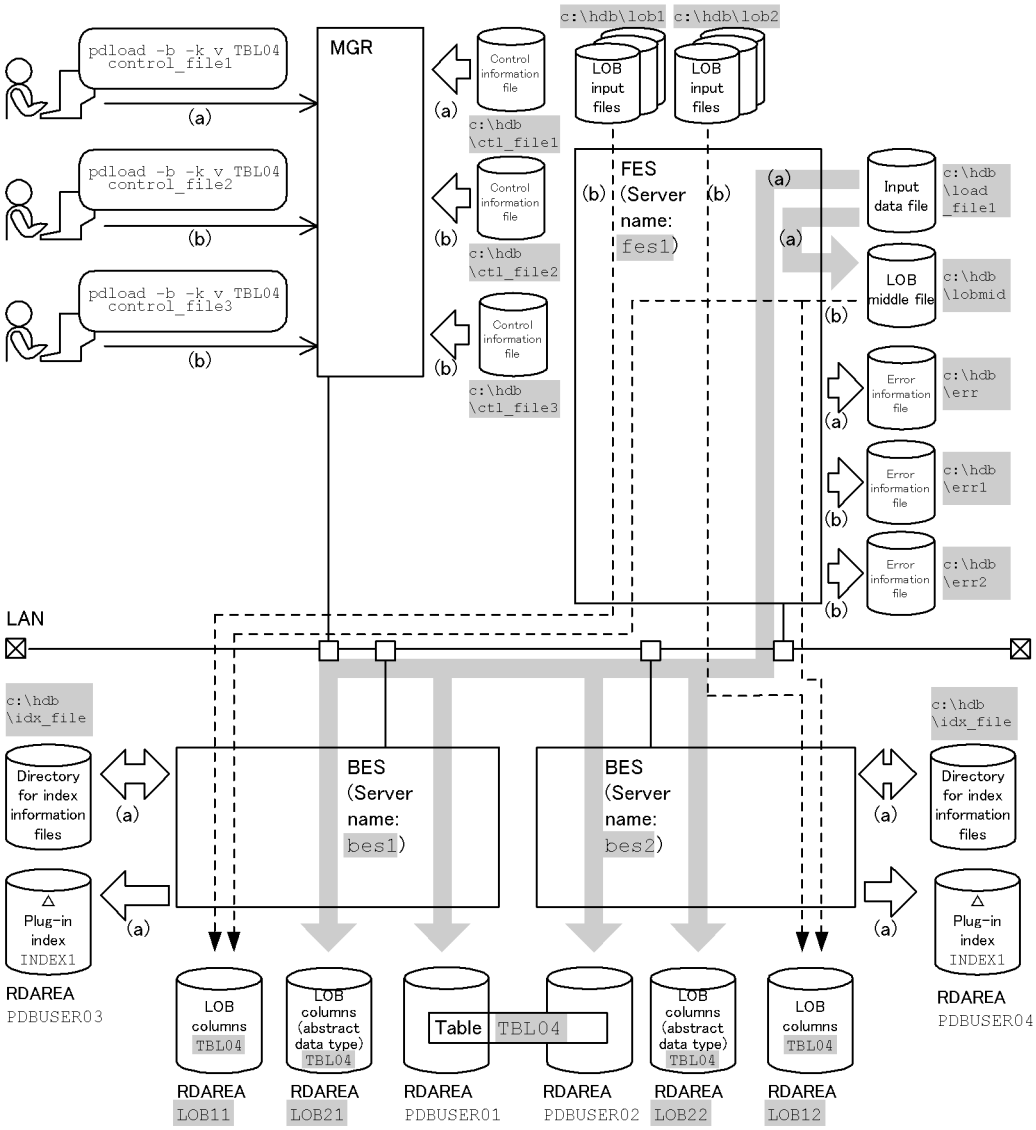
Following is the information about the constructor function of the SGMLTEXT type:

- Name of the function: SGMLTEXT, input: BLOB, output: SGMLTEXT type

Overview



Relationship between input/output files and RDAREAs



: Value specified in the database load utility

Note: (a) and (b) indicate the flow of data when commands (a) and (b), respectively, are executed.

(a) Loading data to the LOB column structure base table and abstract data type

Explanation of the command

The example loads data to the LOB column structure base table and abstract data type of a table with an abstract data type (`TBL04`).

`-b`: Specification for the input data file in the binary format

`-k v`: Specifies that the input data file is to contain the LOB parameters and the names of the LOB input files for the data to be stored in the LOB columns.

`c:\hdb\ctl_file1`: Name of the control information file

Contents of the control information file (c:\hdb\ctl_file1)

```
source fes1:c:\hdb\load_file1 err=c:\hdb\err      1
idxwork bes1 c:\hdb\idx_file                      2
idxwork bes2 c:\hdb\idx_file                      2
lobmid c:\hdb\lobmid                              3
```

Explanation:

1. Specifies the input data file:

   `fes1`: Name of the server containing the input data file

   `c:\hdb\load_file1`: Name of the input data file

   `c:\hdb\err`: Name of the error information file

2. Specifies the directory in which index information files are to be created:

   `bes1`, `bes2`: Names of the servers used to create index information files

   `c:\hdb\idxfile`: Name of the directory in which index information files are created

3. Specifies the LOB middle files:

   `c:\hdb\lobmid`: Name of the LOB middle file

(b) Loading data to LOB columns

Explanation of the command

The example loads data to the LOB columns of the table with abstract data type (`TBL04`) concurrently at each back-end server.

`-b`: Specification for the input data file in the binary format

`-k v`: Specifies that LOB input files are to be provided for each LOB data item for the LOB columns and that an input data file is to be provided for the input

parameters of the constructor function that generates values for the columns of the abstract data type.

`c:\hdb\ctl_file2`, `c:\hdb\ctl_file3`: Names of the control information files

Contents of the control information file (c:\hdb\ctl_file2)

```
lobdata c:\hdb\lob1                                    1
lobmid LOB11 fes1:c:\hdb\lobmid error=c:\hdb\err1      2
```

Explanation:

1. Specifies the input data file:

   `c:\hdb\lob1`: Name of the directory containing the LOB input files

2. Specifies the LOB middle file:

   `LOB11`: Name of the user LOB RDAREA containing the LOB columns

   `fes1`: Name of the server containing the LOB middle file

   `c:\hdb\lobmid`: Name of the LOB middle file

   `c:\hdb\err1`: Name of the error information file

Contents of the control information file (c:\hdb\ctl_file3)

```
lobdata c:\hdb\lob2                                    1
lobmid LOB12 fes1:c:\hdb\lobmid error=c:\hdb\err2      2
```

Explanation:

1. Specifies the LOB input files:

   `c:\hdb\lob2`: Name of the directory containing the LOB input files

2. Specifies the LOB middle file:

   `LOB12`: Name of the user LOB RDAREA containing the LOB columns

   `fes1`: Name of the server containing the LOB middle file

   `c:\hdb\lobmid`: Name of the LOB middle file

   `c:\hdb\err2`: Name of the error information file

Chapter

6. Interactive SQL Execution Utility

This chapter explains the interactive SQL execution utility that can interactively execute SQL statements on a HiRDB database. Execution of the interactive SQL execution utility requires the HiRDB SQL Executor, which is a separate program product.

This chapter contains the following section:

6.1 Overview

Executor

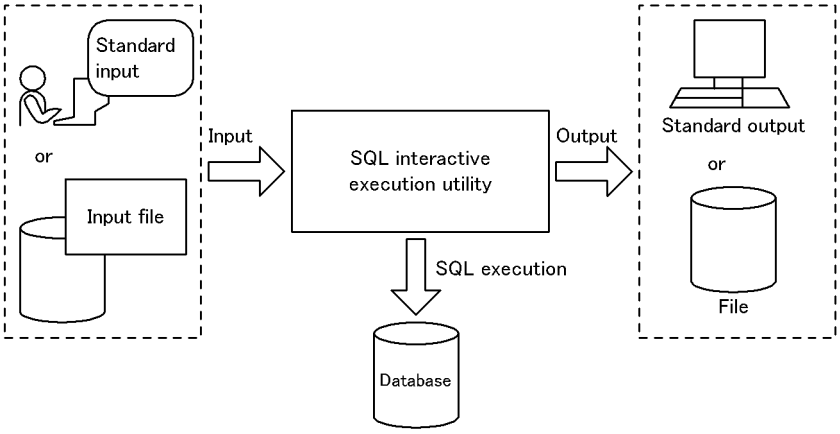
Users who have CONNECT privileges can execute the interactive SQL execution utility. However, the required privileges may vary with the type of SQL to be executed. For details about required SQL execution privileges, see the manual *HiRDB Version 8 SQL Reference*.

6.1.1 Function

The interactive SQL execution utility is used to execute SQLs interactively on a HiRDB database. This utility simplifies entry and execution of SQL statements; it can also be used to specify commands special to the utility, such as for formatting the execution results. The separate HiRDB SQL Executor program product is required in order to execute this utility.

SQLs input from the standard input or an input file are executed, and the execution results are output to the standard output or to a file. Figure 6-1 provides an overview of the interactive SQL execution utility.

Figure 6-1: Overview of interactive SQL execution utility



To execute the interactive SQL execution utility, from **Start**, choose **Program**, then **HiRDB SQL Executer**.

For details about the interactive SQL execution utility, see *README* in the HiRDB SQL Executer.

Chapter

7. Database Structure Modification Utility (pdmod)

This chapter explains the database structure modification utility (pdmod) that can modify the physical structure of a database, including adding an RDAREA, expanding the database, or reinitializing the database.

This chapter contains the following sections:

7.1 Functions of the database structure modification utility

Executor: HiRDB administrator

The database structure modification utility (pdmod) is used to modify the physical structure of a database, such as by adding, expanding, or reinitializing RDAREAs. This utility resolves RDAREA space shortages that may be caused by frequent additions to and updating of tables. Table 7-1 shows the relationships between the functions of the database structure modification utility (pdmod) and the types of RDAREAs.

Table 7-1: Relationships between functions and RDAREA types

RDAREA type / Function	MDir	DDir	DDic	DDic LOB	Usr	Usr LOB	List	Reg	Reg LOB
RDAREA addition	NE	NE	Y	Y[1]	Y	Y	Y	NE	NE
RDAREA expansion	Y	Y	Y	Y	Y	Y	Y	Y	Y
RDAREA reinitialization	NE	NE	NE	Y[2]	Y	Y	Y	Y	Y
RDAREA deletion	NE	NE	NE[3]	NE	Y	Y	Y	NE	NE
Moving RDAREA	NE	NE	NE	NE	Y	Y	NE	NE	NE
Dictionary table attribute definition modification	—	—	—	—	—	—	—	—	—
RDAREA attribute modification	NE	NE	NE	NE	Y	Y	Y	NE	NE
RDAREA change from HiRDB/Single Server structure to HiRDB/ Parallel Server structure.	Y	Y	Y	Y	Y	Y	Y	Y	Y
Auditor registration	—	—	—	—	—	—	—	—	—
Audit trail table creation	—	—	—	—	Y	—	—	—	—

Legend:

MDir: Master directory RDAREA

DDir: Data directory RDAREA

DDic: Data dictionary RDAREA

DDic LOB: Data dictionary LOB RDAREA

Usr: User RDAREA

Usr LOB: User LOB RDAREA

List: List RDAREA

Reg: Registry RDAREA

Reg LOB: Registry LOB RDAREA

Y: Can be executed.

NE: Cannot be executed.

—: Not applicable.

[1] In the case of RDAREA addition, two RDAREAs must be added.

[2] Can be executed only for an RDAREA in which stored procedure of stored function objects are stored.

[3] Can be executed for a data dictionary RDAREA in which a database state analyzed table and database management table are stored.

7.2 Command format

(1) Format

```
pdmod -a control-statement-filename
```

(2) Options

- `-a` *control-statement-filename* ～ <pathname> ((up to 255 characters))

  For a HiRDB/Single Server

  Specify the file containing control statements for the database structure modification utility (`pdmod`) in an absolute path name.

  For a HiRDB/Parallel Server

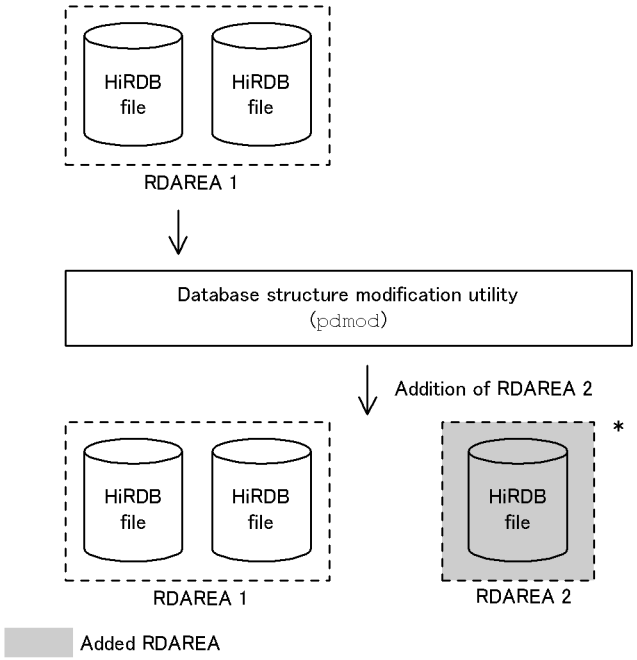
  Specify the file containing control statements for the database structure modification utility (`pdmod`) in an absolute path name. The control statement file should be created for the server machine where the dictionary server is located.

7.3 Adding an RDAREA

7.3.1 Function

RDAREA addition is a function that adds a new RDAREA when there are not enough existing RDAREAs for adding new tables and indexes and modifying table definitions. Figure 7-1 provides an overview of the use of the database structure modification utility (pdmod) for RDAREA addition.

Figure 7-1: Overview of database structure modification utility (pdmod): RDAREA addition



Prerequisites

1. Before adding an RDAREA, you need to initialize the HiRDB file system areas in which the HiRDB files that constitute the new RDAREA are to be created (by executing the pdfmkfs command).
2. When adding an RDAREA, make sure that the total number of RDAREAs does not exceed the maximum number of RDAREAs (pd_max_rdarea_no) or the maximum number of HiRDB files

constituting the RDAREAs (pd_max_file_no) specified in the system common definition. If either of these values is exceeded, you must use the pdchgconf command to change the system common definition or terminate the HiRDB system normally and then change the system common definition.

3. A shared RDAREA that is added is regarded as having been added to every back-end server (even if a back-end server has no shared RDAREAs, the system assumes that one RDAREA has been added). If the maximum permissible number of RDAREAs or the maximum permissible number of HiRDB files constituting RDAREAs is exceeded at any of the back-end servers, the shared RDAREA cannot be added at all.
4. Addition of shared RDAREAs is applicable only to user RDAREAs. A shared RDAREA can be added only to a HiRDB file system area whose usage purpose is SDB.

7.3.2 Examples

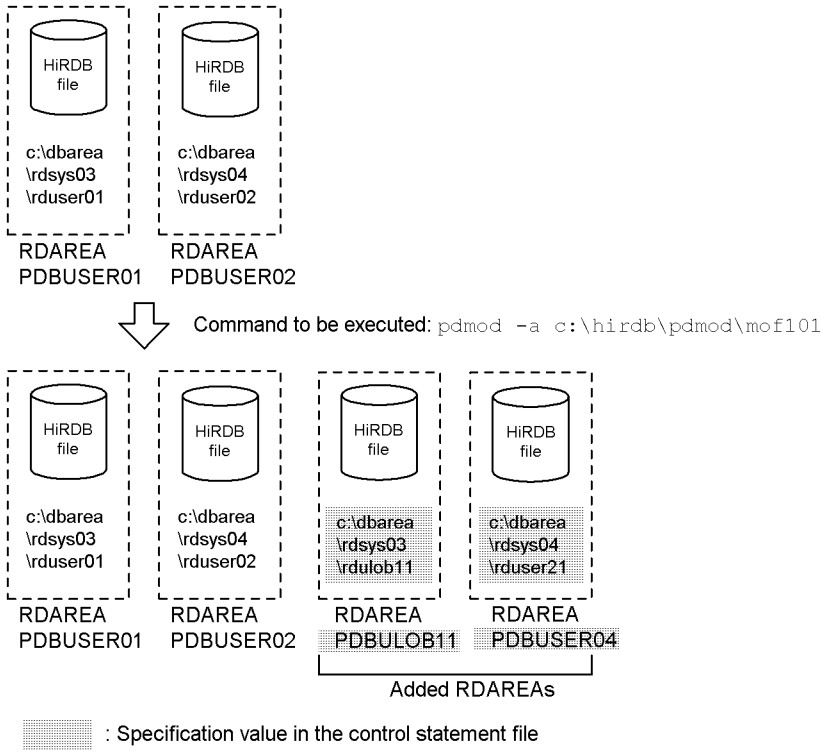
This section presents examples of using the database structure modification utility (adding an RDAREA).

(1) Adding RDAREAs (1)

This example adds the following RDAREAs:

- PDBULOB11 (user LOB RDAREA)
- PDBUSER04 (user RDAREA)

Overview



(a) HiRDB/Single Server

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
/* Adding a LOB RDAREA */
create rdarea PDBULOB11   ..................................................1
         globalbuffer gb01   ...............................................2
         for LOB used by PUBLIC   ..........................................3
         page 8192 characters   ............................................4
         storage control segment 1 pages   .................................5
         file name "c:\dbarea\rdsys03\rdulob11"   ..........................6
           initial 50 segments ;   .........................................7

/* Adding a user RDAREA */
create rdarea PDBUSER04   ..................................................1
         globalbuffer gb02   ...............................................2
         for user used by "userid"   .......................................3
         page 4096 characters   ............................................4
         storage control segment 50 pages   ................................5
         file name "c:\dbarea\rdsys04\rduser21"   ..........................6
           initial 100 segments ;   ........................................7
```

Explanation

1. Specifies the name of the RDAREA to be added.
2. Specifies the name of the global buffer to which the RDAREA is to be allocated.
3. Specifies the type of RDAREA.
4. Specifies the page length.
5. Specifies the segment size.
6. Specifies the name of the HiRDB file that constitutes the RDAREA.
7. Specifies the number of HiRDB file segments.

(b) HiRDB/Parallel Server

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
/* Adding a user LOB RDAREA */
create rdarea PDBULOB11   ..................................................1
         globalbuffer gb01   ...............................................2
         for LOB used by PUBLIC   ..........................................3
         server name bes1   ................................................4
         page 8192 characters   ............................................5
         storage control segment 1 pages   .................................6
         file name "c:\dbarea\rdsys03\rdulob11"   ..........................7
           initial 50 segments ;   .........................................8

/* Adding a user RDAREA */
create rdarea PDBUSER04   ..................................................1
         globalbuffer gb02   ...............................................2
         for user used by "userid"   .......................................3
         server name bes2   ................................................4
         page 4096 characters   ............................................5
         storage control segment 50 pages   ................................6
         file name "c:\dbarea\rdsys04\rduser21"   ..........................7
           initial 100 segments ;   ........................................8
```

Explanation

1. Specifies the name of the RDAREA to be added.
2. Specifies the name of the global buffer to which the RDAREA is to be allocated.
3. Specifies the type of RDAREA.
4. Specifies the page length.
5. Specifies the name of the server that manages the RDAREA.
6. Specifies the segment size.

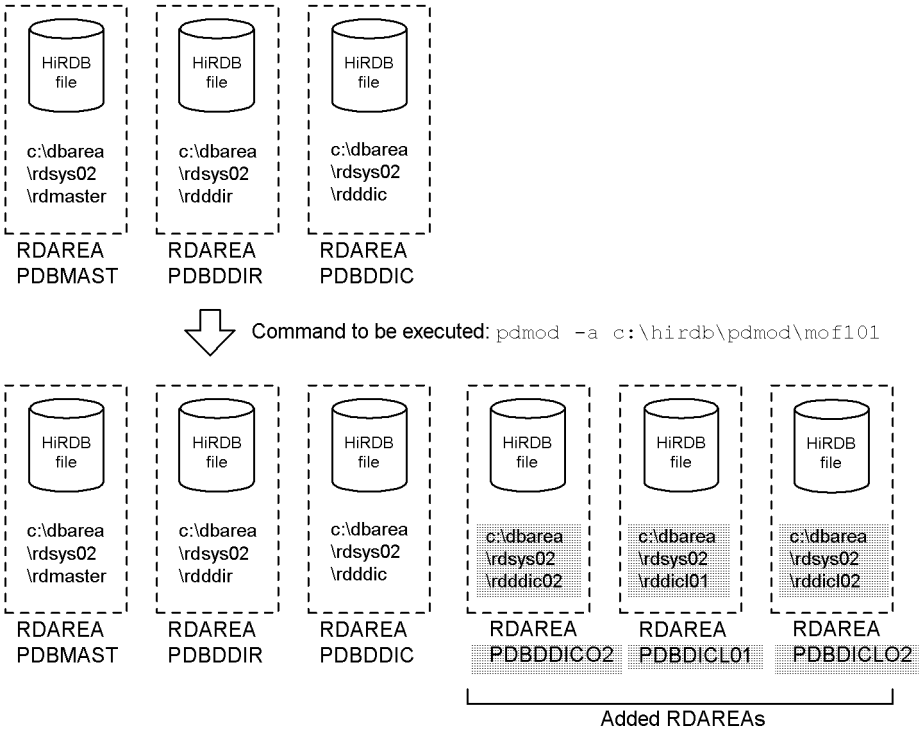
7. Specifies the name of the HiRDB file that constitutes the RDAREA.
8. Specifies the number of HiRDB file segments.

(2) Adding RDAREAs (2)

This example adds the following RDAREAs:

- `PDBDDIC02` (data dictionary RDAREA)
- `PDBDICL1` (data dictionary LOB RDAREA)
- `PDBDICL2` (data dictionary LOB RDAREA)

Overview



(a) HiRDB/Single Server

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
/* Adding a data dictionary RDAREA */
create rdarea PDBDDIC02   ..................................................1
         globalbuffer gbp01   ..............................................2
         for datadictionary of routines   ..................................3
         page 4096 characters   ............................................4
         storage control segment 30 pages   ................................5
         file name "c:\dbarea\rdsys02\rdddic02"   ..........................6
           initial 150 segments ;   ........................................7

/* Adding a data dictionary LOB RDAREA */
create rdarea PDBDICL01   ..................................................1
         globalbuffer gbp01   ..............................................2
         for LOB used by HiRDB(SQL_ROUTINES)   .............................3
         page 8192 characters   ............................................4
         storage control segment 1 pages   .................................5
         file name "c:\dbarea\rdsys02\rddicl1"   ...........................6
           initial 200 segments ;   ........................................7

/* Adding a data dictionary LOB RDAREA */
create rdarea PDBDICL02   ..................................................1
         globalbuffer gbp01   ..............................................2
         for LOB used by HiRDB(SQL_ROUTINES)   .............................3
         page 8192 characters   ............................................4
         storage control segment 1 pages   .................................5
         file name "c:\dbarea\rdsys02\rddicl2"   ...........................6
           initial 2000 segments ;   .......................................7
```

Explanation

1. Specifies the name of the RDAREA to be added.
2. Specifies the name of the global buffer to which the RDAREA is to be allocated.
3. Specifies the type of RDAREA.
4. Specifies the page length.
5. Specifies the segment size.
6. Specifies the name of the HiRDB file that constitutes the RDAREA.
7. Specifies the number of HiRDB file segments.

(b) HiRDB/Parallel Server

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
/* Adding a data dictionary RDAREA */
create rdarea PDBDDIC02   ..................................................1
         globalbuffer gbp01   ..............................................2
         for datadictionary of routines   ..................................3
         server name dics   ................................................4
         page 4096 characters   ............................................5
         storage control segment 30 pages   ................................6
         file name "c:\dbarea\rdsys02\rdddic02"   ..........................7
           initial 150 segments ;   ........................................8

/* Adding a data dictionary LOB RDAREA */
create rdarea PDBDICL01   ..................................................1
         globalbuffer gbp01   ..............................................2
         for LOB used by HiRDB(SQL_ROUTINES)   .............................3
         server name dics   ................................................4
         page 8192 characters   ............................................5
         storage control segment 1 pages   .................................6
         file name "c:\dbarea\rdsys02\rddicl1"   ...........................7
           initial 200 segments ;   ........................................8

/* Adding a data dictionary LOB RDAREA */
create rdarea PDBDICL02   ..................................................1
         globalbuffer gbp01   ..............................................2
         for LOB used by HiRDB(SQL_ROUTINES)   .............................3
         server name dics   ................................................4
         page 8192 characters   ............................................5
         storage control segment 1 pages   .................................6
         file name "c:\dbarea\rdsys02\rddicl2"   ...........................7
           initial 2000 segments ;   .......................................8
```

Explanation

1. Specifies the name of the RDAREA to be added.
2. Specifies the name of the global buffer to which the RDAREA is to be allocated.
3. Specifies the type of RDAREA.
4. Specifies the name of the server that manages the RDAREA.
5. Specifies the page length.
6. Specifies the segment size.
7. Specifies the name of the HiRDB file that constitutes the RDAREA.
8. Specifies the number of HiRDB file segments.

7.3.3 Control statement (create rdarea statement)

This section explains the operands of the `create rdarea` statement. In the following table, each number corresponds to the number assigned to each operand.

No.	Operand
1	`create [shared]`
2	`rdarea` *RDAREA-name*
3	`[globalbuffer` *global-buffer-name*`]`

4	`for {datadictionary of {routines|dbmanagement}`
	`|user used by {PUBLIC|`*authorization-identifier*`[,`*authorization-identifier*`]...}`
	`|LOB used by {HiRDB (`*data-dictionary-table-name*`)`
	`|PUBLIC|`*authorization-identifier*`[,`*authorization-identifier*`]...}`
	`|list}`

5	`[server name` *server-name*`]`
6	`[open attribute {INITIAL|DEFER|SCHEDULE}]`
7	`[page` *page-length* `characters]`
8	`[storage control segment` *segment-size* `pages]`
9	`[max entries` *maximum-registered-lists-count*`]`
10	`[extension {use` *extension-segments-count* `segments|`<u>`nouse`</u>`}]`
11	`file name "`*HiRDB-file-system-area-name*`\`*HiRDB-filename*`"`
12	`initial` *HiRDB-file-segments-count* `segments`
—	`[file name "`*HiRDB-file-system-area-name*`\`*HiRDB-filename*`"`
—	`initial` *HiRDB-file-segments-count* `segments]...`
—	`;`

Note: Nos. 11 and 12 must be specified at the end of the `create rdarea` statement.

(1) shared

For a HiRDB/Parallel Server (only), specifies that a shared RDAREA is to be added.

Rules

1. To specify `shared`, `Y` must have been specified in the `pd_sharedrdarea_use` operand in the system definition. If this specification is missing, an error results.
2. Specifying `shared` in the case of a HiRDB/Single Server results in an error.

3. An RDAREA created as a shared RDAREA can be referenced from any server, not just the server specified in the `server name` operand (can be referenced from all back-end servers).

(2) RDAREA-name ～ <identifier> ((1-30))

Specifies a name for the RDAREA that is to be added. `ALL` cannot be specified as an RDAREA name. If the RDAREA name is enclosed in double quotation marks (`"`), it is treated as being case sensitive; otherwise, it is treated as all uppercase letters. If the RDAREA name contains a blank, it must be enclosed in double quotation marks.

(3) globalbuffer global-buffer-name ～ <identifier> ((1-16))

If you want to temporarily allocate a global buffer to the added RDAREA, specify the name of the global buffer in the HiRDB system (for a HiRDB/Parallel Server, in the server where the RDAREA is added).

You can use the `pdbufls` command to check the global buffer.

Rules

1. You cannot specify a global buffer for indexes or LOBs.
2. If you are adding a data dictionary RDAREA, be sure to specify this operand.
3. If you specify the number of global buffer sectors, the size of the specified global buffer must be greater than the page length of the RDAREA being added. You can use the `pdbufls` command to check the global buffer.
4. At HiRDB startup, the size of the global buffer is set to the maximum length of the corresponding RDAREA.
5. The global buffer allocated with this operand is no longer effective the next time HiRDB is started normally or restarted after planned termination. Therefore, you need to modify information about the global buffer allocation in the system common definitions at that time.
6. If HiRDB fails to allocate the global buffer, the RDAREA is not added.

(4) for {datadictionary of {routines|dbmanagement}|user used by {PUBLIC|authorization-identifier[,authorization-identifier]...}|LOB used by {HiRDB (data-dictionary-table-name)|PUBLIC|authorization-identifier[,authorization-identifier]...}|list}

Specifies the type of RDAREA to be added.

`datadictionary of routines`

Specifies that the RDAREA to be added is a data dictionary RDAREA.

This data dictionary RDAREA will store the data dictionary tables related to stored procedures, stored functions, and triggers (`SQL_ROUTINES`,

SQL_ROUTINE_RESOURCES, and SQL_ROUTINE_PARAMS).

When a data dictionary RDAREA is added, a data dictionary LOB RDAREA must also be added.

`datadictionary of dbmanagement`

Specifies that the RDAREA to be added is a data dictionary RDAREA for storing a database state analyzed table and database management table.

This data dictionary RDAREA is required when the facility for predicting reorganization time is used.

`user used by {PUBLIC|`*authorization-identifier*`[,`*authorization-identifier*`] . . . }`

Specifies that the RDAREA to be added is a user RDAREA.

`PUBLIC`

Specifies that the user RDAREA to be added is a public RDAREA.

authorization-identifier`[,`*authorization-identifier*`] . . .` ∼ <identifier> ((1-8))

Specifies that the user RDAREA to be added is a private RDAREA, and specifies the authorization identifiers of the users who can use the user RDAREA.

Rules:

1. The system treats an authorization identifier enclosed in double quotation marks (") as case sensitive; otherwise, the system treats it as all uppercase letters.
2. By specifying an authorization identifier, you can assign RDAREA usage privileges to the private RDAREA.
3. You can specify a maximum of 255 authorization identifiers.
4. You cannot specify PUBLIC, MASTER, HiRDB, or ALL as the authorization identifier.

`LOB used by{HiRDB(`*dictionary-table-name*`)|`
`PUBLIC|`*authorization-identifier*`[,`*authorization-identifier*`] . . . }`

Specifies that the RDAREA to be added is a data dictionary LOB RDAREA or user LOB RDAREA.

`HiRDB (`*dictionary-table-name*`)`

Specifies that the RDAREA to be added is a data dictionary LOB RDAREA; SQL_ROUTINES must be specified as the name of the data dictionary table.

Stored procedure definition source statements and objects are stored in such RDAREAs. Two `create rdarea` statements must be specified to create

this type of RDAREA, one for storing stored procedure definition source statements and one for storing stored procedure objects.

Notes:

1. If you add a data dictionary LOB RDAREA, a data dictionary table related to the stored procedure is created. To store this table in a separate data dictionary RDAREA from other data dictionary tables, you can add a data dictionary RDAREA at the same time.
2. By adding a data dictionary LOB RDAREA, you can use abstract data type definitions, stored functions, and stored procedures.
3. If such a data dictionary LOB RDAREA already exists, specification of this operand results in an error.
4. When adding a data dictionary LOB RDAREA, you can specify only data dictionary RDAREAs in the same control statement file.

`PUBLIC`

Specifies that the user LOB RDAREA to be added is a public RDAREA.

authorization-identifier [,*authorization-identifier*] . . . ～ <identifier> ((1-8))

Specifies that the user LOB RDAREA to be added is a private RDAREA, and specifies the authorization identifiers of the users who can use the user LOB RDAREA.

Rules:

1. The system treats an authorization identifier enclosed in double quotation marks (`"`) as case sensitive; otherwise, the system treats it as all uppercase letters.
2. By specifying an authorization identifier, you can assign RDAREA usage privileges to the private RDAREA.
3. You can specify a maximum of 255 authorization identifiers.
4. You cannot specify `PUBLIC`, `MASTER`, `HiRDB`, or `ALL` as the authorization identifier.

`list`

Specifies that the list RDAREA is to be added.

(5) server name server-name ～ <identifier> ((1-8))

HiRDB/Single Server

This operand cannot be specified for a HiRDB/Single Server.

HiRDB/Parallel Server

Specifies the name of the server that is to manage the added RDAREA.

For a data dictionary RDAREA or data dictionary LOB RDAREA, the server name of the data dictionary server must be specified. If the name of any other server is specified, the name of the dictionary server is assumed.

For a user RDAREA or user LOB RDAREA, the name of a back-end server must be specified. If the name of any other server is specified, an error results.

When a shared RDAREA is added, the server specified in this operand becomes an updatable back-end server.

(6) open attribute {INITIAL|DEFER|SCHEDULE}

Specifies the RDAREA's open timing.

Criteria

Normally, specify `INITIAL` unless a special operation mode is used.

To reduce the HiRDB startup time when there are many RDAREAs, specify `DEFER`.

Rules

1. This operand takes effect only when `Y` is specified in the `pd_rdarea_open_attribute_use` operand in the system common definitions.
2. This operand is applicable to user RDAREAs, user LOB RDAREAs, and list RDAREAs. This operand is ignored if specified for any other type of RDAREA (the RDAREA open timing for all other RDAREA types is `INITIAL`).
3. When this operand is omitted, the system assumes the value of the `pd_rdarea_open_attribute` operand in the system common definitions.
4. The value of this operand takes effect at the next startup, regardless of the HiRDB startup mode.
5. If the HiRDB file system area assigned to the added RDAREA is shared by other RDAREAs, all such RDAREAs must have the same open timing. Otherwise, an expected result may not be obtained from the RDAREAs sharing the HiRDB file system area.

`INITIAL`

The RDAREA's HiRDB file system area is to be opened at the time of system startup and the RDAREA information is to be made resident in memory; thus, high-speed operation is available from the first SQL.

Notes

1. The RDAREA's initial status at system startup is open and remains

unchanged until an operation command is executed, unless it is placed in error shutdown status due to an error.

2. The INITIAL open attribute does not permit access to a closed RDAREA.

DEFER

The RDAREA's HiRDB file system area is not to be opened at the time of system startup; rather, it is to be opened and its RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed. Thus, high-speed operation is available beginning with the second access to the HiRDB file system area.

Notes

1. The RDAREA's initial status at system startup is closed and changes to open the first time an RDAREA in its HiRDB file system area is accessed. The HiRDB file system area's RDAREAs remain in open status thereafter until an operation command is executed, unless they are placed in error shutdown status due to an error.
2. The DEFER open attribute permits access to a closed RDAREA.

SCHEDULE

The RDAREA's HiRDB file system area is not to be opened at the time of system startup; rather, it is to be opened and the RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed from within a transaction. When this attribute is used, transaction workload increases because a HiRDB file system area opened by a transaction is closed when the transaction terminates and open processing must be performed on the HiRDB file system area the first time each transaction accesses one of its RDAREAs.

Notes

1. The RDAREA's initial status at system startup is closed and changes to open when the RDAREA is first accessed by a transaction. When the transaction is terminated, all the RDAREAs opened by it are closed.

   You can use the pdopen command to keep RDAREAs in open status until they are placed in shutdown close status. You can also use other operation commands to change the RDAREA status. However, if an error is detected, RDAREAs are placed in error shutdown status.
2. The SCHEDULE open attribute permits access to a closed RDAREA.

(7) page page-length characters ～ <unsigned integer> ((4096-30720)) <<4096 or 8192>>

Specifies the page length of the HiRDB file that constitutes the RDAREA, expressed as a multiple of 2,048 bytes. If you have specified the sector length during creation of the HiRDB file system area, specify a value that is a multiple of 2,048 and is also a multiple of the sector length.

The following RDAREAs will have fixed-length pages:

- Data dictionary LOB RDAREA and user LOB RDAREA

  Specify 8,192. If any other value is specified or the operand is omitted, 8,192 is assumed.

- Data dictionary RDAREA for storing a database state analyzed table and database management table

  Specify 4,096. If any other value is specified or the operand is omitted, 4,096 is assumed.

For details about the page lengths of other RDAREAs, see the *HiRDB Version 8 Installation and Design Guide* and specify the appropriate value. This page length is used by HiRDB as the smallest unit of I/O operations on the RDAREA.

(8) storage control segment segment-size pages ～ <unsigned integer> ((1-16000)) <<1 or 50>>

Specifies in pages the size of one segment.

For a data dictionary LOB RDAREA or a user LOB RDAREA `1` must be specified; if any other value is specified, `1` is assumed. For details about the segment sizes of other RDAREAs, see the *HiRDB Version 8 Installation and Design Guide* and specify the appropriate value.

(9) [max entries maximum-registered-lists-count] ～ <unsigned integer> ((500-50000 and a multiple of 500)) <<500>>

Specifies the maximum number of lists that can be created in the list RDAREA, expressed as a multiple of 500. If the specified value is not a multiple of 500, the system rounds it up to the next multiple of 500. An error results if you specify this operand for any RDAREA other than the list RDAREA.

(10) [extension {use extension-segments-count segments|nouse}]

Specifies whether or not to apply automatic extension to the RDAREA that is being added.

Automatic extension of RDAREA is a facility that extends an RDAREA automatically in the event of a shortage of RDAREA space if there is enough space in the HiRDB file system area. If you use this facility for the RDAREA and a shortage of unused segments occurs, the system allocates new unused segments to the RDAREA. These

new unused segments are added at the end of the HiRDB file constituting the RDAREA.

Prerequisites

1. You need to specify the `-e` option (specifying the number of extensions) for the HiRDB file system area containing the RDAREA.
2. There must be enough space in the HiRDB file system area that contains the last HiRDB file constituting the RDAREA.

`use` *extension-segments-count* `segments`

Specifies that automatic extension of an RDAREA is to be applied.

The automatic extension of an RDAREA is applicable to data dictionary RDAREAs, data dictionary LOB RDAREAs, user RDAREAs, and user LOB RDAREAs. For any other type of RDAREA, the system assumes `nouse` even if `use` is specified.

For an RDAREA other than a LOB RDAREA, automatic extension occurs when there are no more used free segments or used segments. For a LOB RDAREA, automatic extension occurs when there are no more unused segments.

extension-segments-count ～ <unsigned integer> ((1-64000))

Specifies the number of extension segments.

The maximum number of HiRDB file extensions is 24. If this value is exceeded, an error occurs. The maximum number of extensions per HiRDB file system area is determined by the value specified when the HiRDB file system area is created. Therefore, you need to define the maximum number of extensions, taking into account the number of files in the HiRDB file system area and the frequency of extension.

`nouse`

Specifies that automatic extension of an RDAREA is not to be applied.

Notes

1. If allocation of unused segments fails due to a shortage of space in the HiRDB file system area, either extend or re-initialize the RDAREA or use the database reorganization utility to reorganize the table.
2. If the number of extensions exceeds the maximum value, integrate the extensions in the HiRDB file system area containing the RDAREA or add a HiRDB file in another HiRDB file system area to the RDAREA.

   To integrate extensions, make a backup copy with `pdfbkup`, initialize the HiRDB file system area with `pdfmkfs`, then restore the HiRDB file system area from its backup copy using `pdfrstr`.

3. The last file is locked from the beginning to the end of the automatic extension process.

(11) file name "HiRDB-file-system-area-name\HiRDB-filename" ～ ((up to 167 characters))

Specifies a file to be allocated to the added RDAREA, in terms of a HiRDB file system area name and a HiRDB file name.

Rules

1. Enclose *HiRDB-file-system-area-name\HiRDB-filename* in double quotation marks (").
2. Do not include a linefeed character inside the double quotation marks.
3. You can allocate a maximum of 16 HiRDB files per RDAREA.
4. The database structure modification utility (pdmod) creates the HiRDB file in the specified HiRDB file system area.
5. *HiRDB-file-system-area-name\HiRDB-filename* must be unique in the HiRDB system.
6. For a list RDAREA, use a HiRDB file system area with a purpose of WORK.
7. The string up to *HiRDB-file-system-area-name* is not case sensitive, but *HiRDB-filename* is case sensitive.

HiRDB-file-system-area-name ～ <pathname>

Specifies the name of the HiRDB file system area.

HiRDB-filename ～ <HiRDB filename> ((1-30 characters))

Specifies the name of the HiRDB file (which cannot begin with pl).

(12) initial HiRDB-file-segments-count segments ～ <unsigned integer>

Specifies the number of segments in this file in the RDAREA. The number of segments is the total number of pages/segment size. (For details about how to estimate the total number of pages, see the *HiRDB Version 8 Installation and Design Guide*.) The size of a HiRDB file cannot exceed 2 GB, regardless of the number of segments.

7.3.4 Notes

1. An added RDAREA becomes available at the following time:
   - Global buffer not allocated by the control statement

     The added RDAREA becomes available the next time HiRDB is started normally. If there is a global buffer created with the -o option specified and the global buffer size is larger than the page size of the added RDAREA, the RDAREA becomes available also at the time of a restart. After the restart,

the RDAREA is closed, so you will have to use the pdopen command to open the RDAREA before it can be used.

- Global buffer allocated by the control statement

  The added RDAREA becomes available upon completion of the RDAREA addition processing. When a shared RDAREA is added, the global buffer must be allocated to all back-end servers.

  In either case, you must modify the system common definitions so that a global buffer is allocated to the newly added RDAREA at the time of the next HiRDB normal startup. For details about the system common definitions, see the *HiRDB Version 8 System Definition*. If the system common definitions are not modified but there is an available global buffer for which the -o option was specified in the pdbuffer operand of the system common definition, that global buffer is allocated automatically to the RDAREA the next time HiRDB is started normally. Also, if there is an available global buffer for which the -o option was specified, or a global buffer is specified with the control statement, that global buffer is allocated at the time of a HiRDB restart.

2. After adding an RDAREA, you should make a backup copy of the added RDAREA, the master directory RDAREA, and the corresponding data dictionary RDAREA, using the database copy utility (pdcopy). For details about the database copy utility, see *18. Database Copy Utility (pdcopy)*.

3. When you add the data dictionary LOB RDAREAs, stored procedures or stored functions become available. When the data dictionary LOB RDAREAs are added, data dictionary tables are also created at the same time. Therefore, you should check to ensure that there is sufficient free space in the data dictionary RDAREAs. If there is not enough space, expand the data dictionary RDAREAs in advance. To check the size of an RDAREA, use the database condition analysis utility.

   If you want to store the data dictionary tables related to stored procedures in a separate data dictionary RDAREA from other data dictionary tables, you can add a data dictionary RDAREA at the same time.

   Once the data dictionary LOB RDAREAs have been added, the stored procedures or stored functions become available when global buffers are allocated to the added RDAREAs.

7.4 Expanding an RDAREA

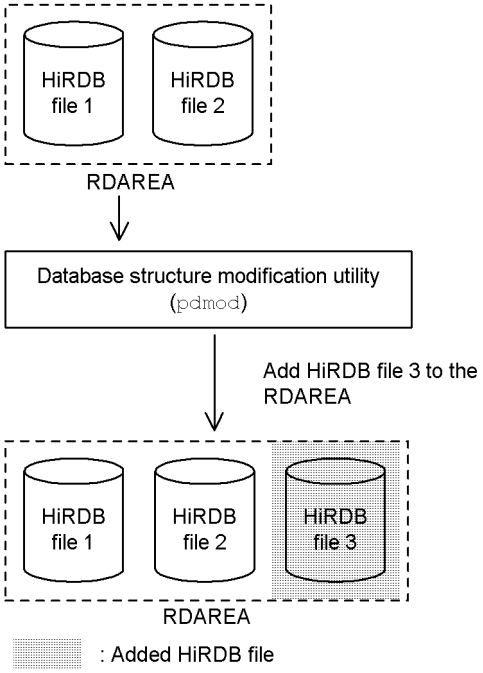
7.4.1 Function

RDAREA expansion is a function that adds HiRDB files to an RDAREA when the initially estimated RDAREA size becomes insufficient. An RDAREA is expanded by adding component HiRDB files.

Files can be added to an RDAREA up to the maximum of 16 files. An RDAREA can be expanded even while a table or index contained in the RDAREA is being referenced by another user. RDAREA expansion has no effect on the data already contained in the RDAREA.

Figure 7-2 provides an overview of the use of the database structure modification utility (pdmod) for RDAREA expansion.

Figure 7-2: Overview of database structure modification utility (pdmod): RDAREA expansion



Note

If you have extended an RDAREA, be sure to make a backup copy of the RDAREA as well as the other RDAREAs that need to be backed up at the same

time. For details about how to back up RDAREAs, see the *HiRDB Version 8 System Operation Guide*.

Prerequisites

1. To expand an RDAREA, first you need to initialize the HiRDB file system areas for creation of HiRDB files.
2. An RDAREA subject to expansion must be in one of the following statuses:
   - Master directory RDAREA

     Shutdown release and open status
   - Data directory RDAREA, data dictionary RDAREA, data dictionary LOB RDAREA, user RDAREA, user LOB RDAREA, registry RDAREA, or registry LOB RDAREA

     • RDAREA open timing set to `INITIAL`: Shutdown release status or open status but shut down by command

     • RDAREA open timing set to `DEFER` or `SCHEDULE`: Shutdown release status, open status but shut down by command, or shutdown release and closed status

     To specify the RDAREA open timing, use the `pd_rdarea_open_attribute` and `pd_rdarea_open_attribute_use` operands in the system common definitions. For details about these operands, see the *HiRDB Version 8 System Definition*. You can check the current RDAREA open timing using the `pddbls` command.
3. When you expand an RDAREA, check that the maximum number of RDAREA component files in the system common definition (`pd_max_file_no`) will not be exceeded. If this value is exceeded, you must use the `pdchgconf` command to change the system common definition or terminate the HiRDB system normally and then change the system common definition.
4. A shared RDAREA that is expanded is regarded as having been expanded at every back-end server (even if a back-end server has no shared RDAREA, the system assumes that an RDAREA has been expanded). If the maximum permissible number of HiRDB files constituting RDAREAs is exceeded at any of the back-end servers, the shared RDAREA cannot be expanded at all.

7.4.2 Examples

This section presents examples of using the database structure modification utility (expanding an RDAREA).

(1) Expanding an RDAREA

This example expands the following RDAREAs:

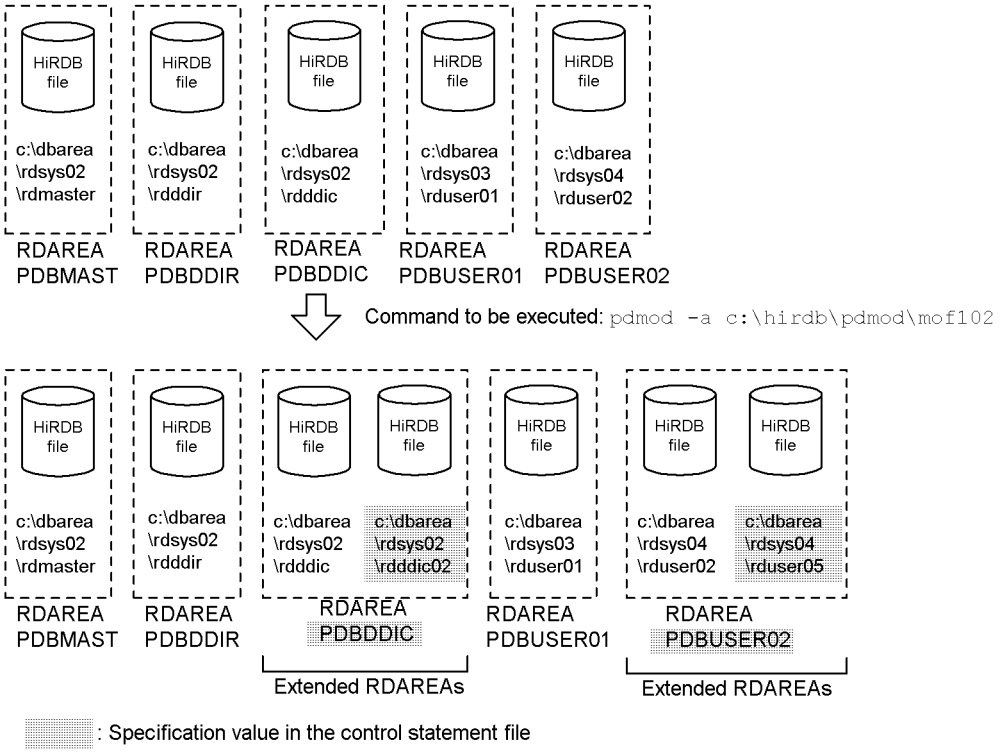
- PDBDDIC

  Allocates a new HiRDB file (c:\dbarea\rdsys02\rdddic02).

- PDBUSER02

  Allocates a new HiRDB file (c:\dbarea\rdsys04\rduser05).

Overview



Contents of the control statement file (c:\hirdb\pdmod\mofl02)

```
/* Expanding a data dictionary RDAREA */
expand rdarea PDBDDIC   ..................................................1
         file name "c:\dbarea\rdsys02\rdddic02"   ........................2
           initial 50 segments ;   .......................................3

/* Expanding a user RDAREA */
expand rdarea PDBUSER02   ................................................1
         file name "c:\dbarea\rdsys04\rduser05"   ........................2
           initial 100 segments ;   ......................................3
```

Explanation

1. Specifies the name of the RDAREA to be expanded.
2. Specifies the name of the RDAREA file to be added to the RDAREA.
3. Specifies the number of HiRDB file segments.

7.4.3 Control statement (expand rdarea statement)

This section explains the operands of the `expand rdarea` statement. In the following table, each number corresponds to the number assigned to each operand.

No.	Operand
1	`expand rdarea` *RDAREA-name*
2	`[extension {use` *extension-segments-count* `segments|nouse}]`
3	`file name "`*HiRDB-file-system-area-name*`\`*HiRDB-filename*`"`
4	`initial` *HiRDB-file-segments-count* `segments`
—	`[file name "`*HiRDB-file-system-area-name*`\`*HiRDB-filename*`"`
—	`initial` *HiRDB-file-segments-count* `segments]...`
—	`;`

Note: Nos. 3 and 4 must be specified at the end of the `expand rdarea` statement.

(1) RDAREA-name ～ <identifier> ((1-30))

Specifies the name of the RDAREA to be expanded. `ALL` cannot be specified as an RDAREA name. If the RDAREA name is enclosed in double quotation marks (`"`), it is treated as being case sensitive; otherwise, it is treated as all uppercase letters. The RDAREA name must be enclosed in double quotation marks if it includes a blank.

(2) extension {use extension-segments-count segments|nouse}]

Specifies whether or not to apply automatic extension to the RDAREA that is being expanded. When this operand is omitted, the system assumes the specification existing before the RDAREA is expanded.

Automatic extension of RDAREA is a facility that extends an RDAREA automatically in the event of a shortage of RDAREA space if there is enough space in the HiRDB file system area. If you use this facility for an RDAREA and a shortage of unused segments occurs, the system allocates new unused segments to the RDAREA. These new unused segments are added at the end of the HiRDB file constituting the RDAREA.

Prerequisites

1. You need to specify the `-e` option (specifying the number of extensions) for the HiRDB file system area containing the RDAREA.

2. There must be enough space in the HiRDB file system area that contains the last HiRDB file constituting the RDAREA.

`use` *extension-segments-count* `segments`

Specifies that automatic extension of RDAREA is to be applied.

The automatic extension of an RDAREA is applicable to data dictionary RDAREAs, data dictionary LOB RDAREAs, user RDAREAs, and user LOB RDAREAs, registry RDAREAs, and registry LOB RDAREAs. For any other type of RDAREA, the system assumes `nouse`, even if `use` is specified.

For an RDAREA other than a LOB RDAREA, automatic extension occurs when there are no more used free segments or used segments. For a LOB RDAREA, automatic extension occurs when there are no more unused segments.

extension-segments-count ～ <unsigned integer> ((1-64000))

Specifies the number of extension segments.

The maximum number of HiRDB file extensions is 24. If this value is exceeded, an error occurs. The maximum number of extensions per HiRDB file system area is determined by the value specified when the HiRDB file system area is created. Therefore, you need to define the maximum number of extensions, taking into account the number of files in the HiRDB file system area and the frequency of extension.

`nouse`

Specifies that automatic extension of RDAREA is not to be applied.

Notes

1. If allocation of unused segments fails due to a shortage of space in the HiRDB file system area, either extend or re-initialize the RDAREA or use the database reorganization utility to reorganize the table.
2. If the number of extensions exceeds the maximum value, integrate the extensions in the HiRDB file system area containing the RDAREA or add HiRDB file in another HiRDB file system area to the RDAREA.

   To integrate extensions, make a backup copy with `pdfbkup`, initialize the HiRDB file system area with `pdfmkfs`, then restore the HiRDB file system area from its backup copy using `pdfrstr`.
3. The last file is locked from the beginning to the end of the automatic extension process.

(3) file name "HiRDB-file-system-area-name\HiRDB-filename" ～ ((up to 167 characters))

Specifies a file to be added to the RDAREA, in terms of a HiRDB file system area

name and a HiRDB file name.

Rules

1. Enclose *HiRDB-file-system-area-name\HiRDB-filename* in double quotation marks (").
2. Do not include a linefeed character inside the double quotation marks.
3. You can allocate a maximum of 16 HiRDB files per RDAREA.
4. The database structure modification utility (pdmod) creates the HiRDB file in the specified HiRDB file system area.
5. *HiRDB-file-system-area-name\HiRDB-filename* must be unique in the HiRDB system.
6. For a list RDAREA, use a HiRDB file system area with a purpose of WORK.
7. If you are extending the RDAREA to a HiRDB file system area with a sector length specified, the page length of the RDAREA must be a multiple of the sector length.
8. The string up to *HiRDB-file-system-area-name* is not case sensitive, but *HiRDB-filename* is case sensitive.

HiRDB-file-system-area-name ～ <pathname>

Specifies the name of the HiRDB file system area.

HiRDB-filename ～ <HiRDB filename> ((1-30 characters))

Specifies the name of the HiRDB file (which cannot begin with pl).

(4) initial HiRDB-file-segments-count segments ～ <unsigned integer>

Specifies the number of segments in this file in the RDAREA file. The number of segments is the total number of pages/segment size. For details about how to estimate the total number of pages, see the *HiRDB Version 8 Installation and Design Guide*. The size of a HiRDB file cannot exceed 2 GB, regardless of the number of segments.

7.4.4 Notes

1. After expanding an RDAREA, you should make a backup copy of the expanded RDAREA, the master directory RDAREA, and the corresponding data dictionary RDAREA, using the database copy utility (pdcopy). For details about the database copy utility, see *18. Database Copy Utility (pdcopy)*.

7.5 Reinitializing an RDAREA

7.5.1 Function

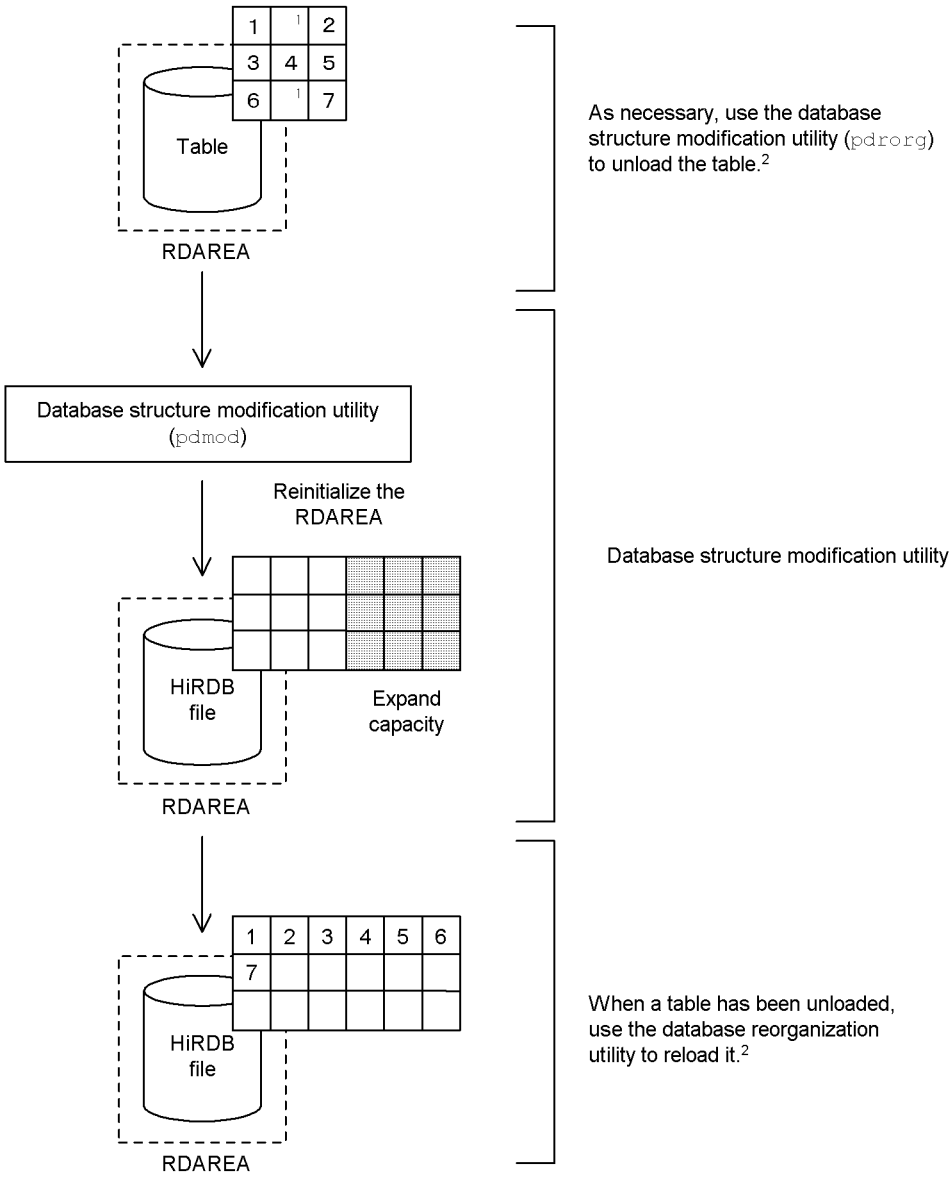
User RDAREA reinitialization is a function that reinitializes an RDAREA for any of the following purposes:

- To change the size, number, or names of the HiRDB files comprising the RDAREA
- To change the segment size and page length of the RDAREA (user RDAREAs only)
- To recover the RDAREA from an error without using its backup

When an RDAREA is reinitialized, table and index definitions are inherited, but all data is deleted. The data must be re-created by the database load utility (`pdload`) or the database reorganization utility (`pdrorg`).

Figure 7-3 provides an overview of the use of the database structure modification utility (`pdmod`) for RDAREA reinitialization.

Figure 7-3: Overview of database structure modification utility (pdmod): RDAREA reinitialization



Note

As soon as an RDAREA has been reinitialized, a backup of the RDAREA should be made.

[1] Free space

[2] For details about unload and reload processing, see Chapter *8. Database Reorganization Utility (pdrorg)*.

Prerequisites

1. An RDAREA subject to reinitialization must be in shutdown and closed status. A shared RDAREA must be in shutdown and closed status at all back-end servers.
2. When reinitializing an RDAREA, be sure that the number of HiRDB files does not exceed the maximum number of HiRDB files constituting the RDAREAs (`pd_max_file_no`) specified in the system common definition. If this value is exceeded, you must use the `pdchgconf` command to change the system common definition or terminate the HiRDB system normally and then change the system common definition.
3. A shared RDAREA that is reinitialized is regarded as having been reinitialized at every back-end server (even if a back-end server has no shared RDAREA to be reinitialized, the system assumes that an RDAREA has been reinitialized). If the maximum permissible number of HiRDB files constituting RDAREAs is exceeded at any of the back-end servers, the shared RDAREA cannot be reinitialized at all.
4. If reinitializing a data dictionary LOB RDAREA, you can execute the utility only on an RDAREA in which stored procedure or stored function objects are stored. After execution, you need to re-create all SQL objects using `ALTER PROCEDURE` (with `ALL` specified) or `ALTER ROUTINE`. For details about `ALTER PROCEDURE` and `ALTER ROUTINE`, see the *HiRDB Version 8 SQL Reference*. You can determine the RDAREA storing the stored procedure or stored function objects by searching the data dictionary table. For details about the data dictionary search method, see the *HiRDB Version 8 UAP Development Guide*.
5. If reinitializing the registry LOB RDAREA, you also need to reinitialize the registry RDAREA (it is not possible to reinitialize the registry LOB RDAREA only). After the reinitialization, you must re-register any registry information that is used by the plug-in module.

7.5.2 Examples

This section presents an example of using the database structure modification utility (reinitializing an RDAREA).

(1) Reinitializing an RDAREA

This example reinitializes the following RDAREAs:

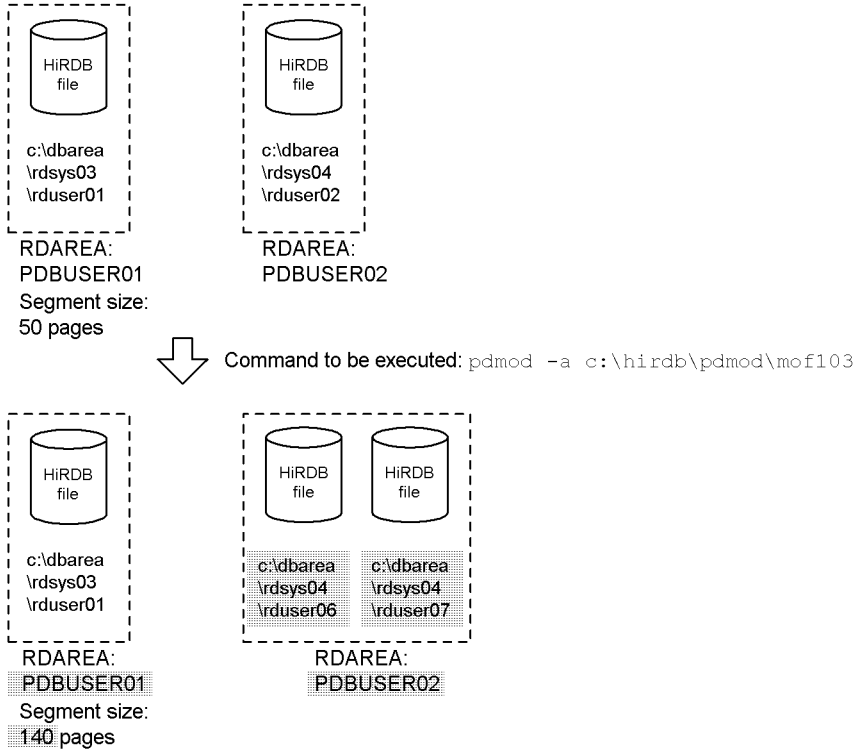
- PDBUSER01

  Change the segment size to 140 pages

- PDBUSER02

  Change the HiRDB file (c:\dbarea\rdsys04\rduser02) allocated to the RDAREA to c:\dbarea\rdsys04\rduser06 and c:\dbarea\rdsys04\rduser07.

Overview



Contents of the control statement file (c:\hirdb\pdmod\mofl03)

```
/* Re-initializing a user RDAREA */
initialize rdarea PDBUSER01   ..................................................1
            storage control segment 140 pages;   ..............................2
```

```
/* Re-initializing a user RDAREA */
initialize rdarea PDBUSER02   ..................................................1
            with reconstruction   ..............................................3
            file name "c:\dbarea\rdsys04\rduser06"   ...........................4
              initial 100 segments   ...........................................5
            file name "c:\dbarea\rdsys04\rduser07"   ...........................4
              initial 60 segments ;   ..........................................5
```

Explanation

1. Specifies the name of the RDAREA to be re-initialized.
2. Specifies the segment size after change.
3. Specifies that all HiRDB files constituting the RDAREA are to be changed.
4. Specifies the name of the HiRDB file to be re-allocated to the RDAREA.
5. Specifies the number of HiRDB file segments.

7.5.3 Control statement (initialize rdarea statement)

This section explains the operands of the `initialize rdarea` statement. In the following table, each number corresponds to the number assigned to each operand.

No.	Operand
1	`initialize rdarea` *RDAREA-name*
2	`[with reconstruction]`
3	`[open attribute {INITIAL|DEFER|SCHEDULE}]`
4	`[page` *page-length* `characters]`
5	`[storage control segment` *segment-size* `pages]`
6	`[max entries` *maximum-lists-count-registered*`]`
7	`[extension {use` *extension-segments-count* `segments|nouse}]`
8	`[file name "`*HiRDB-file-system-area-name*`\`*HiRDB-filename*`"`
9	`initial` *HiRDB-file-segments-count* `segments]`
—	`[file name "`*HiRDB-file-system-area-name*`\`*HiRDB-filename*`"`
—	`initial` *HiRDB-file-segments-count* `segments]...]`
—	`;`

Note: Nos. 8 and 9 must be specified at the end of the `initialize rdarea` statement.

(1) RDAREA-name ～ <identifier> ((1-30))

Specifies the name of the RDAREA that is to be reinitialized. `ALL` cannot be specified as an RDAREA name. If the RDAREA name is enclosed in double quotation marks (`"`), it is treated as being case sensitive; otherwise, it is treated as all uppercase letters. The RDAREA name must be enclosed in double quotation marks if it includes a blank.

(2) with reconstruction

Specifies that all HiRDB files comprising the RDAREA are to be reconstructed. When this operand is specified, all the HiRDB files comprising the RDAREA to be reconstructed must be specified in `file name` operands. Any existing HiRDB file definitions that are not specified in `file name` operand definitions become invalid; HiRDB files that will no longer be used should be deleted. When this operand is omitted, the value in effect before execution of RDAREA reinitialization is inherited. To change an HiRDB file definition without specifying this operand, the name of the applicable HiRDB file must be specified in a file name operand and the new values must be specified in the applicable operands.

(3) open attribute {INITIAL|DEFER|SCHEDULE}

Specifies the RDAREA open timing.

Criteria

Normally, specify `INITIAL` unless a special operation mode is used.

To reduce the HiRDB startup time when there are many RDAREAs, specify `DEFER`.

Rules

1. This operand takes effect only when `Y` is specified in the `pd_rdarea_open_attribute_use` operand in the system common definitions.
2. This operand is applicable to user RDAREAs, user LOB RDAREAs, and list RDAREAs. This operand is ignored if specified for any other type of RDAREA (the RDAREA open timing for all other RDAREA types is `INITIAL`).
3. When this operand is omitted, the system assumes the value existing before the RDAREA reinitialized.
4. The value of this operand takes effect at the next startup, regardless of the HiRDB startup mode.
5. If the HiRDB file system area assigned to the corresponding RDAREA is shared by other RDAREAs, all such RDAREAs must have the same open timing. Otherwise, an expected result may not be obtained from the RDAREAs sharing the HiRDB file system area.

INITIAL

The RDAREA's HiRDB file system area is to be opened at the time of system startup and the RDAREA information is to be made resident in memory; thus, high-speed operation is available from the first SQL.

Notes

1. The RDAREA's initial status at system startup is open and remains unchanged until an operation command is executed, unless it is placed in error shutdown status due to an error.
2. The INITIAL open attribute does not permit access to a closed RDAREA.

DEFER

The RDAREA's HiRDB file system area is not to be opened at the time of system startup; rather, it is to be opened and its RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed. Thus, high-speed operation is available beginning with the second access to the HiRDB file system area.

Notes

1. The RDAREA's initial status at system startup is closed and changes to open the first time an RDAREA in its HiRDB file system area is accessed. The HiRDB file system area's RDAREAs remain in open status thereafter until an operation command is executed, unless they are placed in error shutdown status due to an error.
2. The DEFER open attribute permits access to a closed RDAREA.

SCHEDULE

The RDAREA's HiRDB file system area is not to be opened at the time of system startup; rather, it is to be opened and the RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed from within a transaction. When this attribute is used, transaction workload increases because a HiRDB file system area opened by a transaction is closed when the transaction terminates, and open processing must be performed on the HiRDB file system area the first time each transaction accesses one of its RDAREAs.

Notes

1. The RDAREA's initial status at system startup is closed and changes to open when the RDAREA is first accessed by a transaction. When the transaction is terminated, all the RDAREAs opened by it are closed.

   You can use the pdopen command to keep RDAREAs in open status

until they are placed in shutdown close status. You can also use other operation commands to change the RDAREA status. However, if an error is detected, RDAREAs are placed in error shutdown status.

2. The SCHEDULE open attribute permits access to a closed RDAREA.

(4) page page-length characters ～ <unsigned integer> ((4096-30720))

Specifies a multiple of 2,048 bytes if the page length for the HiRDB files that are to constitute the RDAREA is to be changed. If you specified a sector length during creation of the HiRDB file system area, specify a multiple of the sector length.

When this operand is omitted, the value in effect before execution of RDAREA reinitialization is inherited.

For a data dictionary LOB RDAREA and user LOB RDAREA, 8192 must be specified; if any other value is specified, 8192 is assumed. For details about the page lengths of other RDAREAs, see the *HiRDB Version 8 Installation and Design Guide* and specify the appropriate value.

The page length specified in this operand is used as the smallest unit for I/O operations performed by HiRDB on the RDAREA.

Rules

1. If an RDAREA to be reinitialized includes a table, if you reduce the page length, the table row may not fit (in which case an error occurs during data storage).
2. If the RDAREA includes a FIX table, an error results if the following value is smaller than the row length (you can check the row length by retrieving ROW_LENGTH from SQL_TABLES in the data dictionary table):

   $(\downarrow \text{page length after modification} \div 1000 \downarrow) \times 1000$
3. An error results if the specified page length is greater than the maximum page length for RDAREAs assigned to the global buffer.
4. If the RDAREA to be re-initialized has an index, an error results if the key length of the index obtained from the specified page size is greater than the following value:

   MIN (page size of RDAREA/2 - 1242, 4036)

   (bytes)

(5) storage control segment segment-size pages ～ <unsigned integer> ((1-16000))

Specifies a new segment size (number of pages) for the RDAREA.

When this operand is omitted, the value in effect before execution of RDAREA

reinitialization is inherited.

For a data dictionary LOB RDAREA or user LOB RDAREA, `1` must be specified; if any other value is specified, `1` is assumed. For details about the segment sizes of other RDAREAs, see the *HiRDB Version 8 Installation and Design Guide* and specify the appropriate value.

(6) max entries maximum-lists-count-registered ～ <unsigned integer> ((500-50000 and a multiple of 500))

Specifies the maximum number of lists that can be created in the list RDAREA, expressed as a multiple of 500. If the specified value is not a multiple of 500, the system rounds it up to the next multiple of 500.

When this operand is omitted, the value in effect before execution of RDAREA reinitialization is inherited.

An error results if you specify this operand for any RDAREA other than the list RDAREA.

(7) extension {use extension-segments-count segments|nouse}

Specifies whether or not to apply automatic extension to the RDAREA that is being reinitialized. When this operand is omitted, the value in effect before execution of RDAREA reinitialization is inherited.

Automatic extension of RDAREA is a facility that extends an RDAREA automatically in the event of a shortage of RDAREA space if there is enough space in the HiRDB file system area. If you use this facility for an RDAREA and a shortage of unused segments occurs, the system allocates new unused segments to the RDAREA. These new unused segments are added at the end of the HiRDB file constituting the RDAREA.

Prerequisites

1. You need to specify the `-e` option (specifying the number of extensions) for the HiRDB file system area containing the RDAREA.
2. There must be enough space in the HiRDB file system area that contains the last HiRDB file constituting the RDAREA.

`use` *extension-segments-count* `segments`

Specifies that automatic extension of RDAREA is to be applied.

The automatic extension of RDAREA is applicable to data dictionary RDAREAs, data dictionary LOB RDAREAs, user RDAREAs, and user LOB RDAREAs. For any other type of RDAREA, the system assumes `nouse`, even if `use` is specified.

For an RDAREA other than a LOB RDAREA, automatic extension occurs when there are no more used free segments or used segments. For a LOB RDAREA, automatic extension occurs when there are no more unused segments.

extension-segments-count ～ <unsigned integer> ((1-64000))

Specifies the number of extension segments.

The maximum number of HiRDB file extensions is 24. If this value is exceeded, an error occurs. The maximum number of extensions per HiRDB file system area is determined by the value specified when the HiRDB file system area is created. Therefore, you need to define the maximum number of extensions, taking into account the number of files in the HiRDB file system area and the frequency of extension.

`nouse`

Specifies that automatic extension of RDAREA is not to be applied.

Notes

1. If allocation of unused segments fails due to a shortage of space in the HiRDB file system area, either extend or re-initialize the RDAREA or use the database reorganization utility to reorganize the table.
2. If the number of extensions exceeds the maximum value, integrate the extensions in the HiRDB file system area containing the RDAREA or add a HiRDB file in another HiRDB file system area to the RDAREA.

   To integrate extensions, make a backup copy with `pdfbkup`, initialize the HiRDB file system area with `pdfmkfs`, then restore the HiRDB file system area from its backup copy using `pdfrstr`.
3. The last file is locked from the beginning to the end of the automatic extension process.

(8) file name "HiRDB-file-system-area-name\HiRDB-filename" ～ ((up to 167 characters))

Specifies a file that is to comprise the reinitialized RDAREA, in terms of a HiRDB file system area name and a HiRDB file name.
HiRDB-file-system-area-name\HiRDB-filename must be enclosed in double quotation marks.

Rules

1. When this operand is omitted, the value in effect before execution of RDAREA reinitialization is inherited.
2. Do not include a linefeed character inside the double quotation marks.
3. You can allocate a maximum of 16 HiRDB files per RDAREA.
4. The database structure modification utility (`pdmod`) creates the HiRDB file in the specified HiRDB file system area. If the HiRDB file system area already contains a HiRDB file, `pdmod` deletes it and then creates a new file;

therefore, an error results if there are no HiRDB file privileges.

5. *HiRDB-file-system-area-name**HiRDB-filename* must be unique in the HiRDB system.
6. For a list RDAREA, use a HiRDB file system area with a purpose of `WORK`.
7. The string up to *HiRDB-file-system-area-name* is not case sensitive, but *HiRDB-filename* is case sensitive.

HiRDB-file-system-area-name ∼ <pathname>

Specifies the name of the HiRDB file system area.

HiRDB-filename ∼ <HiRDB filename> ((1-30 characters))

Specifies the name of the HiRDB file (which cannot begin with `pl`).

(9) initial HiRDB-file-segments-count segments ∼ <unsigned integer>

Specifies the number of segments in this file in the RDAREA.

When this operand is omitted, the value in effect before execution of RDAREA reinitialization is inherited.

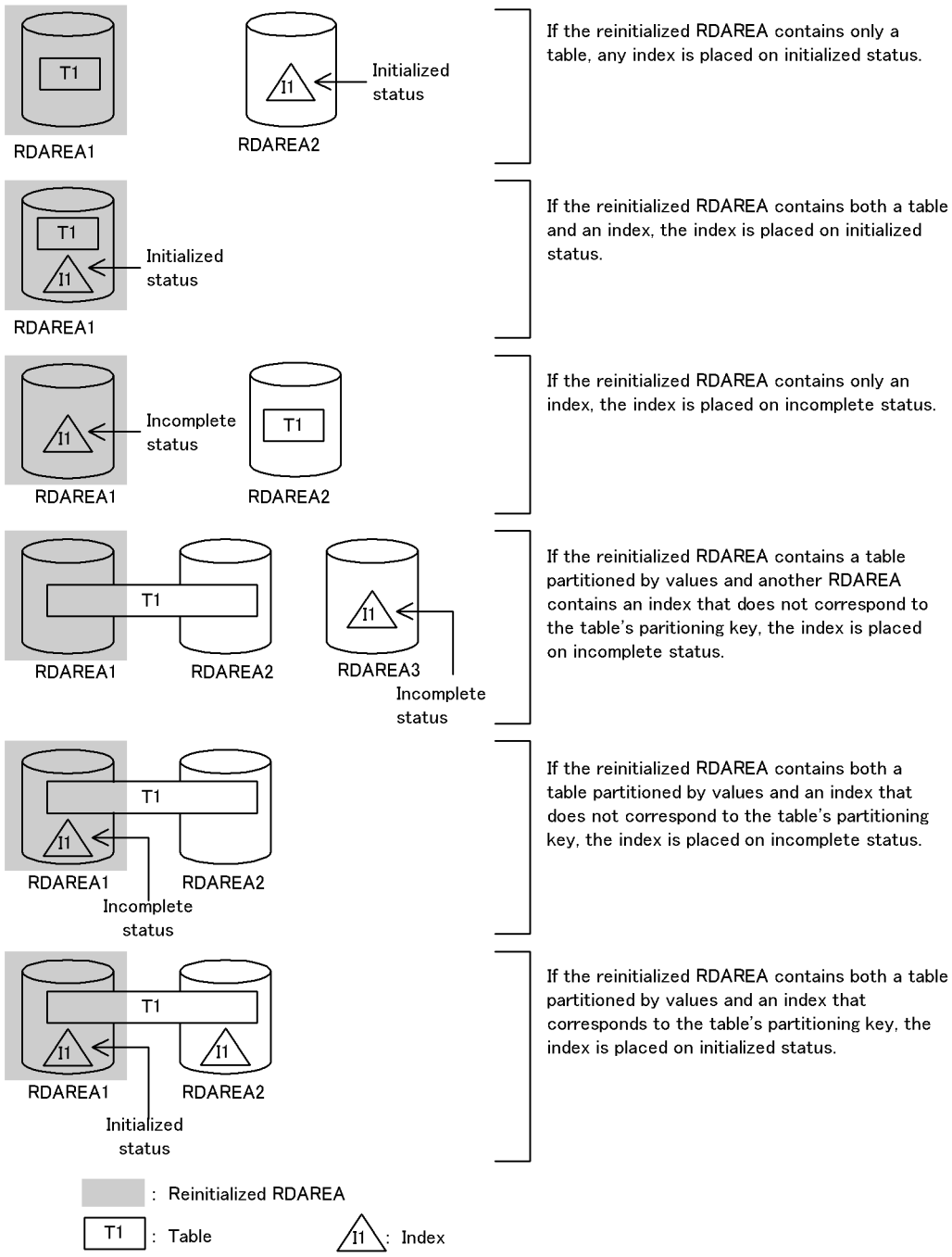
The number of segments is the total number of pages/segment size. (For details about how to estimate the total number of pages, see the *HiRDB Version 8 Installation and Design Guide*.) The size of a HiRDB file cannot exceed 2 GB, regardless of the number of segments.

7.5.4 Table and index status after reinitialization

Figures 7-4 and 7-5 show the table and index status after re-initialization for an RDAREA containing a table and index. Figure 7-6 shows the conditions of a re-initialized table, columns of abstract data type of the LOB attribute, plug-in columns, and a plug-in index that are in the RDAREA to be re-initialized.

If the `KFPX14255-W` or `KFPX24242-W` message is issued during RDAREA re-initialization and the cause of the error is an invalid RDAREA status, see the *HiRDB Version 8 System Operation Guide* for details about increasing the size of the RDAREA and changing its attributes.

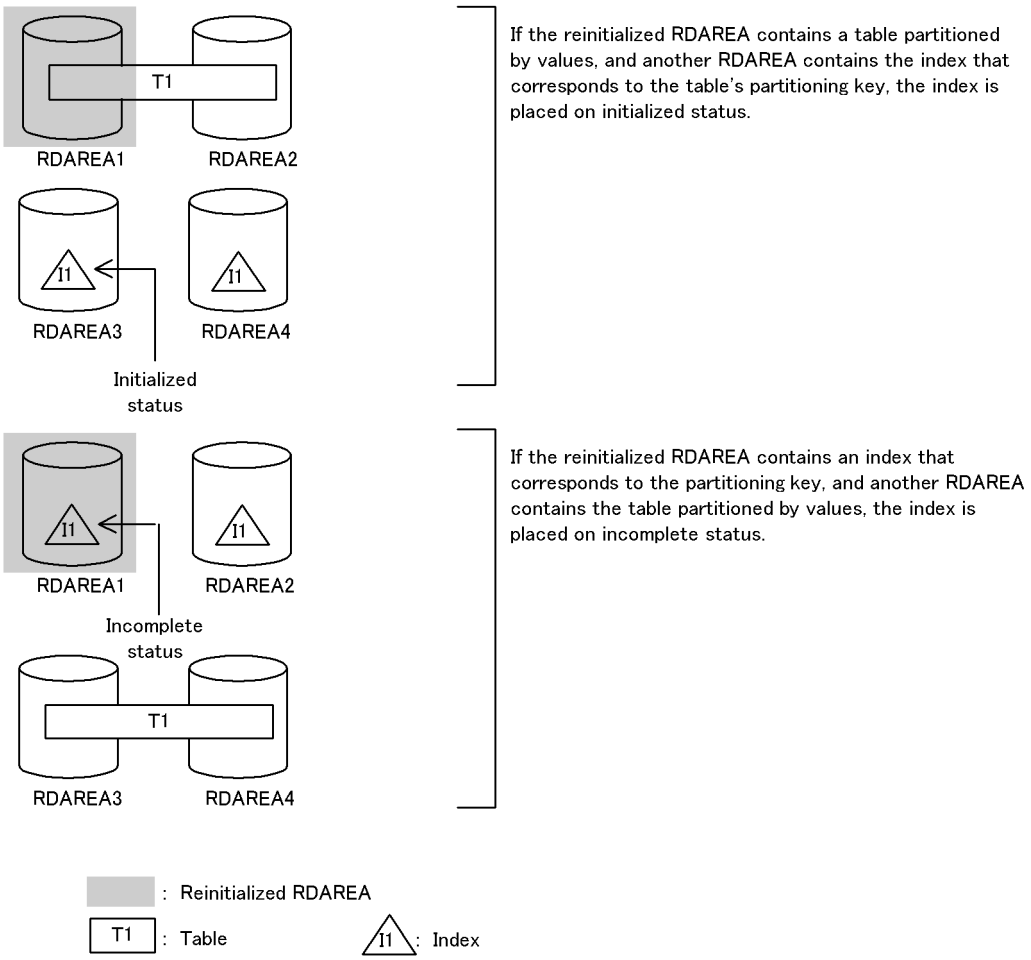
Figure 7-4: Table and index status after reinitialization (1/2)



Note

Initialized status means that there is a definition but there is no data; an index on this status is usable. *Incomplete status* means that the table and index no longer conform; an index on this status mode is not usable.

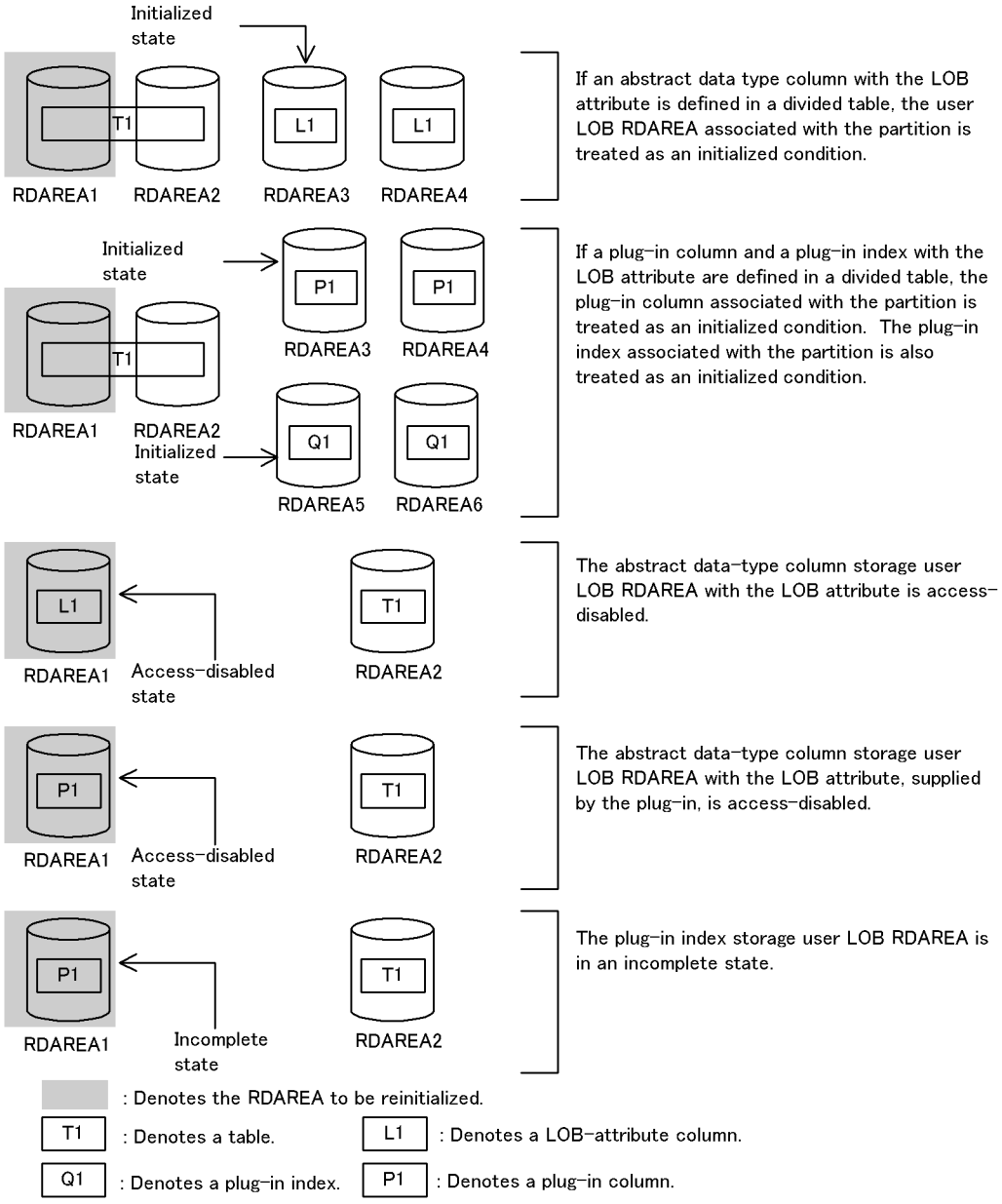
Figure 7-5: Table and index status after reinitialization (2/2)



Note

Initialized status means that there is a definition but there is no data; an index on this status is usable. *Incomplete status* means that the table and index no longer conform; an index on this status is not usable.

Figure 7-6: Conditions of a reinitialized table, columns of abstract data type of the LOB attribute, plug-in-supplied columns of abstract data type, and a plug-in index



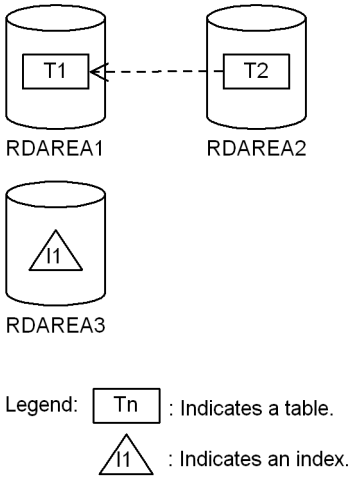
Note

Initialized state refers to a condition in which an index is defined without data, so that the index can be used. *Access-disabled state* refers to a condition in which data has been deleted from the index, so that the index consists solely of a definition. In this condition, the index cannot be used. *Incomplete state* refers to a condition in which the table and the index are mismatched, so that the index cannot be used.

If referential constraints have been defined for a table stored in the RDAREA that is to be re-initialized, the check pending status is changed. The check pending status is managed by the data dictionary tables (`CHECK_PEND` columns in the `SQL_TABLES` and `SQL_REFERENTIAL_CONSTRAINTS` tables) and by the table information in the RDAREA. For details about the check pending status, see the manual *HiRDB Version 8 Installation and Design Guide*.

The following examples show the check pending status changes when `USE` is specified in the `pd_check_pending` operand in the system definition and RDAREAs are re-initialized.

1. When there is an RDAREA that stores only indexes



When `T2` (referencing table) references `T1` (referenced table) and `RDAREA3` stores only the index for the primary key of `T1`, the check pending status is changed during re-initialization as follows:

RDAREA to be re-initialized	Check pending status of T2 (referencing table)	
	Data dictionary table	Table information in RDAREA2
`RDAREA1`	P	P

RDAREA to be re-initialized	Check pending status of T2 (referencing table)	
	Data dictionary table	Table information in RDAREA2
RDAREA2	N	N
RDAREA3	--	--

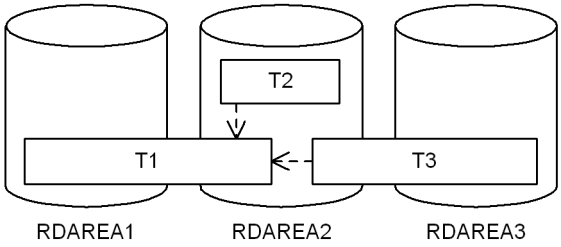
Legend:

P: Check pending status is set.

N: Check pending status is released.

--: Check pending status remains unchanged.

2. For a row-partitioned table



Legend: Tn : Indicates a table.

When T2 (referencing table) and T3 (referencing table) both reference T1 (referenced table), and when T1 and T3 are both row-partitioned tables, the check pending status is changed during re-initialization as follows:

RDAREA to be re-initialized	Check pending status of T2 (referencing table)		Check pending status of T3 (referencing table)		
	Data dictionary table	Table information in RDAREA2	Data dictionary table	Table information in RDAREA2	Table information in RDAREA3
RDAREA1	P	P	P	P	P
RDAREA2	N	N	P	N	P
RDAREA3	--	--	--	--	N

Legend:

P: Check pending status is set.

N: Check pending status is released.

--: Check pending status remains unchanged.

7.5.5 Output of execution results listing

When an RDAREA is re-initialized, an execution results listing is output. The following shows an example of an execution results listing (table and index information):

```
                                 1          2          9
pdmod -information-             yyyy-mm-dd  hh:mm:ss   dbl
 initialize rdarea name = 3 USER_PUBLIC_1
  initialized table
  4 USERA.STOCK
  USERA.EMPLOYEE
  initialized index           rdarea name
  5 USERA.EMPLOYEENO         6 USER_PUBLIC_1
     USERA.PNAME               USER_PUBLIC_2
  unfinished index           rdarea name
   7 USERA.NAME              8 USER_PUBLIC_1
 initialize LOB data(table name)  rdarea name
  10 USERA.STOCK            11 USER_LOB_1
```

Explanation

1. Date pdmod was executed

   yyyy-mm-dd

   year month date

2. Time pdmod was executed

   hh:mm:ss

   hour minute second

3. Name of reinitialized RDAREA

4. Table identifier in reinitialized RDAREA

   authorization-identifier.table-identifier

5. Initialized index

   authorization-identifier.index-identifier

6. Name of RDAREA from which index data was deleted

7. Index placed in incomplete status (unusable status) after the index data was deleted

8. RDAREA name of index placed in incomplete status (unusable status)

9. HiRDB identifier
10. Table identifier of LOB column structure base table
11. Name of user LOB RDAREA

7.5.6 Notes

1. After reinitializing an RDAREA, you should make a backup copy of the expanded RDAREA, the master directory RDAREA, and the corresponding data dictionary RDAREA, using the database copy utility (`pdcopy`). For details about the database copy utility, see Chapter *18. Database Copy Utility (pdcopy)*.
2. If an error occurs while an RDAREA is being reinitialized, the system does not restore the RDAREA. In this case, you must restore it, because its HiRDB files may have already been deleted or the database may have become corrupted during initialization.
3. When reinitialization of an RDAREA terminates normally, the RDAREA is placed in command shutdown and closed status. Before using such an RDAREA, you need to use an operation command to place it in shutdown release and open status.
4. If you change the number of HiRDBs or rename the HiRDBs, the system deletes the HiRDB files that are no longer used.
5. The following describes the data stored in the RDAREA, the definition information, and the user's action after re-initialization.

   RDAREA containing a table

   All data in the RDAREA is deleted and only the table definition information is inherited. If the table is partitioned, only the data stored in the RDAREA subject to reinitializing is deleted. If the table has an index, the index key data corresponding to the deleted table data is also deleted, so that only the definition information is inherited. If the table is partitioned but its index does not correspond to the partitioning key, the index is placed in the unfinished status.

   If the table contains a LOB column, the corresponding LOB data is also deleted.

   If the table contains an abstract data type column, the system treats it as a regular table. However, if the table contains an abstract data type column with the LOB attribute, the system also deletes the data for the abstract data type column in the corresponding user LOB RDAREA.

   If the table has a plug-in index, the system also deletes the data for the plug-in index and inherits only the definition information.

   If the indexes and LOB columns of the table stored in the RDAREA that is

subject to re-initialization are stored in another RDAREA, the RDAREA containing the indexes and LOB columns must be in shutdown release and open status.

- User's action:

  After the reinitialization, restore the data using the database load utility, database reorganization utility, or a UAP. Note that a UAP cannot access the database if there is an unfinished index (a `KFPA11879-E` error). If this happens, take one of the following actions:

  • Either use the database reorganization utility (`pdrorg`) to reorganize the table with the index defined (batch index creation mode), or re-create the index.

  • Delete the table and the index data by initializing the table with the index defined (by executing the `PURGE TABLE` SQL statement).

RDAREA containing an index

All key data for the index is deleted from the RDAREA. If the data corresponding to the index exists in other RDAREAs, the index is placed in the unfinished status.

- User's action:

  If the index is successfully initialized after the reinitialization process, you can use the index as is. If the index is in the unfinished status, take one of the following actions:

  • Either use the database reorganization utility (`pdrorg`) to reorganize the table with the index defined (batch index creation mode), or re-create the index.

  • Delete the table and the index data by initializing the table with the index defined (by executing the `PURGE TABLE` SQL statement).

  • Reinitialize the RDAREA that contains the table with the index defined. In this case, the RDAREA containing the index must be in the shutdown release and open status.

User LOB RDAREA

All LOB data in the RDAREA is deleted. However, data in the LOB column structure base table associated with this LOB data is retained as is. If data in the LOB column structure base table is retained, the LOB column is treated as data with a length of 0.

- User's action:

  After the reinitialization, restore the database using the database load utility, the database reorganization utility, or a UAP.

User LOB RDAREA storing a column of abstract data type with the LOB attribute

All data in the RDAREA is deleted and the area becomes access-disabled. If a retrieval is made from the abstract data type column structure base table with the LOB attribute, the error message KFPA11891-E is output. This is because the abstract data type column structure base table with the LOB attribute (the part of a table containing an abstract data type from which all abstract data type data is removed) associated with the data remains undeleted.

- User's action:

  To access the storage user LOB RDAREA for the access-disabled abstract data type column with the LOB attribute, take one of the following actions:

  • Initialize the abstract data type column structure base table with the LOB attribute (by executing the PURGE TABLE SQL statement) to delete data.

  • Execute the database load utility (pdload) in the creation mode (-d).

  • Reinitialize the abstract data type column structure base table storage RDAREA with the LOB attribute. In this case, the abstract data type storage RDAREA with the LOB attribute must be in the shutdown release state and open status.

User LOB RDAREA storing a plug-in index

All data in the RDAREA is deleted. After the reinitialization, the plug-in index is placed in the unfinished status. If an attempt is made to access an unfinished plug-in index, the error message KFPA11879-E is output.

- User's action:

  Restore the plug-in index in the unfinished status by executing the database reorganization utility specifying the index re-creation option.

6. If a KFPX24231-W message is issued during the reinitialization of a registry RDAREA, and the registry LOB RDAREA is not reinitialized, you need to also reinitialize the registry LOB RDAREA. After the reinitialization, you must re-register the registry information used by the plug-in.
7. If you have created a list on the basis of a table stored in the RDAREA subject to reinitialization, re-create the list.
8. If the RDAREA to be re-initialized stores a referenced table, referencing table, or check constraint table, the check pending status is changed. The check pending status is managed by the data dictionary tables and the table information in the RDAREA. During re-initialization of the RDAREA, the check pending status is changed as described below. For details about the check pending status, see the manual *HiRDB Version 8 Installation and Design Guide*.

- When an RDAREA storing a referenced table (a table in which a primary key has been defined) is re-initialized, any table referencing that table (that is, a table in which a foreign key referencing that primary key has been defined) in other RDAREAs is placed in check pending status. If the value of the `pd_check_pending` operand in the system definition is `NOUSE`, the tables are not placed in check pending status. When an RDAREA to be re-initialized stores only indexes for the primary key, the check pending status remains unchanged.

  If the system was unable to place the referencing table in check pending status, it displays the `KFPX24242-W` message. In such a case, possible causes are as follows:

  - The RDAREA storing the referencing table is in shutdown status.
  - The referencing table to be placed in check pending status is being accessed by another program.
  - Another I/O error or RDAREA access error occurred.

  If this message is displayed, eliminate the cause of the error and set the check pending status forcibly.

- If an RDAREA storing a referencing table is re-initialized and the referencing table is in check pending status, the check pending status is released. However, in the following cases, only the table information in the RDAREA is released from check pending status; the data dictionary table is not released from check pending status:

  - The referencing table is partitioned.
  - The value of `pd_check_pending` in the system definition is set to `NOUSE`.

  If the data dictionary table has not been released from check pending status, use `pdconstck` to perform integrity checking and then release the data dictionary table from check pending status.

- If an RDAREA storing a check constraint table is re-initialized and the table for which check constraints have been defined is in check pending status, the check pending status is released. However, in the following cases, only the table information in the RDAREA is released from check pending status; the data dictionary table is not released from check pending status:

  - The table for which the check constraints have been defined is partitioned.
  - The value of `pd_check_pending` in the system definition is set to `NOUSE`.

  If the data dictionary table has not been released from check pending status, use `pdconstck` to perform integrity checking and then release the data

dictionary table from check pending status.

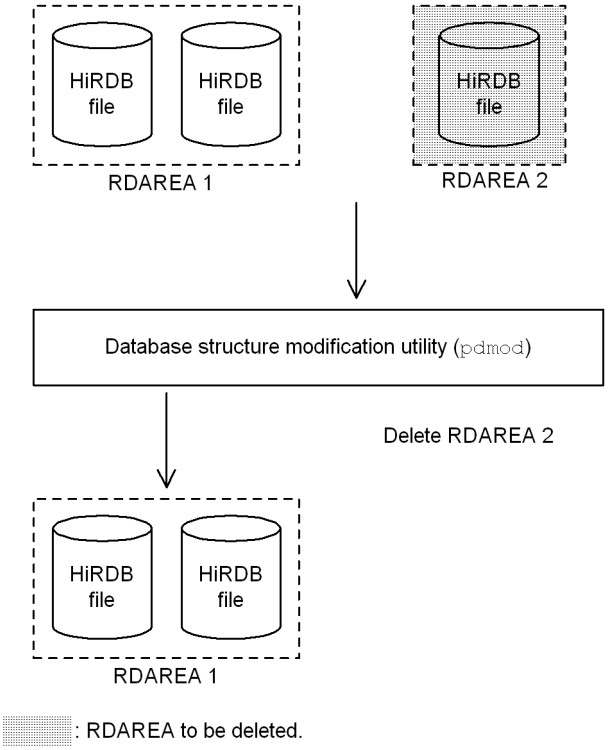
9. Do not re-initialize an RDAREA that contains a table that is subject to data extraction during data linkage. If such an RDAREA is re-initialized, subsequent HiRDB Datareplicator linkage becomes invalid.
10. An RDAREA containing a falsification prevented table cannot be re-initialized.

7.6 Deleting an RDAREA

7.6.1 Function

This function deletes RDAREAs that are no longer needed. An RDAREA can be redefined by first deleting it with this function and then adding a newly defined RDAREA. Figure 7-7 provides an overview of the use of the database structure modification utility (pdmod) for RDAREA deletion.

Figure 7-7: Overview of database structure modification utility (pdmod): RDAREA deletion



Prerequisites

1. An RDAREA that stores a table, index, or LOB column cannot be deleted.
2. An RDAREA subject to deletion must be in the shutdown and closed status. A shared RDAREA must be in shutdown and closed status at all back-end servers.

7.6.2 Example

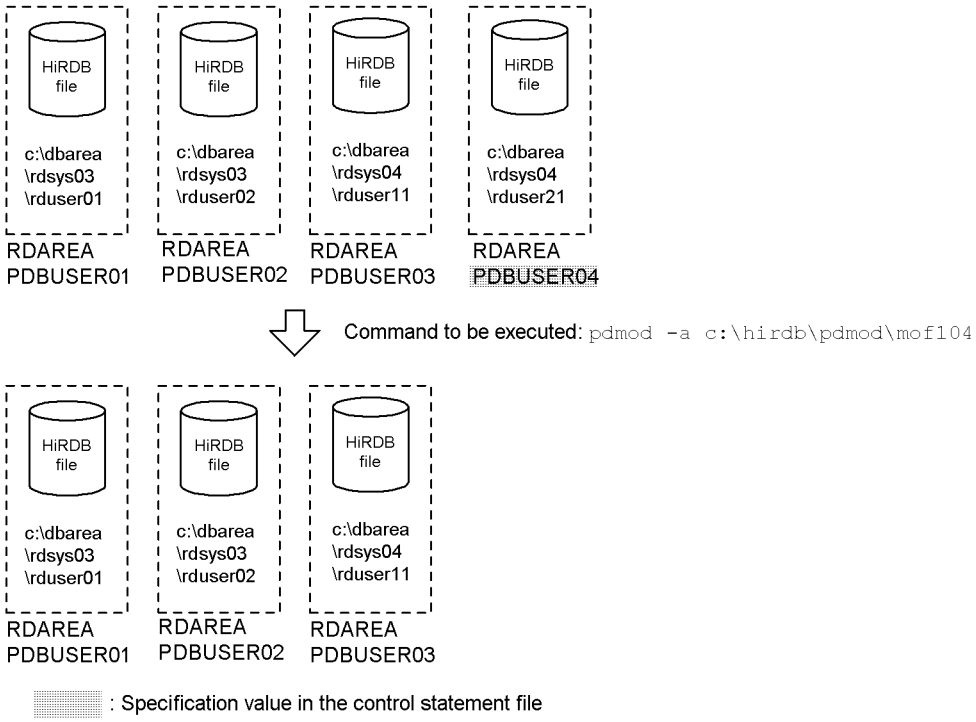
This section presents an example of using the database structure modification utility (deleting an RDAREA).

(1) Deleting an RDAREA

This example deletes the following RDAREA:

- PDBUSER04

Overview



Contents of the control statement file (c:\hirdb\pdmod\mofl04)

```
/* Deleting a user RDAREA */
remove rdarea PDBUSER04 ;
```

Explanation

Specifies the name of the RDAREA to be deleted.

7.6.3 Control statement (remove rdarea statement)

This section explains the operand of the `remove rdarea` statement. In the following table, the number corresponds to the explanation of the operand.

No.	Operand
1	`remove rdarea` *RDAREA-name*`;`

(1) RDAREA-name ～ <identifier> ((1-30))

Specifies the name of the RDAREA that is to be deleted. `ALL` cannot be specified as the RDAREA name. If the RDAREA name is enclosed in double quotation marks (`"`), it is handled as case sensitive; otherwise, it is handled as all uppercase letters. An RDAREA name must be enclosed in double quotation marks if it includes a blank.

An RDAREA that stores a table, index, or LOB column cannot be deleted.

7.6.4 Notes

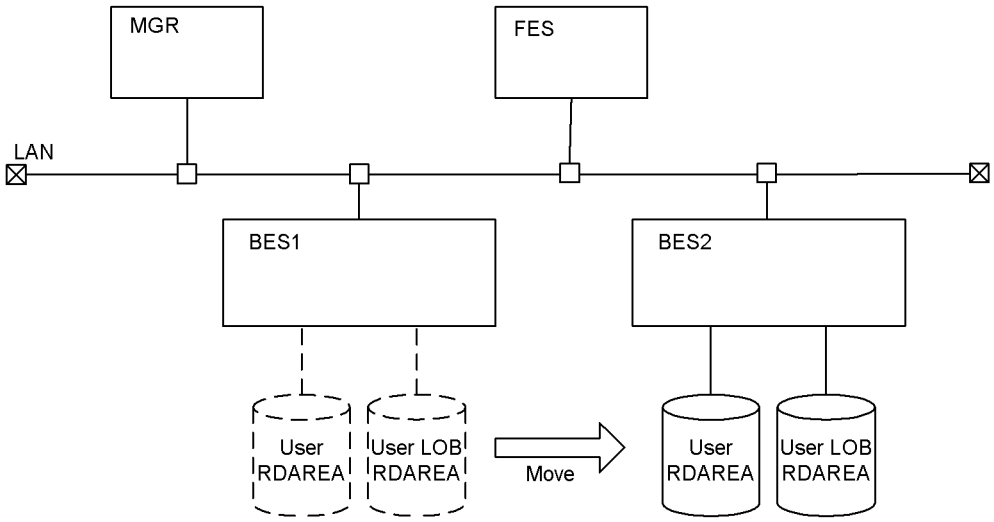
1. If you delete an RDAREA, the HiRDB files used in that RDAREA are also deleted.
2. If an error occurs during RDAREA deletion process, the RDAREA subject to deletion is not restored. In this case, re-execute the `remove rdarea` statement on the corresponding RDAREA because the HiRDB files may have already been deleted.
3. Once an RDAREA has been deleted, you need to modify the `pdbuffer` operand in the system common definitions after HiRDB terminates normally. Otherwise, more memory is used than is actually needed. For details about the system common definitions, see the *HiRDB Version 8 System Definition*.
4. If a data dictionary RDAREA containing a database state analyzed table and a database management table is deleted, the database state analyzed table and database management table are also deleted.

7.7 Moving an RDAREAs

7.7.1 Function

For a HiRDB/Parallel Server, this function moves RDAREA to another server. Figure 7-8 provides an overview of moving RDAREAs.

Figure 7-8: Overview of moving RDAREAs



Prerequisites

1. The RDAREAs to be moved must be in shutdown and closed status. A shared RDAREA must be in shutdown and closed status at all back-end servers.
2. Before moving RDAREAs, use `pdcopy` to make a backup of the RDAREAs to be moved, master directory RDAREA, and data dictionary RDAREA.
3. Before moving RDAREAs, you should check and, if necessary, revise the resource requirements for the target server.
4. When an RDAREA is moved, an error results unless all its related RDAREAs are also moved. Related RDAREAs include the following:
   - For a non-partitioned table, related RDAREAs are all RDAREAs that contain the table, indexes, and LOB data.
   - For a row-partitioned table, related RDAREAs are all RDAREAs that contain the tables corresponding to the partitioning conditions, indexes, and LOB data.

If there is any non-partitioning key index (including a non-partitioning key index used for partitioning in a server), you must delete it. After you finish moving an RDAREA, re-create the non-partitioning key index. If `UNIQUE` is specified for the non-partitioning key index, you may not be able to re-create it. For details, see *UNIQUE* in *CREATE INDEX* in the manual *HiRDB Version 8 SQL Reference*.

If a primary key has been defined other than the non-partitioning keys, you must delete the target table after saving its data, re-define the table after moving the RDAREA, and then recover the data. Note that if a primary key has been defined other than the partitioning keys, you may not be able to re-define the table due to the unique index.

7.7.2 Example

This section explains an example of using the database structure modification utility (moving RDAREAs).

(1) Moving RDAREAs

This example moves the following RDAREAs to back-end server `bes2`:

- `PDBUSER01`, `PDBUSER02`, `PDBULOB01`

Command to be executed

```
pdmod -a c:\hirdb\pdmod\mofl01
```

Contents of the control statement file (`c:\hirdb\pdmod\mofl01`)

```
/* Moving RDAREAs */
 move rdarea PDBUSER01,PDBUSER02,PDBULOB01   .......1
             to bes2 ;   ..........................2
```

Explanation

1. Specifies the names of the RDAREAs to be moved.
2. Specifies the name of the target back-end server.

7.7.3 Control statement (move rdarea statement)

This section explains the operands of the `move rdarea` statement. In the following table, the number corresponds to the explanation of the operand.

No.	Operand
1	`move rdarea` *RDAREA-name*`[,`*RDAREA-name*`]...`
2	`to` *server-name*
—	`;`

(1) RDAREA-name

～ <identifier> ((1-30))

Specifies the name of an RDAREA that is to be moved.

If multiple RDAREAs are specified, they must all be in the same server.

You cannot specify `ALL` for the RDAREA name. The system treats an RDAREA name enclosed in double quotation marks (`"`) as case sensitive; otherwise, the system treats the RDAREA name as all uppercase letters. You must enclose an RDAREA name in double quotation marks if it contains a space.

(2) to server-name

～ <identifier> ((1-8))

Specifies the name of the target server (back-end server).

An error results if the specified server name is not for a back-end server or the server does not exist in HiRDB.

7.7.4 Output of execution results listing

When RDAREAs are moved, an execution results listing is output.

The following is an example of the execution results listing that is output:

```
                              1          2          3
pdmod -information-           yyyy-mm-dd  hh:mm:ss     db1
                        4       5
   move rdarea from bes1 to bes2
     rdarea name = RDUSER1 6
       table name
      7 USERA.STOCK
         USERA.EMPLOYEE
       index name
      8 USERA.EMPLOYEE_NUMBER
         USERA.PRODUCT_NAME
       LOB data(table name)
      9 USERA.STOCK
```

Explanation

1. pdmod execution date

   yyyy-mm-dd (*year-month-date*)

2. pdmod execution time

   hh:mm:ss (*hour:minute:second*)

3. HiRDB identifier
4. Name of the source server
5. Name of the target server
6. Name of an RDAREA that was moved
7. Table contained in the moved RDAREA

   authorization-identifier.table-identifier

8. Index or plug-in index contained in the moved RDAREA

   authorization-identifier.index-identifier

9. LOB column or column of abstract data type of the LOB attribute contained in the moved RDAREA

   authorization-identifier.table-identifier

7.7.5 Notes

1. Once you start moving RDAREAs, do not execute any UAP or utility until the HiRDB restart has been completed. If you do not restart HiRDB after moving RDAREAs, operation cannot be guaranteed.
2. You must delete the HiRDB file that constitutes the moved source RDAREA and create the HiRDB file that constitutes the target RDAREA. The path name of the HiRDB file that constitutes the target RDAREA must be the same as the source HiRDB file.
3. Once an RDAREA has been moved, any routine using a table in the source RDAREA will be disabled. You must re-create such routines with the ALTER ROUTINE SQL statement.
4. After moving RDAREAs, use pdcopy to make a backup of the moved RDAREAs, master directory RDAREA, and data dictionary RDAREA.

7.8 Modifying data dictionary table attribute definitions

7.8.1 Function

This function lets you modify a data dictionary table's attribute definitions. The data dictionary table attribute definitions include:

- Dictionary table referencing privilege
- Use of mixed-character data in dictionary tables

(a) Dictionary table referencing privilege

The information that can be retrieved by accessing a dictionary table can be restricted based on the user's privilege. The specification of the `dicinf` operand in the database initialization utility (`define system` statement) can be modified.

(b) Use of mixed-character data in dictionary tables

This function enables mixed-character data to be used in dictionary tables. If `nouse` (mixed-character data not permitted in dictionary tables) is specified in the `dictionary datatype mchar` operand of the database initialization utility, the specification can be modified so that mixed-character data can be used.

7.8.2 Example

This section presents an example of using the database structure modification utility (modifying data dictionary table attribute definitions).

(1) Modifying data dictionary table attribute definitions

When data dictionary tables are accessed, this example limits the information available to users according to users' privileges. It allows mixed character string data in the data dictionary table. This example assumes that a HiRDB/Single Server is used.

Command to be executed

```
pdmod -a c:\hirdb\pdmod\mofl01
```

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
/* Modifying data dictionary table attribute definitions */
alter system dicinf limited   .....................1
        dictionary datatype mchar use ;   .........2
```

Explanation

1. When data dictionary tables are accessed, this statement limits the information available to users as appropriate to the users' privileges.
2. This operand enables mixed character string data to be used in the data dictionary.

7.8.3 Control statement (alter system statement)

This section explains the operands of the `alter system` statement. In the following table, each number corresponds to the number assigned to each operand.

No.	Operand
1	`alter system [dicinf {unlimited|limited}]`
2	`[dictionary datatype mchar use]`
—	`;`

(1) dicinf {unlimited|limited}

Specifies whether or not to limit the information that can be retrieved by accessing data dictionary tables based on the user's privileges (specification of the reference privilege for data dictionary tables). For details about how to specify the reference privilege for data dictionary tables, see the *HiRDB Version 8 System Operation Guide*.

`unlimited`

All information can be retrieved when a data dictionary table is accessed.

`limited`

The information that can be retrieved when a data dictionary table is accessed is to be restricted based on the user's privileges (users with DBA privilege can retrieve all information; users without DBA privilege can retrieve only the information available to their assigned privilege).

Rules

1. If a view table has been defined that references a data dictionary table, or an SQL object routine has been defined, you cannot change the data dictionary table reference privileges.
2. To modify the data dictionary table reference privileges, you need to delete all view tables that reference the data dictionary table beforehand.
3. If an SQL object routine has been defined that references data dictionary tables, the system issues a warning message indicating that the SQL object will be disabled, then terminates the processing normally. In this case, you need to re-create the routine's disabled SQL

object using ALTER PROCEDURE or ALTER ROUTINE. You can identify the disabled SQL object by searching the data dictionary table. For details about ALTER PROCEDURE or ALTER ROUTINE, see the *HiRDB Version 8 SQL Reference*. For details about how to search a data dictionary table, see the *HiRDB Version 8 UAP Development Guide*.

4. When limited is specified, there is an increase in the number of UAP access tables used to retrieve data dictionary tables. Therefore, you must increase the number of concurrently accessible base tables (value of pd_max_access_tables in the system common definition). Five base tables must be accessible concurrently for each data dictionary table.

(2) dictionary datatype mchar use

Specifies that mixed-character data can be used in dictionary tables. When this operand is specified, the data type of dictionary table columns becomes MVARCHAR.

Criteria

If Kanji characters are used in a table name or column names, the data type is changed to MVARCHAR.

Rules

1. If a view table is defined that references data dictionary tables, you cannot specify this operand. To specify this operand, you need to delete such view tables beforehand.
2. If an SQL object routine has been defined that references data dictionary tables, the system issues a warning message indicating that the SQL object will be disabled, then terminates the processing normally. In this case, you need to re-create the routine's disabled SQL object using ALTER PROCEDURE or ALTER ROUTINE. You can identify the disabled SQL object by searching the data dictionary table. For details about ALTER PROCEDURE or ALTER ROUTINE, see the *HiRDB Version 8 SQL Reference*. For details about how to search a data dictionary table, see the *HiRDB Version 8 UAP Development Guide*.

7.8.4 Note

An error results if the alter system statement is executed while a UAP, utility, or operation command is accessing a data dictionary table.

7.9 Modifying RDAREA attributes

7.9.1 Function

This function modifies the following RDAREA attributes:

- RDAREA open timing
- Automatic extension of RDAREA

Prerequisites

An RDAREA subject to attribute modification must be in the shutdown and closed status.

7.9.2 Example

This section presents an example of using the database structure modification utility (modifying RDAREA attributes).

(1) Modifying RDAREA attributes

For a HiRDB/Single Server, this example modifies the open timing of the following RDAREAs:

- `PDBUSER01` (user RDAREA)
- `PDBULOB11` (user LOB RDAREA)

Command to be executed

```
pdmod -a c:\hirdb\pdmod\mofl01
```

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
/* Modifying RDAREA attributes */
alter rdarea PDBUSER01 open attribute SCHEDULE ;
alter rdarea PDBULOB11 open attribute SCHEDULE ;
```

Explanation

This command changes the RDAREAs' (`PDBUSER01` and `PDBULOB11`) open timing to `SCHEDULE`.

7.9.3 Control statement (alter rdarea statement)

This section explains the operands of the `alter rdarea` statement. In the following table, each number corresponds to the number assigned to each operand.

No.	Operand
1	`alter rdarea` *RDAREA-name*
2	`open attribute {INITIAL|DEFER|SCHEDULE}]`
3	`[extension {use` *extension-segments-count* `segments|nouse}]`
—	`;`

(1) RDAREA-name ～ <identifier> ((1-30))

Specifies the name of the RDAREA whose open timing is to be changed. `ALL` cannot be specified for RDAREA-name. If the RDAREA name is enclosed in double quotation marks (`"`), it is handled as case sensitive; otherwise, it is handled as all uppercase letters. An RDAREA name must be enclosed in double quotation marks if it includes a blank.

(2) open attribute {INITIAL|DEFER|SCHEDULE}

Specifies the RDAREA open timing.

Criteria

Normally, specify `INITIAL` unless some special operation mode is used.

To reduce the HiRDB startup time when there are many RDAREAs, specify `DEFER`.

Rules

1. This operand takes effect only when `Y` is specified in the `pd_rdarea_open_attribute_use` operand in the system common definitions.
2. This operand is applicable to user RDAREAs, user LOB RDAREAs, and list RDAREAs. This operand is ignored if specified for any other type of RDAREA (the RDAREA open timing for all other RDAREA types is `INITIAL`).
3. When this operand is omitted, the system assumes the value existing before the RDAREA attributes were modified.
4. The value of this operand takes effect at the next startup, regardless of the HiRDB startup mode.
5. If the HiRDB file system area assigned to the corresponding RDAREA is shared by other RDAREAs, all such RDAREAs must have the same open timing. Otherwise, an expected result may not be obtained from the RDAREAs sharing the HiRDB file system area.

INITIAL

The RDAREA's HiRDB file system area is to be opened at the time of system startup and the RDAREA information is to be made resident in memory; thus, high-speed operation is available from the first SQL.

Rules

1. The RDAREA's initial status at system startup is open and remains unchanged until an operation command is executed, unless it is placed in error shutdown status due to an error.
2. The INITIAL open attribute does not permit access to a closed RDAREA.

DEFER

The RDAREA's HiRDB file system area is not to be opened at the time of system startup; rather, it is to be opened and its RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed. Thus, high-speed operation is available beginning with the second access to the HiRDB file system area.

Rules

1. The RDAREA's initial status at system startup is closed and changes to open the first time an RDAREA in its HiRDB file system area is accessed. The HiRDB file system area's RDAREAs remain in open status thereafter until an operation command is executed, unless they are placed in error shutdown status due to an error.
2. The DEFER open attribute permits access to a closed RDAREA.

SCHEDULE

The RDAREA's HiRDB file system area is not to be opened at the time of system startup; rather, it is to be opened and the RDAREA information is to be made resident in memory the first time an RDAREA in the HiRDB file system area is accessed from within a transaction. When this attribute is used, transaction workload increases because a HiRDB file system area opened by a transaction is closed when the transaction terminates, and open processing must be performed on the HiRDB file system area the first time each transaction accesses one of its RDAREAs.

Rules

1. The RDAREA's initial status at system startup is closed and changes to open when the RDAREA is first accessed by a transaction. When the transaction is terminated, all the RDAREAs opened by it are closed. You can use the pdopen command to keep RDAREAs in open status until they are placed in shutdown close status. You can also use other

operation commands to change the RDAREA status. However, if an error is detected, RDAREAs are placed in error shutdown status.

2. The SCHEDULE open attribute permits access to a closed RDAREA.

(3) extension {use extension-segments-count segments|nouse}

Specifies whether or not to apply automatic extension to the RDAREA for which attributes are being modified. When this operand is omitted, the value existing before attribute modification takes effect.

Automatic extension of RDAREA is a facility that extends automatically an RDAREA in the event of a shortage of RDAREA space if there is enough space in the HiRDB file system area. If you use this facility for an RDAREA and a shortage of unused segments occurs, the system allocates new unused segments to the RDAREA. These new unused segments are added at the end of the HiRDB file constituting the RDAREA.

Prerequisites

1. You need to specify the -e option (specifying the number of extensions) for the HiRDB file system area containing the RDAREA.
2. There must be enough space in the HiRDB file system area that contains the last HiRDB file constituting the RDAREA.

use *extension-segments-count* segments

Specifies that automatic extension of RDAREA is to be applied.

The automatic extension of RDAREA is applicable to data dictionary RDAREAs, data dictionary LOB RDAREAs, user RDAREAs, and user LOB RDAREAs. For any other type of RDAREA, the system assumes nouse, even if use is specified.

For an RDAREA other than a LOB RDAREA, automatic extension occurs when there are no more used free segments or used segments. For a LOB RDAREA, automatic extension occurs when there are no more unused segments.

extension-segments-count ～ <unsigned integer> ((1-64000))

Specifies the number of extension segments.

The maximum number of HiRDB file extensions is 24. If this value is exceeded, an error occurs. The maximum number of extensions per HiRDB file system area is determined by the value specified when the HiRDB file system area is created. Therefore, you need to define the maximum number of extensions taking into account the number of files in the HiRDB file system area and the frequency of extension.

nouse

Specifies that automatic extension of RDAREA is not to be applied.

Notes

1. If allocation of unused segments fails due to a shortage of space in the HiRDB file system area, either extend or re-initialize the RDAREA or use the database reorganization utility to reorganize the table.
2. If the number of extensions exceeds the maximum value, integrate the extensions in the HiRDB file system area containing the RDAREA or add an HiRDB file in another HiRDB file system area to the RDAREA.

   To integrate extensions, make a backup copy with `pdfbkup`, initialize the HiRDB file system area with `pdfmkfs`, then restore the HiRDB file system area from its backup copy using `pdfrstr`.
3. The last file is locked from the beginning to the end of the automatic extension process.

7.10 Changing an RDAREA from a HiRDB/Single Server structure to a HiRDB/Parallel Server structure

7.10.1 Function

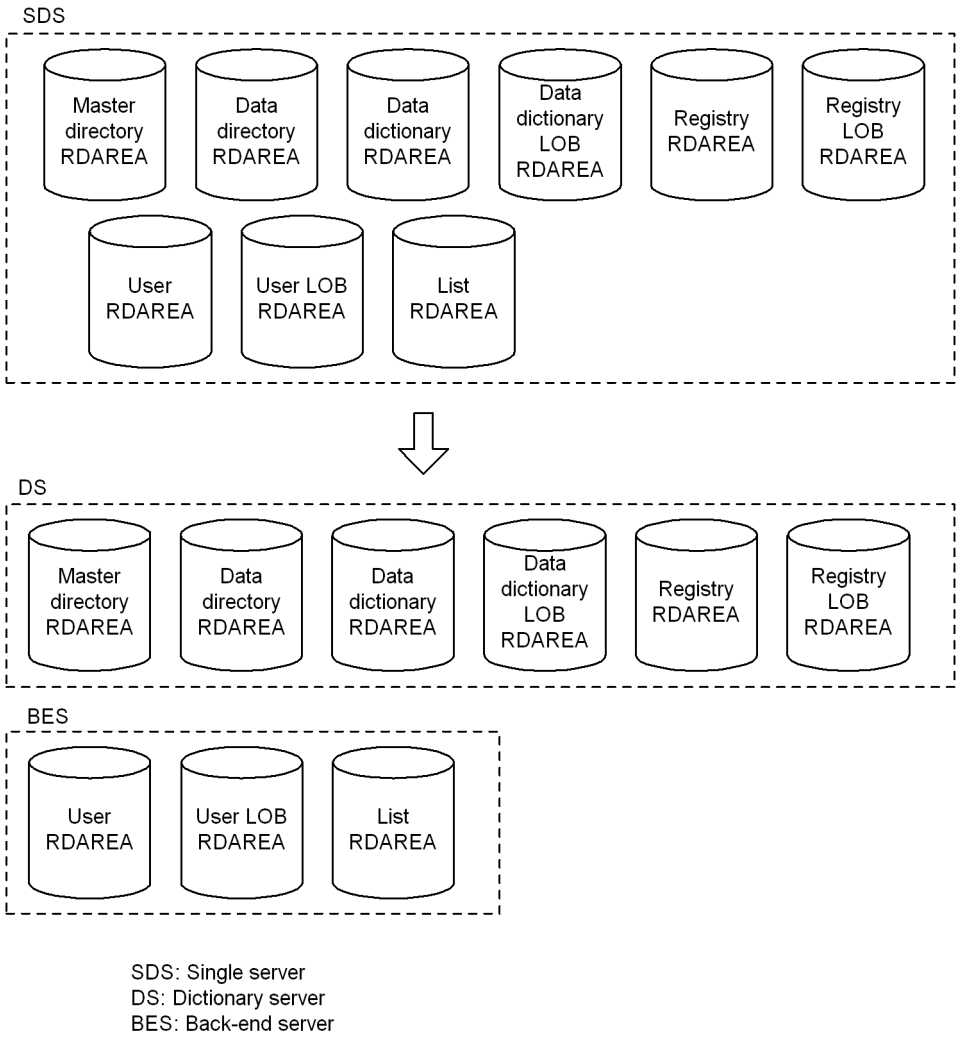
When a HiRDB system is converted from a HiRDB/Single Server to a HiRDB/Parallel Server, this function changes the RDAREAs in the HiRDB/Single Server structure to RDAREAs in the HiRDB/Parallel Server structure.

All RDAREAs are changed as follows:

- Master directory, data directory, data dictionary, and data dictionary LOB RDAREAs become data dictionary server RDAREAs in the HiRDB/Parallel Server
- User, user LOB, and list RDAREAs become back-end server RDAREAs in the HiRDB/Parallel Server

Figure 7-9 provides an overview of the use of the database structure modification utility (`pdmod`) to change RDAREAs from a HiRDB/Single Server structure to a HiRDB/Parallel Server structure.

Figure 7-9: Overview of database structure modification utility (pdmod): changing from HiRDB/Single Server structure to HiRDB/Parallel Server structure



SDS: Single server
DS: Dictionary server
BES: Back-end server

7.10.2 Example

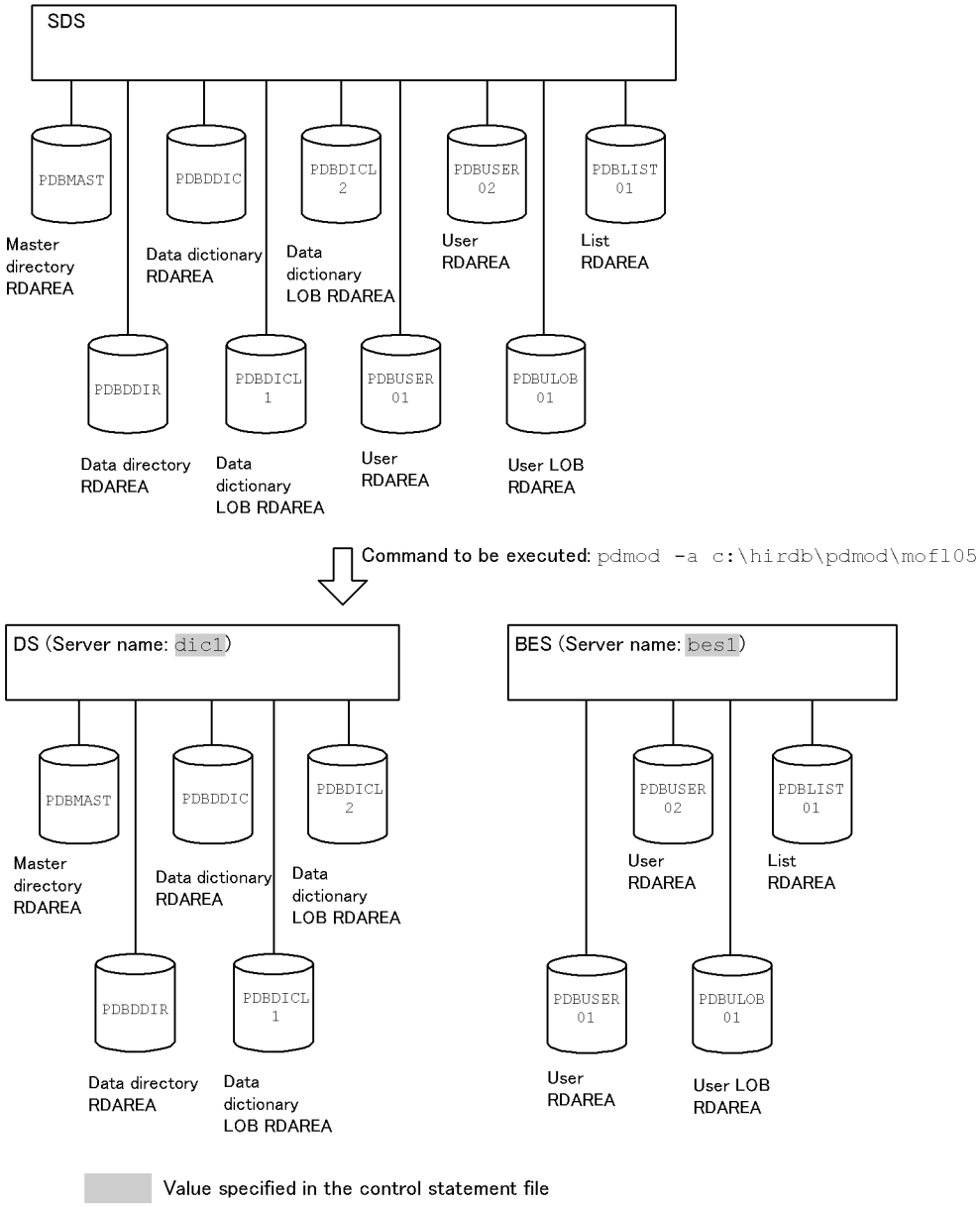
This section presents an example of using the database structure modification utility (changing an RDAREA from a HiRDB/Single Server structure to a HiRDB/Parallel Server structure).

(1) Changing RDAREAs from a HiRDB/Single Server structure to a HiRDB/Parallel Server structure

This example changes the following RDAREAs from a HiRDB/Single Server structure to a HiRDB/Parallel Server structure:

- PDBMAST
- PDBDDIR
- PDBDDIC
- PDBDICL1
- PDBDICL2
- PDBUSER01
- PDBUSER02
- PDBULOB01
- PDBLIST01

Overview


SDS
PDBMAST
PDBDDIC
PDBDICL 2
PDBUSER 02
PDBLIST 01
Master directory RDAREA
Data dictionary RDAREA
Data dictionary LOB RDAREA
User RDAREA
List RDAREA
PDBDDIR
PDBDICL 1
PDBUSER 01
PDBULOB 01
Data directory RDAREA
Data dictionary LOB RDAREA
User RDAREA
User LOB RDAREA
Command to be executed: pdmod -a c:\hirdb\pdmod\mofl05
DS (Server name: dic1)
BES (Server name: bes1)
PDBMAST
PDBDDIC
PDBDICL 2
PDBUSER 02
PDBLIST 01
Master directory RDAREA
Data dictionary RDAREA
Data dictionary LOB RDAREA
User RDAREA
List RDAREA
PDBDDIR
PDBDICL 1
PDBUSER 01
PDBULOB 01
Data directory RDAREA
Data dictionary LOB RDAREA
User RDAREA
User LOB RDAREA
Value specified in the control statement file

Contents of the control statement file (c:\hirdb\pdmod\mofl05)

```
/* Changing RDAREAs from a HiRDB/Single Server structure to a HiRDB/Parallel Server structure */
alter HiRDB mode to parallel
        dictionary server name dic1   ..............1
        server name bes1 others ;   ................2
```

Explanation

1. Name of the dictionary server after reconfiguration
2. Name of the back-end server after reconfiguration

7.10.3 Control statement (alter HiRDB mode to parallel statement)

This section explains the operands of the `alter HiRDB mode to parallel` statement. In the following table, each number corresponds to the number assigned to each operand.

No.	Operand
—	`alter HiRDB mode to parallel`
1	`dictionary server name` *server-name*
2	`server name` *server-name*
3	{*RDAREA-name*[,RDAREA-name]...| others}
—	[server name *server-name*
—	{*RDAREA-name*[,*RDAREA-name*]...| others}]...
—	;

(1) dictionary server name server-name ～ <identifier> ((1-8))

Specifies the name of the dictionary server after the RDAREAs are changed from a HiRDB/Single Server to a HiRDB/Parallel Server. Master directory, data directory, data dictionary, and data dictionary LOB RDAREAs are placed in the specified dictionary server.

(2) server name server-name ～ <identifier> ((1-8))

Specifies the name of the back-end server after the RDAREAs are changed from a HiRDB/Single Server to a HiRDB/Parallel Server. User, user LOB, and list RDAREAs are placed in the specified back-end server.

(3) {RDAREA-name[,RDAREA-name]... \|others}

RDAREA-*name*[,RDAREA-*name*]... ～ <identifier> ((1-30))

Specifies the names of the RDAREAs to be allocated to the specified back-end server. ALL cannot be specified as an RDAREA name. If an RDAREA name is enclosed in double quotation marks ("), it is handled as case sensitive; otherwise, it is handled as all uppercase letters. An RDAREA name must be enclosed in double quotation marks if it includes a blank.

others

Specifies that all RDAREAs are to be allocated to the specified back-end server.

7.10.4 Notes

1. After changing RDAREAs from HiRDB/Single Server structure to HiRDB/Parallel Server structure, be sure to terminate the HiRDB/Single Server normally.
2. If you partition a row-partitioned table into multiple back-end servers, the system does not partition its non-partitioning key indexes, resulting in a mismatch in the non-partitioning key indexes when the HiRDB/Parallel Server is started. Therefore, redefine such non-partitioning key indexes so that they are partitioned into multiple back-end servers.
3. If you are placing a dictionary server in another (added) server machine after migrating from HiRDB/Single Server to HiRDB/Parallel Server, move the master directory RDAREAs, data directory RDAREAs, data dictionary RDAREAs, data dictionary LOB RDAREAs, registry RDAREAs, and registry LOB RDAREAs to the server machine where the dictionary server is located before starting the HiRDB/Parallel Server.
4. If you are changing the server structure specifying RDAREAs, and the RDAREA for the table is different from the RDAREA for the table's indexes, place each RDAREA in the same server machine. However, you should delete a non-partitioned index of a row-partitioned table beforehand.
5. If a stored procedure or a user-defined stored function is registered, all such stored procedures and user-defined stored functions become disabled after the migration. For any stored procedures or user-defined stored functions that have been disabled after the migration, you need to re-create all SQL objects by starting the HiRDB/Parallel Server and executing ALTER PROCEDURE (with ALL specified) or ALTER ROUTINE. For details about ALTER PROCEDURE and ALTER ROUTINE, see the *HiRDB Version 8 SQL Reference*.
6. If the registry facility is used, re-register the registry manipulation stored procedure after the migration using the registry facility initialization utility (pdreginit) specifying -k renew. If you are using the following plug-in, re-register the plug-in by executing the pdplgrgst -a command:
   - HiRDB Image Search Plug in
7. During the migration, do not change the HiRDB identifier in the system common

definition `pd_system_id`.

8. While shared tables and shared indexes are defined, do not change RDAREAs from a HiRDB/Single Server configuration to a HiRDB/Parallel Server configuration. Before you can change the configuration, you must delete all shared tables and shared indexes defined in HiRDB.

7.11 Registering an auditor (security audit facility)

7.11.1 Function

For use of the security audit facility, this function registers a user with the audit privilege (auditor) into a HiRDB that is to be subject to security audits. For details about the security audit facility, see the *HiRDB Version 8 System Operation Guide*.

Prerequisites

1. You can register only one auditor per HiRDB. If an auditor has already been registered, an attempt to register another auditor results in an error.
2. For the auditor, specify either a new user or an existing user who does not have the DBA privilege. Registering an existing user who has the DBA privilege results in an error.
3. When an auditor is registered, a schema is also defined at the same time. Therefore, there is no need to execute CREATE SCHEMA before registering an auditor.

7.11.2 Example

This section explains an example of using the database structure modification utility (registering an auditor).

(1) Registering an auditor

Register the authorization identifier HIRAUD as an auditor

Command to be executed

```
pdmod -a c:\hirdb\pdmod\mofl01
```

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
create auditor for HIRAUD ;   ..............................1
```

Explanation

1. Registers the authorization identifier HIRAUD as an auditor.

7.11.3 Control statement (create auditor statement)

This section explains the operands of the `create auditor` statement. In the following table, the number corresponds to the explanation of the operand.

No.	Operand
1	`create auditor for` *authorization identifier*
-	`;`

(1) authorization-identifier

～ <string of alphanumeric symbols beginning with an alphabetic symbol> ((1-8))

Specifies the authorization identifier of the user who is to be registered as the auditor.

You must specify in this operand the authorization identifier of either a new user or an existing user who does not have the `DBA` privilege. Specifying a user who has the `DBA` privilege or a user who is in password-invalid account lock status results in an error.

To treat the authorization identifier as being case sensitive, enclose the entire authorization identifier in double quotation marks (`"`). If the authorization identifier is not enclosed in double quotation marks, it is treated as being all uppercase letters.

7.11.4 Notes

1. If you register a new user as the auditor, that user's authorization identifier is used as the password immediately after registration. If you register an existing user as the auditor and no password is set for that user, that user's authorization identifier is also used as the password immediately after registration. In either case, use the `GRANT` statement to change the password.
2. If you register a new user as the auditor while the Directory Server linkage facility is being used, you also need to register the user in the Directory Server.

7.12 Creating an audit trail table (security audit facility)

7.12.1 Function

This function creates the audit trail table that is required in order to use the security audit facility.

Prerequisites

1. Before creating an audit trail table, you must register an auditor.
2. You can create one audit trail table per HiRDB. If an audit trail table has already been created, an attempt to create another one will result in an error.
3. For the RDAREA that will store the audit trail table, you must specify a user RDAREA that can be accessed only by the auditor. An error results if you specify a shared RDAREA, public user RDAREA, or RDAREA that can be accessed by a user who is not the auditor.

7.12.2 Example

This section explains an example of using the database structure modification utility (creating an audit trail table).

(1) Creating an audit trail table

This example creates the audit trail table SQL_AUDIT_TRAIL in the RDAREA RD00AUD.

Command to be executed

```
pdmod -a c:\hirdb\pdmod\mofl01
```

Contents of the control statement file (c:\hirdb\pdmod\mofl01)

```
create audit table SQL_AUDIT_TRAIL   ..............................1
                in RD00AUD ;   ...................................2
```

Explanation

1. Creates the audit trail table SQL_AUDIT_TRAIL.
2. Creates the audit trail table in the RDAREA RD00AUD.

7.12.3 Control statement (create audit table statement)

This section explains the operands of the `create audit table` statement. In the following table, the number corresponds to the explanation of the operand.

No.	Operand
1	`create audit table` [*table-identifier*]
2	`in` *RDAREA-name*
-	`;`

(1) table-identifier

～ <identifier> ((1-30)) <<`SQL_AUDIT_TRAIL`>>

Specifies a table identifier for the audit trail table. If this information is omitted, `SQL_AUDIT_TRAIL` is assumed.

To treat the table identifier as being case sensitive, enclose the entire identifier in double quotation marks (`"`). If the identifier is not enclosed in double quotation marks, it is treated as being all uppercase letters.

You cannot specify `in` as the table identifier.

(2) RDAREA-name

～ <string of alphanumeric symbols, underscore, and space beginning with an alphabetic symbol> ((1-30))

Specifies the name of the RDAREA (user RDAREA) that is to store the audit trail table. This RDAREA must be accessible only by the auditor. An error results if the specified RDAREA can be accessed by a user who is not the auditor.

To treat the RDAREA name as being case sensitive, enclose the entire name in double quotation marks (`"`). If the name is not enclosed in double quotation marks, it is treated as being all uppercase letters.

7.13 Rules and notes

(1) Rules

(a) Utility execution

- The database structure modification utility (`pdmod`) can be executed only when HiRDB is running.
- The database structure modification utility (`pdmod`) should be executed on a server machine where the single-server or the system manager is located.

(b) Control statements

- Multiple control statements can be specified simultaneously. The database structure modification utility completes processing at each control statement. Should any control statement cause an error, the database structure modification utility continues with the processing by skipping that statement. The control statements that have been executed successfully are not rolled back, even when other control statements produced errors. Therefore, if an error occurs in the database structure modification utility, resolve the error based on standard error output and messages that are output to the message log, and re-execute only those control statements that caused an error.
- A comment can be provided in a control statement by enclosing it with a slash-asterisk (`/*`) and an asterisk-slash (`*/`).
- If you selected `utf-8` as the character encoding in the `pdntenv` command, you can use a control statements file that contains a BOM. However, only ASCII characters are permitted for comments in the control statements file. If character encoding other than ASCII is used, `pdinit` may not function correctly.

(2) Notes

(a) Utility execution results

The results of execution of the database structure modification utility can be verified by the following methods:

- `pddbls` command (with `-r ALL -a` specification)
- Retrieving the dictionary table
- Executing the database condition analysis utility

(b) In the event a HiRDB file system area is corrupted

If the execution of the database structure modification utility is interrupted due to a power failure or the `pdkill` command, in some cases the HiRDB file system area can be damaged. If a conflict is detected by executing the `pdfls` command, use one of the

following procedures to perform recovery:

- Use the `pdfmkfs` command to recreate the HiRDB file system area. For each RDAREA, execute the database structure modification utility on the HiRDB files that were stored.
- Execute the `pdfbkup` and `pdfrstr` commands to back up and restore the HiRDB file system area.

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)