

PRELIMINARY RELEASE



HiCommand™ Tuning Manager Agent for IBM® DB2® Universal Database™ Reference Guide

PRELIMINARY RELEASE

© 2005 Hitachi, Ltd., Hitachi Data Systems Corporation, ALL RIGHTS RESERVED

Notice: No part of this publication may be reproduced or transmitted in any form or by any electronic or mechanical means, including photocopying and recording, or stored in a database or retrieval system for any purpose, without the express written permission of Hitachi Data Systems Corporation (hereinafter referred to as "Hitachi Data Systems").

Hitachi Data Systems reserves the right to make changes to this document at any time without notice and assumes no responsibility for its use. Hitachi Data Systems products and services can only be ordered under the terms and conditions of Hitachi Data Systems' applicable agreements, including license agreements. All of the features described in this document may not be currently available. Refer to the most recent product announcement or contact your local Hitachi Data Systems sales office for information on feature and product availability.

This document contains the most current information available at the time of publication. When new and/or revised information becomes available, this entire document will be updated and distributed to all registered users.

Trademarks

Hitachi Data Systems is a registered trademark and service mark of Hitachi, Ltd., and the Hitachi Data Systems design mark is a trademark and service mark of Hitachi, Ltd. HiCommand is a trademark of Hitachi, Ltd.

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

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

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

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

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

Linux is a registered trademark of Linus Torvalds.

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

Microsoft SQL Server is a product name of Microsoft Corp.

ORACLE is a registered trademark of Oracle Corporation.

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

PRELIMINARY RELEASE

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

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

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

The following program products contain some parts whose copyrights are reserved by Sun Microsystems, Inc.: P-9D13-Y6312.

The following program products contain some parts whose copyrights are reserved by UNIX System Laboratories, Inc.: P-9D13-Y6312.

All other brand or product names are or may be trademarks or service marks of and are used to identify products or services of their respective owners.

Notice of Export Controls

Export of technical data contained in this document may require an export license from the United States government and/or the government of Japan. Please contact the Hitachi Data Systems Legal Department for any export compliance questions.

Document Revision Level

Revision	Date	Description
MK-95HC097-P	May 2005	Preliminary Release

Source Documents for this Revision

Following are the list of source files that were used to create this document:

- SSD-94HC061-Ed.2, 03.31.05, version 4.1
- HTM_DB2-review-form-f

Preface

The *HiCommand™ Tuning Manager Agent for IBM® DB2® Universal Database™ Reference Guide* is for use with the AIX and Linux 3 systems. It describes the functionality and records of HiCommand Tuning Manager Agent for IBM® DB2® Universal Database™, and it is intended for users who want to:

- learn about the functionality and records of HiCommand™ Tuning Manager Agent for IBM® DB2® Universal Database™
- set up a HiCommand™ Tuning Manager system, and operate that system to collect performance data for storage subsystems

In addition, this manual assumes that readers have a basic knowledge of SANs (storage area networks).

Software Version

This document revision applies to software version 4.1.

Document Conventions

Conventions for Storage Capacity Values

Storage capacity values displayed by HiCommand Tuning Manager Agent for IBM® DB2® Universal Database™ are calculated based on the following values:

- 1 KB (kilobyte) = 1,024 bytes
- 1 MB (megabyte) = 1,024² bytes
- 1 GB (gigabyte) = 1,024³ bytes
- 1 TB (terabyte) = 1,024⁴ bytes

Referenced Documents

- *HiCommand™ Tuning Manager Administration Guide* (MK-92HC021)
- *HiCommand™ Tuning Manager Performance Reporter User's Guide* (MK-93HC033)
- *HiCommand™ Tuning Manager Installation Guide* (MK-93HC028)
- *HiCommand™ Tuning Manager Reference Guide* (MK-92HC022)
- *HiCommand™ Tuning Manager User's Guide* (MK-92HC013)

Comments

Please send us your comments on this document. Make sure to include the document title, number, and revision. Please refer to specific section(s) and paragraph(s) whenever possible.

- E-mail: doc.comments@hds.com
- Fax: 858-695-1186
- Mail:
Technical Writing, M/S 35-10
Hitachi Data Systems
10277 Scripps Ranch Blvd.
San Diego, CA 92131

Thank you! (All comments become the property of Hitachi Data Systems Corporation.)

Contents

Chapter 1	Overview of HTM Agent for DB2	1
1.1	Features of HTM Agent for DB2.....	2
1.1.1	Collecting DB2 Performance Data	2
1.1.2	Collecting Performance Data According to its Characteristics.....	3
1.1.3	Saving Performance Data	3
1.1.4	Defining Reports.....	4
1.2	Collecting and Managing Performance Data.....	5
Chapter 2	Installing and Setting up HTM Agent for DB2	7
2.1	Reviewing the Procedure from Installation to Starting Operations.....	8
2.2	Reviewing the Applicable Operating Systems and Monitored Programs	8
2.3	Installing the Prerequisite Program.....	8
2.4	Installing HTM Agent for DB2	9
2.4.1	Checking the Current Version of HTM Agent for DB2	11
2.5	Uninstalling HTM Agent for DB2	12
2.6	Post-installation Setup.....	13
2.6.1	Specifying the LANG Environment Variable	13
2.6.2	Setting up an Instance Environment.....	14
2.6.3	Specifying the Connection-target Tuning Manager.....	16
2.7	Canceling Setup of an Instance Environment	17
2.7.1	Deleting the Instance Environment on the Agent Machine	17
2.7.2	Deleting the Agent Information on the Tuning Manager Machine	18
2.8	Updating an Instance Environment	19
Chapter 3	Working with the Solution Set.....	21
3.1	Introducing the Solution Set	22
3.2	Reviewing the Format of Report Explanations.....	23
3.3	Reviewing the Organization of Report Folders.....	24
3.4	Reviewing the List of Reports	25
3.4.1	Basic Information on Database	26
3.4.2	Basic Information on Database Manager	27
3.4.3	Bufferpool I/O Status	29
3.4.4	Bufferpool I/O Trend	30
3.4.5	30	
3.4.6	Bufferpool Stat on Database	31
3.4.7	Cache Hit Rate.....	33
3.4.8	Cache Hit Rate Status	33
3.4.9	Cache Hit Rate Trend.....	34
3.4.10	Cache on Database	35
3.4.11	Sort Status on Database	36
3.4.12	SQL Statement on Database	37

Chapter 4	Working with Records	39
4.1	Data Model	40
4.2	Format of Record Contents	40
4.2.1	Function 40	
4.2.2	Default and Changeable Values	41
4.2.3	Key Fields	42
4.2.4	Lifetime	42
4.2.5	Record Size	42
4.2.6	Fields	42
4.3	List of Common Key Fields	44
4.4	List of Data Types	45
4.5	Field Values	46
4.5.1	Data Source	46
4.5.2	Restriction	47
4.5.3	Delta	47
4.6	Fields Added Only When Data is Stored in the Store Database	50
4.7	Notes on Records	51
4.8	List of Records for HTM Agent for DB2	51
4.8.1	Basic Information on Database (PD_DBID)	52
4.8.2	Basic Information on Database Manager (PD_PD)	55
4.8.3	Basic Information on Database Manager Interval (PI_PI)	58
4.8.4	Bufferpool Stat on Database (PD_DBPD)	61
4.8.5	Bufferpool Stat on Database Interval (PI_DBPI)	66
4.8.6	Cache on Database (PD_DCAD)	71
4.8.7	Cache on Database Interval (PI_DCAI)	74
4.8.8	DB2 Configuration (PD_DCFE)	77
4.8.9	Sort Status on Database (PD_DSOD)	80
4.8.10	SQL Statement on Database (PD_DSQD)	83
Chapter 5	Messages	87
5.1	Message Format	88
5.1.1	Format of Output Messages	88
5.1.2	Format of Message Explanations	89
5.1.3	Notes for System Administrators	89
5.2	Message Output Destinations	90
5.3	List of Messages Output to syslog	91
5.4	Messages	92
Chapter 6	Troubleshooting	95
6.1	Following Procedures for Handling Problems	96
6.2	Troubleshooting	97
6.2.1	Setting up or Starting a Service	98
6.2.2	Executing Commands	100
6.2.3	Collecting and Managing Performance Data	101
6.3	Checking Log Information	102
6.3.1	About the Common Message Log	103

6.4	Collecting Data when a Problem Occurs	104
6.4.1	Collecting Log information about the OS (In UNIX®)	104
6.4.2	Collecting Information about the Collection Manager or Agent.....	105
6.4.3	Collecting Operation Information.....	105
6.4.4	Collecting Error and Other Information.....	105
6.5	Data Collection Procedure (for UNIX®)	106
6.5.1	Collecting Data by Executing the jpcras Command.....	106
6.6	Calling the Hitachi Data Systems Support Center	107
Appendix A Notes on Miscellaneous Settings, Properties, and other Information		109
A.1	System Configuration	109
A.2	Performance Reporter Display.....	109
A.3	Estimating System Requirements	110
A.3.1	Estimating Memory Requirements	110
A.3.2	Estimating Disk Space Requirements	111
A.3.2.2	Estimating Disk Space Requirements for the Store Database	111
A.4	Kernel Parameters.....	113
A.5	List of Identifiers	113
A.6	List of Processes	114
A.7	List of Port Numbers	115
A.7.1	Port numbers of Collection Manager and Agent	115
A.7.2	Firewall Passage Directions.....	116
A.8	Properties of HTM Agent for DB2.....	117
A.8.1	List of Agent Store Service Properties	117
A.8.2	List of Agent Collector Service Properties.....	119
A.9	List of Files and Directories	120
A.9.1	List of Files and Directories Common to Collection Manager and the Agent	120
A.9.2	List of Files and Directories of HTM Agent for DB2	124
Acronyms and Abbreviations.....		127

List of Tables

Table 2.1	LANG Environment Variable Values Supported by HTM Agent for DB2	13
Table 2.2	Instance Information of HTM Agent for DB2	14
Table 2.3	Organization of Directories for Instance Environments	15
Table 3.1	Report Folders	24
Table 3.2	List of Reports	25
Table 3.3	Report Data for Basic Information on Database	26
Table 3.4	Report Data for Basic Information on Database Manager	27
Table 3.5	Report Data for Bufferpool I/O Status	29
Table 3.6	Report Data for Bufferpool I/O Trend	30
Table 3.7	Report Data for Bufferpool Stat on Database	31
Table 3.8	Report Data for Cache Hit Rate	33
Table 3.9	Report Data for Cache Hit Rate Status	33
Table 3.10	Report Data for Cache Hit Rate Trend	34
Table 3.11	Report Data for Cache on Database	35
Table 3.12	Report Data for Sort Status on Database	36
Table 3.13	Report Data for SQL Statement on Database	37
Table 4.1	Default and Changeable Values	41
Table 4.2	List of Common Key Fields	44
Table 4.3	List of Data Types	45
Table 4.4	Performance Data Collected by HTM Agent for DB2	47
Table 4.5	Fields that are Added when Data is Stored in the Store Database	50
Table 4.6	Values for Each PI Record Type	50
Table 4.7	List of Records for HTM Agent for DB2	51
Table 4.8	Basic Information on Database (PD_DBID) Default and Changeable Values	52
Table 4.9	Basic Information on Database (PD_DBID) Fields	52
Table 4.10	Basic Information on Database Manager (PD_PD) Default and Changeable Values	55
Table 4.11	Basic Information on Database Manager (PD_PD) Fields	55
Table 4.12	Basic Information on Database Manager Interval (PI_PI) Default and Changeable Values	58
Table 4.13	Basic Information on Database Manager Interval (PI_PI) Fields	58
Table 4.14	Bufferpool Stat on Database (PD_DBPD) Default and Changeable Values	61
Table 4.15	Bufferpool Stat on Database (PD_DBPD) Fields	61
Table 4.16	Bufferpool Stat on Database Interval (PI_DBPI) Default and Changeable Values	66
Table 4.17	Bufferpool Stat on Database Interval (PI_DBPI) Fields	66
Table 4.18	Cache on Database (PD_DCAD) Default and Changeable Values	71
Table 4.19	Cache on Database (PD_DCAD) Fields	71
Table 4.20	Cache on Database Interval (PI_DCAI) Default and Changeable Values	74
Table 4.21	Cache on Database Interval (PI_DCAI) Fields	74
Table 4.22	DB2 Configuration (PD_DCFE) Default and Changeable Values	77
Table 4.23	DB2 Configuration (PD_DCFE) Fields	77
Table 4.24	Sort Status on Database (PD_DSOD) Default and Changeable Values	80
Table 4.25	Sort Status on Database (PD_DSOD) Fields	80
Table 4.26	SQL Statement on Database (PD_DSQD) Default and Changeable Values	83
Table 4.27	SQL Statement on Database (PD_DSQD) Fields	83

PRELIMINARY RELEASE

Table 5.1	Output Destinations of HTM Agent for DB2 Messages	90
Table 5.2	Messages Output to syslog	91
Table 5.3	Messages that are Issued by HTM Agent for DB2	92
Table 6.1	List of Common Problems	97
Table 6.2	Handling Problems Related to Setup or Service Startup	98
Table 6.3	Handling Errors Related to the Execution of Tuning Manager Commands.....	100
Table 6.4	Types of Log Information.....	102
Table 6.5	Names of Storage Directories for the Trace Logs	103
Table 6.6	Log information about the OS (UNIX®)	104
Table 6.7	Collecting Information about Collection Manager and the Agent.....	105
Table A.1	Memory Requirements	110
Table A.2	Process Memory Requirements	110
Table A.3	Disk Space Requirements for the Entire System.....	111
Table A.4	Disk Space Requirements for the Store Database for Each Record Type	111
Table A.5	Default Values for the Retention Period and the Number of Records	112
Table A.6	List of Identifiers of HTM Agent for DB2	113
Table A.7	List of Processes of HTM Agent for DB2.....	114
Table A.8	Port Numbers Used by Collection Manager and Agent.....	115
Table A.9	Firewall Passage Directions.....	116
Table A.10	List of Agent Store Service Properties of HTM Agent for DB2	117
Table A.11	List of Agent Collector Service Properties of HTM Agent for DB2	119
Table A.12	List of Files and Directories Common to Collection Manager and Agent (continues on the following pages).....	120
Table A.13	List of Files and Directories of HTM Agent for DB2	124

PRELIMINARY RELEASE

Chapter 1 Overview of HTM Agent for DB2

This chapter provides an overview of HTM Agent for DB2.

- Features of HTM Agent for DB2 (see section 1.1)
- Collecting and Managing Performance Data (see section 1.2)

1.1 Features of HTM Agent for DB2

HTM Agent for DB2 is a program that collects and manages performance data, enabling you to monitor DB2 performance. HTM Agent for DB2 has the following features:

- **Analysis for DB2 Operating Status**

HTM Agent for DB2 provides the information needed for easy analysis of DB2s operating status, by collecting and summarizing performance data (such as CPU usage rate and number of processes) obtained from the DB2 instance being monitored, and graphically displaying any trends or changes.

- **DB2 Problem Reporting**

HTM Agent for DB2 provides a graphic display of information needed to determine the cause of a problem.

Note: Tuning Manager is a required component for the use HTM Agent for DB2.

1.1.1 Collecting DB2 Performance Data

HTM Agent for DB2 collects performance data, such as memory usage and the pool hit rate, for DB2 running on the target host. The collected data is used to display information about the entire SAN environment in the HTM Main Console.

Using Performance Reporter, which is one of the Tuning Manager program products, you can arrange and display performance data in graphical formats called *reports*. The report format facilitates comprehensive analysis of the operating status of the selected DB2 instance. There are two report classifications: Real-time reports and Historical reports:

- **Real-time reports**

Real-time reports show the current status of the DB2 system being monitored, and are used primarily for checking the current status and problems of the system. To display a real-time report, the system directly uses the most up-to-date real time performance data collected.

- **Historical reports**

Historical reports show the statuses of the DB2 system being monitored, from some time in the past up to the present. Historical reports are used primarily for analysis of system trends. To display historical reports, the system uses performance data stored in a database of HTM Agent for DB2.

1.1.2 Collecting Performance Data According to its Characteristics

HTM Agent for DB2 collects performance data in *records*. Each record consists of smaller units called *fields*. Records and fields are referred to collectively as the *data model*.

Records are classified into three types according to their characteristics. The record types that are used to collect performance data are predefined in HTM Agent for DB2. You use Performance Reporter to specify the performance data records to be collected.

HTM Agent for DB2 supports the following two record types:

- **Product Interval record type** (referred to hereafter as the *PI record type*)
For records of the PI record type, the system collects performance data for a specified interval, such as the number of processes in one minute. You can use these records to analyze changes or trends in the system status over time.
- **Product Detail record type** (referred to hereafter as the *PD record type*)
For records of the PD record type, the system collects performance data that indicates the system status at a specific point in time, such as detailed information about the currently active processes. You can use these records to obtain the system status at a particular time.

For more information about record types, see Chapter 4.

1.1.3 Saving Performance Data

By storing collected performance data into a special database, you can save the performance data up to the current date, and can analyze trends (from the past to the current date) in DB2 operating states. This special database is called the *Store database* of HTM Agent for DB2. Historical reports are used for analysis of trends.

Use Performance Reporter to select the performance data records to be stored in the Store database. For details about how to use Performance Reporter to select records, see the *HiCommand™ Tuning Manager Performance Reporter Reference*.

1.1.4 Defining Reports

HTM Agent for DB2 provides a *solution set*, in which necessary information for reports is predefined. This solution set facilitates the setup for monitoring of the DB2 operating status, since it does not require you to make complicated definitions. You can also customize the solution set as appropriate to your environment.

You can define reports from Performance Reporter. The Performance Reporter Report Wizard can be used to define reports, either by creating new reports or customizing the reports of the solution set.

- For details about the solution set, see Chapter 3.
- For details about how to define reports, see the chapter on defining and using reports, in the *HiCommand™ Tuning Manager User's Guide*, as well as the chapter on using reports, in the *HiCommand™ Tuning Manager - Performance Reporter Reference*.

1.2 Collecting and Managing Performance Data

The procedures for collecting and managing performance data depend on the record type used to store the performance data. The records for HTM Agent for DB2 are classified into the following two types:

- PI record type
- PD record type

For details about how to collect and manage performance data, see the following sections:

- **Performance data collection procedure**

For details about the performance data collection procedure, see the chapter with the overview of the data handled by programs in the Tuning Manager series, in the *HiCommand™ Tuning Manager User's Guide*.

For details about the values of the collected performance data, see Chapter 4.

- **Performance data management procedure**

For details about the performance data management procedure for the PI and PD record types, see the chapter with the overview of the data handled by programs in the Tuning Manager series, in the *HiCommand™ Tuning Manager User's Guide*.

You can use Performance Reporter to define which performance data is to be displayed in a report, from among all records collected and managed by the Agent. For more information about the definition procedure, see the chapter that describes Store database management, in the *HiCommand™ Tuning Manager User's Guide*.

Chapter 2 Installing and Setting up HTM Agent for DB2

This chapter describes the procedures for installing and setting up HTM Agent for DB2.

- Reviewing the Procedure from Installation to Starting Operation (see section 2.1)
- Reviewing the Applicable Operating Systems and Monitored Programs (see section 2.2)
- Installing the Prerequisite Program (see section 2.3)
- Installing HTM Agent for DB2 (see section 2.4)
- Uninstalling HTM Agent for DB2 (see section 2.5)
- Post-installation Setup (see section 2.6)

2.1 Reviewing the Procedure from Installation to Starting Operations

The following shows the procedure from installation to the time you start operating the Tuning Manager series programs:

1. Install and set up Tuning Manager and the Agents.
2. Start Tuning Manager.
3. Start each Agent.
4. Start operation of the programs in the Tuning Manager series.

Document References

- For details about the Agent commands (for the programs in the Tuning Manager series) that are used during installation and setup, see the *HiCommand™ Tuning Manager User's Guide*.
- For details about installation and setup of the programs for collecting and viewing performance data, see the following manuals:
 - *HiCommand™ Tuning Manager Administration Guide*
 - *HiCommand™ Tuning Manager Performance Reporter Reference*
 - *HiCommand™ Tuning Manager Installation Guide*
 - *HiCommand™ Tuning Manager Reference Guide*

For details about the procedure for starting each Agent, see the *HiCommand™ Tuning Manager User's Guide*.

- For details about the procedures for installing and uninstalling Tuning Manager, see the *HiCommand™ Tuning Manager Administration Guide* or *HiCommand™ Tuning Manager Installation Guide*.
- For details about the procedure for setting up an entire system that operates programs in the Tuning Manager series, see the chapter that describes setup for Windows®, in the *HiCommand™ Tuning Manager User's Guide*.

2.2 Reviewing the Applicable Operating Systems and Monitored Programs

HTM Agent for DB2 can run on the AIX and Linux 3 operating systems, and it can monitor the DB2 Universal Database program.

2.3 Installing the Prerequisite Program

In order to manage HTM Agent for DB2, HiCommand Tuning Manager version 4.1 or later is required in the Tuning Manager series system. For details about the procedures for installing Tuning Manager, see the *HiCommand™ Tuning Manager Administration Guide* or the *HiCommand™ Tuning Manager Installation Guide*.

2.4 Installing HTM Agent for DB2

This section describes installation of HTM Agent for DB2 on a UNIX® host.

Before performing the installation, please note the following points:

- When you install HTM Agent for DB2 for the first time, make sure that the installation directory contains no other files or directories.
- Before performing an overwrite installation to upgrade HTM Agent for DB2, check the version of the existing HTM Agent for DB2. To check the version, use the `jpctminfo` command. For details about how to use the `jpctminfo` command for checking the version, see section 2.4.1.
- If you install HTM Agent for DB2 on a host where Tuning Manager has already been installed, the local host always becomes the Agent's connection-target Tuning Manager host. To connect the HTM Agent for DB2 to another Tuning Manager host, make sure that Tuning Manager does not exist on the local host where HTM Agent for DB2 is to be installed.
- You must define a `jpchosts` file, hosts file, or domain name system (DNS) environment so that the IP address of each host can be resolved from each machine on which Tuning Manager and HTM Agent for DB2 are installed. To determine the host names, execute the `uname -n` command. Run HTM Agent for DB2 only in an environment where an IP address can be resolved from its own host name. If the IP address cannot be resolved, change the environment settings so that the IP address can be resolved from its own host name.

To install HTM Agent for DB2:

1. At the host where HTM Agent for DB2 is to be installed, log in as a user with the root user permissions, or use the `su` command to change to the superuser who has root user permissions.

Before proceeding, make sure that you now have root privileges.

2. Check whether any services of Tuning Manager series programs are running on the local host. If so, stop all the active services.

Display the service information to confirm that no services are running. For details about displaying service information and stopping services, see the chapter that describes the procedures for operating Tuning Manager series programs, in the *HiCommand™ Tuning Manager User's Guide*.

If any service of a Tuning Manager series program is active, the service will be automatically terminated during installation.

3. If any other application programs are running, terminate them.

4. Insert the CD-ROM for installing HTM Agent for DB2.

Set the CD-ROM for installing HTM Agent for DB2 in the CD-ROM drive and mount it on a desired directory. To mount the CD-ROM manually, enter the following command (in Linux, CD-ROMs are usually mounted automatically):

- For AIX (when the CD-ROM is loaded into the `/cdrom` directory):
`mount -r -v cdrfs /dev/cd0 /cdrom`
- For Linux (when the CD-ROM is loaded into the `/mnt/cdrom` directory):
`mount -r -o mode=0544 /dev/cdrom /mnt/cdrom`

5. Execute the installation command with the required option specified.

For AIX, use the `installp` command and for Linux, use the `rpm` command. The following examples show how to execute the command:

- In AIX® (when the CD-ROM is loaded into the `/cdrom` directory):

For a new installation:

```
installp -acgNX -d /cdrom HTMDB2
```

For an overwrite installation:

```
installp -acFNX -d /cdrom HTMDB2
```

- In Linux (when the CD-ROM is loaded into the `/mnt/cdrom` directory):

For a new installation:

```
rpm -ivh /mnt/cdrom/HTMDB2.rpm
```

For an overwrite installation:

```
rpm -Uvh --replacepks /mnt/cdrom/HTMDB2.rpm
```

Note: You can also install HTM Agent for DB2 by transferring the contents of the installation CD-ROM to the installation-target host (ensure that the contents (files) are archived using an archive command such as the `tar` command before transferring them to the installation target host).

In such a case, replace the directory indicated in step 5, where the installation command is mounted, with the directory to which the contents were transferred, and then execute the command.

2.4.1 Checking the Current Version of HTM Agent for DB2

To check the version of the existing HTM Agent for DB2, use the `jpctminfo` command, which is stored in the following directory:

```
/opt/jp1pc/tools/
```

This command displays product information about Tuning Manager programs that are installed on the host where the command is executed. You can use this command to check the version and patch history. For details about how to use the `jpctminfo` command, see the chapter that explains commands in the *HiCommand Tuning Manager User's Guide*.

- To check the version of HTM Agent for DB2, use this command: `jpctminfo agtr`
The following example illustrates version information that is displayed by the `jpctminfo agtr` command:

```
PRODUCT HiCommand Tuning Manager - Agent for IBM(R) DB2(R) Universal Database(TM)
VERSION 4.1.0-00(04-10-00)
```

- To display the patch history, use this command: `jpctminfo agtr -p`
If patches have been installed, executing the `jpctminfo agtr -p` command displays the versions and installation dates of the patches, as shown in the following example:

```
PRODUCT HiCommand Tuning Manager - Agent for IBM(R) DB2(R) Universal Database(TM)
VERSION 4.1.0-02(04-10-02)
PATCH   DATE
041001  2005/01/31
041002  2005/02/28
```

However, if no patches have been installed, a message will indicate that there is no patch installation history, as shown in the following example:

```
PRODUCT HiCommand Tuning Manager - Agent for IBM(R) DB2(R) Universal Database(TM)
VERSION 4.1.0-00(04-10-00)
KAVF24903-I There is no patch history information.
```

2.5 Uninstalling HTM Agent for DB2

This section explains the procedure for uninstalling HTM Agent for DB2 from a UNIX[®] host.

Note: Uninstalling HTM Agent for DB2 does not delete the port number defined in the `services` file.

To uninstall HTM Agent for DB2:

1. At the host where HTM Agent for DB2 is to be uninstalled, log in as a user with the root user permissions, or use the `su` command to change to the superuser who has `root` user permissions.

Before proceeding, make sure that you now have root privileges.

2. Check whether any services of Tuning Manager series programs are running on the local host. If any such services are running, stop them.

Display the service information to confirm that no services are running. For details about displaying service information and stopping services, see the chapter that describes the procedures for operating Tuning Manager series programs, in the *HiCommand™ Tuning Manager User's Guide*.

If any service of a Tuning Manager series program is active, the service will be automatically terminated during uninstallation.

3. If any other application programs are running, terminate them.
4. Enter the uninstallation command with the required option specified.
 - For AIX[®], use the `installp` command:

```
installp -u HTMDB2
```
 - For Linux, use the `rpm` command:

```
rpm -e HTMDB2
```

2.6 Post-installation Setup

Before you can use HTM Agent for DB2, you must set up the program named HTM Agent for DB2.

- For details about setting up Tuning Manager and the options for HTM Agent for DB2, see the chapter that describes setup for UNIX®, in the *HiCommand™ Tuning Manager User's Guide*.
- For details about setting up from Performance Reporter, see the *HiCommand™ Tuning Manager Performance Reporter Reference*.

2.6.1 Specifying the LANG Environment Variable

Table 2.1 lists the LANG environment variable values supported by HTM Agent for DB2.

Before you specify the LANG environment variable, make sure that the correct language environment has been installed and constructed. If the language environment is incorrect, encoding errors may occur or definition data may be replaced with invalid data.

Note: The LANG environment variable specified at the time of service startup or command execution determines the language used for the common message log. Therefore, character strings in the code sets of multiple languages, such as English and Japanese, may coexist.

Table 2.1 LANG Environment Variable Values Supported by HTM Agent for DB2

OS	Language		LANG Environment Variable Values
AIX	Japanese	Shift-JIS codes	<ul style="list-style-type: none"> Ja_JP Ja_JP.IBM-932
		EUC codes	<ul style="list-style-type: none"> ja_JP ja_JP.IBM-eucJP
	English (with no Japanese)		C
Linux	Japanese	Shift-JIS codes	Not applicable
		EUC codes	<ul style="list-style-type: none"> ja_JP.eucJP ja_JP.ujis
	English (with no Japanese)		C

2.6.2 Setting up an Instance Environment

You must specify instance information for the DB2 that is to be monitored by HTM Agent for DB2. Table 2.2 lists the instance information items that are to be specified using the `jpcinssetup` command. Please check this information before you start the setup procedure. For details about DB2 instance information, see the DB2 documentation.

Table 2.2 Instance Information of HTM Agent for DB2

Item	Description
<code>db2_instance</code>	Specify the instance name for DB2. Up to 8 single-byte alphanumeric characters can be specified. However, space characters and the following symbols cannot be used: ! " % & ' () + , - . / : ; < = > ? [\] ' { } ~
<code>db2_node</code>	Specify the node name for DB2. Up to 8 single-byte alphanumeric characters can be specified. However, space characters and the following symbols cannot be used: ! " % & ' () + , - . / : ; < = > ? [\] ' { } ~
<code>db2_name</code>	Specify the database name for DB2. Up to 8 single-byte alphanumeric characters can be specified. However, space characters and the following symbols cannot be used: ! " % & ' () + , - . / : ; < = > ? [\] ' { } ~
<code>db2_user</code>	Specify a user name that has SYSADM, SYSCTRL or SYSMANT privileges and is allowed to connect to the DB2 database. Up to 8 single-byte alphanumeric characters can be specified. However, space characters and the following symbols cannot be used: ! " % & ' () + , - . / : ; < = > ? [\] ' { } ~
<code>db2_password</code>	Specify the password for the user specified by <code>db2_user</code> . Up to 64 single-byte alphanumeric characters can be specified. However, space characters and the following symbols cannot be used: ! " % & ' () + , - . / : ; < = > ? [\] ' { } ~
<code>attach_retry</code>	Specify the number of DB2 instance connection retries. The default is 0. A value from 0 to 99 (the number of retries) can be specified.
<code>attach_waittime</code>	Specify the interval for DB2 instance connection retries. The default is 0. A value from 0 to 600 (seconds) can be specified.

To construct an instance environment:

1. Execute the `jpcinssetup` command with the service key and instance name specified. For example, to construct an instance environment for the instance named `db2inst`, execute the following command:

```
jpcinssetup agtr -inst db2inst
```

Use an instance name that is unique among all the names of the Agent instances on the network. You cannot use `sql` as the instance name. For details about the `jpcinssetup` command, see the chapter that describes commands in the *HiCommand Tuning Manager User's Guide*.

2. Specify DB2 instance information.
Enter the information shown in Table 2.2, in accordance with the command's instructions. You cannot omit any requested items. To use a displayed value (the default), press the Enter key.
3. To use multiple instances, repeat steps 1 and 2 for each instance.

2.6.2.1 About the Constructed Environment

Once you have entered all items, the instance environment is constructed.

- **About the organization of directories for instance environments**

Instance environments are constructed under the following directory:

`/opt/jp1pc/agtr`

Table 2.3 shows the organization of the directories for instance environments.

- **About the service ID for an instance environment**

The following is the service ID for an instance environment:

`product-ID function-ID instance-number instance-name [host-name]`

For details about service IDs, see the chapter that describes the operations of Tuning Manager series programs, in the *HiCommand Tuning Manager User's Guide*.

Table 2.3 Organization of Directories for Instance Environments

Directory and File Names			Description
agent	<i>instance-name</i>	<code>jpcagt.ini</code>	The service startup initialization file of Agent Collector
		<code>jpcagt.ini.model</code> See <i>Note</i>	The model file for a service startup initialization file of the Agent Collector service
		<code>log</code>	The directory for storing internal log files of the Agent Collector service
store	<i>instance-name</i>	<code>jpcsto.ini</code>	The service startup initialization file of the Agent Store service
		<code>jpcsto.ini.model</code> See <i>Note</i>	The model file for a service startup initialization file of the Agent Store service
		<code>*.DAT</code>	The definition file for a data model
		<code>dump</code>	The default database export directory
		<code>backup</code>	The default database backup directory
		<code>log</code>	The directory for storing internal log files of the Agent Store service

Note: Use this file to restore the settings in effect before an instance environment was constructed.

2.6.3 Specifying the Connection-target Tuning Manager

At the host where HTM Agent for DB2 is installed, you must specify the Tuning Manager that manages that HTM Agent for DB2 using the `jpcnshostname` command.

Reviewing the Setup Notes

- One Agent host can have only one connection-target Tuning Manager. If more than one Agent is installed on the same host, you cannot specify different connection-target Tuning Managers for different Agents.
- If HTM Agent for DB2 and Tuning Manager are installed on the same host, the Tuning Manager on the local host is the connection-target Tuning Manager. In such a case, you cannot change the connection-target Tuning Manager to a Tuning Manager on a remote host.
- Before you start the setup procedure, you must terminate any Tuning Manager series program that is running on the local host. For details about stopping services, see the chapter that describes the procedures for operating Tuning Manager series programs, in the *HiCommand™ Tuning Manager User's Guide*.

If a Tuning Manager series program is active during execution of the `jpcnshostname` command, a message is displayed that asks you to terminate the program.

To specify the connection-target Tuning Manager:

Execute the `jpcnshostname` command with the host name of the connection-target Tuning Manager specified.

For example, if the connection-target Tuning Manager is on host `host01`, specify the following: `jpcnshostname -s host01`

2.7 Canceling Setup of an Instance Environment

Canceling setup of HTM Agent for DB2 involves the following tasks:

- Deleting the instance environment on the Agent machine
- Deleting the Agent information on the Tuning Manager machine

To cancel setup of multiple instance environments, you must repeat the procedure for each environment.

2.7.1 Deleting the Instance Environment on the Agent Machine

To delete an instance environment on the Agent machine:

1. Use the `jpcinslist` command to find the instance name of HTM Agent for DB2. The following shows the command format: `jpcinslist agtr`

For example, if the instance name is `db2inst`, the command displays `db2inst`.

2. Stop the HTM Agent for DB2 service in the instance environment if it is active.

For details about how to stop a service, see the chapter that explains the operations of Tuning Manager series programs in the *HiCommand Tuning Manager User's Guide*.

3. Use the `jpcinsunsetup` command to delete the instance environment. The following shows the command format: `jpcinsunsetup agtr -inst instance-name`

For example, to delete an instance environment whose instance name is `db2inst`, execute the following command: `jpcinsunsetup agtr -inst db2inst`

If the command is successful, the directories created during instance environment setup are deleted. If a service with the specified instance name is active, a message appears asking whether the service is to be stopped.

For details about the `jpcinsunsetup` command, see the chapter that describes commands in the *HiCommand Tuning Manager User's Guide*.

2.7.2 Deleting the Agent Information on the Tuning Manager Machine

To delete the Agent information on the Tuning Manager machine:

1. On the Tuning Manager machine, execute the `jpcctrl delete` command to delete the Agent information.

For example, to delete the Agent information when the instance name is `db2inst`, the host name is `host03`, the Agent Collector service ID is `RA1db2inst[host03]`, and the Agent Store service ID is `RS1db2inst[host03]`, execute the following command:

```
jpcctrl delete R?1db2inst[host03] host=host03
```

For details on the `jpcctrl delete` command, see the chapter that explains commands in the *HiCommand Tuning Manager User's Guide*.

2. Restart the Collection Manager service.

For details about how to start services, see the chapter that explains operations for running the Tuning Manager series, in the *HiCommand Tuning Manager User's Guide*.

3. Restart Performance Reporter.

2.8 Updating an Instance Environment

Before you update an instance environment, you will find the instance name using the `jpcinslist` command. Afterward, you will update the instance environment using the `jpcinssetup` command.

For details about:

- Updating an instance environment, see the chapter that explains setup in the *HiCommand Tuning Manager User's Guide*.
- The `jpcinslist` and `jpcinssetup` commands, see the chapter that describes commands in the *HiCommand Tuning Manager User's Guide*.

To update an instance environment:

1. Find the instance name by executing the `jpcinslist` command where you also specify the service key indicating the HTM Agent for DB2 that is running in the instance environment.

For example, to check the instance name of HTM Agent for DB2, execute the following command: `jpcinslist agtr`

If the specified instance name is `db2inst`, the command displays `db2inst`.

2. Refer to Table 2.2 (section 2.6.2) to check the information to be updated.
3. If the HTM Agent for DB2 service is active in the instance environment that is to be updated, stop the service.

For details about stopping services, see the chapter that describes the procedures for operating Tuning Manager series programs in the *HiCommand Tuning Manager User's Guide*.

If the service is still active in the instance environment that is to be updated when you execute the `jpcinssetup` command, a confirmation message displays and allows you to stop the service. Stopping the service resumes update processing. Otherwise, update processing is cancelled.

4. Execute the `jpcinssetup` command where you also specify the service key and instance name of the HTM Agent for DB2 in the instance environment that you wish to update.

For example, if you are updating the instance environment for the HTM Agent for DB2 with the instance name `db2inst`, execute the following command:

```
jpcinssetup agtr -inst db2inst
```

PRELIMINARY RELEASE

5. Update the instance information for DB2: Enter the information shown in Table 2.2 (section 2.6.2) in accordance with the command's instructions. The current settings are displayed (except for the value of `db2_password`). To use a displayed value, press the **Enter** key. When all entries are completed, the instance environment is updated.
6. Restart the service in the updated instance environment.
For details about starting services, see the chapter that describes the procedures for operating Tuning Manager series programs in the *HiCommand Tuning Manager User's Guide*.
7. To update multiple instance environments, repeat the procedure for each instance environment.

Chapter 3 Working with the Solution Set

This chapter describes the solution set for HTM Agent for DB2, which Performance Reporter uses.

- Introducing the Solution Set (see section 3.1)
- Reviewing the Format of Report Explanations (see section 3.2)
- Reviewing the Organization of Report Folders (see section 3.3)
- Reviewing the List of Reports (see section 3.4)

3.1 Introducing the Solution Set

The Tuning Manager products enable you to define reports by using the:

- Agent-provided reports
- Report wizard or commands in Performance Reporter to customize reports provided by an Agent
- Report wizard or commands in Performance Reporter to define new reports

The reports provided by an Agent constitute a *solution set*. Because all required information is already defined for the reports in a solution set, you can use them as provided, or customize them as appropriate for your environment. This enables you to complete the preparations for monitoring the operating status of a desired program without having to use the Report wizard or commands to define new reports.

This chapter describes the reports in the solution set that is defined by HTM Agent for DB2.

For details about using a solution set, see the chapter that describes the manipulation of reports, in the *HiCommand™ Tuning Manager Performance Reporter Reference*.

3.2 Reviewing the Format of Report Explanations

This section describes the format used to explain reports. The manual lists the reports in alphabetical order. Each report contains the following parts:

Overview

Provides an overview of the information that can be displayed in the report.

Storage location

Indicates the storage location of the report.

Record

Indicates the record that contains the performance data used in the report. To display a historical report, you must specify information in advance in order to collect the record indicated in this column. Before you display a report, use the `jpcasrec output` command to confirm that the record indicated in this column is set as `Log=Yes`. For a real-time report, you do not need to make this specification. For details about the Performance Reporter commands, see the chapter that describes commands in the *HiCommand™ Tuning Manager Performance Reporter Reference*.

Note: The maximum number of records that can be displayed in a report is set in advance. For details about how to change the maximum number of records that can be displayed, see the chapter that describes how to perform operations on reports, in the *HiCommand™ Tuning Manager Performance Reporter Reference*.

Report Data

This is a table that contains some or all of the following information:

- **Fields**
Describes the fields used in the report.
- **Drilldown reports (report level)**
Lists other reports in the solution set that are related to this report. For details about displaying a drilldown report, see the chapter that describes the manipulation of reports, in the *HiCommand™ Tuning Manager Performance Reporter Reference*. Note that some reports do not have any drilldown reports.
- **Drilldown reports (field level)**
Describes reports in the solution set that are associated with fields used in this report. For details about displaying a drilldown report, see the chapter that describes the manipulation of reports, in the *HiCommand™ Tuning Manager Performance Reporter Reference*. Note that some reports do not have any drilldown reports.

Note: In this manual, if a report only contains one field and no drilldown reports, the field data is listed beneath the 'Field' heading.

3.3 Reviewing the Organization of Report Folders

The following example shows the organization of the report folders for HTM Agent for DB2. Angle brackets enclose folder names.

```

<DB2 Universal Database>
+-- <Monthly Trend>
|   +-- Bufferpool I/O Trend
|   +-- Cache Hit Rate Trend
+-- <Status Reporting>
|   +-- <Daily Trend>
|       +-- Bufferpool I/O Status
|       +-- Cache Hit Rate Status
|   +-- <Real-Time>
|       +-- Basic Information on Database
|       +-- Basic Information on Database Manager
|       +-- Bufferpool Stat on Database
|       +-- Sort Status on Database
+-- <Troubleshooting>
    +-- <Real-Time>
        +-- Cache on Database
        +-- SQL Statement on Database
    +-- <Recent Past>
        +-- Cache Hit Rate
    
```

Table 3.1 describes each folder.

Table 3.1 Report Folders

Folder	Contents/Use
Monthly Trend	Contains reports that display daily information for the past month. Use it to check monthly trends in the system.
Status Reporting	Contains reports for displaying daily information. Use this folder to check the overall status of the system. You can also display real-time reports in addition to historical reports. <ul style="list-style-type: none"> ▪ Daily Trend folder This folder contains reports that display information for the past 24 hours, as well as hourly information for the past 24 hours. Use it to check the daily status of the system. ▪ Real-Time folder This folder contains real-time reports for checking the system status
Troubleshooting	Contains reports for displaying information that is useful for resolving problems. In the event of a system problem, use the reports in this folder to check the cause of the problem. <ul style="list-style-type: none"> ▪ Real-Time folder This folder contains real-time reports for checking the current system status. ▪ Recent Past folder This folder contains historical reports for displaying minute-by-minute information for the past hour.

3.4 Reviewing the List of Reports

Table 3.2 lists the reports that are defined in the solution set in alphabetical order:

Table 3.2 List of Reports

Report name	Displayed information	Storage location
Basic Information on Database	Basic information on the database	Reports/DB2 Universal Database/Status Reporting/Real Time/Basic Information on Database
Basic Information on Database Manager	Basic information on the database at the database manager level	Reports/DB2 Universal Database/Status Reporting/Real Time/Basic Information on Database Manager
Bufferpool I/O Status	Buffer pool I/O status on an hourly basis over the last 24 hours	Reports/DB2 Universal Database/Status Reporting/Daily Trend/Bufferpool I/O Status
Bufferpool I/O Trend	Buffer pool I/O status on a daily basis over the last month	Reports/DB2 Universal Database/Monthly Trend/Bufferpool I/O Trend
Bufferpool Stat on Database	Buffer pool information at the database level	Reports/DB2 Universal Database/Status Reporting/Real Time/Bufferpool Stat on Database
Cache Hit Rate	Cache hit rate on a per-minute basis over the last hour	Reports/DB2 Universal Database/Troubleshooting/Recent Past/Cache Hit Rate
Cache Hit Rate Status	Cache hit rate on an hourly basis over the last 24 hours	Reports/DB2 Universal Database/Status Reporting/Daily Trend/Cache Hit Rate Status
Cache Hit Rate Trend	Cache hit rate on a daily basis over the last month	Reports/DB2 Universal Database/Monthly Trend/Cache Hit Rate Trend
Cache on Database	Cache information at the database level	Reports/DB2 Universal Database/Troubleshooting/Real Time/Cache on Database
Sort Status on Database	Sort information at the database level	Reports/DB2 Universal Database/Status Reporting/Real Time/Sort Status on Database
SQL Statement on Database	SQL Statement information at the database level	Reports/DB2 Universal Database/Troubleshooting/Real Time/SQL Statement on Database

3.4.1 Basic Information on Database

Overview

The Basic Information on Database report displays basic information on the database in real time.

Storage Location

Reports/DB2 Universal Database/Status Reporting/Real Time/Basic Information on Database

Record

Basic Information on Database (PD_DBID)

Table 3.3 Report Data for Basic Information on Database

Fields	
Field Name	Description
Agents Top	The maximum number of agents for applications at the database level.
Appl Section Inserts	The number of SQL sections inserted by an application from its SQL work area.
Appl Section Lookups	The number of SQL sections referenced by an application from its SQL work area.
Appls Cur Cons	The number of applications that are currently connected to the database.
Appls In DB2	The number of applications that are currently connected to the database, and for which the database manager is currently processing requests.
Coord Agents Top	The maximum number of coordinator agents that can run concurrently.
DB Heap Top	Memory usage
DB Name	The real name of the database for which information is collected or to which the application is connected.
DB Status	The current status of the database. The valid values are: <ul style="list-style-type: none"> ▪ 0 (SQLM_DB_ACTIVE: The database is active.) ▪ 1 (SQLM_DB_QUIESCE_PEND: The database is in the quiescent pending state.) ▪ 2 (SQLM_DB_QUIESCED: The database is in the quiescent state.) ▪ 3 (SQLM_DB_ROLLFWD: A roll forward is in progress on the database.)
Hash Join Overflows	The number of times that hash join data exceeded the available sort heap space.
Hash Join Overflows Rate	Ratio of hash_join_overflows and hash_join_small_overflows.
Hash Join Small Overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
Num Assoc Agents	At the database level, the number of subagents for all applications.
Total Hash Joins	The total number of hash joins executed.
Total Hash Loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.

3.4.2 Basic Information on Database Manager

Overview

The Basic Information on Database Manager report displays basic information on the database at the database manager level, in real time.

Storage Location

Reports/DB2 Universal Database/Status Reporting/Real Time/Basic Information on Database Manager

Record

Basic Information on Database Manager (PD_PD)

Table 3.4 Report Data for Basic Information on Database Manager

Fields	
Field Name	Description
Agents Created Empty Pool	The number of agents created because the agent pool was empty.
Agents Created Pool Rate	Agents assigned from the agent pool created because the agent pool was empty.
Agents From Pool	The number of agents assigned from the agent pool.
Agents Registered	The number of agents (coordinator agents and subagents) registered in the database manager instance that is being monitored.
Agents Registered Top	The maximum number of agents (coordinator agents and subagents) that the database manager has ever registered, at the same time, since it was started.
Agents Waiting On Token	The number of agents waiting for a token so that they can execute a transaction in the database manager.
Agents Waiting Top	The maximum number of agents (coordinator agents and subagents) that the database manager has ever registered, at the same time, since it was started. The maximum number of agents that have ever been simultaneously waiting for a token since the database manager was started.
Comm Private Mem	The amount of private memory that the database manager instance has currently committed at the time of the snapshot.
Cons In Exec Total	The total number of applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
ConsTotal	The total number of applications that are currently connected to the database manager instance or the database being monitored.
DB2 Status	The current status of the database manager instance. This is always <code>SQLM DB2 ACTIVE</code> .
Local Cons	The number of local applications that are currently connected to a database within the database manager instance being monitored.
Local Cons In Exec	The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
Post Threshold Hash Joins	The total number of times that a hash join heap request was limited due to concurrent use of shared or private sort heap space.

PRELIMINARY RELEASE

Fields	
Field Name	Description
Rem Cons In	The number of current connections initiated from remote clients to the database manager instance that is being monitored.
Rem Cons In Exec	The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
Sort Heap Allocated	The total number of allocated pages of sort heap space allocated for all sorts at the level chosen, at the time the snapshot was taken.

3.4.3 Bufferpool I/O Status

Overview

The Bufferpool I/O Status report displays a summary of the buffer pool I/O status on an hourly basis over the last 24 hours.

Storage Location

Reports/DB2 Universal Database/Status Reporting/Daily
Trend/Bufferpool I/O Status

Record

Bufferpool Stat on Database Interval (PI_DBPI)

Table 3.5 Report Data for Bufferpool I/O Status

Fields	
Field Name	Description
Pool Async Data Reads Rate	The ratio of the number of data pages read asynchronously into the buffer pool by a prefetcher to the number of read requests that required I/O to get data pages into the buffer pool.
Pool Async Data Writes Rate	The ratio of the number of times that a buffer pool data page was physically written to a disk by either an asynchronous page cleaner or a prefetcher to the number of times a buffer pool data page was physically written to the disk.
Pool Async Index Reads Rate	The ratio of the number of index pages read asynchronously into the buffer pool by a prefetcher to the number of physical read requests to get index pages into the buffer pool.
Pool Async Index Writes Rate	The ratio of the number of times a buffer pool index page was physically written to a disk by either an asynchronous page cleaner or a prefetcher to the number of times a buffer pool index page was physically written to a disk.
Pool Data L Reads	The number of logical read requests for data pages that have gone through the buffer pool.
Pool Data P Reads	The number of read requests that required I/O to get data pages into the buffer pool.
Pool Data Writes	The number of times a buffer pool data page was physically written to a disk.
Pool Index L Reads	The number of logical read requests for index pages that have gone through the buffer pool.
Pool Index P Reads	The number of physical read requests to get index pages into the buffer pool.
Pool Index Writes	The number of times a buffer pool index page was physically written to a disk.

3.4.4 Bufferpool I/O Trend

Overview

The Bufferpool I/O Trend report displays a summary of the buffer pool I/O status on a daily basis over the last month.

Storage Location

Reports/DB2 Universal Database/Monthly Trend/Bufferpool I/O Trend

Record

Bufferpool Stat on Database Interval (PI_DBPI)

Table 3.6 Report Data for Bufferpool I/O Trend

Fields	
Field Name	Description
Pool Async Data Reads Rate	The ratio of the number of data pages read asynchronously into the buffer pool by a prefetcher to the number of read requests that required I/O to get data pages into the buffer pool.
Pool Async Data Writes Rate	The ratio of the number of times that a buffer pool data page was physically written to a disk by either an asynchronous page cleaner or a prefetcher to the number of times a buffer pool data page was physically written to the disk.
Pool Async Index Reads Rate	The ratio of the number of index pages read asynchronously into the buffer pool by a prefetcher to the number of physical read requests to get index pages into the buffer pool.
Pool Async Index Writes Rate	The ratio of the number of times a buffer pool index page was physically written to a disk by either an asynchronous page cleaner or a prefetcher to the number of times a buffer pool index page was physically written to a disk.
Pool Data L Reads	The number of logical read requests for data pages that have gone through the buffer pool.
Pool Data P Reads	The number of read requests that required I/O to get data pages into the buffer pool.
Pool Data Writes	The number of times a buffer pool data page was physically written to a disk.
Pool Index L Reads	The number of logical read requests for index pages that have gone through the buffer pool.
Pool Index P Reads	The number of physical read requests to get index pages into the buffer pool.
Pool Index Writes	The number of times a buffer pool index page was physically written to a disk.

3.4.5

3.4.6 Bufferpool Stat on Database

Overview

The Bufferpool Stat on Database report displays information on the buffer pool at the database level, in real time.

Storage Location

Reports/DB2 Universal Database/Status Reporting/Real Time/Bufferpool Stat on Database

Record

Bufferpool Stat on Database (PD_DBPD)

Table 3.7 Report Data for Bufferpool Stat on Database

Fields	
Field Name	Description
Pool Async Data Reads	The number of pages read asynchronously into the buffer pool.
Pool Async Data Reads Rate	The ratio of the number of data pages read asynchronously into the buffer pool by a prefetcher to the number of read requests that required I/O to get data pages into the buffer pool.
Pool Async Data Writes	The number of times a buffer pool data page was physically written to a disk by either an asynchronous page cleaner or a prefetcher.
Pool Async Data Writes Rate	The ratio of the number of times that a buffer pool data page was physically written to a disk by either an asynchronous page cleaner or a prefetcher to the number of times a buffer pool data page was physically written to the disk.
Pool Async Index Reads	The number of index pages asynchronously read into the buffer pool by a prefetcher.
Pool Async Index Reads Rate	The ratio of the number of index pages read asynchronously into the buffer pool by a prefetcher to the number of physical read requests to get index pages into the buffer pool.
Pool Async Index Writes	The number of times a buffer pool index page was written to a disk by either an asynchronous page cleaner or a prefetcher.
Pool Async Index Writes Rate	The ratio of the number of times a buffer pool index page was physically written to a disk by either an asynchronous page cleaner or a prefetcher to the number of times a buffer pool index page was physically written to a disk.
Pool Async Read Time	The total elapsed time spent reading by database manager prefetchers.
Pool Async Read Time Avg	The average of the total elapsed time spent reading by database manager prefetchers.
Pool Async Total Reads	The total number of data and index pages read asynchronously into the buffer pool by prefetchers.
Pool Async Total Writes	The number of times a buffer pool data page or index page was physically written to a disk by an asynchronous page cleaner or a prefetcher.
Pool Async Write Time	The total elapsed time spent by database manager page cleaners writing data or index pages from the buffer pool to a disk.

PRELIMINARY RELEASE

Fields	
Field Name	Description
Pool Async Write Time Avg	The average of the total elapsed time spent by database manager page cleaners writing data or index pages from the buffer pool to a disk.
Pool Data Hit Rate	Rate of data page hits.
Pool Data L Reads	The number of logical read requests for data pages that have gone through the buffer pool.
Pool Data P Reads	The number of read requests that required I/O to get data pages into the buffer pool.
Pool Data Writes	The number of times a buffer pool data page was physically written to a disk.
Pool Drty PG Steal Clns	The number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.
Pool Drty PG Steal Clns Rate	The ratio of the number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database to the total number of page cleaner invocations.
Pool Drty PG Thrsh Clns	The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.
Pool Index Hit Rate	Rate of index page hits.
Pool Index L Reads	The number of logical read requests for index pages that have gone through the buffer pool.
Pool Index P Reads	The number of physical read requests to get index pages into the buffer pool.
Pool Index Writes	The number of times a buffer pool index page was physically written to a disk.
Pool Lns Gap Clns	The number of times a page cleaner was invoked because the used logging space had reached a predefined criterion for the database.
Pool Read Time	The total amount of time spent processing read requests that caused data or index pages to be physically read from a disk or the buffer pool.
Pool Read Time Avg	The average of the total amount of time spent processing read requests that caused data or index pages to be physically read from a disk or the buffer pool.
Pool Total Hit Rate	Total hit rate for the buffer pool.
Pool Total Writes	The number of times writing from the buffer pool to a disk was executed.
Pool Write Time	The total amount of time spent writing data or index pages physically from the buffer pool to a disk.
Pool Write Time Avg	The average of the total amount of time spent writing data or index pages physically from the buffer pool to a disk.

3.4.7 Cache Hit Rate

Overview

The Cache Hit Rate report displays a summary of the cache hit rate on a per-minute basis over the last hour.

Storage Location

Reports/DB2 Universal Database/Troubleshooting/Recent Past/Cache Hit Rate

Record

Cache on Database Interval (PI_DCAI)

Table 3.8 Report Data for Cache Hit Rate

Fields	
Field Name	Description
Catcache Hit Rate	Rate of catalog cache hits.
Pkg Cache Hit Rate	Rate of package cache hits.

3.4.8 Cache Hit Rate Status

Overview

The Cache Hit Rate Status report displays a summary of the cache hit rate on an hourly basis over the last 24 hours.

Storage Location

Reports/DB2 Universal Database/Status Reporting/Daily Trend/Cache Hit Rate Status

Record

Cache on Database Interval (PI_DCAI)

Table 3.9 Report Data for Cache Hit Rate Status

Fields	
Field Name	Description
Catcache Hit Rate	Rate of catalog cache hits.
Pkg Cache Hit Rate	Rate of package cache hits.

3.4.9 Cache Hit Rate Trend

Overview

The Cache Hit Rate Trend report displays a summary of the cache hit rate on a daily basis over the last month.

Storage Location

Reports/DB2 Universal Database/Monthly Trend/Cache Hit Rate Trend

Record

Cache on Database Interval (PI_DCAI)

Table 3.10 Report Data for Cache Hit Rate Trend

Fields	
Field Name	Description
Catcache Hit Rate	Rate of catalog cache hits.
Pkg Cache Hit Rate	Rate of package cache hits.

3.4.10 Cache on Database

Overview

The Cache on Database report displays cache information at the database level, in real time.

Storage Location

Reports/DB2 Universal Database/Troubleshooting/Real Time/Cache on Database

Record

Cache on Database (PD_DCAD)

Table 3.11 Report Data for Cache on Database

Fields	
Field Name	Description
Cat Cache Inserts	The number of times that the system tried to insert table descriptors or authorization information into the catalog cache.
Cat Cache Lookups	The number of times that the catalog cache was referenced in order to obtain table descriptor or authorization information.
Cat Cache Needed Minimize	The minimum size (pages) of the required catalog cache, in KB.
Cat Cache Overflows	The number of times that the catalog cache overflowed out of the bounds of its allocated memory.
Cat Cache Size Top	The largest size reached by the catalog cache, in KB.
Catcache Hit Rate	Rate of catalog cache hits.
Pkg Cache Hit Rate	Rate of package cache hits.
Pkg Cache Inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache.
Pkg Cache Lookups	The number of times that an application searched for a section or package in the package cache.
Pkg Cache Needed Minimize	The minimum size (pages) of the required package cache, in KB.
Pkg Cache Num Overflows	The number of times that the package cache overflowed out of the bounds of its allocated memory.
Pkg Cache Size Top	The largest size reached by the package cache, in KB.

3.4.11 Sort Status on Database

Overview

The Sort Status on Database report displays sort information at the database level, in real time.

Storage Location

Reports/DB2 Universal Database/Status Reporting/Real Time/Sort Status on Database

Record

Sort Status on Database (PD_DSOD)

Table 3.12 Report Data for Sort Status on Database

Fields	
Field Name	Description
Active Sorts	The number of sorts in the database that currently have a sort heap allocated.
Sheapthres	Sort heap threshold.
Sheapthres Shr	Sort heap threshold for shared sorts.
Sort Heap Allocated	The total number of pages of sort heap space allocated for all sorts at the level chosen, at the time the snapshot was taken.
Sort Heap Allocated Avg	The average of the number of pages of sort heap space allocated for all sorts at the level chosen, at the time the snapshot was taken.
Sort Overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
Sort Overflows Rate	The ratio of the sorts that ran out of sort heap and may have required disk space for temporary storage.
Sort Shrheap Allocated	Sort shared heap currently allocated.
Sort Shrheap Allocated Rate	Shared sort memory usage of the database.
Sort Shrheap Top	Maximum watermark for the sort shared heap .
Total Sort Time	The total elapsed time for all sorts that have been executed (in milliseconds).
Total Sort Time Avg	The average of the total elapsed time for all sorts that have been executed (in milliseconds).
Total Sorts	The total number of sorts that have been executed.

3.4.12 SQL Statement on Database

Overview

The SQL Statement on Database report displays SQL statement information at the database level, in real time.

Storage Location

Reports/DB2 Universal Database/Troubleshooting/Real Time/SQL Statement on Database

Record

SQL Statement on Database (PD_DSQD)

Table 3.13 Report Data for SQL Statement on Database

Fields	
Field Name	Description
Commit SQL STMTS	The total number of SQL COMMIT statements that have been attempted.
DDL SQL STMTS	The number of SQL Data Definition Language (DDL) statements that were executed.
DDL SQL STMTS Rate	The ratio of the SQL Data Definition Language (DDL) statements that were executed.
Dynamic SQL STMTS	The number of dynamic SQL statements that were attempted.
Failed SQL STMTS	The number of SQL statements that were attempted, but failed.
Priv Workspace Num Overflows	The number of times that the private workspace overflowed out of the bounds of its allocated memory.
Priv Workspace Section Inserts	The number of SQL sections inserted into the private workspace by applications.
Priv Workspace Section Lookups	The number of SQL sections that were referenced by applications in the agent private workspace.
Priv Workspace Size Top	The largest size reached by the private workspace.
Rollback SQL STMTS	The total number of SQL ROLLBACK statements that have been attempted.
Select SQL STMTS	The number of SQL SELECT statements that were executed.
Select SQL STMTS Rate	The ratio of the SELECT statements to all statements.
Shr Workspace Hit Rate	Rate of shared workspace hits.
Shr Workspace Num Overflows	The number of times the shared workspace overflowed out of the bounds of its allocated memory.
Shr Workspace Section Inserts	The number of SQL sections inserted into the shared workspace by applications.
Shr Workspace Section Lookups	The number of SQL sections referenced by applications in the shared workspace.
Shr Workspace Size Top	The maximum size reached by the shared workspace.
Static SQL STMTS	The number of static SQL statements that were attempted.
Throughput SQL STMTS	The total number of successful SQL statements.

PRELIMINARY RELEASE

Fields	
Field Name	Description
UID SQL STMTS	The number of SQL UPDATE, INSERT and DELETE statements that were executed.
UID SQL STMTS Rate	The ratio of the SQL UPDATE, INSERT and DELETE statements that were executed.

Chapter 4 Working with Records

This chapter describes the records for HTM Agent for DB2, which are used by Performance Reporter. For details about collecting performance data, see the chapter that provides an overview of data handled by the Tuning Manager products or the chapter that describes management of the Store database, in the *HiCommand™ Tuning Manager User's Guide*.

- Data Model (see section 4.1)
- Format of Record Explanations (see section 4.2)
- List of Common Key Fields (see section 4.3)
- List of Data Types (see section 4.4)
- Field Values (see section 4.5)
- Fields Added Only When Data is Stored in the Store Database (see section 4.6)
- Notes on Records (see section 4.7)
- List of Records for HTM Agent for DB2 (see section 4.8)

4.1 Data Model

Each Agent's records and fields are referred to collectively as a *data model*. There is a specific version number for each Agent and its data model. The data model version for HTM Agent for DB2 04-10 is version 3.0.

For details about the data model, see the chapter that provides an overview of the data handled by the Tuning Manager series programs, in the *HiCommand™ Tuning Manager User's Guide*.

4.2 Format of Record Contents

The records for HTM Agent for DB2 contain six subsections:

- Function
- Default and changeable values
- Key field
- Lifetime
- Record size
- Fields

4.2.1 Function

Provides an overview of the performance data that is stored in the record and includes important information that should be noted.

4.2.2 Default and Changeable Values

Consists of a table of the default values for the performance data under the collection conditions that are defined for the record, and indicates whether the values can be changed by the user. You can use Performance Reporter's `jpcasrec output` command to display each record's values. For details about the items in the table, see the chapter that describes the commands, in the *HiCommand™ Tuning Manager Performance Reporter Reference*.

The following table uses the explanation format to describe each item and its value.

Table 4.1 Default and Changeable Values

Item	Default Value	Changeable
Collection Interval	<p>Performance data collection interval (in seconds)</p> <p>The default value is recommended. Specify one of the following if you wish to change the value. Specifiable values:</p> <ul style="list-style-type: none"> ▪ 0 ▪ A multiple of 60 that is also a whole divisor of 3600, from 60 to 3,600 ▪ A multiple of 3,600 that is also a whole divisor of 86,400, from 3,600 to 86,400 <p>If a value other than those described above is specified, performance data might not be stored correctly. If 0 is set, performance data is not stored.</p>	<p>Whether the value is changeable.</p> <p>Legend: Yes: Changeable No: Not changeable</p>
Collection Offset See <i>Note</i>	Offset value for starting performance data collection (in seconds).	
Log	<p>Whether the collected performance data is stored in the Store database:</p> <p>Yes: Store (however, if <code>Collection Interval=0</code> is set, collected performance data is not stored).</p> <p>No: Do not store.</p>	
LOGIF	Conditions for storing collected performance data in the Store database	

Note: The range of values is from 0 to 32,767 (inclusive) seconds (within the value range specified for Collection Interval). This is used to distribute the collection processing workload because data collection is concentrated when multiple data items are collected. The data collection time that is recorded is the same as for the Collection Interval regardless of the value of Collection Offset.

If you change the value of Collection Offset, you should take into account the collection processing workload.

4.2.3 Key Fields

Key fields constitute record keys and are used for storing and searching records in the Store database. Key fields are listed in *view-name (manager-name)* format.

4.2.4 Lifetime

Indicates the period during which consistency is guaranteed for the performance data that is collected in the record. For details about the lifetime, see the chapter that provides an overview of the data handled by Tuning Manager series programs, in the *HiCommand™ Tuning Manager User's Guide*.

4.2.5 Record Size

Indicates the amount of performance data that can be collected and stored in each record at one time.

4.2.6 Fields

Provides a table that describes the fields of each record. The table contains the following items:

- **View name (Manager name)**

- View name

- Field name that is displayed with Performance Reporter.

- Manager name

- Field name (Tuning Manager name) to be specified in SQL statements when SQL statements are used from Tuning Manager to access the field data stored in the Store database.

- In SQL statements, specify a Manager name by appending the record ID assigned for the record at the beginning of the field name. For example, to specify the DB Name (DB_NAME) field of the Basic Information on Database (PD_DBID) record, specify PD_DBID_DB_NAME.

- **Description**
Explanation of the performance data that is stored in the field. For details, see the DB2 manual.
- **Format**
Data type of the field value, such as `char` or `float`. For details about the data types, see section 4.4.
- **Delta**
Whether the collected data represents the cumulative change in quantitative values. For details about delta, see section 4.5.
- **Supported Version**
Indicates the DB2 version that can use the field.
If a version number is shown, the field is supported by that version and all subsequent versions. *All* means that all DB2 versions support the field. *Not supported* means that the field is not supported by DB2.
- **Restriction**
Indicates the presence or absence of restrictions for the corresponding field value. For information on field value restrictions, see section 4.5.
- **Data source**
Method used to obtain the field value or the source of the data. For details about field values, see section 4.5.

4.3 List of Common Key Fields

There are two types of key fields: One is common among all records, while the other is unique to each record. This section describes the key fields that are common to all records, as listed in Table 4.2. For details about key fields specific to a particular record, see the description of each record.

Table 4.2 List of Common Key Fields

View Name (Manager Name)	Description
Agent Host (DEVICEID)	Instance name and host name of the operating Agent
Agent Instance (PROD_INST)	Instance name of the Agent
Agent Type (PRODID)	Product ID of the Agent
Date (DATE)	Record creation date (GMT)
Date and Time (DATETIME)	A combination of the DATE and TIME fields
Drawer Type (DRAWER_TYPE)	The drawer type. Valid values are as follows: m: minute H: hour D: date w: week M: month Y: year
Record Type (INPUT_RECORD_TYPE)	Identifier that indicates the record type (4 bytes)
Time (TIME)	Record creation time (GMT)

4.4 List of Data Types

Table 4.3 lists the data types for field values, and their corresponding C and C++ data types. The values shown in the *Format* column of the record field tables are those shown below in the *Field* column under *Data type*.

Table 4.3 List of Data Types

Data Type			
Field	C and C++	Byte	Description
char(<i>n</i>)	char()	1	Character data (0x20 to 0x7e)
double	double	8	Numeric value (1.7E±308 (15 digits))
float	float	4	Numeric value (3.4E±38 (7 digits))
long	long	4	Numeric value (-2,147,483,648 to 2,147,483,647)
short	short	2	Numeric value (-32,768 to 32,767)
string(<i>n</i>)	char[]	Number in parentheses	Character string with a length of <i>n</i> bytes. The last character is null.
time_t	unsigned long	4	Numeric value (0 to 4,294,967,295)
timeval	Structure	8	Numeric value (first 4 bytes are seconds, next 4 bytes are microseconds)
ulong	unsigned long	4	Numeric value (0 to 4,294,967,295)
utime	Structure	8	Numeric value (first 4 bytes are seconds, next 4 bytes are microseconds)
word	unsigned short	2	Numeric value (0 to 65,535)
(Not applicable)	unsigned char	1	Numeric value (0 to 255)

4.5 Field Values

This section describes the values that are stored in the fields:

- Data source
- Restriction
- Delta

4.5.1 Data Source

Each field contains a value obtained from a Tuning Manager product or program being monitored, or the value derived through the application of a formula. In the tables, the *Data source* column indicates the source of the value or the formula used to produce the value.

When a field's value is obtained by processing performance data acquired from DB2, the character string in the *Data source* column of HTM Agent for DB2 indicates the method used to obtain the value that is set in the field. The following are two such examples.

- **When uppercase characters are shown**
Uppercase characters indicate the key character string used to acquire the performance data that is stored in the DB2.
- **When Agent Collector is shown**
Agent Collector means that the value stored in the field was obtained from the Agent Collector service.

4.5.2 Restriction

Each field contains a value obtained from DB2 for the monitoring target, or a value based on other such values and calculated according to a given formula. The presence or absence of restrictions for the field value is indicated in the *Restriction* column.

- **When a hyphen (-) is shown**

There is no restriction on the value of each field.

- **When Yes is shown**

Depending on the monitoring target, the value data obtained from DB2 may exceed 32 bits. When the data exceeds 32 bits, there is a restriction on the value stored in the field. For example, if the value 0x1122334455667788 has been obtained, the value inserted in the field will be as indicated in the following table.

OS	Data Storage Method	Higher Address	Lower Address	Value Stored in the Field
AIX (pSeries)	Big-endian architecture	0x11 0x22 0x33 0x44	0x55 0x66 0x77 0x88	0x55 0x66 0x77 0x88
Linux 3 (Intel x86)	Little-endian architecture	0x55 0x66 0x77 0x88	0x11 0x22 0x33 0x44	0x55 0x66 0x77 0x88

4.5.3 Delta

For data collected as an accumulated value, the expression of quantitative change is called *delta*. For example, if the performance data value obtained during the first collection is 3 and the performance data value obtained during the second collection is 4, then the accumulated value is 7 and the change is 1. In the tables, the *Delta* column indicates whether each field's value is a delta value.

Table 4.4 explains the various types of performance data collected by HTM Agent for DB2.

Table 4.4 Performance Data Collected by HTM Agent for DB2

Record Type	Delta	Data Type	Value of Delta is Set to be Displayed See <i>Note</i>	Record Value
PI record type	Yes	Real-time data	Yes	The displayed value indicates the amount of change between amounts of changes.
			No	The displayed value indicates the amount of change.

PRELIMINARY RELEASE

Record Type	Delta	Data Type	Value of Delta is Set to be Displayed See <i>Note</i>	Record Value
PI record type	Yes	Historical data	Not applicable	The displayed value indicates the amount of change.
	No	Real-time data	Yes	The displayed value is the value at the time of data collection.
			No	The displayed value is the value at the time of data collection.
		Historical data	Not applicable	The displayed value is the value at the time of data collection.
PD record type	Yes	Real-time data	Yes	The displayed value indicates the amount of change.
			No	The displayed value indicates the accumulated value
		Historical data	Not applicable	The displayed value indicates the accumulated value
	No	Real-time data	Yes	The displayed value is the value at the time of data collection.
			No	The displayed value is the value at the time of data collection.
		Historical data	Not applicable	The displayed value is the value at the time of data collection.

Note: the following settings are applied in Performance Reporter:

- Indicate delta value checkbox is selected in the [New Report > Indication Settings (Realtime)] window.
- The Indicate delta value checkbox is selected in the [Show Options] window for a real-time report.
- The indicate-delta-value attribute of realtime-indication-settings, in the parameter file that is used as an argument for the `jpccrdef create` command, is specified as *TRUE*.

Please note the following points about the collection of performance data:

- In order for a record of the PI record type to be saved, the performance data must be collected at least twice.

In records of the PI record type, performance data is collected at each collection interval specified by Performance Reporter's `jpcasrec update` command. Note, however, that performance data is not stored in the Store database at the performance data collection time set by Performance Reporter's `jpcasrec update` command.

Historical data for records of the PI record type requires two data collections in order to obtain the value (called delta) resulting from calculating the difference between two sets of data. Thus, it takes up to twice as much time as the specified time value before the historical data is stored in the Store database.

For example, if the `jpcasrec update` command of Performance Reporter is used to set the collection interval for performance data to 300 seconds (5 minutes) at 18:32, the first data collection will begin at 18:35, and the next data collection will begin at 18:40. This enables historical data to be created from the data that was collected at 18:35 and 18:40, which is then stored in the Store database 8 minutes later.

- A real-time report displays data values starting with the first collection. Note, however, that 0 is displayed as the first collected value in reports requiring previously collected data. The values displayed for the second and subsequent data collections vary depending on the report.
- In the following cases, the value of the collected data is displayed, at the time of the second collection and for all subsequent data collections.
 - In a real-time report of the PI record type, when a delta value is not set to be displayed.
 - In a real-time report of the PD record type, when a delta value is set to be displayed.
- In the following case, the difference between the data obtained at the first and second collections is displayed at the time of the second collection. For the third collection and all subsequent data collections, the value of the collected data is displayed.

In a real-time report of the PI record type, when a delta value is set to be displayed.

4.6 Fields Added Only When Data is Stored in the Store Database

The following table lists the fields that are added only when data is stored in the Store database.

Table 4.5 Fields that are Added when Data is Stored in the Store Database

View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction
Agent Host (DEVICEID)	Names of instance and host where Agent is running The format is as follows: <i>instance-name [host-name]</i>	string(256)	No	All	— (The field's value is obtained without processing the performance data acquired from DB2)
Agent Instance (PROD_INST)	Instance name of Agent	string(256)			
Agent Type (PRODIG)	Product ID of Agent (1-byte identifier)	char			
Date (DATE)	Record creation date (GMT) See <i>Note</i>	time_t			
Date and Time (DATETIME)	Combination of the Date (DATE) and Time (TIME) fields	time_t			
Drawer Type (DRAWER_TYPE)	For a PI-type record, the data summarization type	char			
GMT Offset (GMT_ADJUST)	Difference (in seconds) between Greenwich Mean Time and local time	utime			
Time (TIME)	Record creation time (GMT) See <i>Note</i>	time_t			

Note: Records of the PI record type summarize data from a specified period of time. The value set in this field indicates the beginning of that period. The following table lists the set values for each record type.

Table 4.6 Values for Each PI Record Type

Type	Set Value for Each Record Type
Minute	The beginning (0 th second) of the minute for which the record was created.
Hour	The beginning (0 th minute and 0 th second) of the hour for which the record was created.
Day	The beginning (0 th hour, 0 th minute, and 0 th second, AM) of the day for which the record was created.
Week	The beginning (0 th hour, 0 th minute, and 0 th second, AM on Monday) of the week for which the record was created.
Month	The beginning (0 th hour, 0 th minute, and 0 th second, AM on the 1 st) of the month for which the record was created.
Year	The beginning (0 th hour, 0 th minute, and 0 th second, AM on January 1 st) of the year for which the record was created.

4.7 Notes on Records

Record generation result when data cannot be obtained

This section explains the record generation result when the data to be stored in the field cannot be obtained. If an error occurred during performance data collection, or if the generated record contains a field that the monitored object does not support, then the record generation result will be as follows: The record is not generated.

Records are not created when HTM Agent for DB2 cannot acquire the performance data to be stored in a field that indicates the DB2 performance value.

4.8 List of Records for HTM Agent for DB2

Table 4.7 lists the records that can be collected by HTM Agent for DB2 and the information that is stored in each record.

Table 4.7 List of Records for HTM Agent for DB2

Record Name	Record ID	Information Stored in Record
Basic Information on Database	(PD_DBID)	Basic information in database units on the database level. See <i>Note</i>
Basic Information on Database Manager	(PD_PD)	Basic information in database units on the database manager level.
Basic Information on Database Manager Interval	(PI_PI)	Information regarding basic information in database units on the database manager level, using constant units.
Bufferpool Stat on Database	(PD_DBPD)	Information about the buffer pool on the database level. See <i>Note</i>
Bufferpool Stat on Database Interval	(PI_DBPI)	Information about the buffer pool on the database level, using a constant interval as a unit. See <i>Note</i>
Cache on Database	(PD_DCAD)	Information about the cache on the database level. See <i>Note</i>
Cache on Database Interval	(PI_DCAI)	Information about the cache on the database level using a constant interval as a unit. See <i>Note</i>
DB2 Configuration	(PD_DCFE)	Information about DB2 configuration parameters.
Sort Status on Database	(PD_DSOD)	Information about sorting on the database level. See <i>Note</i>
SQL Statement on Database	(PD_DSQD)	Information about SQL Statement on the database level. See <i>Note</i>

Note: Database level information is collected only when connected to one or more applications.

4.8.1 Basic Information on Database (PD_DBID)

Function

Stores basic information about database units on the database level.

Table 4.8 Basic Information on Database (PD_DBID) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

While one or more applications are connected to the database.

Record size

- Fixed part: 1,030 bytes
- Variable part: 0 bytes

Table 4.9 Basic Information on Database (PD_DBID) Fields (continues on the following pages)

Basic Information on Database (PD_DBID)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Agents Top (AGENTS_TOP)	On the application level, the maximum number of agents used when executing a statement. On database level, the maximum number of agents for an application.	ulong	No	All	—	SQLM_ELM_AGENTS_TOP
Appl Section Inserts (APPL_SECTION_INSERTS)						
Appl Section Lookups (APPL_SECTION_LOOKUPS)	Number of SQL section references by applications from the SQL work area.	ulong	No	All	—	SQLM_ELM_APPL_SECTION_LOOKUPS

PRELIMINARY RELEASE

Basic Information on Database (PD_DBID)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Appls Cur Cons (APPLS_CUR_CONS)	Number of applications connected to this database.	ulong	No	All	—	SQLM_ELM_APPLS_CUR_CONS
Appls In DB2 (APPLS_IN_DB2)						
Coord Agents Top (COORD_AGENTS_TOP)	Maximum number of coordinator agents that can operate simultaneously.	ulong	No	All	—	SQLM_ELM_COORD_AGENTS_TOP
DB Heap Top (DB_HEAP_TOP)						
DB Name (DB_NAME)	The real name of the database where the information is collected, or the real name of the database at the application's connection target.	string (9)	No	All	—	SQLM_ELM_DB_NAME
DB Path (DB_PATH)						
DB Status (DB_STATUS)	Current database status.	ulong	No	All	—	SQLM_ELM_DB_STATUS
Hash Join Overflows (HASH_JOIN_OVERFLOW)						
Hash Join Overflows Rate (HASH_JOIN_OVERFLOW_RATE)	Ratio of hash_join_small_overflows and hash_join_overflows.	float	No	All	—	(HASH_JOIN_SMALL_OVERFLOW / HASH_JOIN_OVERFLOW) *100
Hash Join Small Overflows (HASH_JOIN_SMALL_OVERFLOW)						
Interval (interval)	Time for information collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Last Backup (LAST_BACKUP)						
	Most recent database backup completion date.	string (27)	No	All	—	SQLM_ELM_LAST_BACKUP

PRELIMINARY RELEASE

Basic Information on Database (PD_DBID)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Num Assoc Agents (NUM_ASSOC_AGENTS)	On the application level, the number of subagents associated with one application. On the database level, the total number of subagents for all applications.	ulong	No	All	—	SQLM_ELM_NUM_ASSOC_AGENTS
Recode Time (RECODE_TIME)	Collection end time for the performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always DBID.	char(8)	No	All	—	Agent Collector
Total Hash Joins (TOTAL_HASH_JOINS)	Total number of executed hash joins.	ulong	No	All	—	SQLM_ELM_TOTAL_HASH_JOINS
Total Hash Loops (TOTAL_HASH_LOOPS)	The total number of times when the single partition of the hash join was larger than the usable sort heap space.	ulong	No	All	—	SQLM_ELM_TOTAL_HASH_LOOPS

4.8.2 Basic Information on Database Manager (PD_PD)

Function

Stores basic information about database manager units on the database manager level.

Table 4.10 Basic Information on Database Manager (PD_PD) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

From the creation to the deletion of the database.

Record size

- Fixed part: 845 bytes
- Variable part: 0 bytes

Table 4.11 Basic Information on Database Manager (PD_PD) Fields (continues on the following pages)

Basic Information on Database Manager (PD_PD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Agents Created Empty Pool (AGENTS_CREATED_EMPTY_POOL)	Number of agents created due to the agent pool being empty.	ulong	No	All	—	SQLM_ELM_AGENTS_CREATED_EMPTY_POOL
Agents Created Pool Rate (AGENTS_CREATED_POOL_RATE)						
Agents From Pool (AGENTS_FROM_POOL)	The number of agents assigned from the agent pool.	ulong	No	All	—	SQLM_ELM_AGENTS_FROM_POOL

PRELIMINARY RELEASE

Basic Information on Database Manager (PD_PD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Agents Registered (AGENTS_REGISTERED)	The number of agents (coordinator agents and subagents) registered with the monitored database manager instance.	ulong	No	All	—	SQLM_ELM_AGENTS_REGISTERED
Agents Registered Top (AGENTS_REGISTERED_TOP)						
Agents Waiting On Token (AGENTS_WAITING_ON_TOKEN)	The number of agents waiting for tokens to execute transactions in the database manager.	ulong	No	All	—	SQLM_ELM_AGENTS_WAITING_ON_TOKEN
Agents Waiting Top (AGENTS_WAITING_TOP)						
Comm Private Mem (COMM_PRIVATE_MEM)	Amount of dedicated memory committed by the database manager instance at the time the snapshot was taken.	ulong	No	All	—	SQLM_ELM_COMM_PRIVATE_MEM
Cons In Exec Total (CONS_IN_EXEC_TOTAL)						
Cons Total (CONS_TOTAL)	Total number of applications connected to the monitored database manager instance or the database.	ulong	No	All	—	(REM_CONS_IN_EXEC + LOCAL_CONS_IN_EXEC)
DB2 Status (DB2_STATUS)						
Interval (interval)	Time for the information collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Local Cons (LOCAL_CONS)						

PRELIMINARY RELEASE

Basic Information on Database Manager (PD_PD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Local Cons In Exec (LOCAL_CONS_IN_EXEC)	Number of local applications that are connected to a database in the monitored database manager instance, and are also processing task units.	ulong	No	All	—	SQLM_ELM_LOCAL_CONS_IN_EXEC
Post Threshold Hash Joins (POST_THRESHOLD_HASH_JOINS)	Total number of times that a hash join heap request was limited because the shared or dedicated sort heap space was concurrently used.	ulong	No	All	—	SQLM_ELM_POST_THRESHOLD_HASH_JOINS
Product Name (PRODUCT_NAME)	Detailed version information on the currently executing DB2 instance.	string (48)	No	All	—	SQLM_ELM_PRODUCT_NAME
Recode Time (RECODE_TIME)	Collection end time for the performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always PD.	char (8)	No	All	—	Agent Collector
Rem Cons In (REM_CONS_IN)	Number of connections started from a remote client to the monitored database manager instance.	ulong	No	All	—	SQLM_ELM_REM_CONS_IN
Rem Cons In Exec (REM_CONS_IN_EXEC)	Number of local applications that are connected to a database in the monitored database manager instance, and are also processing task units.	ulong	No	All	—	SQLM_ELM_REM_CONS_IN_EXEC
Service Level (SERVICE_LEVEL)	Current corrected service level for the DB2 instance.	string (48)	No	All	—	SQLM_ELM_SERVICE_LEVEL
Sort Heap Allocated (SORT_HEAP_ALLOCATED)	The total number of sort heap space pages allocated to all sort operations at the selected level, at the time the snapshot was taken.	ulong	No	All	—	SQLM_ELM_SORT_HEAP_ALLOCATED

4.8.3 Basic Information on Database Manager Interval (PI_PI)

Function

This record stores the information for a certain interval based on basic information at the database manager level, in database units.

Table 4.12 Basic Information on Database Manager Interval (PI_PI) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	Yes	
LOGIF	(Blank)	

Key fields

None.

Lifetime

From the creation to the deletion of the database.

Record size

- Fixed part: 857 bytes
- Variable part: 0 bytes

Table 4.13 Basic Information on Database Manager Interval (PI_PI) Fields (continues on the following pages)

Basic Information on Database Manager Interval (PI_PI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Agents Created Empty Pool (AGENTS_CREATED_EMPTY_POOL)	Number of agents created due to the agent pool being empty.	ulong	No	All	—	SQLM_ELM_AGENTS_CREATED_EMPTY_POOL
Agents Created Pool Rate (AGENTS_CREATED_POOL_RATE)						
Agents From Pool (AGENTS_FROM_POOL)	The number of agents assigned from the agent pool.	ulong	No	All	—	SQLM_ELM_AGENTS_FROM_POOL
Agents Registered	The number of agents	ulong	No	All	—	SQLM_ELM_AGENTS

PRELIMINARY RELEASE

Basic Information on Database Manager Interval (PI.PI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
(AGENTS_REGISTERED)	(coordinator agents and subagents) registered with the monitored database manager instance.					_REGISTERED
Agents Registered Top	The maximum number of agents (coordinator agents and subagents) registered simultaneously with the database manager since startup.	ulong	No	All	—	SQLM_ELM_AGENTS_REGISTERED_TOP
(AGENTS_REGISTERED_TOP)						
Agents Waiting On Token	The number of agents waiting for tokens to execute a transactions in the database manager.	ulong	No	All	—	SQLM_ELM_AGENTS_WAITING_ON_TOKEN
(AGENTS_WAITING_ON_TOKEN)						
Agents Waiting Top	The maximum number of agents simultaneously waiting for tokens since startup.	ulong	No	All	—	SQLM_ELM_AGENTS_WAITING_TOP
(AGENTS_WAITING_TOP)						
Comm Private Mem	Amount of dedicated memory committed by the database manager instance at the time the snapshot was taken.	ulong	No	All	—	SQLM_ELM_COMM_PRIVATE_MEM
(COMM_PRIVATE_MEM)						
Cons In Exec Total	Total number of applications that are connected to a database in the monitored database manager instance, and are also processing task units.	ulong	No	All	—	(REM_CONS_IN_EXEC + LOCAL_CONS_IN_EXEC)
(CONS_IN_EXEC_TOTAL)						
Cons Total	Total number of applications connected to monitored database manager instance or database.	ulong	No	All	—	(REM_CONS_IN + LOCAL_CONS)
(CONS_TOTAL)						
DB2 Status	Current status of the database manager instance.	ulong	No	All	—	SQLM_ELM_DB2_STATUS
(DB2_STATUS)						
Interval	Time for the information collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
(interval)						
Local Cons	Number of local applications connected to the database in the monitored database manager instance.	ulong	No	All	—	SQLM_ELM_LOCAL_CONS
(LOCAL_CONS)						
Local Cons In Exec	Number of local applications that are connected to a database in the monitored database manager instance, and are also processing task units.	ulong	No	All	—	SQLM_ELM_LOCAL_CONS_IN_EXEC
(LOCAL_CONS_IN_EXEC)						
Post Threshold	Total number of times that a	ulong	No	All	—	SQLM_ELM_POST_T

PRELIMINARY RELEASE

Basic Information on Database Manager Interval (PI.PI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Hash Joins (POST_THRESHOLD_HASH_JOINS)	hash join heap request was limited because the shared or dedicated sort heap space was concurrently used.					HRESHOLD_HASH_JOINS
Product Name (PRODUCT_NAME)	Detailed version information on the currently executing DB2 instance.	string (48)	No	All	—	SQLM_ELM_PRODUCT_NAME
Recode Time (RECODE_TIME)	Collection end time for the performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always PI.	char(8)	No	All	—	Agent Collector
Rem Cons In (REM_CONS_IN)	Number of connections started from a remote client to the monitored database manager instance.	ulong	No	All	—	SQLM_ELM_REM_CONS_IN
Rem Cons In Exec (REM_CONS_IN_EXEC)	Number of local applications that are connected to a database in the monitored database manager instance, and are also processing task units.	ulong	No	All	—	SQLM_ELM_REM_CONS_IN_EXEC
Service Level (SERVICE_LEVEL)	Current corrected service level for the DB2 instance.	string (48)	No	All	—	SQLM_ELM_SERVICE_LEVEL
Sort Heap Allocated (SORT_HEAP_ALLOCATED)	The total number of sort heap space pages allocated to all sort operations at the selected level, at the time the snapshot was taken.	ulong	No	All	—	SQLM_ELM_SORT_HEAP_ALLOCATED

4.8.4 Bufferpool Stat on Database (PD_DBPD)

Function

Stores information about the buffer pool on the database level.

Table 4.14 Bufferpool Stat on Database (PD_DBPD) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

From the creation to the deletion of the database.

Record size

- Fixed part: 825 bytes
- Variable part: 0 bytes

Table 4.15 Bufferpool Stat on Database (PD_DBPD) Fields (continues on the following pages)

Bufferpool Stat on Database (PD_DBPD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Interval (interval)	Interval at which the information is collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Pool Async Data Reads (POOL_ASYNC_DATA_READS)	Number of asynchronous page reads into the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_ASYNC_DATA_READS
Pool Async Data Reads Rate (POOL_ASYNC_DATA_READS_RATE)	Ratio of the prefetcher's asynchronous data page reads into the buffer pool to the number of read requests requiring I/O for entering data pages into the buffer pool.	float	No	All	—	(POOL_ASYNC_DATA_READS / POOL_DATA_P_READS) * 100

PRELIMINARY RELEASE

Bufferpool Stat on Database (PD_DBPD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Async Data Writes (POOL_ASYNC_DATA_WRITES)	The number of physical writes of buffer pool data pages to disk, written by the asynchronous page cleaner or the prefetcher.	ulong	No	All	Y	SQLM_ELM_POOL_ASYNC_DATA_WRITES
Pool Async Data Writes Rate (POOL_ASYNC_DATA_WRITES_RATE)	Ratio of physical disk writes for buffer pool data pages written by the asynchronous page cleaner or prefetcher, to the number of physical disk writes for buffer pool data pages.	float	No	All	—	(POOL_ASYNC_DATA_WRITES / POOL_DATA_WRITES) * 100
Pool Async Index Reads (POOL_ASYNC_INDEX_READS)	Number of asynchronous index page reads by the prefetcher into the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_ASYNC_INDEX_READS
Pool Async Index Reads Rate (POOL_ASYNC_INDEX_READS_RATE)	Ratio of asynchronous index page reads by the prefetcher into the buffer pool, to the number of physical read requests to enter index pages into the buffer pool.	float	No	All	—	(POOL_ASYNC_INDEX_READS / POOL_INDEX_P_READS) * 100
Pool Async Index Writes (POOL_ASYNC_INDEX_WRITES)	Number of physical disk writes for buffer pool index pages by the asynchronous page cleaner or the prefetcher.	ulong	No	All	Y	SQLM_ELM_POOL_ASYNC_INDEX_WRITES
Pool Async Index Writes Rate (POOL_ASYNC_INDEX_WRITES_RATE)	Ratio of physical disk writes for buffer pool index pages by the asynchronous page cleaner or the prefetcher, to the number of physical disk writes for buffer pool index pages.	float	No	All	—	(POOL_ASYNC_INDEX_WRITES / POOL_INDEX_WRITES) * 100
Pool Async Read Time (POOL_ASYNC_READ_TIME)	Total elapsed time that was required by the database manager's prefetcher for read operations.	ulong	No	All	Y	SQLM_ELM_POOL_ASYNC_READ_TIME
Pool Async Read Time Avg (POOL_ASYNC_READ_TIME_AVG)	Average of the total elapsed time that was required by the database manager's prefetcher for read operations.	float	No	All	—	POOL_ASYNC_READ_TIME / (POOL_ASYNC_DATA_READS + POOL_ASYNC_INDEX_READS)

PRELIMINARY RELEASE

Bufferpool Stat on Database (PD_DBPD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Async Total Reads (POOL_ASYNC_TOTAL_READS)	Total of data pages and index pages read in asynchronous mode by the prefetcher into the buffer pool.	ulong	No	All	—	POOL_ASYNC_DATA_READS + POOL_ASYNC_INDEX_READS
Pool Async Total Writes (POOL_ASYNC_TOTAL_WRITES)	Number of physical disk writes for buffer pool data pages or index pages by the asynchronous page cleaner or the prefetcher.	ulong	No	All	—	POOL_ASYNC_DATA_WRITES + POOL_ASYNC_INDEX_WRITES
Pool Async Write Time (POOL_ASYNC_WRITE_TIME)	Total elapsed time, for writing buffer pool data pages or index pages from the buffer pool to disk, that was required by the page cleaner of the database manager.	ulong	No	All	Y	SQLM_ELM_POOL_ASYNC_WRITE_TIME
Pool Async Write Time Avg (POOL_ASYNC_WRITE_TIME_AVG)	Average of the total elapsed time that was required by the database manager's page cleaner for writing data pages or index pages from the buffer pool to disk.	float	No	All	—	POOL_ASYNC_WRITE_TIME / (POOL_ASYNC_DATA_WRITES + POOL_ASYNC_INDEX_WRITES)
Pool Data From Estore (POOL_DATA_FROM_ESTORE)	Number of data pages copied from extended storage to the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_DATA_FROM_ESTORE
Pool Data Hit Rate (POOL_DATA_HIT_RATE)	Data page hit rate.	float	No	All	—	(1 - (POOL_DATA_P_READS / POOL_DATA_L_READS)) * 100
Pool Data L Reads (POOL_DATA_L_READS)	Number of logical read requests for data pages that passed through the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_DATA_L_READS
Pool Data P Reads (POOL_DATA_P_READS)	Number of read requests for which I/O was required to enter data pages into the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_DATA_P_READS
Pool Data To Estore (POOL_DATA_TO_ESTORE)	Number of data pages copied from the buffer pool to extended storage.	ulong	No	All	Y	SQLM_ELM_POOL_DATA_TO_ESTORE
Pool Data Writes (POOL_DATA_WRITES)	Number of physical disk writes for buffer pool data pages.	ulong	No	All	Y	SQLM_ELM_POOL_DATA_WRITES

PRELIMINARY RELEASE

Bufferpool Stat on Database (PD_DBPD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Drty PG Steal Clns (POOL_DRTY_PG_STEAL_CLNS)	Number of page cleaner calls for synchronous writes required by database victim buffer replacement.	ulong	No	All	Y	SQLM_ELM_POOL_DRTY_PG_STEAL_CLNS
Pool Drty PG Steal Clns Rate (POOL_DRTY_PG_STEAL_CLNS_RATE)						
Pool Drty PG Steal Clns Rate (POOL_DRTY_PG_STEAL_CLNS_RATE)	Ratio of the number of page cleaner calls for synchronous writes required by database victim buffer replacement, to the total number of cleaner calls.	float	No	All	—	$\frac{\text{POOL_DRTY_PG_STEAL_CLNS}}{\text{POOL_DRTY_PG_STEAL_CLNS} + \text{POOL_DRTY_PG_THRSH_CLNS} + \text{POOL_LSN_GAP_CLNS}}$
Pool Drty PG Thrsh Clns (POOL_DRTY_PG_THRSH_CLNS)	Number of page cleaner calls due to the buffer pool reaching the dirty page threshold for the database.	ulong	No	All	Y	SQLM_ELM_POOL_DRTY_PG_THRSH_CLNS
Pool Index From Estore (POOL_INDEX_FROM_ESTORE)						
Pool Index From Estore (POOL_INDEX_FROM_ESTORE)	Number of buffer pool index pages copied from extended storage.	ulong	No	All	Y	SQLM_ELM_POOL_INDEX_FROM_ESTORE
Pool Index Hit Rate (POOL_INDEX_HIT_RATE)	Index page hit rate.	float	No	All	—	$\left(1 - \left(\frac{\text{POOL_INDEX_P_READS}}{\text{POOL_INDEX_L_READS}} \right) \right) * 100$
Pool Index L Reads (POOL_INDEX_L_READS)						
Pool Index L Reads (POOL_INDEX_L_READS)	Number of logical read requests for index pages that passed through the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_INDEX_L_READS
Pool Index P Reads (POOL_INDEX_P_READS)	Number of physical read requests to enter index pages into the buffer pool.	ulong	No	All	Y	SQLM_ELM_POOL_INDEX_P_READS
Pool Index To Estore (POOL_INDEX_TO_ESTORE)						
Pool Index To Estore (POOL_INDEX_TO_ESTORE)	Number of index pages copied from the buffer pool to extended storage.	ulong	No	All	Y	SQLM_ELM_POOL_INDEX_TO_ESTORE
Pool Index Writes (POOL_INDEX_WRITES)	Number of physical disk writes for buffer pool index pages.	ulong	No	All	Y	SQLM_ELM_POOL_INDEX_WRITES
Pool Lsn Gap Clns (POOL_LSN_GAP_CLNS)						
Pool Lsn Gap Clns (POOL_LSN_GAP_CLNS)	Number of times the page cleaner was called because the logging space in use reached the defined standard for the database.	ulong	No	All	Y	SQLM_ELM_POOL_LSN_GAP_CLNS

PRELIMINARY RELEASE

Bufferpool Stat on Database (PD_DBPD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Read Time (POOL_READ_TIME)	Total time for read request processing, where physical reads of data or index pages, from a disk or the buffer pool, were necessary.	ulong	No	All	Y	SQLM_ELM_POOL_READ_TIME
Pool Read Time Avg (POOL_READ_TIME_AVG)						
Pool Read Time Avg (POOL_READ_TIME_AVG)	Average time for read request processing, where physical reads of data or index pages, from a disk or the buffer pool, were necessary.	float	No	All	—	POOL_READ_TIME / (POOL_DATA_P_READS + POOL_INDEX_P_READS)
Pool Total Hit Rate (POOL_TOTAL_HIT_RATE)						
Pool Total Hit Rate (POOL_TOTAL_HIT_RATE)	Total buffer pool hit rate.	float	No	All	—	(1 - ((POOL_DATA_P_READS + POOL_INDEX_P_READS) / (POOL_DATA_L_READS + POOL_INDEX_L_READS))) * 100
Pool Total Writes (POOL_TOTAL_WRITES)						
Pool Total Writes (POOL_TOTAL_WRITES)	Number of disk writes from the buffer pool.	ulong	No	All	—	POOL_DATA_WRITES + POOL_INDEX_WRITES
Pool Write Time (POOL_WRITE_TIME)						
Pool Write Time (POOL_WRITE_TIME)	Total time taken for physical writes of data pages or index pages from the buffer pool to disk.	ulong	No	All	Y	SQLM_ELM_POOL_WRITE_TIME
Pool Write Time Avg (POOL_WRITE_TIME_AVG)						
Pool Write Time Avg (POOL_WRITE_TIME_AVG)	Average time taken for physical writes of data pages or index pages from the buffer pool to disk.	float	No	All	—	POOL_WRITE_TIME / (POOL_DATA_WRITES + POOL_INDEX_WRITES)
Recode Time (RECODE_TIME)						
Recode Time (RECODE_TIME)	Collection end time for performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)						
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always DBPD.	char(8)	No	All	—	Agent Collector

4.8.5 Bufferpool Stat on Database Interval (PI_DBPI)

Function

Stores basic information about the buffer pool on the database level, using a constant interval as a unit.

Table 4.16 Bufferpool Stat on Database Interval (PI_DBPI) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	Yes	
LOGIF	(Blank)	

Key fields

None.

Lifetime

While one or more applications are connected to the database.

Record size

- Fixed part: 969 bytes
- Variable part: 0 bytes

Table 4.17 Bufferpool Stat on Database Interval (PI_DBPI) Fields (continues on the following pages)

Bufferpool Stat on Database Interval (PI_DBPI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Interval (interval)	Interval at which the information is collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Pool Async Data Reads (POOL_ASYNC_DATA_READS)	Number of asynchronous page reads into the buffer pool.	ulong	Yes	All	Y	SQLM_ELM_POOL_ASYNC_DATA_READS
Pool Async Data Reads Rate (POOL_ASYNC_DATA_READS_RATE)	Ratio of the prefetcher's asynchronous data page reads into the buffer pool to the number of read requests requiring I/O for entering data pages into the buffer pool.	float	No	All	—	(POOL_ASYNC_DATA_READS / POOL_DATA_P_READS) * 100

PRELIMINARY RELEASE

Bufferpool Stat on Database Interval (PI_DBPI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Async Data Writes (POOL_ASYNC_DATA_WRITES)	The number of physical writes of buffer pool data pages to disk, written by the asynchronous page cleaner or the prefetcher.	ulong	Yes	All	Y	SQLM_ELM_POOL_ASYNC_DATA_WRITES
Pool Async Data Writes Rate (POOL_ASYNC_DATA_WRITES_RATE)	Ratio of physical disk writes for buffer pool data pages written by the asynchronous page cleaner or prefetcher, to the number of physical disk writes for buffer pool data pages.	float	No	All	—	(POOL_ASYNC_DATA_WRITES / POOL_DATA_WRITES) * 100
Pool Async Index Reads (POOL_ASYNC_INDEX_READS)	Number of asynchronous index page reads by the prefetcher into the buffer pool.	ulong	Yes	All	Y	SQLM_ELM_POOL_ASYNC_INDEX_READS
Pool Async Index Reads Rate (POOL_ASYNC_INDEX_READS_RATE)	Ratio of asynchronous index page reads by the prefetcher into the buffer pool, to the number of physical read requests to enter index pages into the buffer pool.	float	No	All	—	(POOL_ASYNC_INDEX_READS / POOL_INDEX_READS) * 100
Pool Async Index Writes (POOL_ASYNC_INDEX_WRITES)	Number of physical disk writes for buffer pool index pages by the asynchronous page cleaner or the prefetcher.	ulong	Yes	All	Y	SQLM_ELM_POOL_ASYNC_INDEX_WRITES
Pool Async Index Writes Rate (POOL_ASYNC_INDEX_WRITES_RATE)	Ratio of physical disk writes for buffer pool index pages by the asynchronous page cleaner or the prefetcher, to the number of physical disk writes for buffer pool index pages.	float	No	All	—	(POOL_ASYNC_INDEX_WRITES / POOL_INDEX_WRITES) * 100
Pool Async Read Time (POOL_ASYNC_READ_TIME)	Total elapsed time that was required by the database manager's prefetcher for read operations.	ulong	Yes	All	Y	SQLM_ELM_POOL_ASYNC_READ_TIME
Pool Async Read Time Avg (POOL_ASYNC_READ_TIME_AVG)	Average of the total elapsed time that was required by the database manager's prefetcher for read operations.	float	No	All	—	POOL_ASYNC_READ_TIME / (POOL_ASYNC_DATA_READS + POOL_ASYNC_INDEX_READS)
Pool Async Total Reads (POOL_ASYNC_TOTAL_READS)	Total of data pages and index pages read in asynchronous mode by the prefetcher into buffer pool.	ulong	Yes	All	—	POOL_ASYNC_DATA_READS + POOL_ASYNC_INDEX_READS

PRELIMINARY RELEASE

Bufferpool Stat on Database Interval (PI_DBPI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Async Total Writes (POOL_ASYNC_TOTAL_WRITES)	Number of physical disk writes for buffer pool data pages or index pages by the asynchronous page cleaner or the prefetcher.	ulong	Yes	All	—	POOL_ASYNC_DATA_WRITES + POOL_ASYNC_INDEX_WRITES
Pool Async Write Time (POOL_ASYNC_WRITE_TIME)						
Pool Async Write Time Avg (POOL_ASYNC_WRITE_TIME_AVG)	Average of the total elapsed time that was required by the database manager's page cleaner for writing data pages or index pages from the buffer pool to disk.	float	No	All	—	POOL_ASYNC_WRITE_TIME / (POOL_ASYNC_DATA_WRITES + POOL_ASYNC_INDEX_WRITES)
Pool Data From Estore (POOL_DATA_FROM_ESTORE)						
Pool Data Hit Rate (POOL_DATA_HIT_RATE)	Data page hit rate.	float	No	All	—	(1 - (POOL_DATA_P_READS / POOL_DATA_L_READS)) * 100
Pool Data L Reads (POOL_DATA_L_READS)						
Pool Data P Reads (POOL_DATA_P_READS)	Number of read requests for which I/O was required to enter data pages into the buffer pool.	ulong	Yes	All	Y	SQLM_ELM_POOL_DATA_P_READS
Pool Data To Estore (POOL_DATA_TO_ESTORE)						
Pool Data Writes (POOL_DATA_WRITES)	Number of physical disk writes for buffer pool data pages.	ulong	Yes	All	Y	SQLM_ELM_POOL_DATA_WRITES
Pool Drty PG Steal Clns (POOL_DRTY_PG_STEAL_CLNS)						

PRELIMINARY RELEASE

Bufferpool Stat on Database Interval (PI_DBPI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Dry PG Steal Clns Rate (POOL_DRTY_PG_STEAL_CLNS_RATE)	Ratio of the number of page cleaner calls for synchronous writes required by database victim buffer replacement, to the total number of cleaner calls.	float	No	All	—	POOL_DRTY_PG_STEAL_CLNS / (POOL_DRTY_PG_STEAL_CLNS + POOL_DRTY_PG_THRSH_CLNS + POOL_LSN_GAP_CLNS)
Pool Dry PG Thrsh Clns (POOL_DRTY_PG_THRSH_CLNS)	Number of page cleaner calls due to the buffer pool reaching the dirty page threshold for the database.	ulong	Yes	All	Y	SQLM_ELM_POOL_DRTY_PG_THRSH_CLNS
Pool Index From Estore (POOL_INDEX_FROM_ESTORE)	Number of buffer pool index pages copied from extended storage.	ulong	Yes	All	Y	SQLM_ELM_POOL_INDEX_FROM_ESTORE
Pool Index Hit Rate (POOL_INDEX_HIT_RATE)	Index page hit rate.	float	No	All	—	(1 - (POOL_INDEX_P_READS / POOL_INDEX_L_READS)) * 100
Pool Index L Reads (POOL_INDEX_L_READS)	Number of logical read requests for index pages that passed through the buffer pool.	ulong	Yes	All	Y	SQLM_ELM_POOL_INDEX_L_READS
Pool Index P Reads (POOL_INDEX_P_READS)	Number of physical read requests to enter index pages into the buffer pool.	ulong	Yes	All	Y	SQLM_ELM_POOL_INDEX_P_READS
Pool Index To Estore (POOL_INDEX_TO_ESTORE)	Number of index pages copied from the buffer pool to extended storage.	ulong	Yes	All	Y	SQLM_ELM_POOL_INDEX_TO_ESTORE
Pool Index Writes (POOL_INDEX_WRITES)	Number of physical disk writes for buffer pool index pages.	ulong	Yes	All	Y	SQLM_ELM_POOL_INDEX_WRITES
Pool Lsn Gap Clns (POOL_LSN_GAP_CLNS)	Number of times the page cleaner was called because the logging space in use has reached the defined standard for the database.	ulong	Yes	All	Y	SQLM_ELM_POOL_LSN_GAP_CLNS
Pool Read Time (POOL_READ_TIME)	Total time for read request processing, where physical reads of data or index pages, from a disk or the buffer pool, were necessary.	ulong	Yes	All	Y	SQLM_ELM_POOL_READ_TIME

PRELIMINARY RELEASE

Bufferpool Stat on Database Interval (PI_DBPI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pool Read Time Avg (POOL_READ_TIME_AVG)	Average time for read request processing, where physical reads of data or index pages, from a disk or the buffer pool, were necessary.	float	No	All	—	$POOL_READ_TIME / (POOL_DATA_P_READS + POOL_INDEX_P_READS)$
Pool Total Hit Rate (POOL_TOTAL_HIT_RATE)	Total buffer pool hit rate.	float	No	All	—	$(1 - ((POOL_DATA_P_READS + POOL_INDEX_P_READS) / (POOL_DATA_L_READS + POOL_INDEX_L_READS))) * 100$
Pool Total Writes (POOL_TOTAL_WRITES)	Number of disk writes from the buffer pool.	ulong	Yes	All	—	$POOL_DATA_WRITES + POOL_INDEX_WRITES$
Pool Write Time (POOL_WRITE_TIME)	Total time taken for physical writes of data pages or index pages from the buffer pool to disk.	ulong	Yes	All	Y	SQLM_ELM_POOL_WRITE_TIME
Pool Write Time Avg (POOL_WRITE_TIME_AVG)	Average time taken for physical writes of data pages or index pages from the buffer pool to disk.	float	No	All	—	$POOL_WRITE_TIME / (POOL_DATA_WRITES + POOL_INDEX_WRITES)$
Recode Time (RECODE_TIME)	Collection end time for performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always DBPI.	char(8)	No	All	—	Agent Collector

4.8.6 Cache on Database (PD_DCAD)

Function

Stores information about the cache on the database level.

Table 4.18 Cache on Database (PD_DCAD) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

While one or more applications are connected to the database.

Record size

- Fixed part: 729 bytes
- Variable part: 0 bytes

Table 4.19 Cache on Database (PD_DCAD) Fields (continues on the following pages)

Cache on Database (PD_DCAD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Cat Cache Inserts (CAT_CACHE_INSERTS)	Number of times that the system tried to insert a table descriptor or permission information into the catalog cache.	ulong	No	All	—	SQLM_ELM_CAT_CACHE_INSERTS
Cat Cache Lookups (CAT_CACHE_LOOKUPS)						
Cat Cache Needed Minisize (CAT_CACHE_NEEDED_MINISIZE)	Minimum required catalog cache size in page (4 KB) units.	ulong	No	All	—	$\left(\frac{\text{CAT_CACHE_SIZE_TOP} + 4,095}{4,096 + 0.9} \right)$ (the result is to be rounded up to the nearest integer)
Cat Cache Overflows						

PRELIMINARY RELEASE

Cache on Database (PD_DCAD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
(CAT_CACHE_OVERFLOW)	cache has overflowed out of the allocated memory environment.					HE_OVERFLOW
Cat Cache Size Top (CAT_CACHE_SIZE_TOP)	Maximum size reached by the catalog cache, in bytes.	ulong	No	All	Y	SQLM_ELM_CAT_CACHE_SIZE_TOP
Catcache Hit Rate (CATCACHE_HIT_RATE)	Catalog cache hit rate.	float	No	All	—	(1 - (CAT_CACHE_INSERTS / CAT_CACHE_LOOKUPS)) * 100
Interval (interval)	Interval at which the information is collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Pkg Cache Hit Rate (PKG_CACHE_HIT_RATE)	Package cache hit rate.	float	No	All	—	(1 - (PKG_CACHE_INSERTS / PKG_CACHE_LOOKUPS)) * 100
Pkg Cache Inserts (PKG_CACHE_INSERTS)	Number of occurrences where package cache inserts were necessary because the requested section could not be used.	ulong	No	All	—	SQLM_ELM_PKG_CACHE_INSERTS
Pkg Cache Lookups (PKG_CACHE_LOOKUPS)	The number of section and package searches in the package cache that were performed by applications.	ulong	No	All	—	SQLM_ELM_PKG_CACHE_LOOKUPS
Pkg Cache Needed Minisize (PKG_CACHE_NEEDED_MINISIZE)	Minimum package cache size required, in page (4 KB) units.	ulong	No	All	—	(PKG_CACHE_SIZE_TOP + 4,095) / 4,096 + 0.9 (the result is to be rounded up to the nearest integer)
Pkg Cache Num Overflows (PKG_CACHE_NUM_OVERFLOW)	Number of times the package cache has overflowed out of the allocated memory environment.	ulong	No	All	—	SQLM_ELM_PKG_CACHE_NUM_OVERFLOW
Pkg Cache Size Top (PKG_CACHE_SIZE_TOP)	Maximum size reached by the package cache, in bytes.	ulong	No	All	Y	SQLM_ELM_PKG_CACHE_SIZE_TOP
Recode Time (RECODE_TIME)	Collection end time for the performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type	Record name. This is always	char (8)	No	All	—	Agent Collector

PRELIMINARY RELEASE

Cache on Database (PD_DCAD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
(INPUT_RECODE_TY PE)	DCAD.)				

4.8.7 Cache on Database Interval (PI_DCAI)

Function

Stores information about the cache on the database level, using a constant interval as a unit.

Table 4.20 Cache on Database Interval (PI_DCAI) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	Yes	
LOGIF	(Blank)	

Key fields

None.

Lifetime

While one or more applications are connected to the database.

Record size

- Fixed part: 753 bytes
- Variable part: 0 bytes

Table 4.21 Cache on Database Interval (PI_DCAI) Fields (continues on the following pages)

Cache on Database Interval (PI_DCAI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Cat Cache Inserts (CAT_CACHE_INSERTS)	Number of times that the system tried to insert a table descriptor or permission information into the catalog cache.	ulong	Yes	All	—	SQLM_ELM_CAT_CACHE_INSERTS
Cat Cache Lookups (CAT_CACHE_LOOKUPS)	Number of lookups in the catalog cache for table descriptor information or permission information.	ulong	Yes	All	-	SQLM_ELM_CAT_CACHE_LOOKUPS

PRELIMINARY RELEASE

Cache on Database Interval (PI_DCAI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Cat Cache Needed Minisize (CAT_CACHE_NEEDED_MINISIZE)	Minimum required catalog cache size in page (4 KB) units.	ulong	Yes	All	—	$(CAT_CACHE_SIZE_TOP + 4,095) / 4,096 + 0.9$ (the result is to be rounded up to the nearest integer)
Cat Cache Overflows (CAT_CACHE_OVERFLOW)	Number of times the catalog cache has overflowed out of the allocated memory environment.	ulong	Yes	All	—	SQLM_ELM_CAT_CACHE_OVERFLOW
Cat Cache Size Top (CAT_CACHE_SIZE_TOP)	Maximum size reached by the catalog cache, in bytes.	ulong	Yes	All	Y	SQLM_ELM_CAT_CACHE_SIZE_TOP
Catcache Hit Rate (CATCACHE_HIT_RATE)	Catalog cache hit rate.	float	No	All	—	$(1 - (CAT_CACHE_INSERTS / CAT_CACHE_LOOKUPS)) * 100$
Interval (interval)	Interval at which the information is collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Pkg Cache Hit Rate (PKG_CACHE_HIT_RATE)	Package cache hit rate.	float	No	All	—	$(1 - (PKG_CACHE_INSERTS / PKG_CACHE_LOOKUPS)) * 100$
Pkg Cache Inserts (PKG_CACHE_INSERTS)	Total number of occurrences where package cache inserts were necessary because the requested section could not be used.	ulong	Yes	All	—	SQLM_ELM_PKG_CACHE_INSERTS
Pkg Cache Lookups (PKG_CACHE_LOOKUPS)	The number of section and package searches in the package cache that were performed by applications.	ulong	Yes	All	—	SQLM_ELM_PKG_CACHE_LOOKUPS
Pkg Cache Needed Minisize (PKG_CACHE_NEEDED_MINISIZE)	Minimum package cache size required, in page (4 KB) units.	ulong	Yes	All	—	$(PKG_CACHE_SIZE_TOP + 4,095) / 4,096 + 0.9$ (the result is to be rounded up to the nearest integer)
Pkg Cache Num Overflows (PKG_CACHE_NUM_OVERFLOW)	Number of times the package cache has overflowed out of the allocated memory environment.	ulong	Yes	All	—	SQLM_ELM_PKG_CACHE_NUM_OVERFLOW

PRELIMINARY RELEASE

Cache on Database Interval (PI_DCAI)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Pkg Cache Size Top (PKG_CACHE_SIZE_TOP)	Maximum size reached by the package cache, in bytes.	ulong	Yes	All	Y	SQLM_ELM_PKG_CACHE_SIZE_TOP
Recode Time (RECODE_TIME)						
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always DCAI.	char(8)	No	All	—	Agent Collector

4.8.8 DB2 Configuration (PD_DCFE)

Function

Stores information about DB2 configuration parameters.

Table 4.22 DB2 Configuration (PD_DCFE) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	3600	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

From the creation to the deletion of the instance.

Record size

- Fixed part: 757 bytes
- Variable part: 0 bytes

Table 4.23 DB2 Configuration (PD_DCFE) Fields (continues on the following pages)

DB2 Configuration (PD_DCFE)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
App Ctl Heap SZ (APP_CTL_HEAP_SZ)	Application control heap size, in page (4 KB) units.	word	No	All	—	SQLF_DBTN_APP_CTL_HEAP_SZ
Appgroup Mem SZ (APPGROUP_MEM_SZ)	Maximum size of the application group memory set, in page (4 KB) units.	ulong	No	All	—	SQLF_DBTN_APPGROUP_MEM_SZ
ApplheapSZ (APPLHEAPSZ)	Application heap size, in page (4 KB) units.	word	No	All	—	SQLF_DBTN_APPLHEAPSZ
Catalogcache SZ (CATALOGCACHE_SZ)	Catalog cache size, in page (4 KB) units.	long	No	All	—	SQLF_DBTN_CATALOGCACHE_SZ
Chngpgs Thresh (CHNGPGS_THRESH)	Threshold value for the number of changed pages.	word	No	All	—	SQLF_DBTN_CHNGPGS_THRESH

PRELIMINARY RELEASE

DB2 Configuration (PD_DCFE)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Database Memory (DATABASE_MEMORY)	Database shared memory size, in page (4 KB) units.	ulong	No	All	—	SQLF_DBTN_DATA BASE_MEMORY
Dbheap (DBHEAP)	Database heap, in page (4 KB) units.	ulong	No	All	—	SQLF_DBTN_DBHE AP
Dft Prefetch SZ (DFT_PREFETCH_SZ)	Default prefetch size.	short	No	All	—	SQLF_DBTN_DFT_ PREFETCH_SZ
Interval (interval)	Interval at which the information is collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM _BOOT_TIME
Locklist (LOCKLIST)	Maximum storage for lock lists, in page (4 KB) units.	ulong	No	All	—	SQLF_DBTN_LOCK LIST
LogbufSZ (LOGBUFSZ)	Log buffer size, in page (4 KB) units.	word	No	All	—	SQLF_DBTN_LOGB UFSZ
Max Connections (MAX_CONNECTIONS)	Maximum number of client connections.	long	No	All	—	SQLF_KTN_MAX_C ONNECTIONS
Max Coordagents (MAX_COORDAGENTS)	Maximum number of coordinator agents.	long	No	All	—	SQLF_KTN_MAX_C OORDAGENTS
Maxappls (MAXAPPLS)	Maximum number of active applications.	word	No	All	—	SQLF_DBTN_MAXA PPLS
Maxcagents (MAXCAGENTS)	Maximum number of concurrent agents.	long	No	All	—	SQLF_KTN_MAXCA GENTS
Maxfilop (MAXFILOP)	Maximum number of open database files for each application.	word	No	All	—	SQLF_DBTN_MAXF ILOP
Maxlocks (MAXLOCKS)	Maximum lock list percentage before escalation.	word	No	All	—	SQLF_DBTN_MAXL OCKS
Maxtotfilop (MAXTOTFILOP)	Maximum total number of open files.	word	No	All	—	SQLF_KTN_MAXTO TFILOP
Min Priv Mem (MIN_PRIV_MEM)	Minimum amount of committed memory, in page (4 KB) units.	ulong	No	All	—	SQLF_KTN_MIN_P RIV_MEM
Mon Heap SZ (MON_HEAP_SZ)	Heap size for the database system monitor, in page (4 KB) units.	word	No	All	—	SQLF_KTN_MON_H EAP_SZ

PRELIMINARY RELEASE

DB2 Configuration (PD_DCFE)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Num Iocleaners (NUM_IOCLEANERS)	Number of asynchronous page cleaners.	word	No	All	—	SQLF_DBTN_NUM_IOCLEANERS
Num Ioservers (NUM_IOSERVERS)						
Num Poolagents (NUM_POOLAGENTS)	Size of the agent pool.	long	No	All	—	SQLF_KTN_NUM_POOLAGENTS
PckcacheSZ (PCKCACHESZ)						
Query Heap SZ (QUERY_HEAP_SZ)	Query heap size.	long	No	All	—	SQLF_KTN_QUERY_HEAP_SZ
Recode Time (RECODE_TIME)						
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always DCFE.	char(8)	No	All	—	Agent Collector
Sortheap (SORTHEAP)						

4.8.9 Sort Status on Database (PD_DSOD)

Function

Stores information about sorting on the database level.

Table 4.24 Sort Status on Database (PD_DSOD) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

While one or more applications are connected to the database.

Record size

- Fixed part: 733 bytes
- Variable part: 0 bytes

Table 4.25 Sort Status on Database (PD_DSOD) Fields (continues on the following pages)

Sort Status on Database (PD_DSOD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Active Sorts (ACTIVE_SORTS)	Number of sorts in the database for which the sort heap is currently allocated.	ulong	No	All	—	SQLM_ELM_ACTIVE_SORTS
Interval (interval)						
Interval (interval)	Interval at which the information is collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Recode Time (RECODE_TIME)	Collection end time for the performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)						
Recode Type (INPUT_RECODE_TYPE)	Record name. This is always DSOD.	char(8)	No	All	—	Agent Collector

PRELIMINARY RELEASE

Sort Status on Database (PD_DSOD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Sheapthres (SHEAPTHRES)	Sort heap threshold.	ulong	No	All	—	SQLF_KTN_SHEAPTHRES
Sheapthres Shr (SHEAPTHRES_SHR)	Sort heap threshold for shared sorts.	ulong	No	All	—	SQLF_DBTN_SHEAPTHRES_SHR
Sort Heap Allocated (SORT_HEAP_ALLOCATED)	Total number of sort heap space pages allocated to all sort operations for the selected level at the time the snapshot was taken.	ulong	No	All	—	SQLM_ELM_SORT_HEAP_ALLOCATED
Sort Heap Allocated Avg (SORT_HEAP_ALLOCATED_AVG)	Average number of pages for sort heap space allocated to all sorts at the selected level, when snapshots are taken.	float	No	All	—	SORT_HEAP_ALLOCATED / ACTIVE_SORTS
Sort Overflows (SORT_OVERFLOW_S)	Total number of sorts which may need temporary memory disk space, as a result of using up the sort heap.	ulong	No	All	Y	SQLM_ELM_SORT_OVERFLOW_S
Sort Overflows Rate (SORT_OVERFLOW_S_RATE)	Ratio of sorts which may need temporary memory disk space, as a result of using up the sort heap.	float	No	All	—	(SORT_OVERFLOW_S / TOTAL_SORTS) * 100
Sort Shrheap Allocated (SORT_SHRHEAP_ALLOCATED)	Currently allocated sort shared heap.	ulong	No	All	—	SQLM_ELM_SORT_SHRHEAP_ALLOCATED
Sort Shrheap Allocated Rate (SORT_SHRHEAP_ALLOCATED_RATE)	Shared sort memory usage for the database.	float	No	All	—	SHEAPTHRES_SHR != 0 ? (SORT_SHRHEAP_ALLOCATED / SHEAPTHRES_SHR) * 100 : (SORT_SHRHEAP_ALLOCATED / SHEAPTHRES) * 100
Sort Shrheap Top (SORT_SHRHEAP_TOP)	Maximum watermark for the sort shared heap.	ulong	No	All	—	SQLM_ELM_SORT_SHRHEAP_TOP
Total Sort Time (TOTAL_SORT_TIME)	Total time elapsed for all executed sorts, in milliseconds.	ulong	No	All	—	SQLM_ELM_TOTAL_SORT_TIME

PRELIMINARY RELEASE

Sort Status on Database (PD_DSOD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Total Sort Time Avg (TOTAL_SORT_TIME_AVG)	Average time elapsed for all executed sorts, in milliseconds.	ulong	No	All	—	TOTAL_SORT_TIME / TOTAL_SORTS
Total Sorts (TOTAL_SORTS)	Total number of sorts executed.	ulong	No	All	Y	SQLM_ELM_TOTAL_SORTS

4.8.10 SQL Statement on Database (PD_DSQD)

Function

Stores information about SQL statements on the database level.

Table 4.26 SQL Statement on Database (PD_DSQD) Default and Changeable Values

Item	Default Value	Changeable?
Collection Interval	60	Yes
Collection Offset	0	
Log	No	
LOGIF	(Blank)	

Key fields

None.

Lifetime

From the creation to the deletion of the database.

Record size

- Fixed part: 765 bytes
- Variable part: 0 bytes

Table 4.27 SQL Statement on Database (PD_DSQD) Fields (continues on the following pages)

SQL Statement on Database (PD_DSQD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Commit SQL STMTS (COMMIT_SQL_STMTS)	Total number of SQL COMMIT statements attempted.	ulong	No	All	Y	SQLM_ELM_COMMIT_SQL_STMTS
DDL SQL STMTS (DDL_SQL_STMTS)	Number of SQL Data Definition Language (DDL) statements executed.	ulong	No	All	Y	SQLM_ELM_DDL_SQL_STMTS
DDL SQL STMTS Rate (DDL_SQL_STMTS_RATE)	Ratio of SQL Data Definition Language (DDL) statements executed.	float	No	All	—	DDL_SQL_STMTS / (DYNAMIC_SQL_STMTS + STATIC_SQL_STMTS) * 100

PRELIMINARY RELEASE

SQL Statement on Database (PD_DSQD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Dynamic SQL STMTS (DYNAMIC_SQL_STMTS)	Number of dynamic SQL statements attempted.	ulong	No	All	Y	SQLM_ELM_DYNAMIC_SQL_STMTS
Failed SQL STMTS (FAILED_SQL_STMTS)						
Interval (interval)	Interval at which the information collected, in seconds.	ulong	No	All	—	RECORD_TIME - CURRENT_SYSTEM_BOOT_TIME
Priv Workspace Num Overflows (PRIV_WORKSPACE_NUM_OVERFLOW)	Number of times the private workspace overflowed out of the boundary for the allocated memory.	ulong	No	All	—	SQLM_ELM_PRIV_WORKSPACE_NUM_OVERFLOW
Priv Workspace Section Inserts (PRIV_WORKSPACE_SECTION_INSERTS)						
Priv Workspace Section Lookups (PRIV_WORKSPACE_SECTION_LOOKUPS)	Number of times applications performed SQL section lookups in the agent private workspace.	ulong	No	All	—	SQLM_ELM_PRIV_WORKSPACE_SECTION_LOOKUPS
Priv Workspace Size Top (PRIV_WORKSPACE_SIZE_TOP)						
Recode Time (RECODE_TIME)	Collection end time for the performance data stored in the record.	time_t	No	All	—	Agent Collector
Recode Type (INPUT_RECODE_TYPE)						
Rollback SQL STMTS (ROLLBACK_SQL_STMTS)	Total number of SQL ROLLBACK statements attempted.	ulong	No	All	Y	SQLM_ELM_ROLLBACK_SQL_STMTS
Select SQL STMTS (SELECT_SQL_STMTS)						

PRELIMINARY RELEASE

SQL Statement on Database (PD_DSQD)						
View Name (Manager Name)	Description	Format	Delta	Supported Versions	Restriction	Data Source
Select SQL STMTS Rate (SELECT_SQL_STMTS_RATE)	Ratio of <code>SELECT</code> statements to all statements.	float	No	All	—	$\text{SELECT_SQL_STMTS} / (\text{DYNAMIC_SQL_STMTS} + \text{STATIC_SQL_STMTS}) * 100$
Shr Workspace Hit Rate (SHR_WORKSPACE_HIT_RATE)	Shared workspace hit rate.	float	No	All	—	$(1 - (\text{SHR_WORKSPACE_SECTION_INSERTS} / \text{SHR_WORKSPACE_SECTION_LOOKUPS})) * 100$
Shr Workspace Num Overflows (SHR_WORKSPACE_NUM_OVERFLOW)	Number of times the shared workspace overflowed out of the boundary for the allocated memory.	ulong	No	All	—	<code>SQLM_ELM_SHR_WORKSPACE_NUM_OVERFLOW</code>
Shr Workspace Section Inserts (SHR_WORKSPACE_SECTION_INSERTS)	Number of times applications performed SQL section inserts into the shared workspace.	ulong	No	All	—	<code>SQLM_ELM_SHR_WORKSPACE_SECTION_INSERTS</code>
Shr Workspace Section Lookups (SHR_WORKSPACE_SECTION_LOOKUPS)	Number of times applications performed SQL section lookups in the shared workspace.	ulong	No	All	—	<code>SQLM_ELM_SHR_WORKSPACE_SECTION_LOOKUPS</code>
Shr Workspace Size Top (SHR_WORKSPACE_SIZE_TOP)	Maximum size attained by a shared workspace.	ulong	No	All	Y	<code>SQLM_ELM_SHR_WORKSPACE_SIZE_TOP</code>
Static SQL STMTS (STATIC_SQL_STMTS)	Number of static SQL statements attempted.	ulong	No	All	Y	<code>SQLM_ELM_STATIC_SQL_STMTS</code>
Throughput SQL STMTS (THROUGHPUT_SQL_STMTS)	Total number of successful SQL statements.	ulong	No	All	—	$(\text{DYNAMIC_SQL_STMTS} + \text{STATIC_SQL_STMTS}) - \text{FAILED_SQL_STMTS}$
UID SQL STMTS (UID_SQL_STMTS)	Number of SQL <code>UPDATE</code> , <code>INSERT</code> , and <code>DELETE</code> statements executed.	ulong	No	All	Y	<code>SQLM_ELM_UID_SQL_STMTS</code>
UID SQL STMTS Rate (UID_SQL_STMTS_RATE)	Ratio of SQL <code>UPDATE</code> , <code>INSERT</code> , and <code>DELETE</code> statements executed.	float	No	All	—	$\text{UID_SQL_STMTS} / (\text{DYNAMIC_SQL_STMTS} + \text{STATIC_SQL_STMTS}) * 100$

Chapter 5 Messages

This chapter describes the format of HTM Agent for DB2 messages, lists the locations to which messages are output, shows which messages are output to syslog, and describes the messages in detail.

- Message Format (see section 5.1)
- Message Output Destinations (see section 5.2)
- List of Messages Output to Syslog (see section 5.3)
- Messages (see section 5.4)

5.1 Message Format

This section shows the format of messages that are issued by HTM Agent for DB2, and the notations that are used in this manual to explain the messages.

5.1.1 Format of Output Messages

The format of the messages issued by HTM Agent for DB2 consists of the message ID, followed by the message text, as shown in the following example:

KAVFnnnnn-Y message text

The message ID indicates the following:

- **K**: System identifier.
- **AVF**: Indicates an Agent message.
- **nnnnn**: Serial number of the message. HTM Agent for DB2 message numbers are in the format 29xxx.
- **Y**: Type of message:
 - **E**: Error
Message issued when the system cancels processing.
 - **W**: Warning
Message issued when the system resumes processing after message output.
 - **I**: Information
Message in which the system provides the user with information.
 - **Q**: Query
Message in which the system prompts the user for a response.

The following is the correspondence of the message types to the syslog priority levels:

- **E**
 - Level: LOG_ERR
 - Description: Error message
- **W**
 - Level: LOG_WARNING
 - Description: Warning message
- **I**
 - Level: LOG_INFO
 - Description: Additional information message.
- **Q** (Not output)

5.1.2 Format of Message Explanations

This section describes the format that is used for message explanations in this manual. Messages are listed in a table, in the order of the message IDs, as shown in the following example. The italics represent information that is variable.

Message ID	Message	Corrective Action
<i>message-ID</i>	<i>message-text</i>	Explanation of the message: (S) Explains the processing performed by the system. (O) Explains the action the operator should take when the message is displayed

5.1.3 Notes for System Administrators

If a problem occurs during operation, collect log information and conduct initial checking according to the procedures in Chapter 6.

When you conduct initial checking to determine the cause of a problem, examine all applicable log information, such as the log information for the OS (`syslog` for UNIX) and the log information output by HTM Agent for DB2. This log information enables you to understand the details of the problem, take appropriate action, and prevent the problem from occurring in the future. You should also make a record of the operations that led to the problem and determine whether the problem is likely to recur.

5.2 Message Output Destinations

Table 5.1 lists the output destinations of the messages issued by HTM Agent for DB2.

Table 5.1 Output Destinations of HTM Agent for DB2 Messages

Message ID	Output Destination			
	syslog	Common Message Log	Standard Output	Standard Error Output
KAVF24901	Not Output	Output	Not Output	Output
KAVF24902				
KAVF24903	Not Output	Not Output	Output	Not Output
KAVF24904	Not Output	Output	Not Output	Output
KAVF24905				
KAVF29001	Output	Output	Not Output	Not Output
KAVF29002				
KAVF29003				
KAVF29004				
KAVF29005				
KAVF29006				
KAVF29010				
KAVF29011				
KAVF29012				
KAVF29013				
KAVF29014				
KAVF29015				
KAVF29016				
KAVF29017				
KAVF29018				
KAVF29019				

5.3 List of Messages Output to syslog

Table 5.2 lists the messages that HTM Agent for DB2 outputs to syslog. The syslog information is output to the syslog file. For the installation location of the syslog file, see the syslog daemon configuration file (default path is /etc/syslogd.conf).

Table 5.2 Messages Output to syslog

Message ID	syslog	
	Facility	Level
KAVF29001-E	LOG_DAEMON	LOG_ERR
KAVF29002-I	LOG_DAEMON	LOG_INFO
KAVF29003-I	LOG_DAEMON	
KAVF29004-E	LOG_DAEMON	LOG_ERR
KAVF29005-E	LOG_DAEMON	
KAVF29006-E	LOG_DAEMON	
KAVF29010-E	LOG_DAEMON	
KAVF29011-E	LOG_DAEMON	
KAVF29012-E	LOG_DAEMON	
KAVF29013-W	LOG_DAEMON	
KAVF29014-E	LOG_DAEMON	LOG_ERR
KAVF29015-E	LOG_DAEMON	
KAVF29016-W	LOG_DAEMON	LOG_WARNING
KAVF29017-E	LOG_DAEMON	LOG_ERR
KAVF29018-E	LOG_DAEMON	
KAVF29019-W	LOG_DAEMON	LOG_WARNING

5.4 Messages

Table 5.3 lists the messages that are issued by HTM Agent for DB2, and how to correct the issue that caused the message.

Table 5.3 Messages that are Issued by HTM Agent for DB2 (continues on the following pages)

Message ID	Message	Corrective Action
KAVF24901-E	An attempt to acquire product information has failed. (<i>servicekey=service-key</i>)	<p>The <code>jpctminfo</code> command failed to acquire the information about the product specified by the indicated service key. Possible causes are as follows:</p> <ul style="list-style-type: none"> ▪ The version of the Tuning Manager program specified in <i>service-key</i> is earlier than 04-00. ▪ The command was executed during installation or uninstallation of a Tuning Manager program. ▪ If the cause is not one of the above, the environment for the Tuning Manager program specified in <i>service-key</i> is incorrect. <p>(S) Stops command execution.</p> <p>(O) If the command was executed during installation or uninstallation of a Tuning Manager program, re-execute the command after the installation or uninstallation finishes. If the environment is incorrect, collect maintenance information, and then overwrite-install the Tuning Manager program indicated by <i>service-key</i>.</p>
KAVF24902-E	There is no product information. (<i>servicekey=service-key</i>)	<p>The Tuning Manager program indicated by <i>service-key</i> in the message and specified in the <code>jpctminfo</code> command has not been installed.</p> <p>(S) Stops command execution.</p>
KAVF24903-I	There is no patch history information.	<p>There is no patch history information for the Tuning Manager program specified in the <code>jpctminfo</code> command.</p> <p>(S) Stops command execution.</p>
KAVF24904-E	An error occurred while reading the patch history information.	<p>An error occurred while the command was reading the patch history information for the specified Tuning Manager program.</p> <p>(S) Stops command execution.</p> <p>(O) A Tuning Manager program is probably being installed or uninstalled. After installation or uninstallation finishes, re-execute the command. If no Tuning Manager program is being installed or uninstalled, collect maintenance data and contact the system administrator.</p>
KAVF24905-E	The service key (<i>service-key</i>) cannot be specified.	<p>An invalid service key was specified.</p> <p>(S) Stops command execution.</p> <p>(O) Check whether a correct service key was specified.</p>

PRELIMINARY RELEASE

Message ID	Message	Corrective Action
KAVF29001-E	The Agent for IBM DB2 Universal Database does not support this DB2 Universal Database version.	HTM Agent for DB2 was executed for an unsupported version of DB2. (S) Terminates processing of the Agent Collector service. (O) Use a supported version of DB2.
KAVF29002-I	Agent Collector has started.	Startup of the Agent Collector service has completed.
KAVF29003-I	Agent Collector has stopped.	The Agent Collector service stopped.
KAVF29004-E	Memory could not be allocated.	Memory is insufficient. (S) Terminates processing of the Agent Collector service. (O) Increase the amount of available memory space.
KAVF29005-E	An attempt to write to file <i>file-name</i> has failed.	The system could not write to the file indicated by <i>file-name</i> . (S) Terminates processing of the Agent Collector service. (O) Check the permissions for the file. Also check whether the file capacity has been exceeded or whether the disk has become full.
KAVF29006-E	The file was not found. (file name = <i>file-name</i>)	The file indicated by <i>file-name</i> could not be found. (S) Terminates processing of the Agent Collector service. (O) Check whether the file indicated by <i>file-name</i> exists.
KAVF29010-E	File format error: <i>file-name</i>	The format of the file indicated by <i>file-name</i> is invalid. (S) Terminates processing of the Agent Collector service. (O) Check the format of the file indicated by <i>file-name</i> .
KAVF29011-E	A message could not be written to the log file <i>log-file-name</i> .	The system could not output a message to the file indicated by <i>log-file-name</i> . (S) Terminates processing of the Agent Collector service. (O) If the file indicated by <i>log-file-name</i> exists in the path, check its write permissions. If the file indicated by <i>log-file-name</i> does not exist in the path, check whether you have read and write permissions for the specified path.
KAVF29012-E	The file does not have read permission. (file name = <i>file-name</i>)	The user does not have read permissions for the file indicated by <i>file-name</i> (the HiRDB system definition file, HiRDB server definition file, or <i>jpcagt.ini</i> file). (S) Terminates processing of the Agent Collector service. (O) Check the permissions for the file.

PRELIMINARY RELEASE

Message ID	Message	Corrective Action
KAVF29013-W	An attempt to initialize Agent Collector has failed.	The system failed to initialize the Agent Collector service. (S) Terminates processing of the Agent Collector service. (O) Check all initialization configurations of the Agent Collector service.
KAVF29014-E	System information could not be acquired.	The system could not acquire system information such as host names. (S) Terminates processing of the Agent Collector service. (O) Check the environment settings of the OS.
KAVF29015-E	An attempt to execute function <i>command-name</i> has failed.	An error occurred during execution of the command indicated by <i>command-name</i> . (S) Terminates processing of the Agent Collector service. (O) Collect maintenance data and contact the system administrator. For details on collecting maintenance data, see the chapter on troubleshooting in the manual <i>HiCommand Tuning Manager User's Guide</i> .
KAVF29016-W	Connection to DB2 is not possible.	The system cannot connect to DB2.
KAVF29017-E	A program error has occurred. Please connect maintenance personnel.	A program error occurred. (O) Collect maintenance data and contact the system administrator. For details on collecting maintenance data, see the chapter on troubleshooting in the manual <i>HiCommand Tuning Manager User's Guide</i> .
KAVF29018-E	<i>SQLCA:DB2-error-message</i>	This is error message information returned by DB2. (O) Check the DB2 manual and eliminate the cause of the error.
KAVF29019-W	An overflow error has occurred: <i>data-source</i>	An overflow occurred in a data source. The uppermost 4 bytes of the 64-bit data have been truncated.

Chapter 6 Troubleshooting

This chapter explains how to handle any problems that may occur while you are using Tuning Manager series programs. This chapter is concerned principally with handling errors relating to an Agent. For information on problem handling for the entire Tuning Manager series system, see the chapter that describes problem handling in the *HiCommand™ Tuning Manager User's Guide*.

- Following Procedures for Handling Problems (see section 6.1)
- Troubleshooting (see section 6.2)
- Checking Log Information (see section 6.3)
- Collecting Data when a Problem Occurs (see section 6.4)
- Data Collection Procedure (for UNIX®) (see section 6.5)
- Calling the Hitachi Data Systems Support Center (see section 6.6)

6.1 Following Procedures for Handling Problems

If you encounter problems while using Tuning Manager products, follow these guidelines:

- **Check the event**

Examine the following:

- The event when the problem occurred
- The message contents (if a message is output)
- The log information, such as the common message log

For details about causes of messages and actions to be taken, see Chapter 5 in this manual, and the chapter that describes messages in the *HiCommand™ Tuning Manager User's Guide*. For details about the log information that is output by the Tuning Manager products, see section 6.3.

- **Collect data**

Collect data to determine the cause of the problem. For details on collecting the necessary data, see sections 6.4 and 6.5.

- **Determine the cause**

Use the collected data to determine the cause and extent of the error, as well as the range of its consequences.

6.2 Troubleshooting

If an error occurs while you are using a Tuning Manager product, you should first check to see if any of the events described in this section have occurred. Table 6.1 lists the principal errors that may occur while you are using a Tuning Manager product.

Table 6.1 List of Common Problems

Classification	Problem	Section
Setting up or starting service	<ul style="list-style-type: none"> ▪ A Tuning Manager program service does not start. ▪ It takes a long time for a service to start once startup is requested. ▪ When a service for another program starts immediately after a service for a Tuning Manager series program stops, communication is not performed correctly. ▪ The error message KAVB05033-E A service could not start is issued during execution of the jpcstart command. ▪ After a message appears indicating that the disk capacity is insufficient, the Master Store service or Agent Store service stops. 	6.2.1
Executing commands	<ul style="list-style-type: none"> ▪ When the jpcctrl list command is executed, the names of services not operating are output. ▪ When the jpcctrl dump command is executed, data other than the specified Store data is output. 	6.2.2
Collecting and managing performance data	<ul style="list-style-type: none"> ▪ Though the data storage time is set for a short period of time, the size of the Store database for Agent does not become smaller. ▪ The message Illegal data was detected in the Store database is output to the common message log. ▪ Agent is active, but no performance data is collected. 	0

6.2.1 Setting up or Starting a Service

Handle errors related to setup or service startup as described in Table 6.2.

Table 6.2 Handling Problems Related to Setup or Service Startup (continues on the next page)

Problem	Possible Causes and Solutions
<p>A Tuning Manager program service does not start.</p>	<p>Tuning Manager has stopped. When Tuning Manager and Agent exist on the same host, the Agent service cannot be started if Tuning Manager for the connection destination has stopped. Make sure that the Collection Manager service has started, and start it if it has not. For details about how to start the service, see the chapter that describes how to operate the Tuning Manager series in the <i>HiCommand Tuning Manager User's Guide</i>.</p> <p>The same port number is set for multiple Tuning Manager program services. When the same port number is set for multiple Tuning Manager program services, none of the Tuning Manager program services can be started. The default is for the system to assign port numbers automatically, in which case there will not be any duplicated port numbers. If you specified specific port numbers for any Tuning Manager programs during Tuning Manager setup, check the specified port numbers. If you specified the same port number for more than one program, make appropriate corrections in the port number specifications. For details about setting port numbers, see the chapter that describes setup in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p> <p>There is an error in the settings for the Store database directory. If a directory that cannot be accessed, or does not exist, is specified as any of the following directories, the Agent Store service will not start. Check the settings of the directory attributes, and revise them as necessary.</p> <ul style="list-style-type: none"> ▪ Store database directory ▪ Store database backup directory ▪ Store database export directory <p>If you specify the same directory for multiple Agent Store services, the Agent Store service cannot start. Check the directory settings, and revise them as necessary.</p> <p>The host name of the machine was changed using a non-permitted procedure. For details about the procedures for renaming a host on a machine, see the chapter that describes setup in the <i>HiCommand™ Tuning Manager User's Guide</i>. If you have used any other method to rename a host, the programs of the Tuning Manager products may not start.</p> <p>There was an error in an instance environment setting. If there was an error during setup in any of the following instance environment settings, the Agent Collector service cannot be started:</p> <ul style="list-style-type: none"> ▪ db2_instance ▪ db2_node ▪ db2_name ▪ db2_user ▪ db2_password <p>Execute the <code>jpccinssetup</code> command to specify the correct settings. For details about the <code>jpccinssetup</code> command, see the chapter that describes commands in the <i>HiCommand Tuning Manager User's Guide</i>.</p>
<p>It takes a long time for a service to start once startup is requested.</p>	<p>The following are some reasons why it may take a long time for service to actually start once you execute the <code>jpccstart</code> command or start a service by choosing the Service icon. If this is caused by one of the following reasons, subsequent service startups should take less time:</p> <ul style="list-style-type: none"> ▪ If you restart the system and then start a service, without specifying that the service is to stop automatically when the system stops, the indexes of the Store database might be rebuilt. This might slow the startup of the service. ▪ If you start a service after adding a new Agent, the indexes of the Store database will be created during the first startup. This might slow the startup of the service.

Problem	Possible Causes and Solutions
<p>When a service for another program starts immediately after a service for a Tuning Manager series program stops, communication is not performed correctly.</p>	<p>Immediately after stopping a Tuning Manager series program service, another program service might be started that uses the same port that the stopped service was using. In this case, communication might not be performed correctly. You can use either of the following techniques to avoid this problem:</p> <p>Fix the port numbers for the Tuning Manager series programs.</p> <p>Assign a fixed port number to each Tuning Manager program. For details about setting port numbers, see the chapters that describe setup, in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p> <p>Set the TCP_TIMEWAIT value.</p> <p>Use an OS command to set the TCP_TIMEWAIT value that specifies a connection wait time.</p> <ul style="list-style-type: none"> ▪ For AIX, specify a connection wait time of at least 75 seconds, as follows: AIX: tcp_timewait:5 ▪ For Linux, you cannot change the default connection wait time (60 seconds). Use the fixed port numbers for the services of the Tuning Manager series programs.
<p>The error message "KAVE05033-E A service could not start" is issued during execution of the jpcstart command</p>	<p>If service cannot be started temporarily when the jpcstart command is executed due to insufficient resources, etc., the KAVE05033-E error message may be output. Use the net start or jpcctrl list command to check whether the service has actually started.</p>
<p>After a message appears indicating that the disk capacity is insufficient, the Master Store service or Agent Store service stops.</p>	<p>If the disk used by the Store database does not have sufficient free space, data storage to the Store database stops. A message appears indicating that free disk space is insufficient, and then the Master Store service or Agent Store service stops.</p> <p>If this message appears, take either of the following actions:</p> <p>Secure sufficient free disk space</p> <p>Estimate the disk space requirement for the Store database, and move the Store database to a disk that has a sufficient amount of free disk space. For details on how to estimate the disk space requirement for the Store database, see Appendix A .3, <i>Estimating System Requirements</i>. For details on how to set up the Store database, see the chapter that describes setup in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p> <p>Change the store conditions for the Store database</p> <p>Change the conditions for storing data in the Store database to adjust its maximum size. For details on how to do this, see the chapter that describes commands in the <i>HiCommand™ Tuning Manager Performance Reporter Reference</i>.</p> <p>If the Master Store service or Agent Store service still fails to start even after taking the above actions, an unrecoverable logical conflict exists in the Store database. In such a case, restore the Store database from backup data, and then restart the Master Store service or Agent Store service. If there is no backup data available, initialize the Store database, and then start the Master Store service or Agent Store service. To initialize the Store database, delete the following files in the Store database storage directory:</p> <ul style="list-style-type: none"> ▪ All files with the extension .DB ▪ All files with the extension .IDX <p>For details about the Store database storage directory, see the chapter that describes setup in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p>

6.2.2 Executing Commands

Handle errors related to execution of Tuning Manager commands as described in Table 6.3.

Table 6.3 Handling Errors Related to the Execution of Tuning Manager Commands

Problem	Possible Causes and Solutions
<p>When the <code>jcctrl list</code> command is executed, the names of services not operating are output.</p>	<p>A Tuning Manager series program was uninstalled without its service information being deleted.</p> <p>Service information for a Tuning Manager series program remains in the database even after the program is uninstalled. Execute the <code>jcctrl delete</code> command to delete the service information. For details about deleting service information, see the chapter that describes how to operate Tuning Manager series programs, in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p> <p>The machine's host name was changed without deleting the service information of the Tuning Manager series program.</p> <p>If the host name of a machine was changed without deleting a Tuning Manager series program's service information, the service information corresponding to the service ID that was associated with the previous host name remains in the database that the Master Manager manages. Execute the <code>jcctrl delete</code> command to delete the service information. For details about deleting service information, see the chapter that describes how to operate the Tuning Manager series in the <i>HiCommand™ Tuning Manager User's Guide</i>. For details about changing the host name, see the chapter that describes setup in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p>
<p>When the <code>jcctrl dump</code> command is executed, data other than the specified Store data is output</p>	<p>Specifying the same export file name for the same Store service in multiple executions of the <code>jcctrl dump</code> command causes the initial output results to be overwritten with the subsequent output results. Each time you execute the <code>jcctrl dump</code> command for the same Store service, specify a different export file name. For details about exporting Store data, see the chapter that describes management of the Store database, in the <i>HiCommand™ Tuning Manager User's Guide</i>.</p>

6.2.3 Collecting and Managing Performance Data

Handle errors related to collection and management of Tuning Manager performance data as follows:

- **Although the data storage time is set for a short period of time, the size of the Store database for Agent does not become smaller.**

If the file capacity of the Store database is already at its limit, the file size will not become smaller even if a shorter data storage period is set. In this case, set a shorter storage period, back up the Store database, and then restore the database again.

For details on setting the data storage period, see the chapter on commands in the *HiCommand Tuning Manager Performance Reporter Reference*. For details on backing up and restoring the Store database, see the chapter on backup and disk management in the *HiCommand Tuning Manager User's Guide*.

- **The message "Illegal data was detected in the Store database" is output to the common message log.**

An unexpected service halt or machine shutdown may result in invalid data in the Store database. Correct this problem as follows:

- If the Store database has been backed up, restore it.
- If the Store database has not been backed up, stop the Agent Store service, delete the corresponding database file (*.DB file or *.IDX file), and then restart the service.

- **Performance data was not collected, even though the Agent was started.**

Take the following actions:

- Check the status of DB2, and if it stopped, restart it.
- Check the settings that were made during the setup of the instance environment.

Execute the `jpcinssetup` command to specify the correct settings. For details about the `jpcinssetup` command, see the chapter that describes the commands in the *HiCommand Tuning Manager User's Guide*.

6.2.3.1 Managing Other Problems

You should check the existing circumstances when other errors occur. If a message is output, read its contents. For details about the log information that is output by the Collection Manager and Agent, see section 6.3.

If you cannot resolve an error by taking any of the steps described from sections 6.2.1 through to 0, or if an error occurs that is not described in these sections, collect data needed to investigate the error and contact the system administrator.

For details about the data you should collect and how to collect it, see sections 6.4 and 6.5.

6.3 Checking Log Information

When an error occurs with the Collection Manager or an Agent, you should check the log information and investigate the problem. The following three types of log information are output during operation of the Collection Manager and Agent:

- System log
- Common message log
- Trace log

Table 6.4 describes the three types of log information and the log options that can be set for each type.

Table 6.4 Types of Log Information

Log Type	Details
System log	A system log contains log information that reports the system status and errors that occurred. Tuning Manager outputs the system log to the <code>syslog</code> file. For details about the output format, see the chapter that describes log information in the <i>HiCommand™ Tuning Manager User's Guide</i> .
Common message log	The common message log contains log information that reports the system status and errors that have occurred. The information output to this log is more detailed than the system log information. For details about the common message log's output destination file name and file size, see section 6.3.1. For details about the output format, see the chapter that describes log information in the <i>HiCommand™ Tuning Manager User's Guide</i> .
Trace log	Whenever an error occurs, the trace log contains log information that is needed to investigate the cause of the error, or to determine the processing time required by each process. The trace log is output to a log file for the particular Collection Manager or Agent service.

6.3.1 About the Common Message Log

The common message log is one of the types of log information for the Collection Manager or an Agent. Listed below is the service name (or control name) and the log file name for the log output source file, and the amount of disk space used.

- **Type of log information**

Common message log

- **Output source**

Collection Manager or Agent

- **File name**

/opt/jp1pc/log/jpclog{01|02}

Note: The value 01 or 02 is appended to the common message log's file name. Log information is first output to the log file whose name ends with 01. When the maximum log file size is reached, the number at the end of the log file name changes from 01 to 02, and a log file with 01 at the end of its file name is created. Log information is then output to the file whose name ends in 01. If a log file with a name ending in 02 already exists, that log file is overwritten.

- **Disk space used (in kilobytes)**

2,048(*2)

Note: The value in parentheses is the number of log files that can be created for a single service. For example, 2,048(*2) indicates that up to two log files, each using 2,048 KB of disk space, can be created. In this case, a total of 4,096 KB of disk space will be used.

6.3.1.1 About the Trace Logs

The trace log is one of the types of log information for the Collection Manager or an Agent. Table 6.5 lists the service names or control names that are the output sources of Agent trace logs, and the names of the storage folders.

Table 6.5 Names of Storage Directories for the Trace Logs

Type of Log Information	Output Source	Directory Name
Trace log	Action Handler	/opt/jp1pc/bin/action/log/
	Collection Manager or Agent commands	/opt/jp1pc/tools/log/
	Agent Collector	/opt/jp1pc/agtr/agent/ <i>instance-name</i> /log/
	Agent Store	/opt/jp1pc/agtr/store/ <i>instance-name</i> /log/

6.4 Collecting Data when a Problem Occurs

If the appropriate action described in section 6.2 is unsuccessful in correcting the error, you should collect necessary data and contact the system administrator to determine the cause of the error. This section describes the data that should be collected in the event of an error.

Collection Manager and Agent provide commands for collecting the needed data in the batch mode. Use the `jpcras` command to collect Agent data. The following tables indicate the data that can be collected by the `jpcras` command.

Note: The data collected by the `jpcras` command depends on the options you specify when you execute the command. For details about the command's options and the data that can be collected, see the chapter that describes commands in the *HiCommand™ Tuning Manager User's Guide*.

6.4.1 Collecting Log information about the OS (In UNIX®)

All data that is listed in the table can be collected by the `jpcras` command.

Table 6.6 Log information about the OS (UNIX®)

Type of Information	Overview	Default File Name
System log	syslog	<ul style="list-style-type: none"> ▪ For AIX N/A ▪ For Linux /var/log/messages
Process information	List of processes	N/A
System file	hosts file	/etc/hosts
	services file	/etc/services
	passwd file	/etc/passwd
OS information	Patch information	Not applicable
	Kernel information	
	Version information	
	Network status	
	Environment variable	
	Host name	
Dump information	core file	

6.4.2 Collecting Information about the Collection Manager or Agent

You need to collect information about the Collection Manager and Agent, as listed below. In the case of a network error, you also need to collect applicable files from the connection-target machine.

All data that is listed in the table can be collected by the `jpccras` command.

Table 6.7 Collecting Information about Collection Manager and the Agent

Type of Information	Overview	Default File Name
Common message log	Message log output from the Collection Manager or Agent	/opt/jp1pc/log/jpclog{01 02}
Configuration information	Each configuration information file	Not applicable
	Output results of the <code>jpcctrl list</code> command	
Version information	Product version	Not applicable
	Historical information	
Database information	Agent Store	/opt/jp1pc/agtr/store/instance-name/*.DB /opt/jp1pc/agtr/store/instance-name/*.IDX
Trace log	Trace information for each program of the Collection Manager or Agent	Not applicable For details about the trace log storage directory, see section 6.3.1.1.

6.4.3 Collecting Operation Information

You should collect the following information about the operation being performed when the error occurred:

- Details of the operation
- Time the error occurred
- Machine configuration (the OS version, host name, and Tuning Manager and Agent configuration, etc.)
- Whether the problem is replicable

6.4.4 Collecting Error and Other Information

If the error occurred during command execution, do the following:

- Obtain messages output to the console.
- Collect command arguments that were specified.

6.5 Data Collection Procedure (for UNIX®)

This section explains how to collect data in the event of an error.

6.5.1 Collecting Data by Executing the `jpcras` Command

Use the `jpcras` command to collect the data needed to determine the cause of an error (note that the procedure described below must be executed by an OS user with the root permission). Execute the command at a server containing Collection Manager, or at the Agent server in which the error occurred.

1. Log in to the host where the service subject to this data collection is installed.
2. Specify in the `jpcras` command the data to be collected and the storage directory for the data, and then execute the command.

The following `jpcras` command stores all the available information in the `/tmp/jpc/mgr` directory:

```
jpcras /tmp/jpc/agt all all
```

The data collected by the data collection command can be stored in the specified directory in a compressed format by using either the `tar` or the `compress` command.

Example file name:

```
jpcrasYYMMDD.tar.Z
```

The date is added at the location indicated by `YYMMDD`.

For details about the `jpcras` command, see the chapter that describes commands in the *HiCommand™ Tuning Manager User's Guide*.

For additional data collection procedures, see sections 6.4.3 and 6.4.4.

6.6 Calling the Hitachi Data Systems Support Center

If you need to call the Hitachi Data Systems Support Center, make sure to provide as much information about the problem as possible, as outlined in this chapter.

The worldwide Hitachi Data Systems Support Centers are:

- Hitachi Data Systems North America/Latin America
San Diego, California, USA
1-800-348-4357
- Hitachi Data Systems Europe
Contact Hitachi Data Systems Local Support
- Hitachi Data Systems Asia Pacific
North Ryde, Australia
61-2-9325-3300

Appendix A Notes on Miscellaneous Settings, Properties, and other Information

A.1 System Configuration

Following are the maximum values for a single Tuning Manager system:

- Maximum number of Tuning Managers: 1
- Maximum number of Agents: 50

Note: One Agent is required for each program that is the target of performance monitoring.

If the system configuration consists of more than 50 Agents, set a Tuning Manager at each key location.

A.2 Performance Reporter Display

One Performance Reporter can display a maximum of 10 report windows.

A.3 Estimating System Requirements

Before configuring a system that will use HTM Agent for DB2, we recommend that you first estimate the performance of the computer to be used, to ensure that the computer is capable of running HTM Agent for DB2. This involves estimating both memory and various disk space requirements.

A.3.1 Estimating Memory Requirements

The memory requirements vary depending on the setup conditions and on the conditions under which HTM Agent for DB2 will be used.

The following table provides general estimates of the memory requirements of HTM Agent for DB2.

Table A.1 Memory Requirements

Condition of HTM Agent for DB2	Required Memory Size (MB)	
	AIX	Linux 3
Operation	$a + (b + c) \times n$	$a + (b + c) \times n$

Legend:

- a*: Memory requirement for the `jpcsh` process
- b*: Memory requirement for the `jpcagtr` process
- c*: Memory requirement for the `jpcsto` process
- n*: Number of instances

The memory requirement varies depending on the OS. The following shows the memory requirements for each process.

Table A.2 Process Memory Requirements

Process Name	Memory Requirements for Processes (MB)	
	AIX	Linux 3
<code>jpcsh</code>	4	8
<code>jpcagtr</code>	16	16
<code>jpcsto</code>	18	24

A.3.2 Estimating Disk Space Requirements

The required disk space varies depending on the number of records that store performance data, as outlined in the following sections.

A.3.2.1 Estimating Disk Space Requirements for the Entire System

Table A.3 Disk Space Requirements for the Entire System

Condition of HTM Agent for DB2	Equation for Estimating Disk Space (MB)	
	AIX	Linux 3
At installation <i>Note:</i> The required disk space for installation is twice as large as the size of the program.	64	28
Operating under initialized conditions	118	65
Operation	$a + b + 84$	$a + b + 36$

Legend:

a: Size of the common message log file.

b: Sum of the disk spaces for each instance.

A.3.2.2 Estimating Disk Space Requirements for the Store Database

All records of one record type are stored in the same file in the Store database. The following table shows disk space requirements of the Store database for each record type.

Note: When you use the `jpcctrl backup` command or the `jpcctrl dump` command, nearly twice the disk space that is calculated from the table below will be required for a backup file or export file.

Table A.4 Disk Space Requirements for the Store Database for Each Record Type

Record Type	Formula for Estimating Disk Space Requirements (bytes)
PI record type	$(X_1 + \dots + X_n + 3,500 \times a)$
PD record type	$(Y_1 + \dots + Y_n + 700 \times b)$

PRELIMINARY RELEASE

Legend:

X: Disk space for each record that collects historical data using PI record type records

X can be calculated by the following formula:

$$X = \{d \times e + (c + 1,900) \times \{(d \times e) / (65,250 - c) + 1\}^{\text{Note 1}}\} \times f \times 1.5$$

Y: Disk space for each record that collects historical data using PD record type records

Y can be calculated by the following formula:

$$Y = \{d \times g + (c + 1,900) \times \{(d \times e) / (65,250 - c) + 1\}^{\text{Note 1}} \times (g/e)^{\text{Note 2}}\} \times 1.5$$

a: Number of records that collect historical data using PI record type records

b: Number of records that collect historical data using PD record type records

c: Size of the fixed part of each record that collects historical data ^{Note 3}

d: Size of the variable part of each record that collects historical data ^{Note 3}

e: Number of instances for each record that collects historical data (1 for single instance record)

f: Number of stored records for each record that collects historical data ^{Note 4}

g: Number of stored records for each record that collects historical data ^{Note 5}

Note 1: In the $\{(d \times e) / (65,250 - c) + 1\}$ calculation, the part below the decimal point is discarded.

Note 2: In the (g/e) calculation, the decimal part is discarded.

Note 3: For details about the sizes of the fixed and variable parts of each record, see Chapter 4.

Note 4: For records of the PI record type, the collected data is summarized automatically over a certain fixed period (minute, hour, day, week, month, and year). Thus, you must take into consideration the number of records to be stored over the applicable period. Table A.5 lists the default values for the retention period and the number of records.

Note 5: For details about the number of stored records, see section A.8.1, *List of Agent Store Service Properties*. The default number of records to be stored for records of each PD record type is 10,000.

Table A.5 Default Values for the Retention Period and the Number of Records

Data Type	Retention Period	Number of Records (If the Collection Interval is 1 Minute)
Minute-by-minute	1 day	1,440
Hourly	7 days	168
Daily	1 year	366
Weekly	1 year	53
Monthly	1 year	12
Yearly	No limit	$(\text{years-collected}) \times 1$

A.4 Kernel Parameters

When using HTM Agent for DB2, you do not need to adjust the kernel parameters.

For details on adjusting the kernel parameters when using Tuning Manager or Performance Reporter in a UNIX® environment, see the appendix that lists the kernel parameters, in the *HiCommand™ Tuning Manager User's Guide*. For details on adjusting the kernel parameters when using DB2, see the DB2 manual.

A.5 List of Identifiers

When operating HTM Agent for DB2 or when extracting performance data from the Store database of HTM Agent for DB2, identifiers are required so that the system can identify HTM Agent for DB2. The following table shows the identifiers of HTM Agent for DB2.

Table A.6 List of Identifiers of HTM Agent for DB2

Use	Name	Identifier	Description
Commands	Product ID	R	The product ID is part of the service ID. The service ID is required for commands that perform operations such as: <ul style="list-style-type: none"> ▪ Checking the system configuration of the Tuning Manager series ▪ Backing up performance data For details about the service ID, see the chapter that explains the operations for running Tuning Manager in the <i>HiCommand™ Tuning Manager User's Guide</i> .
	Service key	agtr	This identifier is needed in order to use commands to start and stop HTM Agent for DB2. For details about the service key, see the chapter that explains the operations for running Tuning Manager in the <i>HiCommand™ Tuning Manager User's Guide</i> .

A.6 List of Processes

The following table lists the processes of HTM Agent for DB2. The value in parentheses following each process name is the number of processes that can run concurrently.

Table A.7 List of Processes of HTM Agent for DB2

Process name (process count)	Function
jpcagtr(<i>n</i>)	The process of the Agent Collector service. One jpcagtr process is started for each instance of HTM Agent for DB2.
jpcah(1)	The process of the Action Handler service. Only one jpcah process is started for each host even if Tuning Manager and multiple Agents are installed on the same host.
jpcsto(<i>n</i>)	The process of the Agent Store service. One jpcsto process is started for each instance of HTM Agent for DB2.
stpqlpr(1) <i>Note:</i> This is a child process of the jpcsto process.	The process for backup/export of the Store database.

A.7 List of Port Numbers

You can change a port number to match your system environment. For details about changing a port number, see the chapter that describes setup in the *HiCommand™ Tuning Manager User's Guide*. The TCP/IP protocol is used.

A.7.1 Port numbers of Collection Manager and Agent

The following table lists the port numbers that are used by Collection Manager and the Agent. Note that except for port numbers 22285 and 22286, all remaining port numbers are assigned if you do not specify port numbers when you execute the `jpcnsconfig port` command. Also, if you do not execute the `jpcnsconfig port` command, port numbers that are not being used by the system are automatically assigned each time the service is restarted.

Table A.8 Port Numbers Used by Collection Manager and Agent

Service Name	Parameter	Port Number
Name Server	<code>jplpcnsvr</code>	22285
Master Manager	<code>jplpcmm</code>	20271
Master Store	<code>jplpcsto</code>	20272
Correlator	<code>jplpcep</code>	20273
Trap Generator	<code>jplpctrap</code>	20274
View Server	<code>jplpcsvr2</code>	20276
View Server (between Performance Reporter and the View Server service)	<code>jplpcsvr</code>	22286
Action Handler	<code>jplpcah</code>	20275
Agent Store	<code>jplpcstor [nnn]</code> See <i>Note 1</i>	See <i>Note 2</i>
Agent Collector	<code>jplpcagtr [nnn]</code> See <i>Note 1</i>	

Notes:

- 1: When multiple instances of the service are created, a sequential number (nnn) is appended to the second and subsequent instances. The first instance does not have a sequential number.
- 2: Each time the service is restarted, port numbers that are not being used by the system are automatically assigned

A.7.2 Firewall Passage Directions

If there is a firewall between Tuning Manager and HTM Agent for DB2, you must set fixed port numbers for all services of Tuning Manager and the Agent. To enable communications to pass through the firewall, port numbers that you set must comply with the directions shown in following table. A single arrow in the “Passage Direction” column indicates the direction from which communications begin. Dual arrows indicate that communications can be initiated from either the Tuning Manager Host or Agent Host.

Table A.9 Firewall Passage Directions

Service Name	Parameter	Passage Directions
Name Server	jp1pcnsvr	Tuning Manager host <-- Agent host
Master Manager	jp1pcmm	Tuning Manager host <-- Agent host
Master Store	jp1pcsto	Tuning Manager host <-- Agent host
Correlator	jp1pcep	Tuning Manager host <-- Agent host
Trap Generator	jp1pctrap	Tuning Manager host <-- Agent host
View Server	jp1pcsvr2	Tuning Manager host <-- Agent host
Action Handler	jp1pcah	Tuning Manager host <--> Agent host
Agent Store	jp1pcstor[<i>nnn</i>] See <i>Note</i>	Agent host <-- Tuning Manager host
Agent Collector	jp1pcagtr[<i>nnn</i>] See <i>Note</i>	Agent host <-- Tuning Manager host

Note: When multiple instances of the service are created, a sequential number (*nnn*) is appended to the second and subsequent instances. The first instance does not have a sequential number.

To execute the `jpctr1 dump` command or the `jpctr1 list` command at an Agent host, execute the command by one of the following methods:

- Specifying the `proxy` option of the `jpctr1 dump` command or the `jpctr1 list` command, so that communications are performed through Tuning Manager. For details about the proxy option of these commands, see the chapter that describes commands in the *HiCommand™ Tuning Manager User's Guide*.
- Setting port numbers so that connections can be initiated between Agent Hosts (Agent host <--> Agent host), and communications can be passed through the firewall:
 - Action Handler
(`jp1pcah` parameter)
 - Agent Store
(`jp1pcstor[nnn]` parameter)
 - Agent Collector
(`jp1pcagtr[nnn]` parameter)

A.8 Properties of HTM Agent for DB2

This section lists the properties of Agent Store of HTM Agent for DB2, and the properties of Agent Collector. You use Performance Reporter commands to output these properties.

A.8.1 List of Agent Store Service Properties

Table A.10 lists the properties of the Agent Store service of HTM Agent for DB2 that are output by executing `jpcaspsv output` commands.

Table A.10 List of Agent Store Service Properties of HTM Agent for DB2

Parameter Name	Element Name		Description
service	—		Outputs the service ID that identifies the Agent for the id attribute.
	product-interval	minute-drawer	Outputs the duration for which minute records of the PI record type are stored. The following values are output: <ul style="list-style-type: none"> ▪ minute ▪ hour ▪ day ▪ 2 days ▪ 3 days ▪ 4 days ▪ 5 days ▪ 6 days ▪ week ▪ month ▪ year
		hour-drawer	Outputs the duration for which hourly records of the PI record type are stored. The following values are output: <ul style="list-style-type: none"> ▪ hour ▪ day ▪ 2 days ▪ 3 days ▪ 4 days ▪ 5 days ▪ 6 days ▪ week ▪ month ▪ year

PRELIMINARY RELEASE

Parameter Name	Element Name		Description
service	product-interval	day-drawer	Outputs the duration for which daily records of the PI record type are stored. The following values are output: <ul style="list-style-type: none"> ▪ day ▪ 2 days ▪ 3 days ▪ 4 days ▪ 5 days ▪ 6 days ▪ week ▪ month ▪ year
		week-drawer	Outputs the duration for which weekly records of the PI record type are stored. The following values are output: <ul style="list-style-type: none"> ▪ week ▪ month ▪ year
		month-drawer	Outputs the duration for which monthly records of the PI record type are stored. The following values are output: <ul style="list-style-type: none"> ▪ month ▪ year
	product-detail	detail-record	Outputs the number of saved records for each record of the PD record type. Outputs the record ID of the PD record type for the id attribute and an integer from 0 to 2,147,483,647 for the max-rec attribute. See <i>Note</i>
	product-log	log-record	Outputs the number of saved records for each record of the PL record type. Outputs the record ID of the PL record type for the id attribute and an integer from 0 to 2,147,483,647 for the max-rec attribute. See <i>Note</i> This is not used for HTM Agent for DB2.

Note: For details about the record ID of each record, see Chapter 4.

A.8.2 List of Agent Collector Service Properties

Table A.11 lists the properties of the Agent Collector service of HTM Agent for DB2 that are output by executing `jpcasrec output` commands.

Table A.11 List of Agent Collector Service Properties of HTM Agent for DB2

Parameter Name	Element Name		Description
service	—		Outputs the service ID that identifies the Agent for the id attribute.
	record	—	Outputs the record ID for the id attribute. See <i>Note</i>
		log	Outputs "Yes" or "No" to indicate whether to record to the Store database.
		collection-interval	Outputs the data collection interval of performance data (in seconds) by the integer from 0 to 2,147,483,647.
		collection-offset	Outputs the offset value for the start of data collection. Output the integer (in seconds) within the range of the value of the collection-interval, from 0 to 32,767.
		logif	Outputs conditions for when records are recorded in the database. Only records that meet the conditions specified will be recorded.

Note: For details about the record ID of each record, see Chapter 4.

A.9 List of Files and Directories

The installation directory of Tuning Manager is: /opt/jp1pc/.

A.9.1 List of Files and Directories Common to Collection Manager and the Agent

Table A.12 lists the files and directories common to the UNIX edition of Collection Manager and the Agent.

Table A.12 List of Files and Directories Common to Collection Manager and Agent (continues on the following pages)

Directory Name	File Name	Description
/opt/jp1pc/	—	HTM common base directory
	jpgc_start	The service auto-startup script file See <i>Note</i>
	jpgc_start.model	The model file for the service auto-startup script file
	jpgc_stop	The service auto-stop script file See <i>Note</i>
	jpgc_stop.model	The model file for the service auto-stop script file
	jpgccomm.ini	The common startup-information file for each component
	jpgccomm.ini.model	The model file for the common startup-information file for each component
	jpgchosts	The host information file
	jpgchosts.model	The model file for the host information file
	jpgcns.ini	The service configuration file
	jpgcns.ini.model	The model file for the service configuration file
	jpgcplist.ini	The product definition file
	jpgcplist.ini.model	The model file for a product definition file
/opt/jp1pc/bin/	—	The storage directory for commands
	agtsolmlt	Internal command
	agtsolrm	
	getpinfo	
	iniedit	
	jpgccvtmdl	
	jpgciniupdate	
	jpgcmkindex	
	mdlevt_bkup	

PRELIMINARY RELEASE

Directory Name	File Name	Description
/opt/jplpc/bin/	mdlcv_t_exec	Internal command
	nscfg	
	pfm_inst	
/opt/jplpc/bin/action/	—	The base directory of the Action Handler service
	jpcah	The executable program of the Action Handler service
	jpcah.ini	The startup-information file of the Action Handler service
	jpcah.ini.model	The model file for the startup-information file of the Action Handler service
/opt/jplpc/bin/action/log/	—	The storage directory for log files of the Action Handler service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jplpc/log/	—	The storage directory for log files
	jpglog01	The common message log file
	jpglog02	The common message log file (alternate file)
/opt/jplpc/nls/\$LANG\$/	—	The storage directory for message catalogs
	jpgcmgrmsg.cat	Internal file
/opt/jplpc/pid/	—	Directory for storing internal control files
/opt/jplpc/tools/	—	The storage directory for commands
	jpgcagtsetup	The commands for adding and setting up Agent
	jpgcctrl	The service manager command
	jpgchasetup	The command for setting up a logical host environment
	jpgcimevt	Internal command
	jpgcinslist	The command for listing instance environments
	jpgcinssetup	The command for setting up an instance environment
	jpgcinsunsetup	The command for canceling setup of an instance environment
	jpgcnconfig	The command for defining the service configuration
	jpgcnshostname	The command for setting the host name
	jpgcras	The command for collecting maintenance materials

PRELIMINARY RELEASE

Directory Name	File Name	Description
	jpcresto	The command for restoring the Store database
	jpcstart	The command for issuing a service start request
	jpcstop	The command for issuing a service stop request
	jpctminfo	The command for displaying product information

PRELIMINARY RELEASE

Directory Name	File Name	Description
/opt/jp1pc/tools/log/	—	The storage directory for command log files
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)

Note: The user copies these files from the model files.

A.9.2 List of Files and Directories of HTM Agent for DB2

Table A.13 lists the files and directories of HTM Agent for DB2.

Table A.13 List of Files and Directories of HTM Agent for DB2 (continues on the next page)

Directory Name	File Name	Description
/opt/jp1pc/agtr/	—	The base directory of HTM Agent for DB2
/opt/jp1pc/agtr/agent/	—	The base directory of the Agent Collector service
	jpcagtr	The execution program of the Agent Collector service
/opt/jp1pc/agtr/agent/ <i>instance-name</i>	-	The root directory of the Agent Collector service (for each instance) See <i>Note 1</i>
	jpcagt.ini	The service startup initialization file of Agent Collector (for each instance) See <i>Note 1</i>
	jpcagt.ini.model	The model file for a service startup initialization file of the Agent Collector service (for each instance) See <i>Note 1</i>
/opt/jp1pc/agtr/agent/ <i>instance-name</i> /log/	—	The directory for storing internal log files of the Agent Collector service (for each instance) See <i>Note 1</i>
/opt/jp1pc/agtr/nls/	—	The storage directory for message catalogs
/opt/jp1pc/agtr/store/	—	The root directory of the Agent Store service
	jpcsto	The executable program of the Agent Store service
	stpqlpr	The executable program for backing up/exporting the Store database
	jpcsto.ini.instmpl	Internal file
	*.DAT	The definition file for a data model
/opt/jp1pc/agtr/store/ <i>instance-name</i> /	—	The base directory of the Agent Store service (for each instance) See <i>Note 1</i>
	*.DB	The performance data file (for each instance) See <i>Note 2</i>
	*.IDX	The index file for a performance data file (for each instance) See <i>Note 2</i>
	*.LCK	The lock file for a performance data file (for each instance) See <i>Note 2</i>
	jpcsto.ini	The service startup initialization file of the Agent Store service (for each instance) See <i>Note 1</i>

PRELIMINARY RELEASE

Directory Name	File Name	Description
/opt/jplpc/agtr/store/instance-name/	jpcsto.ini.model	The model file for a service startup initialization file of the Agent Store service (for each instance) See Note 1
	*.DAT	The definition file for a data model (for each instance) See Note 1
/opt/jplpc/agtr/store/instance-name/backup/	—	The default database backup directory (for each instance) See Note 1
/opt/jplpc/agtr/store/instance-name/dump/	—	The default database export directory (for each instance) See Note 1
/opt/jplpc/agtr/store/instance-name/log/	—	The directory for storing internal log files of the Agent Store service (for each instance) See Note 1
/opt/jplpc/setup/	—	The storage directory for setup files
	jpcagtru.Z	The archive file for setting up HTM Agent (UNIX)
	jpcagtrw.EXE	The archive file for setting up HTM Agent (Windows)

Notes:

- 1: Created by executing the jpcinssetup command.
- 2: Created when the Agent Store service starts.

Acronyms and Abbreviations

API	application programming interface
BMP	bitmap
CD- ROM	compact disk - read-only memory
DLL	dynamic linking library
DNS	domain name system
FQDN	fully qualified domain name
I/O	input/output
IPF	Itanium® Processor Family
LAN	local area network
MB	megabytes
MPP	massively parallel processor
OS	operating system
SAN	storage area network
SMP	symmetric multi-processor
UDP	user datagram protocol

