HITACHI
DATA SYSTEMS

HiCommand® Tuning Manager Agent Administration Guide

MK-92HC013-10

Trademarks

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

The following program products contain some parts whose copyrights are reserved by Sun Microsystems, Inc.: P-9D13-Y6312, P-9D13-Y6322, P-9D13-Y6332, P-9D13-Y6342, P-9D13-Y6372, P-9D13-Y6382.

The following program products contain some parts whose copyrights are reserved by UNIX System Laboratories, Inc.: P-9D13-Y6312, P-9D13-Y6322, P-9D13-Y6332, P-9D13-Y6342, P-9D13-Y6372, P-9D13-Y6382.

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-92HC013-07	December 2005	Revision 7, supersedes and replaces revision 6
MK-92HC013-08	June 2006	Revision 8, supersedes and replaces revision 7
MK-92HC013-09	August 2006	Revision 9, supersedes and replaces revision 8
MK-92HC013-10	December 2006	Revision 10, supersedes and replaces revision 9

Preface

The *HiCommand® Tuning Manager Agent Administration Guide* describes the functions of collecting and managing performance data. It is intended for system/account administrators who have a basic knowledge of both SANs (Storage Area Networks) and NAS (Network Attached Storage) and who are responsible for:

- Store database management
- Backup and disk management
- Cluster system set up and maintenance
- Data collection (system configuration detail records, log information, and workgroup information)

Note: The use of HiCommand Tuning Manager and all other Hitachi Data Systems® services and products is governed by the terms of your agreement(s) with Hitachi Data Systems.

Software Version

This document revision applies to software version 5.5.

Convention for Storage Capacity Values

Storage capacity values displayed by HiCommand Tuning Manager are calculated based on the following values:

1 KB (kilobyte) = 1,024 bytes
1 MB (megabyte) = $1,024^2$ bytes
1 GB (gigabyte) = $1,024^3$ bytes
1 TB (terabyte) = $1,024^4$ bytes

Referenced Documents

- *HiCommand Tuning Manager Server Administration Guide*, MK-92HC021
- *HiCommand Tuning Manager User's Guide*, MK-92HC022
- *HiCommand Tuning Manager Hardware Reports Reference*, MK-95HC111
- *HiCommand Tuning Manager Operating System Reports Reference*, MK-95HC112
- *HiCommand Tuning Manager Application Reports Reference*, MK-95HC113
- *HiCommand Tuning Manager Messages Reference*, MK-95HC114
- *HiCommand Tuning Manager Command Line Interface Guide*, MK-96HC119
- *HiCommand Tuning Manager Installation Guide*, MK-96HC141

Readme and Release Notes Contents

These files can be found on the installation CD. They contain requirements and notes for use of HiCommand Tuning Manager that may not be fully described in the manual. Be sure to review these files before installing HiCommand Tuning Manager.

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 6 Preparing for Failover in a Cluster System

Chapter 7 Collecting Log Information

Chapter 8 Collecting Workgroup Information

Chapter 9 Checking Service Status Using the Status Management Function

Chapter 10 Error Handling Procedures

List of Figures

List of Tables

Chapter 1 Starting and Stopping Collection Manager and Agent Services

This chapter explains the operations required to use the Tuning Manager series, such as how to start and stop Collection Manager and Agent services, and how to use service information.

Note that in this manual, the word **service** is used to indicate Collection Manager and Agent services.

This chapter covers the following topics:

- Understanding Services (see section 1.1)
- Types of Service (see section 1.2)
- Starting and Stopping a Service (see section 1.3)
- Starting an Agent in Stand-Alone Mode (see section 1.4)
- Using Service Information (see section 1.5)
- Precautions for Operations (see section 1.6)
- Notes on Disconnecting the Communication Line (see section 1.7)

1.1 Understanding Services

Collection Manager and Agent are made up of services, and can display and set the properties of each service.

A service is the control process for some functionality.

For details on the services used to accumulate and view performance data (in Main Console), see the *HiCommand® Tuning Manager Server Administration Guide* and the *HiCommand Tuning Manager User's Guide*.

When monitoring multiple storage subsystems or multiple instances in the database, an Agent service can be operated by multiple instances (monitored individually) depending on the instance environment structure.

1.2 Types of Service

The service names displayed in the Windows® Administration Tool differ from the names of the corresponding Tuning Manager services. Table 1.1describes the correspondence between Tuning Manager service names and Windows service names.

Table 1.1 Correspondence Between Service Names and Windows Service Names

Service Name		Windows Service Name
Collection Manager service	Name Server	PFM – Name Server
	Master Manager	PFM – Master Manager
	Master Store	PFM – Master Store
	View Server	PFM – View Server
	Correlator	PFM – Correlator
	Trap Generator	PFM – Trap Generator
Collection Manager Service Agent Service	Action Handler (see ***Note 1***)	PFM – Action Handler
	Status Server (see ***Note 2***)	PFM – Status Server
Agent Service	Agent Collector	(1) For Agent for RAID: PFM – Agent for RAID *instance-name* (2) For Agent for RAID Map: PFM – Agent for RAIDMap (3) For Agent for Platform (Windows): PFM – Agent for Windows (4) For Agent for SAN Switch: PFM – Agent for SANSwitch *instance-name* (5) For Agent for NAS: PFM – Agent for NAS *instance-name* (6) For Agent for Oracle®: PFM – Agent for Oracle *instance-name* (7) For Agent for Microsoft® SQL Server®: PFM – Agent for Microsoft SQL Server *instance-name* (8) For Agent for Microsoft Exchange Server: PFM - Agent for MSExchange
	Agent Store	(1) For Agent for RAID: PFM – Agent Store for RAID *instance-name* (2) For Agent for RAID Map: PFM – Agent Store for RAIDMap (3) For Agent for Platform (Windows): PFM – Agent Store for Windows (4) For Agent for SAN Switch: PFM – Agent Store for SANSwitch *instance-name* (5) For Agent for NAS: PFM – Agent Store for NAS *instance-name* (6) For Agent for Oracle PFM – Agent Store for Oracle *instance-name* (7) For Agent for Microsoft SQL Server: PFM – Agent Store for Microsoft SQL Server *instance-name* (8) For Agent for Microsoft Exchange Server: PFM - Agent Store for MSExchange

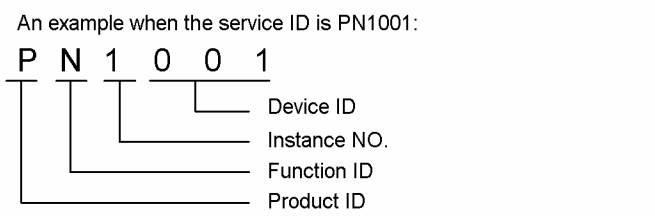
Note 1: Among the hosts that constitute the Tuning Manager series system, there is one Action Handler service at each Tuning Manager host and Agent host. If the Tuning Manager and Agent exist on the same host, or if a multiple number of Agents exist on the same host, only one Action Handler service will be allocated on the host.

Note 2: Among the hosts that constitute the Tuning Manager system, one Status Server service of a version that supports the status management function exists at each Tuning Manager host and Agent host. If the Tuning Manager and Agent exist on the same host, or if a multiple number of Agents exist on the same host, only one Status Server service will be allocated on the host.

1.2.1 Service ID

A unique ID is assigned to the Collection Manager and Agent services. This ID is called a **service ID**, and is specified when commands are used to check the system configuration for Tuning Manager series, or to back up performance data from individual agents.

A service ID consists of the following components:



The following describes the service ID components:

Product ID: The **product ID** is a one-byte identifier that indicates which service belongs to which Collection Manager and Agent program product. For Collection Manager services, the Action Handler service, or the Status Server service, the product ID is `P`. For details on each Agent product ID, see the list of IDs in the appendix.

- **Function ID**: The function ID is a one-byte identifier that indicates the function type of this service.

  Function IDs, their corresponding service names, and overviews of the function indicated by the function IDs are listed in Table 1.2.

- **Instance NO.**: The instance number is a one-byte identifier that indicates the management number that is used for internal processing.

- **Device ID**: The device ID consists of 1-255 bytes that indicate the location where this service is being run, such as the host in the Tuning Manager system. The device ID differs depending on the service.

  Note: Table 1.3 lists the service names and the corresponding device IDs.

Table 1.2 Function IDs, Service Names, and Function Overview

Function ID	Service Name	Function Overview
N	Name Server	Function that manages service configuration information that is within the system
M	Master Manager	Main function for Collection Manager
P	View Server	Internal function
E	Correlator	Function that controls event distribution between services
C	Trap Generator	Function that issues SNMP traps
H	Action Handler	Function that executes an action
A	Agent Collector	Function that collects performance data
S	Master Store	Internal function
	Agent Store	Function that stores performance data
T	Status Server	Function that manages the status of a service

Table 1.3 Service Name and Device ID

Service Name	Specified Device ID Contents
Name Server	Fixed at **001**.
Master Manager	Fixed at **001**.
Master Store	Fixed at **001**.
View Server	Host name is specified.
Correlator	Fixed at **001**.
Status Server	Host name is specified.
Trap Generator	Host name is specified.
Action Handler	Host name is specified.
Agent Collector	For an Agent for which an instance environment was not set up in the pre-operation setup, the host name is specified. For an Agent for which an instance environment was set up in the pre-operation setup, *instance-name*[*host-name*] is specified.
Agent Store	For an Agent for which an instance environment was not set up in the pre-operation setup, the host name is specified. For an Agent for which an instance environment was set up in the pre-operation setup, *instance-name*[*host-name*] is specified.

Examples:

- **Service ID for the Name Server service**: For the Name Server service, the product ID is specified as `P`, function ID as `N`, and device ID as `001`. The following is the service ID when the instance number is `1`: `PN1001`
- **Service ID for the View Server service**: For the View Server service, the product ID is specified as `P`, function ID as `P`, and device ID as *host-name*. The following is the service ID when the instance number is `1` and host name is `host01`: `PP1host01`
- **Service ID for the Agent Store service** (when an Agent instance is not created): For the Agent Store service of Agent for Platform (Windows), the product ID is specified as `T`, function ID as `S`, and the device ID as *host-name*. The following is the service ID when the instance number is `1` and host name is `host02`: `TS1host02`
- **Service ID for the Agent Store service** (when an Agent instance is created): For the Agent Store service of Agent for Oracle, the product ID is specified as `O`, function ID as `S`, and the device ID as *instance-name*`[`*host-name*`]`. The following is the service ID when the instance number is `1`, instance name is `oracleA`, and host name is `host03`: `OS1oracleA[host03]`

1.2.2 Service Key

When executing commands, for example to start and stop services for Collection Manager and Agent, identifiers called **service keys** are used.

Table 1.4 contains a list of service keys. For details on the Agent service keys, see Appendix A.

Table 1.4 Service Keys

Service Key	Meaning
all	Indicates all Collection Manager and Agent services.
mgr	Indicates the Collection Manager service.
act	Indicates the Action Handler service.
stat	Indicates the Status Server service.

1.3 Starting and Stopping a Service

This section describes how to start and terminate services of Collection Manager and Agent.

The following OS user permissions are required for performing the operations described in this section:

- For Windows: administrator permissions
- For UNIX®: root user permissions

You must install, configure, and start the monitoring target before you start an Agent service.

When performing the following operations, you do not need to be aware of the order in which Collection Manager and Agent services start and stop:

- Starting and stopping services automatically
- Using the service start command (`jpcstart`) and service stop command (`jpcstop`) to start and stop all services

When using administration tools in Windows to change service properties and then start services, start the services in the following order.

For Collection Manager services:

- Status Server (see ***Note***)
- Name Server
- Master Manager
- Master Store
- Correlator
- Trap Generator
- View Server
- Action Handler

For Agent services:

- Status Server (see ***Note***)
- Action Handler
- Agent Store
- Agent Collector

Note: This service starts only when the status management function is enabled.

When terminating a service, reverse the start sequence described above.

Observe the following precautions if terminating the Tuning Manager first:

- When terminating the Agent after terminating the Tuning Manager: If you restart only Tuning Manager, the inactive Agent is recognized as active. The actual Agent status will not be recognized by the Tuning Manager.
- When not terminating the Agent after terminating the Tuning Manager: Even while Tuning Manager is inactive, the Agent collects performance data and saves the data into the Agent. Then, later when the Tuning Manager is restarted, the Tuning Manager retrieves the data collected by the Agent. Using this functionality, Tuning Manager attempts to restore the continuous collection of performance data. However, the performance data collected by the Agent will be deleted from the oldest. Therefore, all or a portion of the collected data might not be displayed if a long period of time passes before Tuning Manager is restarted. The period of time the data will be retrievable depends on the environment and the data being collected.

The Agent services can be started even if no Collection Manager services are running. However, the functionality available for the Tuning Manager series programs will be restricted. We recommend that you start the Collection Manager services before attempting to use the Agent services. For information on starting Agent independently, see section 1.4.

If startup of a service fails, use the common message log to check why the service did not start correctly. After the error is corrected, restart the service.

Note that even if startup of the Agent Collector service fails, the Agent Store service remains active. If this occurs, use the `jpcstop` command to stop the Agent Store service, check why the Agent Collector service did not start correctly, and then resolve the problem.

We recommend that you stop the Agent services before attempting to stop the Collection Manager services.

1.3.1 Starting a Service Manually

Use the `jpcstart` command to start services manually. This command enables you to start just the services on the host where you are logged in. Use the `jpcstart` command to start the following services:

- All Collection Manager and Agent services on a host
- The Collection Manager service on a host
- The Action Handler service on a host
- The Status Server service on a host
- The specific Agent service on a host

To start the services manually:

1. Log on to a host where Tuning Manager or Agent is installed.
2. Specify the service key that indicates the service you want to start, and execute the `jpcstart` command. For example, if you want to start all services on the host, specify as below and execute the command:

```
jpcstart all
```

If you want to start an Agent by instance in an instance environment, execute the `jpcstart` command, specifying a desired instance name. For example, to start an Agent for Oracle service with an instance name of `oracleA`, execute the command as follows:

```
jpcstart agto inst=oracleA
```

For details on the `jpcstart` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

1.3.2 Starting a Service Automatically

This section describes the methods used to start each service automatically for each OS.

Windows systems

In Windows systems, the services are configured to start automatically when the system is restarted by default. This section explains how to cancel the automatic startup and reapply the automatic startup.

To change the service start settings:

1. Log on to a host where Tuning Manager or Agent is installed.
2. Select the name of a service for which you want to change the settings. (The service name indicated here is a Windows service name. For information on the correspondence of service names between Windows, and Collection Manager and Agent, see section 1.1.)

   In Windows 2000, from the **Start** menu, click **Settings**, **Control Panel**, **Administrative Tools**, and then **Services**. In the displayed window, select the desired service.

   In Windows Server® 2003, from the **Start** menu, click **Administrative Tools**, and then **Services**. In the displayed window, select the desired service.
3. Select the startup type. To cancel automatic startup, select **Manual** for the startup type. To use automatic startup, select **Automatic** for the startup type.

   Note: Do not change the service account settings. If you do, the service may not operate properly.

UNIX systems

In UNIX systems, to automatically start services during system startup, use the service automatic start script file for the Tuning Manager series system. In AIX® systems, use the automatic start script file for AIX. In Linux®, the automatic start functionality of the Agent service is enabled when you start the service on runlevel 3 (OS default) or 5. If you attempt to start services on runlevel 1 or 2, the services do not start automatically.

The following shows the procedure for specifying automatic service startup.

Note: If you are using HP-UX®, Solaris™, or Linux systems, perform steps 1 to 3. If you are using AIX, perform steps 1 to 4.

1. Log on to a host where Tuning Manager or Agent is installed.
2. Execute the following command to move to the /opt/jp1pc directory:

   ```
   cd /opt/jp1pc
   ```

3. Specify the service automatic start script file for the Tuning Manager series system.

   The following are the names of the .model file for the service automatic start script and the service automatic start script file:

   - The .model file name for the service automatic start script:

     ```
     jpc_start.model
     ```

   - The service automatic start script file name:

     ```
     jpc_start
     ```

   Copy the .model file for the service automatic start script to the service automatic start script file, and then add execution permissions. Execute the following commands:

   ```
   cp -p jpc_start.model jpc_start
   chmod 555 jpc_start
   ```

4. If you are using AIX, register an automatic start script file for AIX.

   The Tuning Manager series programs provide an automatic start script file for AIX to execute the service automatic start script file for the Tuning Manager series programs specified in step 3. Register this automatic start script file in the AIX setup file.

   - Automatic start script file:

     ```
     /etc/rc.jp1_pc
     ```

   - AIX Setup file:

     ```
     /etc/inittab
     ```

   To register an automatic start script file in the AIX setup file:

   a) Use the mkitab command to register the /etc/rc.jp1_pc file in the /etc/inittab setup file:

      ```
      mkitab "jp1pc:2:wait:/etc/rc.jp1_pc >/dev/console 2>&1"
      ```

   b) Use the lsitab command to make sure that the /etc/rc.jp1_pc file is registered in the /etc/inittab setup file:

      ```
      lsitab jp1pc
      jp1pc:2:wait:/etc/rc.jp1_pc >/dev/console 2>&1
      ```

      When you use the mkitab command, the entry for the /etc/rc.jp1_pc file is added as the last line of the /etc/inittab setup file. If the /etc/inittab setup file already contains entries for programs to be linked through action execution, edit the /etc/inittab setup file so that these entries come after the line for the /etc/rc.jp1_pc file.

      Also, uninstallation does not delete the lines registered in the /etc/inittab setup file.

To cancel the registration when uninstalling an Agent:

a) Use the `rmitab` command to cancel registration of the `/etc/rc.jp1_pc` file in the `/etc/inittab` setup file as follows:

```
rmitab jp1pc
```

b) Use the `lsitab` command to make sure that the `/etc/rc.jp1_pc` file is not registered in the `/etc/inittab` setup file as follows:

```
lsitab jp1pc
```

If no files are registered to the `/etc/rc.jp1_pc` file, nothing is displayed when executing the above command. Make sure that nothing is displayed.

When you want to automatically start only a specific service

The service automatic start script file can only be used to start services on physical hosts, and cannot be used to start services on logical hosts.

To automatically start a specific service only, edit this file as follows:

Before

```
nohup /opt/jp1pc/tools/jpcstart all -nochk 2> /dev/null 1> /dev/null &
```

After

```
nohup /opt/jp1pc/tools/jpcstart act -nochk 2> /dev/null 1> /dev/null
nohup /opt/jp1pc/tools/jpcstart service-key -nochk 2> /dev/null 1> /dev/null &
```

Notes:

- Add the first line only when Action Handler startup is required. Do not include an ampersand (`&`) at the end of the first line.
- For *service-key* in the second line, specify the service key for the service to be started automatically.

Note:

If automatic service startup is specified in a version earlier than 3.50, the problem explained below may occur.

If Agent Service terminates abnormally, such as due to a power outage, the Store database index will be rebuilt when the Agent Store service is started, which means that startup will take a long time. During this period, the Agent Collector service might not be able to start because it will not be able to communicate with the Agent Store service.

The model file (`jpc_start.model`) for the service automatic start script, which is registered by the Tuning Manager series programs of this version, supports this problem. Therefore, reconfigure the automatic start settings by using this model file. Note that we recommend that you back up the automatic start script file before the reconfiguration.

With respect to a service starting using the reconfigured automatic start script file, the operations are changed as described below:

- Because startup of the Collection Manager and Agent services is executed in the background, startup processing for these services might still be in progress even though OS startup has completed.
- Because the Collection Manager and Agent services are started synchronously, startup completion may take longer than it did before reconfiguration.

– The following message, which used to be displayed on the console before reconfiguration, is no longer displayed. To make sure that the Collection Manager and Agent services have started, see the common message log.

  KAVE06007-I The service will now start. (service=*service-name*, lhost=*logical-host-name*, inst=*instance-name*)

1.3.3 Stopping a Service Manually

The jpcstop command is used to terminate services manually. This command can terminate only the following services on the logged-in host:

- All Collection Manager and Agent services on a host
- The Collection Manager services on a host
- The Action Handler services on a host
- The Status Server service on a host
- Specific Agent services on a host

Check the operating status of the service on the host before terminating a service manually. The jpcctrl list command is used for checking the service operating status. This command can check the operating status of all services in the Tuning Manager series system or services on specific hosts.

To check service-operating statuses and terminate the services manually:

1. Log on to a host where Tuning Manager or Agent is installed.
2. Execute the jpcctrl list command. For example, specify and execute the following command to check all services operating on the local host on the entire Tuning Manager series system:

   ```
   jpcctrl list "*"
   ```

   For details on information that can be displayed by executing the jpcctrl list command, see section 1.5.1.
3. Specify the service key of the service you want to terminate, and execute the jpcstop command. For example, to terminate all services on the local host, execute the following command:

   ```
   jpcstop all
   ```

   To terminate an Agent running in an instance environment for a particular instance, execute the jpcstop command, specifying its instance name. For example, to terminate Agent for Oracle service with an instance name oracleA, execute the following command:

   ```
   jpcstop agto inst=oracleA
   ```

To stop a specific Collection Manager or Agent service, check the Host Name, ServiceID, and Service Name displayed by the jpcctrl list command to determine whether the service running on the local host is a Collection Manager service or Agent service, and then execute the jpcstop command with the appropriate service key specified.

For details on the `jpcctrl list` command and `jpcstop` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

Note: If the Agent Collector service is busy collecting performance data, executing the `jpcstop` command might not be able to stop the Agent Collector service. If the following message is not output to the standard output even though the command process has finished, wait a few minutes and re-execute the command.

```
KAVE06008-I The service will now stop. (service=service-name,
lhost=logical-host-name, inst=instance-name)
```

1.3.4 Stopping a Service Automatically

This section describes the methods used to stop each service automatically for each OS.

Windows systems

In Windows systems, the services terminate automatically when the system terminates. Therefore, specifying the automatic termination function is not required.

UNIX systems

In UNIX systems, to automatically terminate services during a system termination, use the service automatic stop script file for the Tuning Manager series system. In AIX systems, also use the automatic stop script file for AIX. In Linux, the automatic stop functionality of the Agent service is effective when starting on runlevel 0 or 6 of the OS regulations. If you attempt to stop services on runlevel 1 or 2, services do not stop automatically.

If you stop the system without stopping the services, inconsistencies might occur in the database. If you do not stop the services manually, make sure to set up the automatic termination.

The following shows how to specify automatic service termination. If you are using HP-UX, Solaris, or Linux systems, perform steps 1 to 3. If you are using AIX, perform steps 1 to 4.

1. Log on to a host where Tuning Manager or Agent is installed.
2. Execute the following command to move to the `/opt/jp1pc` directory:

   ```
   cd /opt/jp1pc
   ```

3. Specify the service automatic stop script file for the Tuning Manager series system.

   The following are the names of the `.model` file for the service automatic stop script and the service automatic stop script file:

   - The `.model` file name for the service automatic stop script: `jpc_stop.model`
   - The service automatic stop script file name: `jpc_stop`

   Copy the `.model` file for the service automatic stop script to the service automatic stop script file, and then add execution permissions. Execute the following commands:

   ```
   cp -p jpc_stop.model jpc_stop
   chmod 555 jpc_stop
   ```

4. If you are using AIX, register the automatic stop script file for AIX.

   In the automatic stop script file for AIX, register the service automatic stop script file for the Tuning Manager series system specified in step 3.

   Automatic stop script file name:

   ```
   /etc/rc.shutdown
   ```

   Add the following lines to the automatic stop script file; there is no need to consider the sequence of stopping the services:

   ```
   if [ -x /opt/jp1pc/jpc_stop ]; then
          /opt/jp1pc/jpc_stop
   fi
   ```

   If the `/etc/rc.shutdown` file does not exist, create it and then specify the file attributes by executing the following commands:

   ```
   chmod 550 /etc/rc.shutdown
   chown root /etc/rc.shutdown
   chgrp shutdown /etc/rc.shutdown
   ```

   Note: Uninstallation does not delete the added lines and the `/etc/rc.shutdown` file. Delete the added lines as required.

1.4 Starting an Agent in Stand-Alone Mode

You can collect Performance data by starting only the Agent even if the Master Manager service and Name Server service of the Collection Manager cannot be started due to an error. The status in which Agent is running independently is called **stand-alone mode**.

If the Master Manager service and Name Server service of the Collection Manager are not running when the Agent is started, the Agent starts in the stand-alone mode and collects performance data. In stand-alone mode, the connection to Tuning Manager is checked once every 5 minutes. Once the Agent starts in stand-alone mode, if Tuning Manager starts and connection confirmation from the Agent is successful, the Agent terminates the stand-alone mode, and moves to the normal mode in which the Agent is connected to Tuning Manager. The performance data accumulated on the Agent in the stand-alone mode can be viewed as a historical report.

Note: An Agent cannot be started independently if it is installed on the same host as Tuning Manager.

1.4.1 Functions Available in Stand-Alone Mode

Table 1.5 lists the functions available in stand-alone mode.

Table 1.5 Functions Available in Stand-Alone Mode

Function	Availability	Service Name
Starting and terminating services, and checking operating status	Yes	Agent Store, Agent Collector, Action Handler
Collecting log data	Yes	Agent Store, Agent Collector
Displaying reports	N/A	Agent Store, Agent Collector
Monitoring performance data by an alarm	N/A	Agent Collector
Performing an action for an alarm event	N/A	Action Handler
Managing the status of a service	Yes	Status Server

1.4.2 Commands Available in Stand-Alone Mode

Table 1.6 lists the commands available in stand-alone mode.

Table 1.6 Commands Available in Stand-Alone Mode

Command	Function	Availability
`jpcctrl backup`	Creates a backup file for data stored in the database of the Master Store service or Agent Store service	Yes (see ***Note 1***)
`jpcctrl clear`	Deletes data stored in the database of the Master Store service or Agent Store service	N/A
`jpcctrl delete`	Deletes service information of an agent registered in the Tuning Manager series programs	N/A
`jpcctrl dump`	Exports data stored in the database of the Master Store service or Agent Store service	Yes (see ***Note 1***)
`jpcctrl list` (when `host` option is not specified)	Lets you check the operating status of services on the local host	Yes
`jpcctrl list` (when specifying another host with `host` option)	Displays the configuration and status for Collection Manager and Agent services	Yes (see ***Note 2***)
`jpcctrl register`	Re-registers collection information for Collection Manager and Agent services	N/A
`jpcras`	Collects information about Collection Manager or Agent	Yes
`jpcstart`	Starts services	Yes
`jpcstop`	Stops services	Yes
`jpcstsetup`	Enables and disables the status management function	Yes

Note 1: This command can be executed in the stand-alone mode only when the `-alone` option is specified.

Note 2: This command can be executed in the stand-alone mode only when the `-stat` option is specified.

For details on the commands and command options, see *HiCommand Tuning Manager Command Line Interface Guide*.

1.5 Using Service Information

Service information that can be displayed using the `jpcctrl list` command may not be deleted when the Tuning Manager series program is uninstalled. If this occurs, the service information must be deleted with other commands. Furthermore, if the activated service information is deleted by mistake, it can be re-registered.

The `jpcctrl delete` command is used for deleting service information and the `jpcctrl register` command is used for re-registering deleted service information.

This section describes the methods for displaying, deleting, and re-registering the service information.

The following OS user permissions are required for the operations described in this section:

- Windows systems: Administrator permissions or Backup Operator permissions
- UNIX systems: root user permissions

1.5.1 Displaying Service Information

To display service information, use the `jpcctrl list` command as follows:

1. Log on to a host where Tuning Manager or Agent is installed.
2. Execute the `jpcctrl list` command, specifying the ID of the service for which you want to display information.

   For example, specify and execute the following command to check the operating status of all services on the host `host01`:

   ```
   jpcctrl list "*" host=host01
   ```

The following table lists the information that can be displayed by executing the `jpcctrl list` command.

Table 1.7 Information Displayable by the jpcctrl list Command

Output Data	Description
Host Name	Host name of the operating service
ServiceID	Service ID
Service Name	Service name
PID	Service's process ID ▪ When the version does not support the status management function: PID is displayed only when `Status` is `Active`. ▪ When the version supports the status management function: PID is displayed when `Status` is `Active`, `Busy`, `S Active`, `S Busy`, `Starting`, or `Stopping`.

Output Data	Description
Port	Communication port number being used by the service ▪ When the version does not support the status management function: The port number is displayed only when `Status` is `Active`. ▪ When the version supports the status management function: The port number is displayed when `Status` is `Active`, `Busy`, `S Active`, or `S Busy`.
Status	Service status ▪ Statuses displayed when the version does not support the status management function, or when the function is supported but not enabled: `Active`: Active `Inactive`: Unable to establish communication, or connection is terminated. `Comm Err`: Communication is possible, but there is no response. `Timeout`: Communication timed out. `Error`: An error other than a communication timeout occurred. ***Note:*** If the status management function is enabled on the same host as a version that supports the function, the status is displayed with an asterisk (`*`) appended to the end. ▪ Statuses displayed when the status management function is enabled on a version that supports this function: `Active`: The service is waiting for a request. `Inactive`: The service has stopped. `Starting`: The service is starting. `Busy`: The service is processing a request. `S Active`: The service is waiting for a request. (stand-alone mode) `S Busy`: The service is processing a request. (stand-alone mode) `Stopping`: The service is stopping. ▪ Statuses displayed when the status server service has stopped. Same as those displayed when the version does not support the status management function.

For details on the `jpcctrl list` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

Note: If the status management function is not enabled, executing the `jpcctrl list` command on the Agent Collector service or Agent Store service may return the **Inactive** or **Timeout** message even though the service is running. This means that the Agent Collector service or Agent Store service is busy collecting performance data. To display the service status correctly, enable the status management function. For details on the status management function, see Chapter 9.

1.5.2 Deleting Service Information

To delete service information, use the `jpcctrl delete` command as follows:

1. Log on to a host where Tuning Manager is installed.

2. Execute the `jpcctrl delete` command, specifying the ID of the service for which you want to delete information. For example, specify and execute the following command to delete information about Agent Store services of Agent for Oracle on the host `host02`:

   ```
   jpcctrl delete OS* host=host02
   ```

3. Restart Tuning Manager to reflect the deleted service information.

For details on the `jpcctrl delete` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

1.5.3 Re-registering Service Information

Service information can be re-registered only from a host where Tuning Manager is installed.

To re-register service information, use the `jpcctrl register` command as follows:

1. Log on to a host where Tuning Manager is installed.
2. Execute the `jpcctrl register` command, specifying the ID of the service for which you want to re-register information. For example, specify and execute the following command to re-register information about the Agent Store services of Agent for Oracle on the host `host02`:

   ```
   jpcctrl register OS* host=host02
   ```

   For details on the `jpcctrl register` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

1.6 Precautions for Operations

This section gives precautions on operating the Tuning Manager series programs.

1.6.1 Changing the Time on the Agent Machine

When changing the current time of the Agent machine, note the following:

- Before changing the time settings of a machine, stop all Agent services installed on the machine. After changing the time settings, restart the Agent services.

  On the host where Agent for RAID Map is installed, if you reset the machine's clock back, delete all files under the following directory before restarting the Agent services.

  For Windows:

  installation-folder`\agte\agent\hldutility\log\*`

  For UNIX:

  `/opt/jp1pc/agte/agent/HLDUtility/log/*`

- When setting the time forward from the currently set time, log data from the time prior to making the change to after the change is made will not be saved.
- When setting the time back to a time prior to the current time, data will be collected and log data will be stored from the time after the change is made. Collected data and log data with dates and times prior to the change will be overwritten. For example, if the current time is 12:00 on July 5, 2005, and this time is changed to 00:00 on July 1, 2005, collected data and log data from 00:00 on July 1 to 12:00 on July 5 will be overwritten. If you want to save the log data and data collected prior to the change, execute the `jpcctrl backup` command to back up the data and log data before changing the time settings.

1.6.2 When Creating a New Database on the Monitoring-target Microsoft SQL Server

If you attempt to create a new database in a Microsoft SQL Server for which Agent for Microsoft SQL Server is collecting data, your attempt might fail due to the conflict with the data collection processing. To avoid this, stop all services of Agent for Microsoft SQL Server instances, and then create the new database.

1.6.3 Restarting the Monitoring-target Microsoft SQL Server

If you restart the monitoring-target Microsoft SQL Server while Agent for Microsoft SQL Server is running, the data collected after the restart will be corrupted. Therefore, to restart Microsoft SQL Server, you must restart the Agent for Microsoft SQL Server service.

1.7 Notes on Disconnecting the Communication Line

When using Tuning Manager in an environment where you are charged for connection time, note that Tuning Manager does not disconnect the line until 70 seconds after you finish communicating with the connection destination.

To disconnect the line immediately after finishing communication, edit the `jpccomm.ini` file as follows:

1. Stop all services of Tuning Manager series programs.
2. Open the `jpccomm.ini` file using a text editor.
3. Change the line connection mode.

   Change the label value in all sections of the `jpccomm.ini` file as follows:

   ```
   NS Keepalive Mode=0
   ```
4. Save the `jpccomm.ini` file, and then close it.
5. Restart the service.

Chapter 2 Overview of Data Handled by Tuning Manager Series Programs

This chapter describes the performance data that is handled by the Tuning Manager series products. It also explains how to collect and manage performance data.

Individual performance data can be viewed and monitored by using Performance Reporter.

- Overview of Data Handled by Tuning Manager Series Products (see section 2.1)
- Overview of Performance Data (see section 2.2)
- Collecting Performance Data (see section 2.3)
- Conditions for Performance Data to be Summarized and Stored (see section 2.4)

2.1 Overview of Data Handled by Tuning Manager Series Products

The Tuning Manager series products include functions for collecting various data and for efficiently managing such data in order to monitor the operation of target systems. This section provides an overview of the data handled by the Tuning Manager series products.

2.1.1 Types of Data Handled by Tuning Manager Series Products

Of the performance data collected by Agents, the Tuning Manager series products store only specified performance data in a database. This database is called a *Store database*.

The Store database manages the data, depending on the characteristics and properties of the data, by summarizing and overwriting existing data to prevent the size of the data from becoming excessive. Therefore, performance can be monitored in certain resources.

Performance data is data collected from target systems that are being monitored. It indicates the operating status of the monitored systems.

There are two types of performance data:

Real-time data

Real-time reports contain performance data that indicates the current statuses of the systems being monitored. These reports are primarily used to check the current statuses of, and problems in, systems. Real-time data is not stored in a Store database.

Historical data

Historical reports contain performance data that indicates how the statuses of the monitored systems have changed from the past to the present time. These reports are primarily used to analyze system trends. Historical data is stored in an Agent database in one of the following two formats, depending on the data properties:

- Summarized record

  Values collected by an Agent are automatically calculated and summarized into data, such as averages and total values, in units of minutes, hours, days, weeks, months, and years, and then stored in a Store database.

  After the fixed retention period elapses, the summarized records are overwritten in chronological order by the latest data. Records whose name starts with `PI` are stored in this format.

- Non-summarized record

  Performance data collected by an Agent is stored into a Store database as is.

When the fixed number of records is exceeded, non-summarized records are overwritten in chronological order by the latest data. Records whose name starts with `PD` and `PL` are stored in this format.

2.1.2 About Data Models

Performance data is collected in the form of *records*. Each record is further divided into units called *fields*. The collective name for records and fields is *data model*. Data models are managed by version.

Figure 2.1 provides a conceptual diagram of a data model, records, and fields.

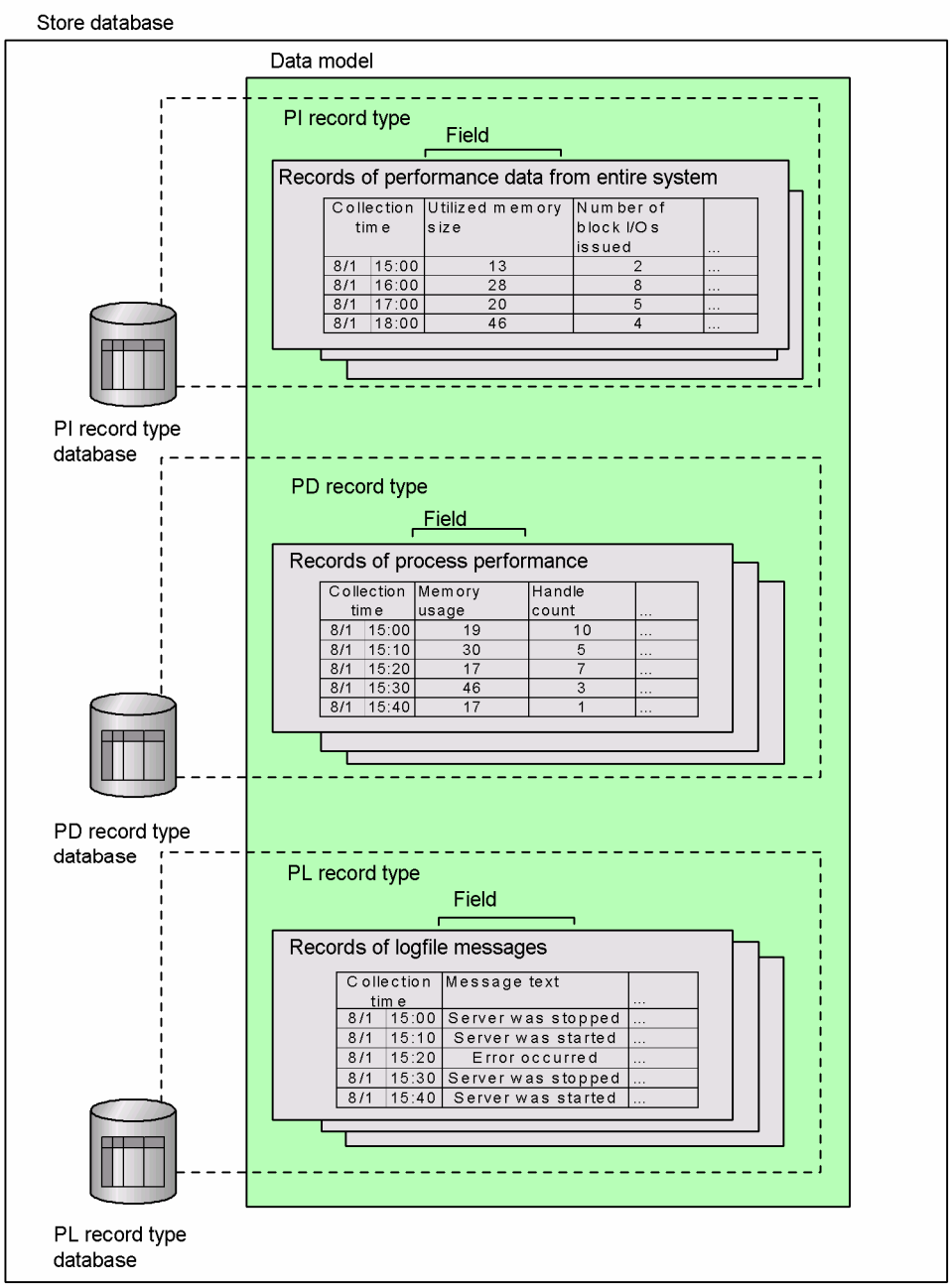


Figure 2.1 Conceptual Diagram of a Data Model, Records, and Fields

2.2 Overview of Performance Data

This section provides an overview of performance data.

2.2.1 Overview of Data

Tuning Manager series products classify performance data into record types based on the characteristics of the data. The table below summarizes the record types.

Table 2.1 Record Types

Name of Record Type		Description	Purpose
Summarized record	Product Interval record type (abbreviated as PI record type)	Collects performance data over specified intervals, such as the number of processes per minute. This record type is suitable for monitoring system performance over extended time periods.	Used to analyze changes and trends in system status over time, such as: ▪ Change in the number of system calls issued over a specified period of time ▪ Change in the capacity of the file system in use
Non-summarized record	Product Detail record type (abbreviated as PD record type)	Collects performance data indicating system status at a specified time, such as detailed information about currently active processes. This record type is suitable for analyzing system status when a problem occurs.	Used to obtain system status at a specified time, such as: ▪ CPU utilization rate by process ▪ Capacity of the current file system in use
	Product Log record type (abbreviated as PL record type)	Records of the PL type contain logs and messages from the system and applications. This record type is useful for checking system messages that are output when a problem occurs.	Use this record type when you want to check messages from the system or applications.

Certain monitored systems may also include additional record types that you can use. For more information about the records types that can be used, the records of each record type, and the data model versions, see the chapter that describes records in the following manuals:

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

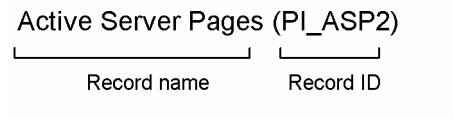
2.2.1.1 Record Name Format and Field Name Format

This section describes the formats for record and field names of performance data. For details about records and fields, see the chapter that describes records in the following manuals:

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

Record name

Each record has a record name and an associated record ID.

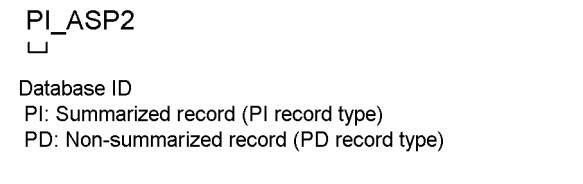


- Record name

  Each record is assigned a record name that indicates the monitored item.

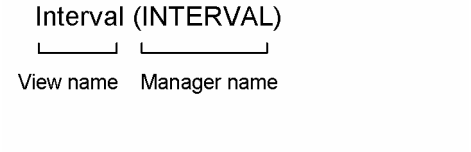
- Record ID

  The first two characters of the record ID represent the database ID of the database in which the record is stored. The database ID indicates the record type.



Field name

Each field making up a record has the following two names associated with it.



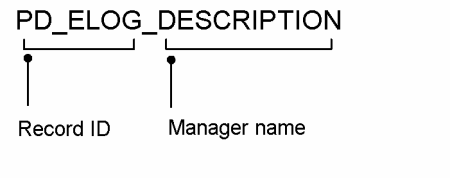
- View name

  This indicates the field name displayed in Performance Reporter.

- Manager name

This indicates the field name used in SQL statements when Tuning Manager employs SQL to utilize field data stored in a Store database. When coding a SQL statement, you add the record ID at the beginning of the field name.

For example, the Description (DESCRIPTION) field of the Event Log (PD_ELOG) record for Agent for Platform (Windows) is written as PD_ELOG_DESCRIPTION.



2.2.1.2 Record Recording Format

There are two types of record recording formats: single instance and multiple instance.

Single instance record

This is a record recorded on a single line.

Multiple instance record

This is a record recorded on multiple lines.

2.2.1.3 About the Lifecycle of Performance Data

This section describes the time period over which the integrity of the performance data is guaranteed.

The Agent collects performance data as follows:

- Real-time data is collected at the data refresh intervals specified in the report definition.
- Historical data is collected at the intervals specified in the Collection Interval property of the relevant record.

If an Agent determines that the collected performance data is the same type as that collected previously, it recognizes the data as belonging to the same field of the same record.

For example, for Process Detail (PD) records on Agent for Platform (Windows), the Agent uses the process name and process ID to determine if the performance data is of the same type.

Performance data collected between the time a process is generated and the time it is terminated is recognized as performance data from the same process. In this case, the integrity of the performance data is guaranteed.

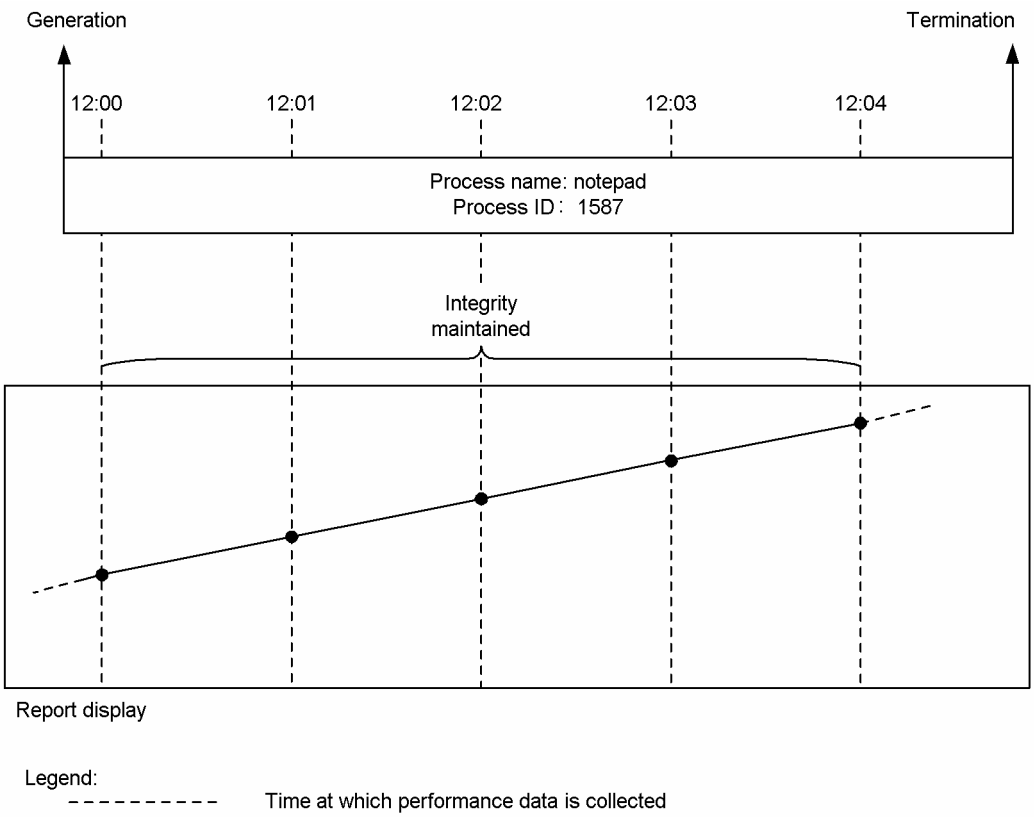


Figure 2.2 Consistency of Performance Data Collected Between the Time a Process Is Generated and the Time It Is Terminated

In contrast, if a process terminates and is then regenerated within a collection interval, and the names and process IDs of these process instances are identical, the performance data is still recognized as having come from the same process. In this case, the integrity of the performance data is not guaranteed.

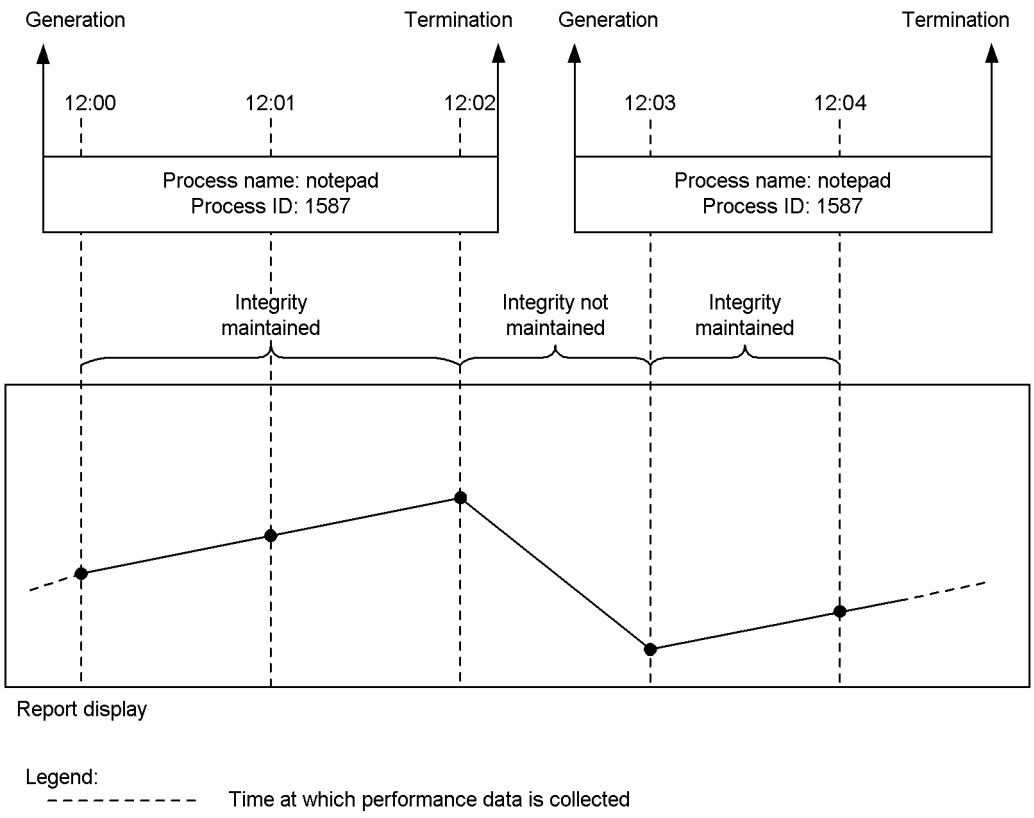


Figure 2.3 Consistency of Performance Data When a Process Terminates and Is Then Regenerated

In other words, the only data whose integrity is guaranteed is performance data of the exact same process instance starting from the time the monitored process is generated and ending when it is terminated.

The time period over which the integrity of performance data is guaranteed in this way is called its *lifetime*.

We recommend you carefully consider the lifetime of the performance data when you specify the performance data collection interval, report display interval, and so on.

2.3 Collecting Performance Data

Performance data is collected by the Agent Collector service and managed as records. Note that performance data collected by the Agent Collector service may or may not be stored in a Store database by the Agent Store service. You can use performance data that is stored in a Store database to display historical reports. Performance data that is not stored in a Store database is used for real-time report display.

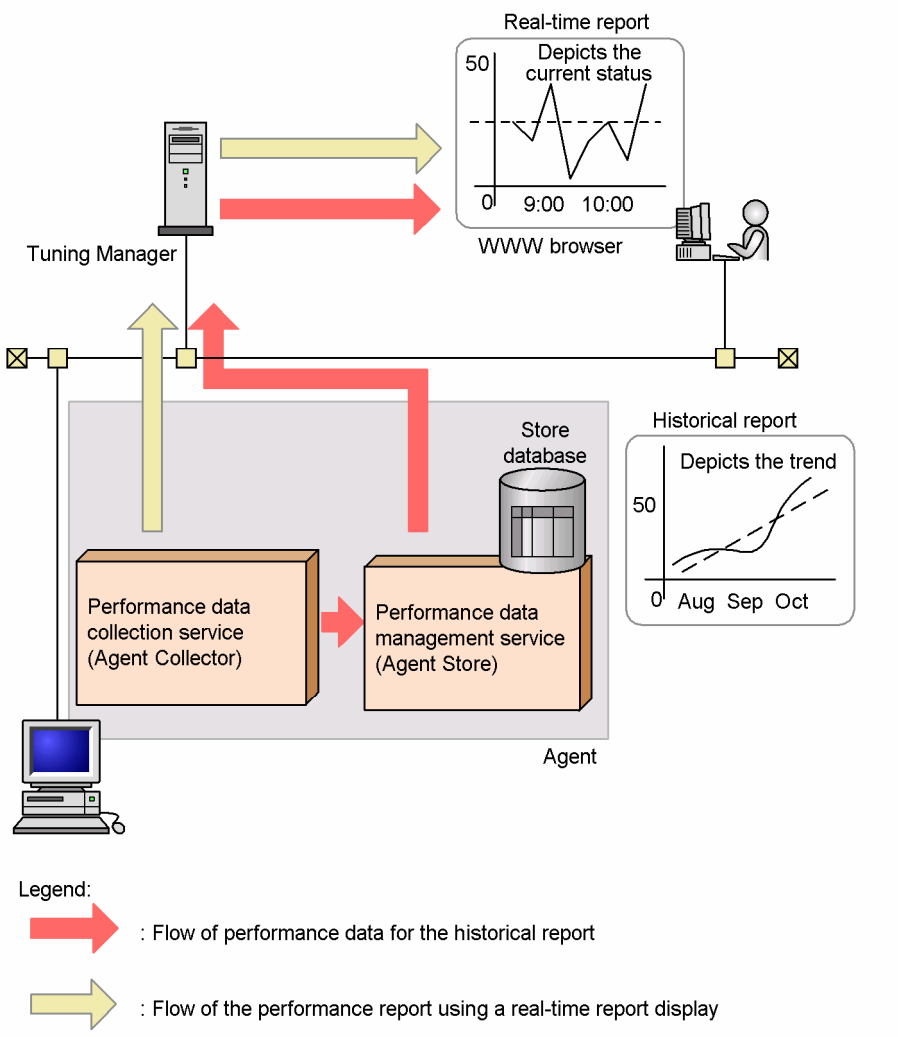


Figure 2.4 Flow of Performance Data From the Time It Is Collected To the Time It Is Used for Report Display

Tuning Manager series products allow you to specify how performance data will be recorded into a Store database.

The data recording method is set using Performance Reporter. You can specify the following options:

- Whether to record the collected performance data into a Store database
- The performance data collection interval
- The offset value for starting the collection of performance data
- Judgment criterion as to whether to record performance data into a Store database

The data recording method differs for each stored record. For details on the recording methods you can set for each record type, see the chapter that describes records (record default values and specifiable values) in the following manuals:

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

For information about how to specify these settings, see the *HiCommand Tuning Manager Command Line Interface Guide*.

Notes:

- Increasing the number of records in which performance data is collected may adversely affect disk capacity and system performance. When configuring the records to be collected, specify only items that must be monitored, based on a consideration of the requirements for collecting performance data, such as required disk capacity and record collection interval.

  For information about required disk capacity, see the description of system requirements in the appendix of the *HiCommand Tuning Manager Installation Guide*.
- Collection of records of the PI record type does not begin at the time specified in Performance Reporter for collecting performance data. Collection of data begins from the next collection start time.
- For the record collection interval, specify either the default value or a value of 60 seconds or greater that is also a divisor of 3,600. If you specify a record collection interval exceeding 3,600 seconds, specify an integral multiple of 3,600 and a divisor of 86,400. If you specify a value for the record collection interval that is less than the default value or if you specify a value less than 60 seconds, the number of open operations on files and utilized memory increases. This prevents Store database processing, which means that the collected performance data can no longer be saved.

The following sections describe how performance data is stored.

2.3.1 Historical Data

The following figure shows the flow of performance data until the data is stored in a Store database.

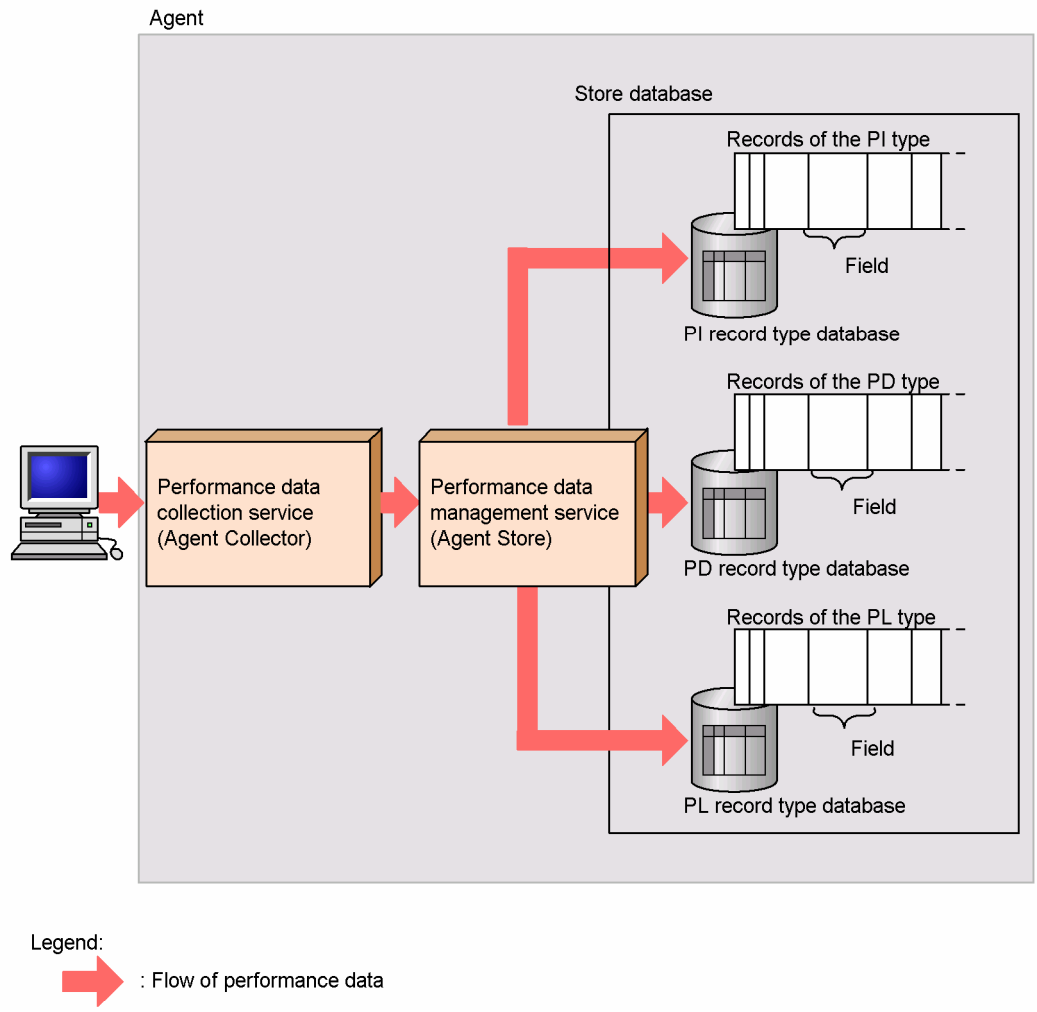


Figure 2.5 Process Flow Until Performance Data is Stored in a Store Database

Performance data is stored at specified intervals in a Store database. The default performance data collection interval differs by record. For information about the default collection interval value, see the chapter that describes records in the following manuals:

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

The following section explains the collection start time and collection method for performance data that is to be stored in a Store database.

2.3.1.1 Collection Start Time for Performance Data

When performance data is stored in a Store database, its collection start time is determined by the performance data collection interval (Collection Interval), the number of seconds that have elapsed since 00:00:00 hours Greenwich Mean Time (GMT) on January 1, 1970, and the value set for `Collection Offset`.

For example, suppose that, in the System Overview (PI) record for Agent for Platform (Windows), the Performance Reporter performance data collection interval is set to 43,200 seconds (12 hours), the Collection Offset value is set to 10 seconds, and the Agent is started at 08:00:00 eastern standard time (EST) (13:00:00 GMT) on August 2nd. In this case, the first data collection begins at 19:00:10 EST on August 3rd. This collection time is derived by adding the 10-second Collection Offset to the time 19:00:00 EST (24:00:00 (or 00:00:00) GMT), which was calculated by multiplying 43,200 seconds (12 hours) by 2 and then adding it to 00:00:00 GMT. The next data collection begins at 07:00:10 EST, which is 12 hours after the first collection, as specified for the data collection interval.

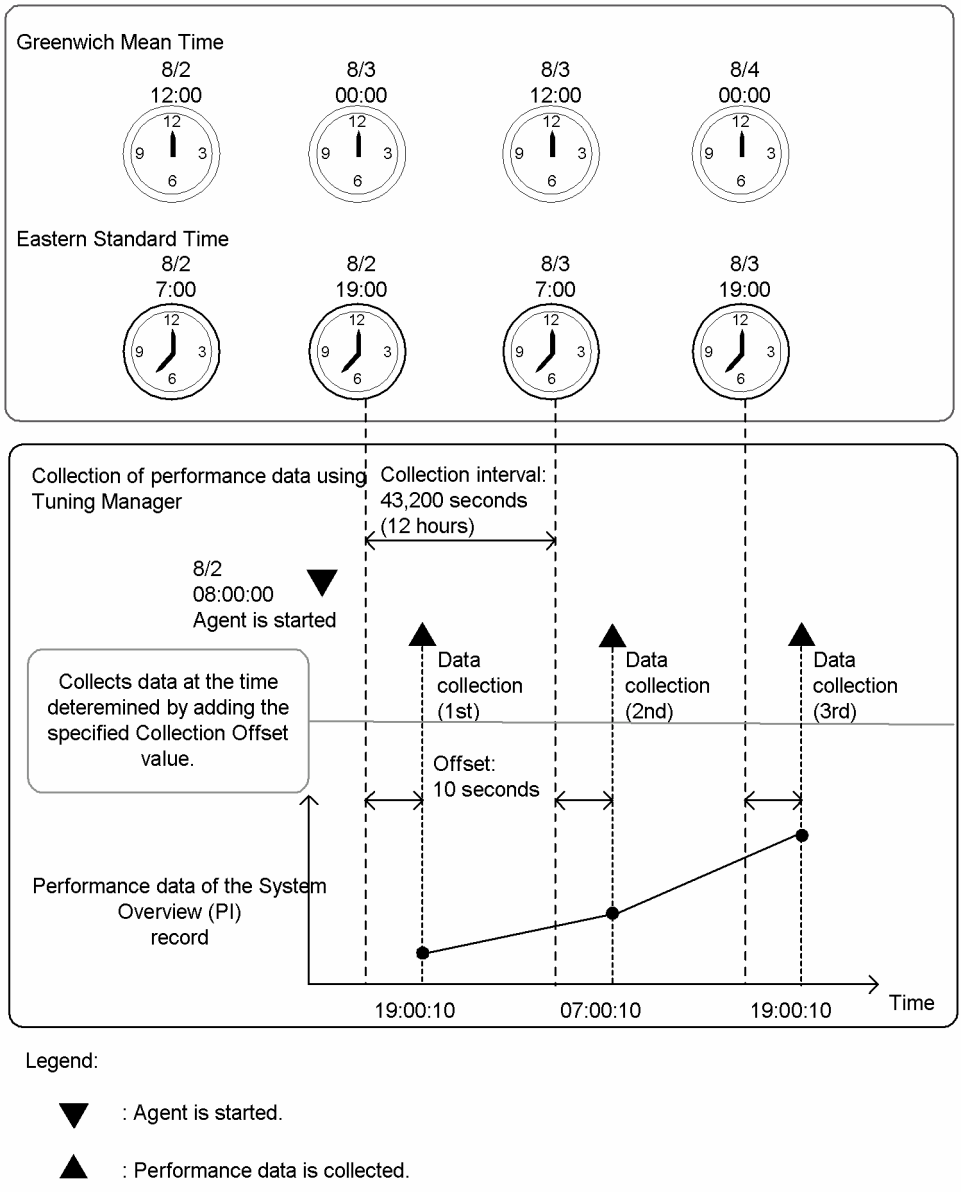


Figure 2.6 Collection Start Time for Performance Data (For Historical Data)

2.3.1.2 Collection Method for Performance Data

By default, the performance data stored in a Store database is only for specific records. To store desired performance data in a Store database, you must use Performance Reporter to specify, for each record, which data is to be stored. For details about how to specify this setting, see the *HiCommand Tuning Manager Command Line Interface Guide*.

The performance data of the PI, PD, and PL record types is stored as follows:

- For PI record type

  Performance data is collected at the collection interval specified in Performance Reporter. However, the first time the Agent performs collection after being started, the performance data is not stored in the Store database. The system stores the performance data starting with the second collection.

- PD and PL record types

  Performance data is collected at the collection interval specified in Performance Reporter. The system stores the performance data, starting with the first collection performed after an Agent is started.

The value to be stored in each field of the records for the PI, PD, and PL types is defined whether to be a delta value. For information about whether a field stores a delta value, see the chapter that describes records (tables listing the fields of each record) in the following manuals:

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

If, in a table of record fields, the delta column for a field shows `Yes`, the field stores the difference in value with respect to the previously measured value. This means that the value is stored in the Store database from the second collection onwards after the Agent starts.

For example, when the delta column for a field storing the I/O count after the system is started shows `Yes`, the field stores the number of I/Os issued over the time between the last collection and the current collection.

The following example indicates how the performance data in a delta field for the PI record type is stored in the Store database:

Example:

In Performance Reporter, the performance data collection interval is set to 3,600 seconds (1 hour) and the Agent is started at 16:30:00 EST (21:30:00 GMT) on August 1st. The first data collection begins at 17:00:00 EST, on August 1st, which is the first multiple of 3,600 seconds (1 hour) to elapse (22:00:00 GMT) after 00:00:00 GMT. The next data collection begins at 18:00:00 EST (23:00:00 GMT), which is 1 hour after the first collection as specified for the data collection interval. Historical data is then created based on the data collected at 17:00:00 EST and 18:00:00 EST, and is then stored in a Store database.

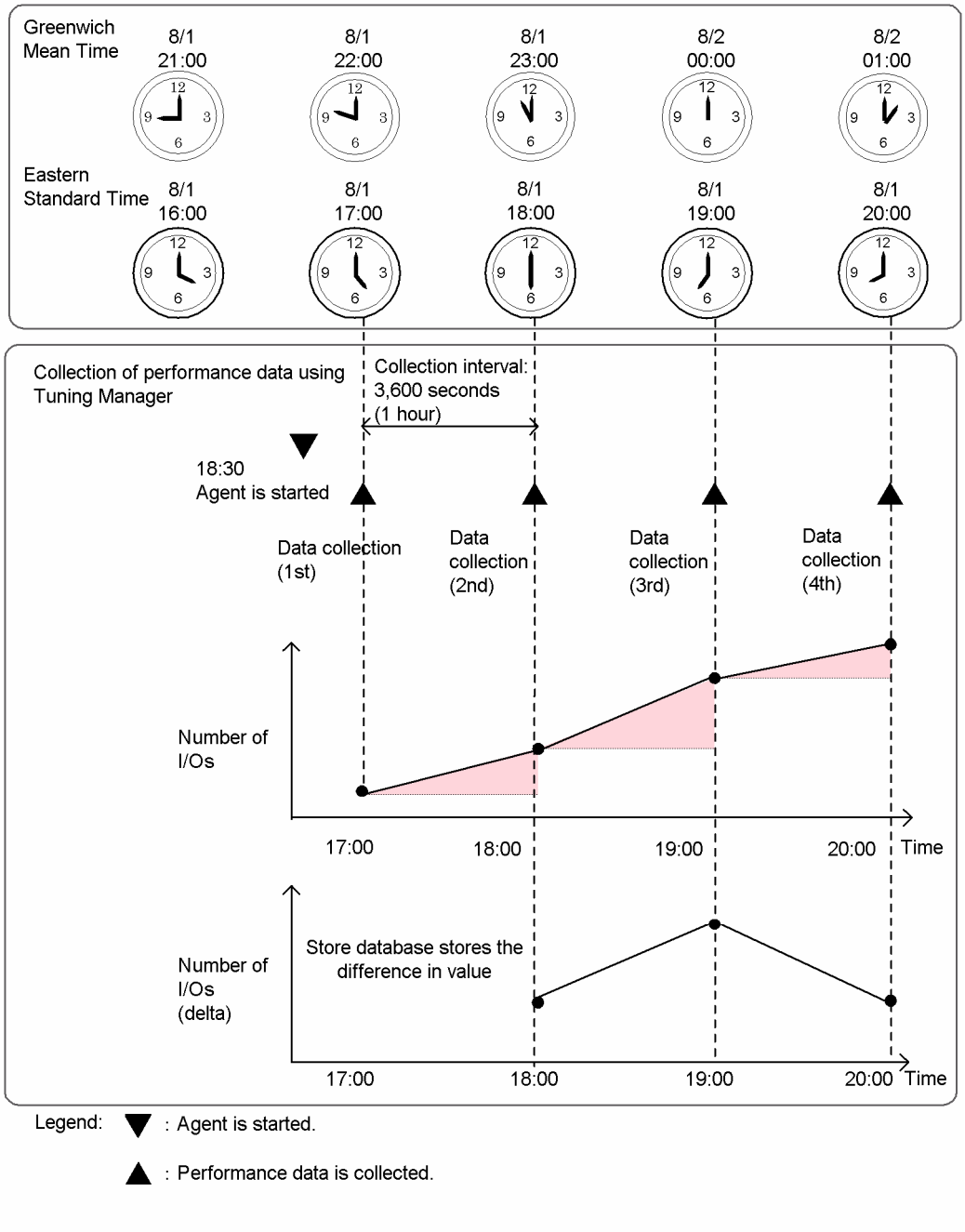


Figure 2.7 Storage Method for Performance Data

2.3.2 Real-time Data

This section describes when to start and how to perform the collection of real-time performance data that is not stored in a Store database.

2.3.2.1 Collection Start Time for Performance Data

When performance data is not stored in a Store database, its collection start time is determined by the time at which a request for displaying real-time reports is sent.

For example, if, in Performance Reporter, you set the real-time report refresh interval for an Agent for Platform (Windows) Content Index Detail (PD_CIND) record to 180 seconds (3 minutes), and display the report at 18:31:00, the first data collection begins at 18:31:00. The next data collection begins at 18:34:00, 3 minutes after the first collection as specified for the report refresh interval.

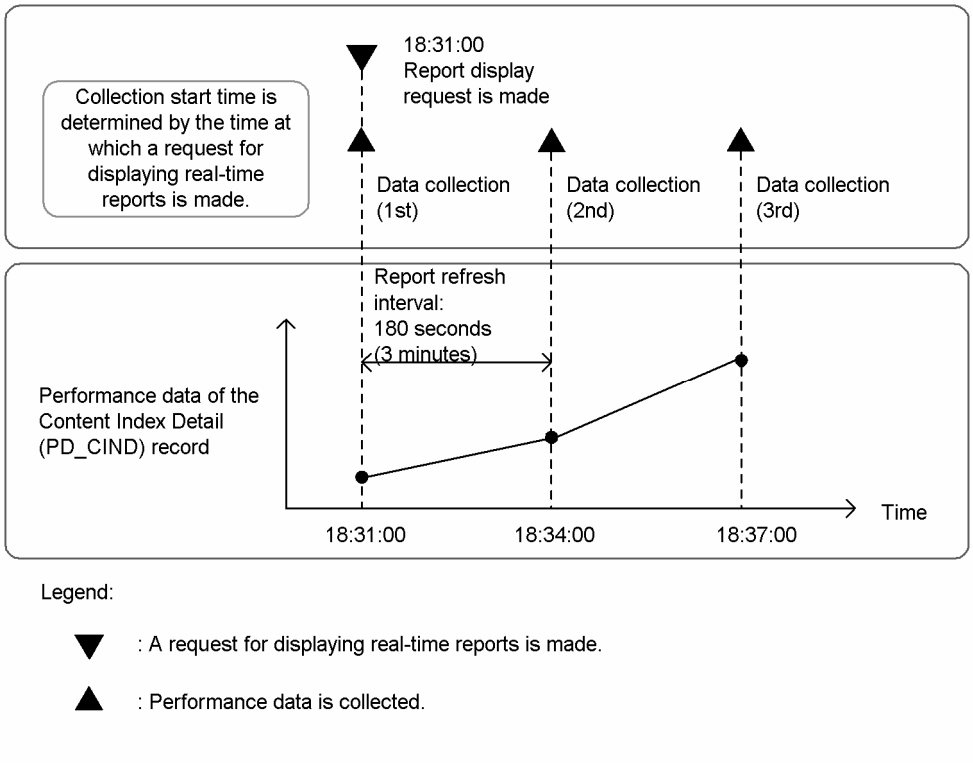


Figure 2.8 Collection Start Time for Performance Data (For Real-time Data)

2.3.2.2 Collection Method for Performance Data

Real-time data is not stored in a Store database. Real-time performance data is collected from the Agent on request.

2.4 Conditions for Performance Data to be Summarized and Stored

For performance data, the conditions for records to be retained and the record summarizing method to be applied when the data retention conditions are fulfilled, differ depending on each record type.

This section explains the conditions you can specify for the data retention conditions for each record type, and how to summarize each record type when all of the data retention conditions are fulfilled.

2.4.1 Records of the PI Record Type

Data is stored in a database that stores records of the PI record type each time performance data collection is performed. For this type of database, performance data is automatically summarized at a fixed period (hourly, daily, weekly, monthly, or yearly). Each field that stores a numeric value is summarized into an average or accumulated value. This summarizing is performed when performance data is collected.

For example, if you specify the record retention period of per-minute data to 1 hour, and 10 bytes of data is collected every minute, data is collected 60 times an hour, yielding 600 bytes of data every hour. If summarized, these 60 data items are averaged and stored as hourly data.

The following figure shows how PI record types are summarized.

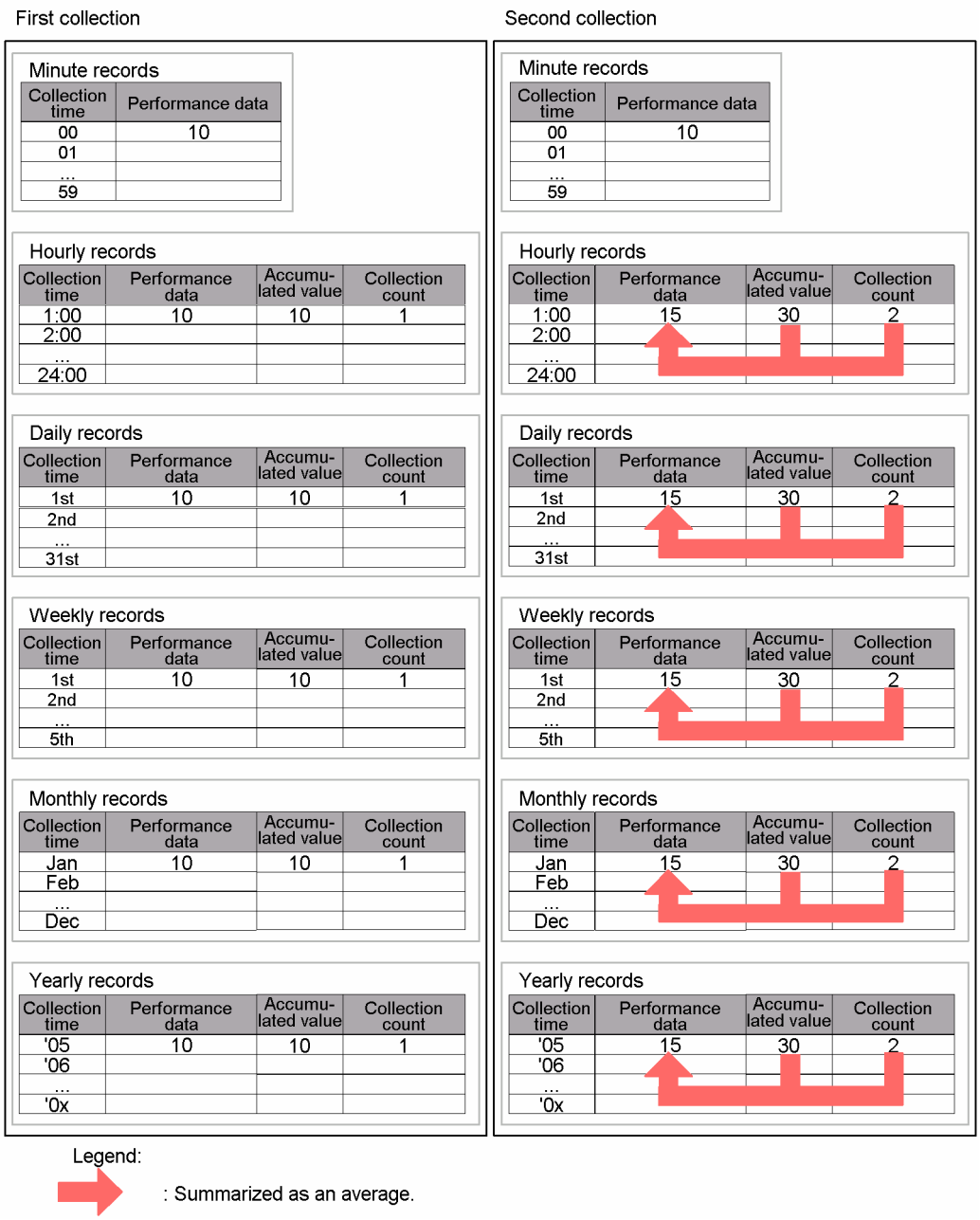


Figure 2.9 PI Method for Summarizing Records

If the specified retention period is reached, the data is totaled and saved in the next summary record. After the data has been totaled, unnecessary records are overwritten. Table 2.2 provides detailed information about each of these time segments.

Table 2.2 PI Record Type Segments

Segment	Description	Default Value of Retention Period
Minute	Up to 60 records an hour, 1,440 records a day, 10,080 records a week, 44,640 records a month, and 527,040 records a year can be saved. Once the system attempts to save a record exceeding these values, or the specified retention period is exceeded, records are overwritten chronologically from old to new.	1 day
Hour	Up to 24 records a day, 168 records a week, 744 records a month, and 8,784 records a year can be saved. Once the system attempts to save a record exceeding these values, or the specified retention period is exceeded, records are overwritten chronologically from old to new.	1 week
Day	Up to 7 records a week, 31 records a month, and 366 records a year can be saved. Once the system attempts to save a record exceeding these values, or the specified retention period is exceeded, records are overwritten chronologically from old to new.	1 year
Week	Up to 5 records a month and 52 records a year can be saved. Once the system attempts to save a record exceeding these values, or the specified retention period is exceeded, records are overwritten chronologically from old to new.	1 year
Month	Up to 12 records a year can be saved. Once the system attempts to save a record exceeding these values, or the specified retention period is exceeded, records are overwritten chronologically from old to new.	1 year
Year	One record a year is saved. Yearly records are accumulated without being updated.	Unlimited

2.4.2 PD-type and PL-type Records

Databases that contain records of the PD or PL type are not eligible for time-based summarizing. For records of the PD or PL type, you can specify the maximum number of records that can be stored.

2.4.3 Collection Offset to Start Collecting Performance Data

If there are many records to be collected or recorded, the collection and recording processing concentrates at certain times, resulting in adverse effects on performance. You can distribute the system workload by specifying an offset value with `Collection Offset` to shift the collection and recording timing for each record.

For example, when two performance data items are to be collected every minute, and `Collection Offset` is set to 0 seconds for one data item and 20 seconds for another, the time Tuning Manager starts collecting each performance data item shifts 20 seconds.

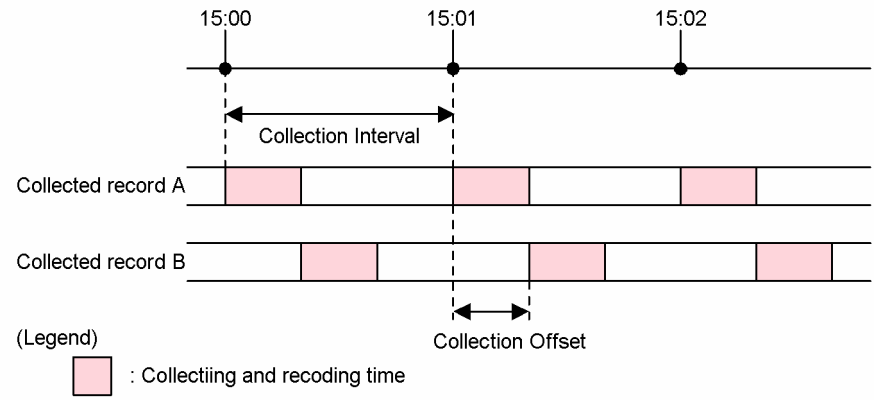


Figure 2.10 Workload Distribution of the Collection and Recording Processing for Performance Data by Specifying an Offset Value

When changing the value of `Collection Offset`, take into account the workload of collection processing.

Chapter 3 Managing the Store Database

This chapter describes how to manage a Store database that stores performance data for Tuning Manager series programs.

This chapter covers the following topics:

- About Store Database Management (see section 3.1)
- Recording Data in the Store Database (see section 3.2)
- Setting Save Conditions of the Store Database (see section 3.3)
- Returning to Default Settings (see section 3.4)
- Exporting the Store Database (see section 3.5)
- Notes on Using the Store Database (see section 3.6)

3.1 About Store Database Management

The Tuning Manager series programs store, in a database, specified performance data that is collected by the agent. This database is called the Store database.

You can use Performance Reporter to set the way you want the performance data collected by the Tuning Manager series programs to be managed. There are two types of Store database management settings:

- Data recording methods
- Save conditions

About Data Recording Methods:

For the methods of recording data in the Store database, you can set whether or not to record performance data in the Store database and the interval at which data will be collected. Accordingly, you can selectively store the performance data you want to save in the Store database.

About Data Saving Conditions:

For the conditions of saving data in the Store database, you can set the maximum number of data entries to be saved and their storage period. Accordingly, you can prevent large increases in the amount of data stored in the Store database.

The following sections describe how to set the Store database to be managed. For an overview of data managed by the Tuning Manager series programs, see Chapter 2.

3.2 Recording Data in the Store Database

You use Performance Reporter to set the methods for recording data in the Store database. Use Performance Reporter to set the following items for each record:

- Whether or not to record collected performance data in the Store database
- The collection interval for performance data
- The offset value for beginning performance data collection
- The conditions used to determine whether or not to record performance data in the database

Depending on the record, it may be possible to set items other than these recording methods. Also, there may be some values that cannot be changed. For details on recording methods that can be set and the default values for each record, see the chapter concerning records (default values and values that can be changed for each record) in the following manuals:

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

To set the recording method for performance data, execute the `jpcasrec output` and `jpcasrec update` commands of Performance Reporter. For details on these commands, see the *HiCommand Tuning Manager Command Line Interface Guide*.

3.3 Setting Save Conditions of the Store Database

In the Tuning Manager series programs, the record storage period and the maximum number of records can be set to prevent large increases in the amount of data stored in the Store database. Data save conditions to be set vary according to the record type. The save conditions that can be set for each record are as follows:

- Record of the PI record type: the record storage period can be set
- Record of the PD record type: the maximum number of records can be set
- Record of the PL record type: the maximum number of records can be set

For details about conditions that can be set as save conditions and the method for summarizing records when the save conditions are met, see section 2.4.

You use Performance Reporter to specify the save conditions for the Store database. To specify the settings, execute the `jpcaspsv output` and `jpcaspsv update` commands of Performance Reporter.

For details on the `jpcaspsv output` and the `jpcaspsv update` commands, see the *HiCommand Tuning Manager Command Line Interface Guide*.

3.4 Returning to Default Settings

If you have changed the settings of the Agent Store service's Store database, which stores performance data, you can return all values to their default values in one step.

To return Agent Store database settings to their default values:

1. Execute the following command to stop the Agent service:

   `jpcstop` ***xxxx***

   Note: ***xxxx*** indicates the service key of each Agent. For details on the service keys of each agent, see Appendix A.

2. Delete the Agent Collector service startup initialization file `jpcagt.ini`, and the Agent Store service startup initialization file `jpcsto.ini`.

   Locations of the jpcagt.ini file

   - Windows: *installation-folder****xxxx**\`agent[\`*instance-name*`]`
   - UNIX: `/opt/jp1pc/`***xxxx***`/agent[/`*instance-name*`]`

   Locations of the jpcsto.ini file

   - Windows: *installation-folder****xxxx**\`store[\`*instance-name*`]`
   - UNIX: `/opt/jp1pc/`***xxxx***`/store[/`*instance-name*`]`

     Note: *instance-name* indicates a directory for operating in the instance environment. For an Agent that monitors an application program that can start a set of multiple services at the same host, the system creates a number of directories equal to the number of instances.

3. Copy the sample file `jpcagt.ini.model` of the Agent Collector service startup initialization file using the name `jpcagt.ini`.

   The `jpcagt.ini.model` file is stored in the same directory as the `jpcagt.ini` file.

4. Copy the sample file `jpcsto.ini.model` of the Agent Store service startup initialization file, using the name `jpcsto.ini`.

   The `jpcsto.ini.model` file is stored in the same directory as the `jpcsto.ini` file.

5. Execute the following command to start the Agent service:

   `jpcstart` ***xxxx***

For details on the `jpcstop` command and the `jpcstart` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

3.5 Exporting the Store Database

In the Tuning Manager series programs, you can export data from the Store database to text files. By modifying exported data using other application programs, you can create more complex reports and perform data analyses that correspond to user goals.

To export data, execute the `jpcctrl dump` command and perform the following steps. For details on this command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

To export the Store database in which performance data is stored:

1. Log in to the host on which the Agent is installed.
2. Execute the `jpcctrl list` command, and confirm that the following services are running:
   - Name Server
   - Master Manager
   - Master Store
3. Execute the `jpcctrl dump` command.

Example:

Execute the following command to export the performance data for Oct. 17th, 2005, 2:00AM - 2:59PM (GMT) to the `pcsr.out` file. The performance data is stored in the Processor Overview (PI_PCSR) record of `host02`, an Agent for Platform (Windows) host.

```
jpcctrl dump TS* host=host02 2005/10/17 02:00 2005/10/17 14:59
pcsr.out PI PCSR
```

If the command execution finishes successfully, the export file for the performance data will be output to the following files:

- Windows:

  *installation-folder****xxxx**\store[*instance-name*]\dump\pcsr.out

- UNIX:

  /opt/jp1pc/***xxxx***/store[/*instance-name*]/dump/pcsr.out

  Notes:

  - *xxxx* indicates the service key of each Agent.
  - *instance-name* indicates a directory for operating in the instance environment. For an Agent that monitors an application program that can start a set of multiple services at the same host, the system creates a number of directories equal to the number of instances.

Do not stop the Master Manager service during export processing. If the Master Manager service is stopped, the following message is displayed:

```
KAVE05267-E The dump command was interrupted because Master
Manager stopped. (dbid=database-ID)
```

Even when this message is displayed, export processing continues uninterrupted. The export processing is executed for the Master Store or Agent Store service that was specified in the argument of the `jpcctrl dump` command. In this case, the monitoring of export processing is terminated only for the `jpcctrl dump` command.

While export processing is underway for the Master Store or Agent Store service, if the Master Manager service is restarted and the `jpcctrl dump` command is re-executed, the following message may be displayed:

```
KAVE05232-E Because backup or export was being processed, the
request was refused.
```

If this message is displayed, wait a while after the export processing is completed, and then re-execute the command.

To determine whether the export processing is completed after canceling the command, check the `DUMP.LOG` file that has been output under the Store database's export destination directory. The message `Ended normally.` indicates that the processing was completed successfully.

3.6 Notes on Using the Store Database

This section explains precautions relating to operation of the Store database when using the Tuning Manager series programs.

3.6.1 Restrictions on the Size of the Store Database

The maximum file size for the Store database when using the Tuning Manager series programs is 2 GB. You cannot run operations that exceed the file size limit set by the `ulimit` command of UNIX or the limit on the file system.

When the file size of the Store database reaches the limit, the Store service stops. If this happens, the following error message is output to the system log (the Windows event log in Windows, or syslog in UNIX) and the common message log:

```
KAVE00182-E  The record data could not be stored because the Store database reached
the writing limit. (record=record-ID, file=file-name)
```

Therefore, when setting the save conditions for the Store database, consider the restriction on the file size of the Store database. For details about setting the save conditions for the Store database, see section 3.3.

3.6.2 Operation Following Abnormal Termination of the Store Service

Note the following when the Store service has terminated abnormally:

- If the Store service terminates abnormally while data is being written to the Store database, at the next startup of the Store service, the system checks the integrity of the database before starting the Store service. Invalid data found during the integrity check cannot be guaranteed.
- If the Store service cannot terminate normally due to a problem such as disconnection of the power, the indexes of the Store database must be rebuilt at restart. It may therefore take a long time for the Store service to start.

3.6.3 Checking the Size of the Store Database and Reorganizing the Database

The Store database consists of a *data file*, which stores data entities, and an *index file*, which manages data indexes in order to increase access speed. When data file records are deleted, the resulting empty area becomes null areas, and the file size is not reduced automatically. Although null areas in the data file are reused, reuse efficiency deteriorates when the number of instances in the performance data to be stored changes each time data is collected. As a result, the size of the Store database may eventually exceed the estimated amount of disk space. It is recommended that you regularly check the size of the Store database and reorganize it to reduce null areas whenever the file size exceeds 90% of the estimated disk space.

The following describes how to check the size of the Store database and how to reorganize it.

3.6.3.1 Checking the Size of the Store Database

At the location where the Store database is stored, check the sizes of all files whose extension is `.DB` or `.IDX`, and total the sizes. If this total size exceeds 90% of the estimated disk space, reorganize the Store database as described below.

3.6.3.2 Reorganizing the Store Database

To reorganize the Store database:

1. Start the Tuning Manager series program service that manages the Store database to be reorganized.

   If the service for the Agent or Collection Manager that manages the Store database to be reorganized is stopped, use the `jpcstart` command to start it.

2. Use the `jpcctrl backup` command to back up the Store database.

   Execute the `jpcctrl backup` command to back up the Store database that is to be reorganized. When the `jpcctrl backup` command is executed, the data in the file (but not the null areas) are extracted and saved.

   Note: To execute the `jpcctrl backup` command, at least about twice as much free space as the total size you obtained in section 3.6.3.1 is required on the disk to which the backup file will be output. Before executing the command, you must make sure that there is enough free space.

3. Stop the Tuning Manager series program service that manages the Store database to be reorganized.

   Use the `jpcstop` command to stop service for the Agent or Collection Manager that manages the Store database to be reorganized.

4. Use the `jpcresto` command to restore the Store database.

Execute the `jpcresto` command to restore the Store database you backed up in step 2.

5. Start the Tuning Manager series program service.

   If necessary, use the `jpcstart` command to start the service you stopped in step 3.

Chapter 4 Monitoring Operations Using Alarms

With Tuning Manager series programs, you can set threshold values for performance data that is being collected and then be notified by means of an alarm if an item in the performance data exceeds a specified threshold value.

This chapter explains how to create alarms and how to use alarms to report problems that have occurred.

- Overview of Alarms (see section 4.1)
- Procedures for setting and using alarms (see section 4.2)
- Preparations before setting alarms (see section 4.3)
- Syntax of an alarm definition file (see section 4.4)
- Setting alarms (see section 4.5)
- Using alarms (see section 4.6)
- Notes on alarms (see section 4.7)

4.1 Overview of Alarms

You can configure Tuning Manager to notify the user whenever performance data being monitored by a monitoring agent reaches a preset threshold.

The entity that defines the system action to be performed when a data item reaches a set threshold is called an **alarm**, and all the alarms defined as a single set constitute what is called an **alarm table**.

When a data item reaches a threshold, the monitoring agent reports this fact by issuing an **alarm event**. The operation that the Tuning Manager series program performs on reception of an alarm event is called an **action**. The following are actions that can be performed by Tuning Manager series programs:

- Send an email notification to the system administrator.
- Execute a recovery program or other command.
- Send an SNMP trap.

Associating an alarm table with a monitoring agent enables Tuning Manager to detect when a threshold is exceeded. Association of an alarm table with a monitoring agent is called **binding**. By binding an alarm table, you can associate its set of alarms to one or more monitoring agents. However, each monitoring agent can have only one set of alarms (one alarm table) bound to it.

4.2 Procedures for Setting and Using Alarms

This section describes how to set and use alarms and the procedures for creating and using alarms.

References:

- For details about the commands for setting and using alarms that are provided by the services of Tuning Manager series programs, see the *HiCommand Tuning Manager Command Line Interface Guide*.
- For details about the alarms for the solution set, see the following manuals:
  - *HiCommand Tuning Manager Hardware Reports Reference*
  - *HiCommand Tuning Manager Operating System Reports Reference*
  - *HiCommand Tuning Manager Application Reports Reference*

4.2.1 How to Set and Use Alarms

Commands are provided for setting and using alarms.

To set alarms, the following methods are available:

- Defining a new alarm table and alarms

  This method creates an alarm table appropriate for the system environment and then defines alarms. You can add more alarms to the alarm table later.

- Using an existing alarm table or alarms

  The following methods are available:

  - Using the solution set

    The solution set is a group of preset alarms defining necessary information that is provided by each Agent. If the solution set is used, the alarms provided in the solution set become effective when the Agent starts.

  - Customizing the solution set

    This method copies the solution set and then customizes the copy as appropriate for the system environment.

  - Using an existing alarm table or alarms

    This method copies an existing alarm table or alarms and then customizes the copy.

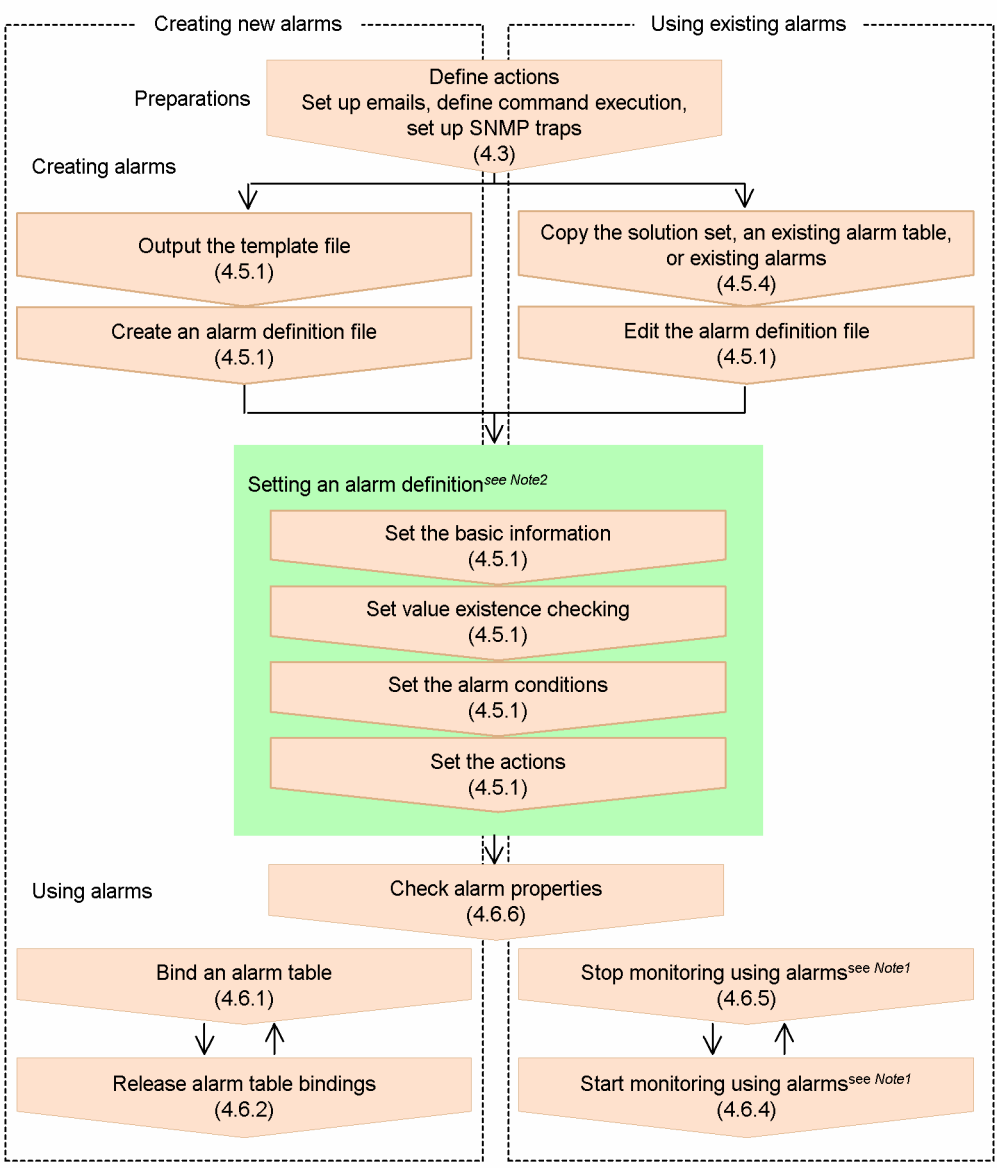
To use alarms, you must associate (bind) an alarm table defined by one of the above methods to the applicable monitoring agents.

Reference:

An alarm definition file created using the `jpcalarm` command can define a maximum of 50 alarms.

4.2.2 Procedures for Setting and Using Alarms

The following figure shows the procedures for setting and using alarms:



Legend: ( ): Subsection to be referenced

Note1: Perform this step as necessary.

Note2: If you are using existing alarms, edit the alarm definitions as appropriate.

Figure 4.1 Procedures for Setting and Using Alarms

4.3 Preparations Before Setting Alarms

This section describes the preparations that must be completed before you can set alarms.

4.3.1 Setting Email Senders

If you want an email to be sent when an alarm event occurs at an Agent, use the `jpcahprp update` command to set the email senders. In the parameter file that is specified in the argument of the `jpcahprp update` command, under `capabilities` you must set `email` to `Yes`.

4.3.2 Setting the Host Used to Execute a Command

If you want a command to execute automatically when an alarm event occurs at an Agent, use the `jpcahprp update` command to change the `Action Handler` property at the host where the command is to be executed. In the parameter file that is specified in the argument of the `jpcahprp update` command, under `capabilities` you must set script to `Yes`.

4.3.3 Settings for Sending an SNMP Trap

If you want to send an SNMP trap when an alarm event occurs at an Agent, use the `jpctgprp create` command to change the `Trap Generator` property. In the parameter file that is specified in the argument of the `jpctgprp create` command, under `trap-generator-definition` you must set the name of the target host in `snmp-host`.

To delete an SNMP trap destination that has been set, you use the `jpctgprp delete` command to remove the SNMP host name from the Trap Generator definition information.

4.4 Syntax of an Alarm Definition File

This section describes the syntax of an alarm definition file that is used for setting alarms.

4.4.1 Terms Used to Explain the Syntax of an Alarm Definition File

The table below lists and describes the terms used to explain the syntax of an alarm definition file and the specification rules.

Table 4.1 Terms Used to Explain Syntax of Alarm Definition File and Definition and Specification Rules

Term	Definition and Specification Rule
Section	Indicates the major settings. A section name must be enclosed in single-byte square brackets, such as `[Alarm Data]`. No characters other than the section name can appear between the opening square bracket and the closing square bracket. The line specifying a section must contain no other characters except for spaces. Note that any spaces preceding the opening square bracket and following the closing square bracket are ignored. ***Example:*** `[Alarm Data]`
Subsection	Indicates an intermediate setting. A subsection name must be enclosed in double single-byte square brackets, such as `[[General]]`. There must not be any spaces between the two opening square brackets or between the two closing square brackets. No characters other than the subsection name can appear between the set of opening square brackets and the set of closing square brackets. The line specifying a subsection must contain no other characters except for spaces. Note that any spaces preceding the pair of opening square brackets and following the pair of closing square brackets are ignored. ***Example:*** `[[General]]`
Subsubsection	Indicates a minor setting. A subsubsection name must be enclosed in triple single-byte square brackets, such as `[[[Message Text]]]`. There must not be any spaces between opening square brackets or between closing square brackets. No characters other than the subsubsection name can appear between the set of opening square brackets and the set of closing square brackets. The line specifying a subsubsection must contain no other characters except for spaces. Note that any spaces preceding the set of opening square brackets and following the set of closing square brackets are ignored. ***Example:*** `[[[Message Text]]]`

Term	Definition and Specification Rule
Label	Indicates a name and a value that are set. A label must be specified on one line, as shown below: *label-name*=*label-value* *label-name* is a name for the value and *label-value* is the actual value. On a line on which a label is specified, all spaces preceding *label-name* and following *label-value* are ignored. ***Example:*** `Product=D4.0`
Comment	Indicates a comment. To specify a comment, enter a hash mark (#) immediately before the comment. All information beginning with # up to a linefeed is treated as a comment. A comment can begin anywhere in a line. To specify # as a part of a character string, enter a single-byte backslash (\) immediately before #, such as \#. ***Example:*** `Product=D4.0 # Agent for RAID`

4.4.2 Notes about Creating an Alarm Definition File

You should note the following about creating an alarm definition file:

- Section and label names are case sensitive. Spaces in a section name or label name are also significant.
- Shift JIS or EUC codes can be used for the definitions, such as for the alarm table name and alarm names. You specify in the `Alarm Definition File Code` label of the alarm definition file the character codes that you will be using.
- If you specify a record or field name in an alarm event occurrence condition, make sure that the specified value has been defined at each Agent. For details about the record and field names, see the chapter in the following manuals that describes records:
  - *HiCommand Tuning Manager Hardware Reports Reference*
  - *HiCommand Tuning Manager Operating System Reports Reference*
  - *HiCommand Tuning Manager Application Reports Reference*
- You can define a maximum of 50 alarms in an alarm definition file. If more than 50 alarms are defined in a single alarm definition file, an error occurs when the alarm definition file is imported or checked.
- If you intend to specify a host name or Action Handler service ID in an alarm definition file, you should first use the `jpcctrl list` command to check the service ID or host name. For details about the `jpcctrl list` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

- A maximum of 50 alarms can be defined in an alarm table. If an existing alarm definition file already contains alarm definitions, you can add new alarm definitions only until the total number of alarm definitions reaches 50. If the number of alarms you have defined for a single alarm definition file is 50 or less but the total number of alarms including existing alarm definitions exceeds 50, an error occurs when the alarm definition file is imported.

4.4.3 Components of an Alarm Definition File

An alarm definition file consists of a header section that indicates the version of the alarm definition file and the character codes that are used and a section for each alarm that is defined.

The following figure shows the components of an alarm definition file.

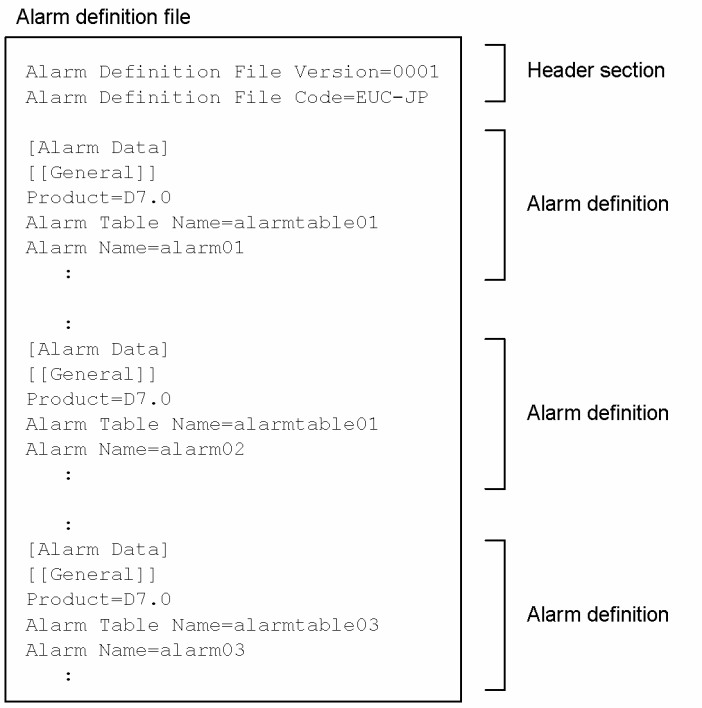


Figure 4.2 Components of an Alarm Definition File

The following describes each component.

4.4.3.1 Alarm Definition File Version Label

The `Alarm Definition File Version` label specifies the syntax version of the alarm definition file. The following table lists and describes the value of the `Alarm Definition File Version` label.

Table 4.2 Value of the Alarm Definition File Version Label

Label Name	Description	Specification	Value
`Alarm Definition File Version`	Syntax version of the alarm definition file	Required	For Tuning Manager 5.5, the value is `0001`.

Specify the `Alarm Definition File Version` label only once at the beginning of the file. The following shows a specification example:

```
Alarm Definition File Version=0001
```

4.4.3.2 Alarm Definition File Code Label

The `Alarm Definition File Code` label specifies the character codes used in the alarm definition file. The following table lists and describes the values specifiable for the `Alarm Definition File Code` label.

Table 4.3 Values of the Alarm Definition File Code Label

Label Name	Description	Specification	Value

`Alarm Definition File Code`	Character codes used in the alarm definition file	Required	`Shift_JIS`
			The alarm definition file is specified in double-byte characters (Shift JIS codes) or in single-byte characters (7-bit ASCII characters).
			`EUC-JP`
			The alarm definition file is specified in double-byte characters (EUC codes) or in single-byte characters (7-bit ASCII characters).
			`C`
			The alarm definition file is specified in characters other than the above.

If the language environment used to execute the `jpcalarm` command does not match the character codes used in the alarm definition file, code conversion is performed on the alarm definition file according to the specified `Alarm Definition File Code` label. The following table shows the correspondence between the language environment during execution of the `jpcalarm` command and the character codes used in the alarm definition file:

Table 4.4 Correspondence Between Language Environment During Execution of the jpcalarm Command and the Character Codes Used in the Alarm Definition File

Specification in Alarm Definition File	Language Environment During Execution of jpcalarm Command		
	Shift JIS	EUC	C
Shift_JIS	--	Shift JIS to EUC	--
EUC-JP	EUC to Shift JIS	--	--
C	--	--	--

Legend:

--: No conversion occurs.

Specify the `Alarm Definition File Code` label only once in the alarm definition file, immediately following the `Alarm Definition File Version` label. The following shows a specification example:

```
Alarm Definition File Code=C
```

4.4.3.3 Alarm Data Section

An `Alarm Data` section specifies an alarm definition. You must create one `Alarm Data` section for each alarm definition.

Specify the `Alarm Data` sections immediately following the `Alarm Definition File Code` label. You can specify a maximum of 50 `Alarm Data` sections in a single alarm definition file.

An `Alarm Data` section consists of multiple subsections. The table below lists and describes the subsections that can be specified in an `Alarm Data` section. You must specify the subsections in the order they are listed in this table.

Table 4.5 Subsections Specifiable in an Alarm Data Section

Subsection Name	Description	Specification
General	Basic information, such as the alarm name	Required
Advanced Setting	Extended information, such as the alarm type and monitoring time	Optional
Check Value Exist	Value to be monitored by a **value existence checking alarm**	Optional see ***Note1***
Alarm Condition Expressions	Condition expressions for alarms other than a **value existence checking alarm**	Optional see ***Note2***
Actions	Actions to be executed by the alarm	Optional
Action Definition E-mail	Settings for sending email when the alarm occurs	Optional see ***Note3***

Subsection Name	Description	Specification
`Action Definition Command`	Settings for executing a command when the alarm occurs	Optional see ***Note4***
`Action Definition JP1 Event`	Uneditable section	Optional

Note1: This subsection cannot be omitted when the `Check Value Exist` label of the `General` subsection specifies that this alarm is a **value existence checking alarm**.

Note2: This subsection cannot be omitted when the `Check Value Exist` label of the `General` subsection specifies that this alarm is not a **value existence checking alarm**.

Note3: This subsection cannot be omitted when the `E-mail` label of the `Actions` subsection specifies that email is to be sent when this alarm occurs.

Note4: This subsection cannot be omitted when the `Command` label of the `Actions` subsection specifies that a command is to be executed when this alarm occurs.

Each subsection consists of subsubsections and labels. The following describes the subsubsections and labels that are specified in each subsection.

General Subsection

The `General` subsection specifies basic information about the alarm definition, such as the type and version of the data model that is used in the alarm definition, the alarm table name, and the alarm name.

The table below lists and describes the labels that can be specified in the `General` subsection and their values. You must specify the labels in the order they are listed in this table.

Table 4.6 Labels Specifiable in the General Subsection and Their Values

Label Name	Description	Specification	Value
`Product`	Type of Agent for which the alarm is defined (product) and the version of the data model	Required	Specifies the product ID of the Agent and the version of the data model. For details about the product ID, see the list of IDs provided in the Appendix. For details about the version of the data model, see the version compatibility listing in the Appendix. For Agent for RAID 5.5, specify `D7.0`.
`Alarm Table Name`	Alarm table name	Required	Up to 64 bytes of double-byte and single-byte characters. If the name contains any single-byte space, enclose the entire name in double quotation marks ("). Any single-byte space following an equal sign (`=`) or preceding a linefeed code is ignored. The specified name cannot begin with `PFM` (not case sensitive).
`Alarm Name`	Alarm name	Required	Up to 20 bytes of double-byte and single-byte characters. If the name contains any single-byte space, enclose the entire name in double quotation marks ("). Any single-byte space following an equal sign (`=`) or preceding a linefeed code is ignored.
`Message Text`	Message text to be sent to the SNMP trap when the alarm occurs	Optional see ***Note1***	0 to 255 bytes of double-byte and single-byte characters. This can be a variable. For a list of permitted variables and their description, see Table 4.7. Any single-byte space following = or preceding a linefeed code is ignored.
`Check Value Exist`	Whether this is an alarm that checks whether the value exists	Optional see ***Note2***	`Y` The alarm checks whether the value exists `N` or omitted The alarm is a normal alarm.

Note1: You can omit the entire label or only the value.

Note2: If you are omitting this specification, you must omit the entire label. You cannot omit only the value.

Table 4.7 Descriptions of Variables Usable in Message Text Subsubsections

Variable	Description
%AIS	Alarm name
%ANS	Name of the agent that binds the alarm table in which this alarm is defined
%ATS	Name of the alarm table in which this alarm is defined
%CVS[n][.p]	Measurement value resulting in alarm notification (satisfying the conditional expression). ▪ *n* If multiple conditional expressions are specified in the `Alarm Condition Expression` subsection, this variable specifies the field position, expressed as 1 or a greater value, where the first field is 1. If 0 is specified or the specified value is greater than the number of conditional expressions, the measurement value in the first field is displayed. ▪ *p* Specifies the number of decimal places to be displayed (value is rounded).
%HNS	Host name of the agent that binds the alarm table defining this alarm
%PTS	Product name set in `Product`
%SCS	Alarm status resulting in message output
%SCT	System time of the host where the agent for which alarm evaluation was performed is installed
%MTS	Value defined in the `Message Text` label of the `General` subsection

The example below defines the following alarm:

- Data model to be used: Data model for Agent for RAID 5.5 (version of the data model is 7.0)
- Alarm table name: `alarmtable01`
- Alarm name: `alarm01`
- Message text to be sent to the SNMP trap when an alarm occurs: `CPU is at %CVS% utilization`
- Whether to set this alarm as a **value existence checking alarm**: No (this is a normal alarm)

The following shows the `General` subsection for this example:

```
   :
[[General]]
Product=D7.0
Alarm Table Name=alarmtable01
Alarm Name=alarm01
Message Text="CPU is at %CVS% utilization"
Check Value Exist=N
   :
```

Advanced Setting Subsection

The `Advanced Setting` subsection specifies extended information about the alarm definition, such as the type of alarm and the alarm monitoring time.

The table below lists and describes the labels that can be specified in the `Advanced Setting` subsection and their values. You must specify the labels in the order they are listed in this table.

Table 4.8 Labels Specifiable in the Advanced Setting Subsection and Their Values

Label Name	Description	Specification	Value
`Active Alarm`	Specifies the alarm status (enabled or disabled).	Optional see ***Note1***	`Y` or omitted The alarm is enabled. `N` The alarm is disabled.
`Regular Alarm`	Specifies whether the alarm is to report regularly (**Notify regularly**).	Optional see ***Note1***	`Y` The alarm reports regularly. `N` or omitted The alarm does not report regularly.
`Evaluate All Data`	Specifies whether the alarm is to evaluate all data (**Evaluate all data**).	Optional see ***Note1***	`Y` The alarm evaluates all data. `N` or omitted The alarm does not evaluate all data.
`Monitoring Regularly`	Specifies whether the alarm is to monitor data regularly (**Monitor regularly**).	Optional see ***Note1***	`Y` or omitted The alarm monitors data regularly. `N` The alarm does not monitor data regularly.
`Monitoring Time`	Monitoring time range when N (data not monitored regularly) is specified in the `Monitoring Regularly` label	Optional see ***Note2***	Specifies the monitoring start and end times connected by a single-byte hyphen (-). Specify the time in the format *HH*`:`*MM*. *HH* Start or end time (hour) in the range from 00 to 23. *MM* Start or end time (minute) in the range from 00 to 59. The specified value must be the local time. To monitor from 7 a.m. to 9 p.m., specify `07:00-21:00`. If `Y` is specified in the `Monitoring Regularly` label or the `Monitoring Regularly` label is omitted, this label is ignored, if specified.
`Damping`	Whether the alarm is to be reported when damping conditions are satisfied	Optional see ***Note1***	`Y` Notify the alarm when the damping conditions are satisfied. `N` or omitted Do not take into account the damping conditions.

Label Name	Description	Specification	Value
Damping Count	If Y is specified in the Damping label to notify the alarm when damping conditions are satisfied, this label specifies the maximum alarm evaluation count and the maximum number of times the threshold can be exceeded before the alarm is notified	Optional see ***Note3***	Specifies the maximum number of times the threshold can be exceeded and the maximum alarm evaluation count, connected by a single-byte forward slash (/). Each value must be an integer in the range 1 to 32767. If the specified maximum alarm evaluation count is less than the maximum number of times the threshold can be exceeded, the system assumes that the maximum alarm evaluation count is the same as the maximum number of times the threshold can be exceeded. If N is specified in the Damping label or the Damping label is omitted, this label is ignored, if specified.

Note1: If you are omitting this specification, you must omit the entire label. You cannot omit only the value.

Note2: If you are omitting this specification, you must omit the entire label. You cannot omit only the value. If N is specified in the Monitoring Regularly label, this label cannot be omitted.

Note3: If you are omitting this specification, you must omit the entire label. You cannot omit only the value. If Y is specified in the Damping label, this label cannot be omitted.

The example below defines the following alarm:

- Enable the alarm.
- Report regularly.
- Do not evaluate all data.
- Specify the range of the alarm's monitoring time by specifying a time from 6 a.m. to 6 p.m. (local time).
- Report the alarm when the threshold is exceeded twice by the end of the third alarm evaluation.

The following shows the Advanced Setting subsection for this example:

```
   :
[[Advanced Setting]]
Active Alarm=Y
Regular Alarm=Y
Evaluate All Data=N
Monitoring Regularly=N
Monitoring Time=06:00-18:00
Damping=Y
Damping Count=2/3
   :
```

Check Value Exist Subsection

The `Check Value Exist` subsection must be specified when the `General` subsection specifies that the alarm being defined is an alarm that checks whether a value exists. The `Check Value Exist` subsection specifies the value that is to be monitored and the records and fields that contain that value.

The table below lists and describes the labels that can be specified in the `Check Value Exist` subsection and their values. You must specify the labels in the order they are listed in this table.

Table 4.9 Labels Specifiable in the Check Value Exist Subsection and Their Values

Label Name	Description	Specification	Value
`Record`	Record name (record ID) to be monitored	Required	Up to 7 bytes of single-byte characters.
`Field`	Field name (Manager name) in the record that is to be monitored	Required	Up to 50 bytes of single-byte characters.
`Value`	Value to be monitored for its existence	Required	Integer, decimal value, or up to 127 bytes of double-byte and single-byte characters. If the value contains any single-byte space, enclose the entire value in double quotation marks ("). The permitted value depends on the field value. When single-byte characters are specified, control characters and the special characters `( ) [ ] < >` and `=` cannot be specified.

The following shows a specification example of the `Check Value Exist` subsection when you want to monitor whether or not a CLPR whose name is DBSERVER is defined in Agent for RAID:

```
   :
[[Check Value Exist]]
Record=PD_CLPC
Field=CLPR_NAME
Value=DBSERVER
   :
```

Alarm Condition Expressions Subsection

The `Alarm Condition Expression`s subsection must be specified when the `General` subsection specifies that the alarm being defined is not an alarm that checks whether a value exists. The `Alarm Condition Expressions` subsection specifies conditions to be used to determine whether the alarm is to be activated.

The table below lists and describes the label that can be specified in the `Alarm Condition Expressions` subsection and its values.

Table 4.10 Label Specifiable in the Alarm Condition Expressions Subsection and Its Value

Label Name	Description	Specification	Value
Condition	Conditional expressions used for alarm monitoring	Required	Specifies the conditional expressions (see ***Note***) each in the following format: *field condition abnormal-value,warning-value* The following describes each value: *field* Specifies the names of the record and field to be monitored, expressed as a character string consisting of the record name and the manager name of the field connected by an underscore (_). The permitted value is up to 50 bytes of single-byte characters. *condition* Specifies the operator used to evaluate the conditional expression. The following operators can be specified: =: The value of the field is equal to the specified value. <: The value of the field is less than the specified value. <=: The value of the field is equal to or less than the specified value. >: The value of the field is greater than the specified value. >=: The value of the field is equal to or greater than the specified value. <>: The value of the field is not equal to the specified value. *abnormal-value,warning-value* Specifies the thresholds to be used as the criteria for the abnormal and warning alarms. The permitted values depend on the field value. Specify *abnormal-value* and *warning-value* connected by a comma (,). For *abnormal-value* and *warning-value*, specify an integer, decimal value, or up to 749 bytes of double-byte and single-byte characters. When single-byte characters are specified, control characters and the special characters ( ) [ ] < > and = cannot be specified. To specify a character string, enclose the entire character string in single-byte double quotation marks ("). To specify multiple, conditional expressions, use AND to connect the above expressions. A maximum of 5 conditional expressions can be specified together. If the alarm is to be exported with different conditions for abnormal value and warning value (such as abnormal value <50, warning value <= 60), the condition for the warning value (if there is no warning value, then the condition for the abnormal value) takes effect during the export operation. When an alarm with no abnormal value or warning value is exported, <*abnormal-value*> or <*warning-value*> becomes blank during the export operation.

Note: The length of the conditional expressions can be obtained from the formula shown below, where the number of conditional expressions connected by AND is *n* ($1 \leq n \leq 5$), $1 \leq i \leq n$. If the value obtained from the following formula exceeds 749 (bytes), an error occurs:

Length of the conditional expression for an abnormal value = $\sum_{i=1}^{n} (a_i+b_i+c_i+4)+(n-1)\times 5$

Length of the conditional expression for a warning value = $\sum_{i=1}^{n} (a_i+b_i+d_i+4)+(n-1)\times 5$

Legend:

ai: Length of <*field-to-be-monitored*> *i* (bytes)

bi: Length of <*condition*> *i* (bytes)

ci: Length of <*abnormal-value*> *i* (bytes) (if a character string is specified, the length without "")

di: Length of <*warning-value*> *i* (bytes) (if a character string is specified, the length without "")

The following shows a specification example of the `Alarm Condition Expressions` subsection when reporting an abnormal alarm if the value of the Read Hit % (READ_HIT_RATE) is below 70.0, and the value of warning alarm reporting is below 85.0 while the value of the Read I/O Count (READ_IO_COUNT) field is 100 or more. (that is, the field of the Logical Device Summary (PI_LDS) record in Agent for RAID):

```
   :
[[Alarm Condition Expressions]]
Condition= PI_LDS_READ_IO_COUNT>=100,100 AND PI_LDS_READ_HIT_RATE<70.0,85.0
   :
```

Actions Subsection

The `Actions` subsection specifies the actions to be executed when an alarm event occurs.

The table below lists and describes the labels that can be specified in the `Actions` subsection and their values. You must specify the labels in the order they are listed in this table.

Table 4.11 Labels Specifiable in the Actions Subsection and Their Values

Label Name	Description	Specification	Value
`Report`	This is an uneditable label.	Optional see ***Note1***	-

`E-mail`	Alarm status that triggers email to be sent (applicable if email is to be sent when an alarm event occurs)	Optional
		see ***Note2***
			`Abnormal`
			Sends email when the alarm status becomes abnormal.
			`Warning`
			Sends email when the alarm status becomes warning.
			`Normal`
			Sends email when the alarm status becomes normal.
			If multiple alarm statuses trigger email transmission, specify the character strings indicating the applicable statuses by connecting them by a single-byte comma (`,`). The character strings indicating statuses can be specified in any order. However, an error occurs if the same character string is specified more than once.
			If `"Y"` is specified as the value of the `Check Value Exist` label in the `General` subsection, `Warning` cannot be specified. If `"Y"` is specified as the value of the `Regular Alarm` label in the `Advanced Setting` subsection, `Normal` cannot be specified.

`Command`	Alarm status that triggers the execution of a command (applicable if a command is to be executed when an alarm event occurs)	Optional
		see ***Note2***
			`Abnormal`
			Executes a command when the alarm status becomes abnormal.
			`Warning`
			Executes a command when the alarm status becomes warning.
			`Normal`
			Executes a command when the alarm status becomes normal.
			If multiple alarm statuses trigger command execution, specify the character strings indicating the applicable statuses by connecting them by a single-byte comma (`,`). The character strings indicating statuses can be specified in any order. However, an error occurs if the same character string is specified more than once.
			If `"Y"` is specified as the value of the `Check Value Exist` label in the `General` subsection, Warning cannot be specified. If `"Y"` is specified as the value of the `Regular Alarm` label in the `Advanced Setting` subsection, `Normal` cannot be specified.

Label Name	Description	Specification	Value
SNMP	Alarm status that triggers the transmission of an SNMP trap (applicable if SNMP traps are to be sent when an alarm event occurs)	Optional see ***Note2***	Abnormal Sends an SNMP trap when the alarm status becomes abnormal. Warning Sends an SNMP trap when the alarm status becomes warning. Normal Sends an SNMP trap when the alarm status becomes normal. If multiple alarm statuses trigger transmission of an SNMP trap, specify the character strings indicating the applicable statuses by connecting them by a single-byte comma (,). The character strings indicating statuses can be specified in any order. However, an error occurs if the same character string is specified more than once. If "Y" is specified as the value of the Check Value Exist label in the General subsection, Warning cannot be specified. If "Y" is specified as the value of the Regular Alarm label in the Advanced Setting subsection, Normal cannot be specified.
JP1 Event	This is an uneditable label.	Optional see ***Note2***	-

Note1: You can omit the entire label or only the value.

Note2: If you are omitting this specification, you must omit the entire label. You cannot omit only the value.

This example below executes the following actions when an alarm event occurs:

- Send email each time the alarm status becomes abnormal or warning.
- Send an SNMP trap each time the alarm status becomes abnormal, warning, or normal.

The following shows the Actions subsection for this example:

```
   :
[[Actions]]
#Report=
E-mail=Abnormal,Warning
Command=Abnormal
SNMP=Abnormal,Warning,Normal
#JP1 Event=N
   :
```

Action Definition E-mail Subsection

If email is to be sent when an alarm event occurs, the Action Definition E-mail subsection specifies the email transmission settings, such as the email address.

The table below lists and describes the labels and subsubsection that can be specified in the `Action Definition E-mail` subsection and their values. You must specify the labels and subsubsection in the order they are listed in this table.

Table 4.12 Labels and Subsubsection Specifiable in the Action Definition E-mail Subsection and Their Values

Label Name or Subsubsection Name	Description	Specification	Value
`E-mail` (label)	Email address	Required	Up to 127 bytes of single-byte characters. To specify multiple addresses, separate them with a single-byte comma (`,`). Make sure that the total length of all the addresses does not exceed 127 bytes.
`Action Handler` (label)	Service ID of the Action Handler service from which email is to be sent	Required	Up to 258 bytes of single-byte characters
`Message Text` (subsubsection)	Email message to be sent	Optional see ***Note***	Up to 1,000 bytes of double-byte and single-byte characters. Variables can be specified in the message. For a list of permitted variables and their descriptions, see Table 4.7. All information up to the first line of the next section or subsection (including lines containing only linefeed codes) is treated as the message text (not including comments).

Note: You can omit the entire label or only the value.

The example below sends email as follows when an alarm event occurs:

- Send email to `xxxx@xxx.xxx`.
- `PH1host01` is the service ID of the Action Handler service from which email is to be sent.
- Send the following message text:

```
Date: %SCT
Host: %HNS

Product: %PTS
```

The following shows the `Action Definition E-mail` subsection for this example:

```
   :
[[Action Definition E-mail]]
E-mail Address=xxxx@xxx.xxx
Action Handler=PH1host01
[[[Message Text]]]
Date: %SCT
Host: %HNS

Product: %PTS
```

Action Definition Command Subsection

If a command is to be executed when an alarm event occurs, the `Action Definition Command` subsection specifies the command execution settings, such as the name of the command to be executed.

The table below lists and describes the labels and subsubsection that can be specified in the `Action Definition Command` subsection and their values. You must specify the labels and subsubsection in the order they are listed in this table.

Table 4.13 Labels and Subsubsection Specifiable in the Action Definition Command Subsection and Their Values

Label Name or Subsubsection Name	Description	Specification	Value
`Command Name` (label)	Name of the command to be executed	Required	Up to 511 bytes of single-byte characters. If the name contains any single-byte space, enclose the entire name in double quotation marks ("). The command name must be expressed as the absolute path or a path relative to the current directory of the executing action handler, or it must be expressed as a command name that is found in the installation directory of the action handler or the directory set in the `PATH` environment variable. Specify the full path if you specify the execution module located in the WOW 64 system directory (SysWOW64) for Windows Server 2003 x64.
`Action Handler` (label)	Service ID of the Action Handler service that executes the command	Required	Up to 258 bytes of single-byte characters. To use the Action Handler service at the local host, specify `LOCAL`.
`Message Text` (subsubsection)	Parameters to be passed to the command	Optional see ***Note***	0 to 2,047 bytes of double-byte and single-byte characters. Variables can be specified for parameters. For a list of permitted variables and their descriptions, see Table 4.7. All information up to the first line of the next section or subsection (including lines containing only linefeed codes) is treated as parameters (not including comments).

Note: You can omit the entire label or only the value.

The example below executes the following command when an alarm event occurs:

- Command name: `/usr/bin/LogOutput`.
- `PH1host01` is the service ID of the Action Handler service that executes the command.
- Pass the following parameters to the command:

```
%SCT %HNS "%MTS"
```

The following shows the Action `Definition Command` subsection for this example:

```
   :
[[Action Definition Command]]
Command Name=/usr/bin/LogOutput
Action Handler=PH1host01
[[[Message Text]]]
%SCT %HNS "%MTS"
```

Action Definition JP1 Event Subsection

This subsection is not editable.

4.5 Setting Alarms

This section describes the procedure for setting alarms, based on an example of using Agent for RAID. For details about each label in the alarm definition file, see section 4.4.

4.5.1 Creating an Alarm Definition File

4.5.1.1 Outputting the Alarm Definition File Template

Before you create an alarm definition file, you must first output the template file that contains all the labels that can be defined in an alarm definition file.

In the following procedure, you output the template file named `/tmp/alarmtmp01.cfg`.

To output the template file:

1. Output the template file.

   In this step, you use the `jpcalarm export` command to output the template file. Execute this command with the `-template` option specified, as shown below:

```
jpcalarm export -f /tmp/alarmtmp01.cfg –template
```

   The following shows the output results:

```
#Alarm Definition File Version=0001
#Alarm Definition File Code=

#[Alarm Data]
#[[General]]
#Product=
#Alarm Table Name=
#Alarm Name=
#Message Text=
#Check Value Exist=N

#[[Advanced Setting]]
#Active Alarm=Y
#Regular Alarm=Y
#Evaluate All Data=N
#Monitoring Regularly=N
#Monitoring Time=
#Damping=N
#Damping Count=

#[[Check Value Exist]]
#Record=
#Field=
#Value=
```

```
#[[Alarm Condition Expressions]]
#Condition=

#[[Actions]]
#Report=
#E-mail=Abnormal,Warning,Normal
#Command=Abnormal,Warning,Normal
#SNMP=Abnormal,Warning,Normal
#JP1 Event=N

#[[Action Definition E-mail]]
#E-mail Address=
#Action Handler=
#[[[Message Text]]]
#Date: %SCT
#Host: %HNS
#
#Product: %PTS
#Agent: %ANS
#
#Alarm: %AIS (%ATS)
#State: %SCS
#
#Message: %MTS
#[[Action Definition Command]]
#Command Name=
#Action Handler=

#[[[Message Text]]]
#
#[[Action Definition JP1 Event]]
#Event ID=
#Action Handler=
#Message=%MTS
#Switch Alarm Level=Y
#Exec Logical Host=
```

A hash mark (#) appears at the beginning of each line in the template file, indicating that these are all comment lines.

4.5.1.2 Creating an Alarm Definition File

In this section, you edit the template file that you output above (`/tmp/alarmtmp01.cfg`) in order to create an alarm definition file.

To create an alarm definition file:

1. Use a text editor to open the `/tmp/alarmtmp01.cfg` file.
2. Define header information for the alarm definition file.

   In this step, you define header information. The header information defines the syntax version of the alarm definition file and the character codes used to create the alarm definition file. This information is defined on the following lines:

```
#Alarm Definition File Version=0001
#Alarm Definition File Code=
   :
```

Delete the hash mark at the beginning of each of these lines, and edit the lines as follows:

```
Alarm Definition File Version=0001
Alarm Definition File Code=C
   :
```

- `Alarm Definition File Version` label

  Defines the syntax version of the alarm definition file.

  For Tuning Manager 5.5, the syntax version is fixed to `0001`, so this value is set as the default in the template file.

- `Alarm Definition File Code` label

  Defines the character codes used for the alarm definition file.

  This example sets `C`, indicating use of English-language character codes.

3. Define an agent type, data model version, alarm table name, and alarm name.

   In this step, you define each alarm. Alarms are defined in the `Alarm Data` section. Create an `Alarm Data` section for each alarm that you define.

   The `Alarm Data` section is composed of several subsections. The following items are defined in the `General` subsection:

   - Agent type
   - Data model version
   - Alarm table name
   - Alarm name

   This information is defined on the following lines:

```
   :
#[Alarm Data]
#[[General]]
#Product=
#Alarm Table Name=
#Alarm Name=
#Message Text=
#Check Value Exist=N
```

   This example defines `Storage Monitoring` as the alarm that monitors the storage subsystem activity status.

   Delete the hash mark at the beginning of each of these lines, and edit them as follows:

```
[Alarm Data]
[[General]]
Product= D7.0
Alarm Table Name="Storage Monitoring"
Alarm Name="Usage Rate (CACHE)"
Message Text="Usage Rate (%CVS%)"
Check Value Exist=N
   :
```

- `Product` label

  Defines the agent type and the data model version. In this label, you define the product ID of the agent followed immediately by the data model version. In this example, `D` is specified as the product ID of Agent for RAID and `7.0` is specified as the version of the data model.

- `Alarm Table Name` label

  Defines a name for the alarm table. An alarm table name beginning with `PFM` cannot be specified. If the name contains any single-byte space, the entire name must be enclosed in double quotation marks ("). This example defines "`Storage Monitoring`".

- `Alarm Name` label

  Defines a name for the alarm. If the name contains any single-byte space, the entire name must be enclosed in double quotation marks ("). This example defines "`Usage Rate (CACHE)`".

- `Message Text` label

  Defines the actual content of the `%MTS` variable to be used to define the message text to be sent by email. If the value contains any single-byte space, the entire value must be enclosed in double quotation marks (").

- `Check Value Exist` label

  Specify `Y` or `N` to set this alarm as a **value existence checking alarm**.

4. Define the conditions under which the alarm is to be generated.

   In this step, you use conditional expressions to define the conditions under which the alarm is to be generated.

   The conditions under which an alarm is to be generated are defined in the `Alarm Condition Expressions` subsection. This information is defined on the following lines:

```
   :
#[[Alarm Condition Expressions]]
#Condition=
   :
```

   For each `Condition` label, an alarm-triggering conditional expression that includes the names of the monitored record and field is entered.

   This example monitors the cache memory usage on the disk array device.

   - Conditional expression for determining the cache memory usage on the disk array device

     The cache memory usage on the disk array device is stored in the Cache Memory Usage (`CACHE_MEMORY_USAGE`) field of the Storage Summary (`PI`) record. We use the values in this field for the evaluation condition.

```
PI_CACHE_MEMORY_USAGE
```

     This example defines that the status is to be treated as abnormal when the cache memory usage on the disk array device exceeds 50% and as warning when it exceeds 30%.

Delete the hash marks (#) from the beginning of the relevant lines, and then specify them in the alarm definition file.

```
   :
[[Alarm Condition Expressions]]
Condition=PI_CACHE_MEMORY_USAGE>50,30
   :
```

5. Define the actions that are to occur when the alarm is generated.

   In this step, you define in the `Actions` subsection the actions that are to occur when the alarm is generated. This information is defined on the following lines:

```
   :
#[[Actions]]
#Report=
#E-mail=Abnormal,Warning,Normal
#Command=Abnormal,Warning,Normal
#SNMP=Abnormal,Warning,Normal
#JP1 Event=N
   :
```

   To send email when the alarm status is abnormal, specify as follows:

```
   :
[[Actions]]
#Report=
E-mail=Abnormal
#Command=Abnormal,Warning,Normal
#SNMP=Abnormal,Warning,Normal
#JP1 Event=N
   :
```

   — `E-mail` label

   Defines the alarm conditions under which an email is to be sent.

   - To send email when an abnormal condition occurs: `Abnormal`

   - To send email when a warning condition occurs: `Warning`

   - To send email when a normal condition occurs: `Normal`

   To define more than one alarm condition under which the action is to be issued, separate the conditions with a comma (,).

6. Define the email destination and the message text.

   In this step, you define the email destination and the email's message contents.

   The email destination is defined in the `Action Definition E-mail` subsection, and the message text is defined in the `Message Text` subsubsection.

   This information is defined on the following lines:

```
   :
#[[Action Definition E-mail]]
#E-mail Address=
#Action Handler=

#[[[Message Text]]]
#Date: %SCT
#Host: %HNS
#
#Product: %PTS
#Agent: %ANS
```

```
#
#Alarm: %AIS (%ATS)
#State: %SCS
#
#Message: %MTS
   :
```

In the following example, we define the email destination as `xxxx@xxx.xxx`, and we indicate that the message text is specified with variables. Delete the hash marks from the relevant lines, and edit them as shown in the following:

```
   :
[[Action Definition E-mail]]
E-mail Address=xxxx@xxx.xxx
Action Handler=PH1host01
   :
[[[Message Text]]]
Date: %SCT
Host: %HNS

Product: %PTS
Agent: %ANS

Alarm: %AIS (%ATS)
State: %SCS

Message: %MTS
   :
```

- `E-mail Address` label

  Defines the email destination.

- `Action Handler` label

  Defines the service ID of the Action Handler service that acts as the source of the email.

- The following explains the message text to be sent, as defined through the use of variables in the `Message Text` subsection.

  - Date and time the alarm was generated
  - Host name of the agent on which the alarm was generated
  - Agent type and data model version
  - Name of agent on which the alarm was generated
  - Alarm name
  - Alarm table name
  - Alarm condition
  - Usage (value defined in the `Message Text` label of the `General` section)

See

Table 4.7 for descriptions of the variables that can be used in the `Message Text` subsubsection.

7. When you are finished editing, save the `/tmp/alarmtmp01.cfg` file.

4.5.2 Verifying an Alarm Definition File

In this section, you learn how to check the validity of the alarm definitions you have created. You use the `jpcalarm check` command to check the alarm definition file.

In the following example, you check not only the syntax of the alarm definition file, but you also check the contents of the file, such as whether the agents defined in the file are set up and whether the records and fields are supported.

To check the alarm definitions:

1. Check if the Name Server, Master Manager, and View Server services are running.

   In this step, you use the `jpcctrl list` command to check whether services of Tuning Manager series programs are running.

   Type the following to display a list of services running on `host01`:

```
jpcctrl list "*" host=host01
```

   If Tuning Manager is running on `host01`, a list similar to the following is output:

```
Host Name ServiceID  Service Name     PID    Port    Status
-------   ---------  ---------------- -----  ----    -------
host01    PC1host01  Trap Generator   1468   1134    Active
host01    PE1001     Correlator       1420   1114    Active
host01    PH1host01  Action Handler   872    1116    Active
host01    PM1001     Master Manager   1388   1104    Active
host01    PP1host01  View Server      1504   1155    Active
host01    PS1001     Master Store     632    1109    Active
host01    PN1001     Name Server      484    8204    Active
```

   In this example, the Name Server, Master Manager, and View Server services are all running.

2. Execute the `jpcalarm check` command.

   Type the command as shown below:

```
jpcalarm check -f /tmp/alarmtmp01.cfg
```

   If there are any errors in the alarm definition file, error messages will be output, indicating the text and line number of each error found in the file.

   When an error is reported, correct it based on the text of the error message.

4.5.3 Modifying Alarm Definitions

You can modify alarm definition information by exporting the alarm definitions to a separate file, editing the definitions in that file, and then importing the edited definitions.

The following commands are used for these operations:

- To export alarm definitions:

  `jpcalarm export` command

- To import alarm definitions:

  `jpcalarm import` command

Note:

The alarms defined in the solution set (the alarm table whose name starts with `PFM`) cannot be edited. To edit these alarms, export the solution set, change the name of the alarm table for the alarm definition file, and then import the file again.

To edit previously defined alarm definitions:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name of the alarm table in which the alarm definitions to be edited are defined.

   For example, to find the names of the alarm tables defined in Agent for RAID, type the command as follows:

   ```
   jpcalarm list -key agtd
   ```

   The following shows a sample output result. In this example, you can see that a solution set and an alarm table named `alarmtable1` are defined.

   ```
   Product ID:D
   Alarm Table Name:
   alarmtable1
     PFM RAID Solution Alarms 7.00
   ```

3. Execute the `jpcalarm list` command to find the names of the alarms whose definitions are to be edited.

   For example, to find the names of the alarms defined in the alarm table named `alarmtable1` of Agent for RAID, type the command as follows:

   ```
   jpcalarm list -key agtd -table alarmtable1
   ```

   The following shows a sample output result:

   ```
   Product ID:D
   DataModelVersion:7.0
   Alarm Table Name:alarmtable1
   Alarm Name:
     Disk 01      [active]
     Disk 02      [active]
   ```

```
The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

4. Execute the `jpcalarm export` command.

   For example, to export all the definition information for the alarms defined in the alarm table named `alarmtable1` of Agent for RAID to a file named `/tmp/alarmtable1.cfg`, type the command as follows:

```
jpcalarm export -f /tmp/alarmtable1.cfg -key agtd -table alarmtable1
```

5. Use a text editor to open the `/tmp/alarmtable1.cfg` file.
6. Edit the `/tmp/alarmtable1.cfg` file.

   For details about how to edit the definitions in the alarm definition file, see section 4.5.1.2.

7. Save the `/tmp/alarmtable1.cfg` file.
8. Execute the `jpcalarm import` command.

   For example, to import the definitions in the `/tmp/alarmtable1.cfg` alarm definition file, type the command as follows:

```
jpcalarm import -f /tmp/alarmtable1.cfg
```

Note: To edit alarms defined in the solution set (alarm table whose name begins with `PFM`):

Use the `jpcalarm export` command to export the alarms defined in the solution set.

Edit the alarm table name in the alarm definition file, and then use the `jpcalarm import` command to import the alarms.

4.5.4 Copying an Alarm Table

You use the `jpcalarm copy` command to copy an alarm table.

Notes:

- When you copy an entire alarm table, the copy destination alarm table is recognized as an alarm table of the same agent as the copy source alarm table. You cannot copy the alarm table for another agent.
- At the copy destination, you cannot specify an alarm table name that begins with `PFM`.

To copy an alarm table:

1. Log in to the host on which Tuning Manager is installed.

2. Execute the `jpcalarm list` command to find the name of the alarm table to be copied.

   For example, to find the names of the alarm tables defined in Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd
```

   The following shows a sample output result. In this example, you can see that only a solution set is defined.

```
Product ID:D
Alarm Table Name:
  PFM RAID Solution Alarms 7.00
```

3. Execute the `jpcalarm copy` command.

   For example, to copy the `PFM RAID Solution Alarms 7.00` solution set to an alarm table named `alarmtable1`, type the command as follows:

```
jpcalarm copy -key agtd -table "PFM RAID Solution Alarms 7.00" -name
alarmtable1
```

4. Execute the `jpcalarm list` command to verify that the alarm table was copied.

   As in step 2, type the command as follows:

```
jpcalarm list -key agtd
```

   The following shows a sample output result. In this example, you can see that a new alarm table named `alarmtable1` has been created.

```
Product ID:D
Alarm Table Name:
  alarmtable1
  PFM RAID Solution Alarms 7.00
```

4.5.5 Deleting an Alarm Table

You use the `jpcalarm delete` command to delete an alarm table.

Note: The solution set (alarm table whose name begins with `PFM`) cannot be deleted.

To delete an alarm table:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name of the alarm table to be deleted.

   For example, to find the names of the alarm tables defined in Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd
```

The following shows a sample output result. In this example, you can see that a solution set and an alarm table named `alarmtable1` are defined.

```
Product ID:D
Alarm Table Name:
  alarmtable1
  PFM RAID Solution Alarms 7.00
```

3. Execute the `jpcalarm delete` command.

   For example, to delete the `alarmtable1` alarm table of Agent for RAID, type the command as follows:

```
jpcalarm delete -key agtd -table alarmtable1
```

4. Execute the `jpcalarm list` command to verify that the alarm table has been deleted.

   As in step 2, type the command as follows:

```
jpcalarm list -key agtd
```

The following shows a sample output result. In this example, you can see that the alarm table named `alarmtable1` has been deleted.

```
Product ID:D
Alarm Table Name:
  PFM RAID Solution Alarms 7.00
```

4.5.6 Deleting an Alarm

You use the `jpcalarm delete` command to delete an individual alarm.

Note: An alarm defined in the solution set (alarm table whose name begins with `PFM`) cannot be deleted.

To delete an individual alarm:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name of the alarm table in which the alarm to be deleted is defined.

   For example, to find the names of the alarm tables defined in Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd
```

The following shows a sample output result. In this example, you can see that a solution set and an alarm table named `alarmtable1` are defined.

```
Product ID:D
Alarm Table Name:
  alarmtable1
  PFM RAID Solution Alarms 7.00
```

3. Execute the `jpcalarm list` command to find the name of the alarm to be deleted.

   For example, to find the names of the alarms defined in the `alarmtable1` alarm table of Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd -table alarmtable1
```

   The following shows a sample output result. In this example, you can see that the `Disk 01` and `Disk 02` alarms are defined in the `alarmtable1` alarm table.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:alarmtable1
Alarm Name:
  Disk 01      [active]
  Disk 02      [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

4. Execute the `jpcalarm delete` command.

   For example, to delete the `Disk02` alarm in the `alarmtable1` alarm table of Agent for RAID, type the command as follows:

```
jpcalarm delete -key agtd -table alarmtable1 -alarm "Disk 02"
```

5. Execute the `jpcalarm list` command to verify that the alarm has been deleted.

   As in step 3, type the command as follows:

```
jpcalarm list -key agtd -table alarmtable1
```

   The following shows a sample output result. In this example, you can see that the `Disk 02` alarm has been deleted.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:alarmtable1
Alarm Name:
  Disk 01       [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

4.6 Using Alarms

This section describes how to use the alarms.

4.6.1 Binding an Alarm Table to a Monitoring Agent

You use the `jpcalarm bind` command to bind an alarm table to an agent.

Note: Only one alarm table can be bound to an agent. When you bind a second alarm table to an agent to which an alarm table has already been bound, the existing alarm table is released and the new alarm table becomes bound to the agent.

To bind an alarm table:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name of the alarm table to be bound.

   For example, to check the names of the alarm tables defined in Agent for RAID, type the command as follows:

   ```
   jpcalarm list -key agtd
   ```

   The following shows a sample output result. In this example, you can see that only a solution set is defined.

   ```
   Product ID:D
   Alarm Table Name:
     PFM RAID Solution Alarms 7.00
   ```

3. Execute the `jpcalarm list` command to find the agents to which the alarm table is already bound.

   For example, to find the agents to which the `PFM RAID Solution Alarm 7.00` solution set of Agent for RAID is bound, type the command as follows:

   ```
   jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
   ```

   The following shows a sample output result. In this example, you can see that the `PFM RAID Solution Alarms 7.00` solution set is bound to instance `instA` and `instB` of host `hostA`.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate      [active]
  Write Cache Hit Rate     [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

4. Execute the `jpcalarm bind` command.

   For example, to bind the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID to instance instC of `host01`, type the command as follows:

```
jpcalarm bind -key agtd -table "PFM RAID Solution Alarms 7.00" -id
DA1instC[host01]
```

5. Execute the `jpcalarm list` command to verify that the alarm table has been bound.

   As in step 3, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

   The following shows a sample output result. In this example, you can see that the `PFM RAID Solution Alarms 7.00` solution set is now bound to instance `instA` and `instB` of host `hostA`, and instance `instC` of `host01`.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate     [active]
  Write Cache Hit Rate    [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
  DA1instC[host01]
```

4.6.2 Releasing Alarm Table Bindings to Monitoring Agents

You use the `jpcalarm unbind` command to release an alarm table's bindings.

To release an alarm table's binding:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name of the alarm table whose bindings are to be released.

For example, to check the names of the alarm tables defined in Agent for RAID type the command as follows:

```
jpcalarm list -key agtd
```

The following shows a sample output result. In this example, you can see that only a solution set is defined.

```
Product ID:D
Alarm Table Name:
  PFM RAID Solution Alarms 7.00
```

3. Execute the `jpcalarm list` command to find the currently bound agents of the alarm table whose bindings are to be released.

   For example, to find the agents to which the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID is bound, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

   The following shows a sample output result. In this example, you can see that the `PFM RAID Solution Alarms 7.00` solution set is bound to instance `instA` and `instB` of host `hostA`, and instance `instC` of `host01`.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate      [active]
  Write Cache Hit Rate     [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
  DA1instC[host01]
```

4. Execute the `jpcalarm unbind` command.

   For example, to release the bindings to all instances whose name begins with `inst` for the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID, type the command as follows:

```
jpcalarm unbind -key agtd -table "PFM RAID Solution Alarms 7.00" -id "DA1inst*"
```

5. Execute the `jpcalarm list` command to verify that the alarm table bindings have been released.

   As in step 3, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

   The following shows a sample output result. In this example, you can see that the `PFM RAID Solution Alarms 7.00` solution set is no longer bound to any agents.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM UNIX Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate      [active]
  Write Cache Hit Rate     [active]

The Bound Agent:
```

4.6.3 Checking the Bindings Between an Alarm Table and Monitoring Agents

You use the `jpcalarm list` command to check the status of an alarm table's bindings.

To check the status of an alarm table's bindings:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name of the alarm table whose bindings are to be checked.

   For example, to check the names of the alarm tables defined in Agent for RAID, type the command as follows:

   ```
   jpcalarm list -key agtd
   ```

   The following shows a sample output result. In this example, you can see that only a solution set is defined.

   ```
   Product ID:D
   Alarm Table Name:
     PFM RAID Solution Alarms 7.00
   ```

3. Execute the `jpcalarm list` command to find the agents to which the alarm table is bound.

   For example, to find the agents to which the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID is bound, type the command as follows:

   ```
   jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
   ```

   The following shows a sample output result. In this example, the solution set is bound to the Agents of instance `instA` and `instB` of host `hostA`, and instance `instC` of `host01`.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate       [active]
  Write Cache Hit Rate      [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
  DA1instC[host01]
```

4.6.4 Starting Alarm Monitoring

You use the `jpcalarm active` command to activate an alarm.

To activate an alarm:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name and status of the alarm to be activated.

   For example, to find the statuses of the alarms in the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

   Below, is a sample output result. If an alarm has been activated, `active` is displayed following the alarm name. If an alarm in not active, `inactive` is displayed following the alarm name. In this example, the `Read Cache Hit Rate` alarm is inactive.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate       [inactive]
  Write Cache Hit Rate      [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

3. Execute the `jpcalarm active` command.

   For example, to activate the `Read Cache Hit Rate` alarm in the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID, type the command as follows:

```
jpcalarm active -key agtd -table "PFM RAID Solution Alarms 7.00" -alarm "Read
Cache Hit Rate"
```

4. Execute the `jpcalarm list` command to verify that the alarm has been activated. As in step 2, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

Below, is a sample output result. In this example, the `Read Cache Hit Rate` alarm is now active.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate       [active]
  Write Cache Hit Rate      [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

4.6.5 Stopping Alarm Monitoring

You use the `jpcalarm inactive` command to deactivate an alarm.

To deactivate an alarm:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command to find the name and status of the alarm to be deactivated.

   For example, to find the statuses of the alarms in the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID, type the command as follows:

   ```
   jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
   ```

   The following shows a sample output result. If an alarm has been activated, `active` is displayed following the alarm name. In this example, you can see that all the alarms are active.

   ```
   Product ID:D
   DataModelVersion:7.0
   Alarm Table Name:PFM RAID Solution Alarms 7.00
   Alarm Name:
     Read Cache Hit Rate       [active]
     Write Cache Hit Rate      [active]
   The Bound Agent:
     DA1instA[hostA]
     DA1instB[hostA]
   ```

3. Execute the `jpcalarm inactive` command.

   For example, to deactivate the `Read Cache Hit Rate` alarm in the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID, type the command as follows:

```
jpcalarm inactive -key agtd -table "PFM RAID Solution Alarms 7.00" -alarm "Read
Cache Hit Rate"
```

4. Execute the `jpcalarm list` command to verify that the alarm has been deactivated.

   As in step 2, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

   The following shows a sample output result. If an alarm has been deactivated, `inactive` is displayed following the alarm name. In this example, you can see that the `Read Cache Hit Rate` alarm is now deactivated.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate      [inactive]
  Write Cache Hit Rate     [active]
The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

4.6.6 Checking the Properties of an Alarm Table

You can display a list of alarm tables defined for an agent. You can also display a list of alarms defined in an alarm table, as well as a list of the agents to which that alarm table is bound.

This section explains how to display definition information for alarm tables.

Note: You cannot display the threshold values of individual alarms or other definition information defined in an alarm. To check definition information for alarms, you must use the `jpcalarm export` command to export the alarm definitions. For details about exporting alarm definitions, see section 4.5.3.

4.6.6.1 Displaying a List of Alarm Tables

You use the `jpcalarm list` command to display a list of alarm tables defined by a particular agent.

To display a list of alarm tables:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command.

   For example, to find the names of the alarm tables defined by HTM - Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd
```

The following shows a sample output result. In this example, you can see that a solution set and an alarm table named `alarmtable1` are defined.

```
Product ID:D
Alarm Table Name:
  alarmtable1
  PFM RAID Solution Alarms 7.00
```

Table 4.14 shows the information that is output when you execute the `jpcalarm list` command with only the `-key` option specified.

Table 4.14 Information Displayed by the jpcalarm list Command (with the -key Option Specified)

Displayed Information	Description
Product ID	Indicates the product ID, which includes the agent type. For details about the product ID of each agent, see Appendix A.
Alarm Table Name	Indicates the name of the alarm table.

4.6.6.2 Displaying Information About Alarms in an Alarm Table

You use the `jpcalarm list` command to display a list of alarms defined in a particular alarm table and a list of agents to which that alarm table is bound.

To display alarm information:

1. Log in to the host on which Tuning Manager is installed.
2. Execute the `jpcalarm list` command.

   For example, to display information about alarms defined in the `PFM RAID Solution Alarms 7.00` solution set of Agent for RAID, type the command as follows:

```
jpcalarm list -key agtd -table "PFM RAID Solution Alarms 7.00"
```

   The following shows a sample output result. In this example, you can see that all the alarms in the solution set are active and that the solution set is bound to instance `instA` and `instB` of host `hostA`.

```
Product ID:D
DataModelVersion:7.0
Alarm Table Name:PFM RAID Solution Alarms 7.00
Alarm Name:
  Read Cache Hit Rate      [active]
  Write Cache Hit Rate     [active]

The Bound Agent:
  DA1instA[hostA]
  DA1instB[hostA]
```

   Table 4.15 shows the information that is output when you execute the `jpcalarm list` command with the `-key` and `-table` options specified.

Table 4.15 Information Displayed by the jpcalarm list Command (with the -key and - table Options Specified)

Displayed Information	Description
Product ID	Indicates the product ID, which includes the agent type. For details about the product ID of each agent, see Appendix A.
DataModelVersion	Indicates the version of the data model.
Alarm Table Name	Indicates the name of the alarm table.
Alarm Name	Indicates the name and status of each alarm: ▪ `active`: Alarm is active ▪ `inactive`: Alarm is not active
The Bound Agent	Indicates the service ID of the agent to which the alarm table is bound.

4.7 Notes on Alarms

This section provides notes on alarms.

4.7.1 Notes on Creating Alarms

- Alarm evaluation time

  If, in several records, you specify monitoring conditions with different monitoring intervals and offsets for an alarm, alarm evaluation is performed only when the monitoring time coincides with the scheduled data collection time. Change the collection interval setting as necessary.

- Saving a record that is to be evaluated as an alarm condition

  It is not necessary to register in the Store database a record that has been selected as an alarm condition.

- Limit on the number of alarms

  A maximum of 50 alarms can be registered in one alarm table.

- Changing the character code type

  If you have used double-byte characters to create alarms, do not change the character code types of Tuning Manager. If you subsequently change the character code type, you can no longer run the previously defined alarms and reports.

  Before changing the character code type, first uninstall, and then reconfigure your environment.

- Notes about setting the `Check Value Exist` label to Y in the `General` subsection

  If the `Check Value Exist` label is set to `Y` in the `General` subsection, the value specified in the conditional expression is not found in the collected data during alarm notification. In this case, even if specified, the `%CVS` variable for the message text or email subject will be replaced with a space character.

- Effect of the number of alarms generated on the number of connected agents

  In the Tuning Manager series, Tuning Manager performs specific processing, such as receiving alarms issued from agents and storing them sequentially in the Store database (Master Store). If an agent issues alarms too frequently or if alarms are issued from many agents simultaneously, Tuning Manager processing delays might result. If processing delays occur, unprocessed alarms may begin accumulating in the Tuning Manager host memory, decreasing available memory and possibly degrading system performance.

To avoid this situation, you should consider damping alarms when you define them so that the number of alarms reported does not exceed the number of alarms that Tuning Manager can process per unit of time. You should also determine in advance the number of agents to be connected to Tuning Manager. You can connect a maximum of 50 Agents to a single Tuning Manager.

4.7.2 Notes on Alarm Evaluation

- Limit on the number of alarm evaluations

  In terms of collecting multiple instance records with agents, a maximum of 32,767 instances can be handled in a single collection. For an alarm bound to agents, up to 32,767 instances are evaluated. Any subsequent instances are not evaluated.

- Alarm evaluation interval

  Alarm evaluation is performed at a set interval. This interval differs for each record of the monitoring agent. For details about the alarm evaluation interval for each record, see the chapters in the following manuals that describe records (description of each record):

  - *HiCommand Tuning Manager Hardware Reports Reference*
  - *HiCommand Tuning Manager Operating System Reports Reference*
  - *HiCommand Tuning Manager Application Reports Reference*

  To change the alarm evaluation interval, use Performance Reporter's `jpcasrec update` command to change the record collection interval.

- Differences in alarm evaluation for alarm condition combinations

  The way in which an alarm is evaluated depends on the alarm conditions and the record type of the alarm being evaluated. Table 4.16 shows the differences in alarm evaluations for various combinations of alarm conditions.

Table 4.16 Differences in Alarm Evaluation Based on Alarm Conditions

Conditional Expression	Record Type	Value of the Regular Alarm Label (see *Note 1*)	Value of the Evaluate All Data Label (see *Note 2*)	Alarm Evaluation (Reported)
In the `General` subsection, `Check Value Exist` label = `N`	Single-line record see ***Note 3***	N	N	▪ When an abnormal condition is satisfied and the previously reported alarm was a condition other than abnormal, an abnormal alarm is reported. ▪ When a warning condition that is not an abnormal condition is satisfied and the previously reported alarm was a condition other than warning, a warning alarm is reported. ▪ If neither of the above conditions is applicable and the previously reported alarm was either abnormal or warning, a normal alarm is reported.
		N	Y	

Conditional Expression	Record Type	Value of the Regular Alarm Label (see *Note 1*)	Value of the Evaluate All Data Label (see *Note 2*)	Alarm Evaluation (Reported)
		Y	N	When either an abnormal or warning condition is satisfied, an alarm that indicates an abnormal (or warning) condition is reported, regardless of whether or not an alarm was reported previously.
		Y	Y	
	Multi-line record see ***Note 4***	N	N	▪ When a single data item that satisfies an abnormal condition is found and the previously reported alarm was a condition other than abnormal, an abnormal alarm is reported for that data item. ▪ When no data item that satisfies an abnormal condition is found but a single data item that satisfies a warning condition is found, and the previously reported alarm was a condition other than warning, a warning alarm is reported for that data item. ▪ If none of the collected data satisfies either of the above conditions and the previously reported alarm was either abnormal or warning, a normal alarm is reported. ***Note***: Because alarm evaluation terminates as soon as a data item that satisfies a condition is found, some of the collected data might not have been evaluated.
		N	Y	▪ If evaluation of all the collected data results in finding one or more data items that satisfy an abnormal condition and the previously reported alarm was a condition other than abnormal, an abnormal alarm is reported for each of these data items. ▪ If evaluation of all the collected data does not result in finding any data item that satisfies an abnormal condition but does result in finding one or more data items that satisfy a warning condition, and the previously reported alarm was a condition other than warning, a warning alarm is reported for each of these data items. ▪ If none of the collected data satisfies either of the above conditions and the previously reported alarm was either abnormal or warning, a normal alarm is reported. ***Note***: Because all the data is evaluated, there might be more than one alarm notification for any one interval.

Conditional Expression	Record Type	Value of the Regular Alarm Label (see *Note 1*)	Value of the Evaluate All Data Label (see *Note 2*)	Alarm Evaluation (Reported)
		Y	N	▪ As soon as a data item that satisfies an abnormal condition is found, an alarm is reported that indicates an abnormal condition based on that data item, regardless of whether or not an alarm was reported previously. ▪ If no data item that satisfies an abnormal condition has been found, as soon as a data item that satisfies a warning condition is found, an alarm is reported that indicates a warning condition based on that data item, regardless of whether or not an alarm was reported previously. ***Note:*** Because alarm evaluation terminates as soon as a data item that satisfies a condition is found, some of the collected data might not have been evaluated.
		Y	Y	For all data, an alarm is reported that indicates that an abnormal or warning condition exists for each data item that satisfies an abnormal (or warning) condition. ***Note***: Because all the data is evaluated, there might be more than one alarm notification for any one interval.
In the `General` subsection, `Check Value Exist` label = `Y`	Multi-line record see ***Note 4***	N	N	If the value specified in the `Alarm Condition Expressions` conditional expression is not found in all the collected data (which means that the condition is not met), the abnormal alarm is notified. ***Note***: An alarm notification indicating non-operation is reported only once. If no data has been collected, alarm evaluation is not performed.
		N	Y	
		Y	N	If the value specified in the `Alarm Condition Expressions` conditional expression is not found in all the collected data (which means that the condition is not met), the abnormal alarm is notified. ***Note***: Alarm notification is performed each time. If no data has been collected, alarm evaluation is not performed.
		Y	Y	

Note 1: The Regular Alarm label is the Regular Alarm label in the Advanced Setting subsection.

Note 2: The Evaluate All Data label is the Evaluate All Data label in the Advanced Setting subsection.

Note 3: A single-line record is a single-instance record.

Note 4: A multi-line record is a multiple-instance record.

The following explains how alarm evaluation is performed for each alarm notification condition:

When a value is set in the `Check Value Exist` section

If you set a value in the `Check Value Exist` subsection, whether or not the specified value exists will be evaluated for all fields in the specified PD and PI record type records. If a value does not exist, only one alarm is reported in an interval.

If an alarm conditional expression is set

When you specify an alarm conditional expression, multiple records are collected in a single interval if the alarm is being evaluated for PD record type records. By default, alarm evaluation terminates as soon as the first data item that satisfies the conditional expression is found. This means that some of the performance data might not be evaluated. To include performance data of the PD record type in the target of alarm evaluation, set the `Evaluate All Data` label to `Y` in the `Advanced Setting` subsection.

If the alarm is being evaluated for a PI record type record, only one record is collected per interval, so alarm evaluation and notification are performed only once per interval.

- Differences in alarm evaluation when damping is set

  With respect to the explanation under the *Differences in alarm evaluation for alarm condition combinations* bullet above, there are additional differences in alarm evaluation when the `Damping` label is set in the `Advanced Setting` subsection. The table below shows the differences in alarm evaluation for alarm condition combinations when the `Damping` label is set.

Table 4.17 Differences in Alarm Evaluation When Damping Is Set

Value of the Damping Label	Value of the Regular Alarm Label (see *Note 1*)	Value of the Evaluate All Data Label (see *Note 2*)	Alarm Evaluation (Reported)
Y	N	N	▪ An alarm is reported only when the alarm condition has changed from the previously reported condition. ▪ An alarm condition is reported based on the data item collected at the time of alarm notification that satisfies the most important condition. ***Note***: The alarm status is the result of evaluating the value set in the `Damping Count` label; therefore, the alarm status might not match the threshold of the reported data.
Y	N	Y	▪ An alarm is reported only when the alarm condition has changed from the previously reported condition. ▪ When the alarm condition is warning or abnormal, an alarm condition is reported based on all data that satisfies the alarm condition at the time of alarm notification. ***Note***: The alarm status is the result of evaluating the value set in the `Damping Count` label; therefore, the alarm status might not match the threshold of the reported data.
Y	Y	N	The data item that satisfies the most important condition collected at the time of alarm notification is reported.
Y	Y	Y	All data items that satisfy an abnormal or warning condition at the time of alarm notification are reported.

Note 1: The Regular Alarm label is the Regular Alarm label in the Advanced Setting subsection.

Note 2: The Evaluate All Data label is the Evaluate All Data label in the Advanced Setting subsection.

The following table shows the timing of alarm notification.

Table 4.18 The Timing of Alarm Notification

Damping	Timing of Alarm Notification
n/m	An alarm is reported if the threshold value is exceeded n times in *m* intervals. Subsequently, an alarm is reported if the threshold value is exceeded *n* times while alarm evaluation is performed *m* times.
n/n	An alarm is reported the first time the threshold value is exceeded. Subsequently, an alarm is reported every *n* times while the threshold value is being exceeded continuously. This is used to prevent alarm notification from being performed continuously while a threshold value is being exceeded.

4.7.3 Notes about Operation

When an Agent is started in the stand-alone mode, alarms cannot be used to monitor performance data, nor can actions be taken in response to alarm events.

Once the connection-target Tuning Manager or the Action Handler service stops, no actions are executed. To execute actions, start the connection-target Tuning Manager and the Action Handler service.

Chapter 5 Backup and Disk Management

This chapter describes backup of the Tuning Manager series system and management of the disk used by the Store database. As a part of the backup plan for the entire system, consider the backup of the Tuning Manager series system, and periodic management of the disk used by the Store database.

This chapter covers the following topics:

- Backup and Restore for Tuning Manager Programs (see section 5.1)
- Backup and Restore for the Store Database (see section 5.2)
- Disk Management for the Store Database (see section 5.3)
- Precautions Regarding Temporary Files (see section 5.4)

5.1 Backup and Restore for Tuning Manager Programs

If a Tuning Manager series program fails to operate due to a disk failure, it may become impossible to restore data that is used in the Tuning Manager series program. In preparation for unexpected events such as these, it is necessary to periodically back up various types of definition information, performance data, service definition information, and so on. Also, if you change the instance name or settings of an instance environment, make a backup of updated information.

This section describes how to back up the various types of definition information of the Tuning Manager series programs.

5.1.1 Backing Up Performance Data and Service Definition Information

The following information must be backed up: Performance data and Service definition information.

The following sections describe backup procedures for this information.

5.1.2 Backing Up and Restoring Performance Data

To back up performance data stored in the Store database, use the `jpcctrl backup` command. Ensure that you periodically back up this data.

For details about the backup procedures, see section 5.2.1.

5.1.3 Backing Up and Restoring Service Definition Information

Ensure that you back up service definition information periodically. When restoring service definition information, overwrite the definition information file that has been backed up.

5.1.3.1 Before You Begin

Before backing up service definition information, review the following notes.

- Terminate all Tuning Manager series program services on the local host.
- If service definition information is only restored on the Agent host, node and other information of instances added after backup will remain in the Tuning Manager host. In this case, execute the `jpcctrl delete` command to delete unnecessary agent information.

5.1.3.2 Definition Information Files To Be Backed Up

The following table shows the definition information files to be backed up for both Windows and UNIX systems.

Table 5.1 Definition Information Files to Be Backed Up (For Windows)

Operating System	File Type	File Name	Description
Windows	Common	*installation-folder**.ini	Common settings file in the Tuning Manager series programs
		installation-folder\bin\action*.ini	Settings file of the Action Handler service
		installation-folder\bin\statsvr*.ini	Settings file of the Status Server service
	Performance Reporter	*Performance-Reporter-installation-folder*\conf*.*	Settings file of Performance Reporter
	Tuning Manager	*installation-folder*\mgr\clator*.ini	Settings file of the Correlator service
		installation-folder\mgr\manager*.ini	Settings file of the Master Manager service
		installation-folder\mgr\manager*.DB	Database file of the Master Manager service
		installation-folder\mgr\manager*.IDX	Index file of the Master Manager service
		installation-folder\mgr\manager*.DAT	Data model file of the Master Manager service
		installation-folder\mgr\store*.ini	Settings file of the Master Store service
		installation-folder\mgr\store*.DAT	Data model file of the Master Store service
		installation-folder\mgr\namesvr*.ini	Settings file of the Name Server service
		installation-folder\mgr\namesvr*.DB	Database file of the Name Server service
		installation-folder\mgr\namesvr*.IDX	Index file of the Name Server service
		installation-folder\mgr\trapgen*.ini	Settings file of the Trap Generator service
		installation-folder\mgr\viewsvr*.ini	Settings file of the View Server service
		installation-folder\mgr\viewsvr\data*	User definition information file of the View Server service

Operating System	File Type	File Name	Description
		installation-folder\mgr\viewsvr\reports*	Report definition information file of the View Server service
	Agent	*installation-folder**xxxx*\agent*.ini (see ***Note 1***)	Settings file of the Agent Collector service
		*installation-folder**xxxx*\agent\instance-name*.ini (see ***Note 1*** and ***Note 2***)	Settings file of the Agent Collector service
		*installation-folder**xxxx*\store*.ini (see ***Note 1***)	Settings file of the Agent Store service
		*installation-folder**xxxx*\store\instance-name*.ini (see ***Note 1*** and ***Note 2***)	Settings file of the Agent Store service

Table 5.2 Definition Information Files to Be Backed Up (For UNIX)

Operating System	File Type	File Name	Description
UNIX	Common	/opt/jp1pc/*.ini	Common settings file in the Tuning Manager series programs
		/opt/jp1pc/bin/action/*.ini	Settings file of the Action Handler service
		/opt/jp1pc/bin/statsvr/*.ini	Settings file of the Status Server service
	Performance Reporter	/opt/HiCommand/TuningManager/PerformanceReporter/conf/*.*	Settings file of Performance Reporter
	Tuning Manager	/opt/jp1pc/mgr/clator/*.ini	Settings file of the Correlator service
		/opt/jp1pc/mgr/manager/*.ini	Settings file of the Master Manager service
		/opt/jp1pc/mgr/manager/*.DB	Database file of the Master Manager service
		/opt/jp1pc/mgr/manager/*.IDX	Index file of the Master Manager service
		/opt/jp1pc/mgr/manager/*.DAT	Data model file of the Master Manager service
		/opt/jp1pc/mgr/store/*.ini	Settings file of the Master Store service
		/opt/jp1pc/mgr/store/*.DAT	Data model file of the Master Store service
		/opt/jp1pc/mgr/namesvr/*.ini	Settings file of the Name Server service
		/opt/jp1pc/mgr/namesvr/*.DB	Database file of the Name Server service

Operating System	File Type	File Name	Description
		/opt/jp1pc/mgr/namesvr/*.IDX	Index file of the Name Server service
		/opt/jp1pc/mgr/trapgen/*.ini	Settings file of the Trap Generator service
		/opt/jp1pc/mgr/viewsvr/*.ini	Settings file of the View Server service
		/opt/jp1pc/mgr/viewsvr/data/*	User definition information file of the View Server service
		/opt/jp1pc/mgr/viewsvr/Reports/*	Report definition information file of the View Server service
	Agent	/opt/jp1pc/*xxxx*/agent/*.ini (see ***Note 1***)	Settings file of the Agent Collector service
		/opt/jp1pc/*xxxx*/agent/*instance-name*/*.ini (see ***Note 1*** and ***Note 2***)	Settings file of the Agent Collector service
		/opt/jp1pc/*xxxx*/store/*.ini (see ***Note 1***)	Settings file of the Agent Store service
		/opt/jp1pc/*xxxx*/store/*instance-name*/*.ini (see ***Note 1*** and ***Note 2***)	Settings file of the Agent Store service

Note 1: *xxxx* indicates the service key of each Agent. For details on service keys of each Agent, see Appendix A.

Note 2: *instance-name* indicates a directory for operating in the instance environment. For an Agent that monitors an application program that can start a set of multiple services at the same host, the system creates a number of directories equal to the number of instances.

5.2 Backup and Restore for the Store Database

This section describes the backup and restoration procedures for the Store database. To back up the Store database, use the `jpcctrl backup` command. To restore the Store database, use the `jpcresto` command.

When backing up or restoring the Store database, the following OS user permissions are required:

- Windows systems: Users with Administrator or Backup Operator permissions
- UNIX systems: Users with a root user permissions

Note: The Store database cannot be restored if either of the following conditions is present:

- The data model version of the Store database that is backed up using the jpcctrl backup command differs from the version of the Store database to be restored.
- The service key of the data to be restored differs from the data backed up with the jpcctrl backup command.

5.2.1 Backing Up the Store Database

To back up the Store database, use the `jpcctrl backup` command. For details on the `jpcctrl backup` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

To back up the Store database:

1. Log in to the host where Tuning Manager is installed.
2. Make sure that the Name Server service, the Master Manager service, and the Agent Store service that manages the performance data that you want to back up, are running.
3. Execute the `jpcctrl backup` command.

   Example:

   To back up performance data in the Store database of Agent for Oracle running on `host02`, execute the following command:

   ```
   jpcctrl backup OS* host=host02
   ```

   By default, when the command is executed, the backup file is created with the following name on the host where the Agent is operating.

   Windows systems:

   ```
   installation-
   folder\xxxx\store[\instance-name]\backup\generation-
   number\database-ID.db
   ```

   (see ***Note 1***, ***Note 2***, and ***Note 3***)

UNIX systems:

/opt/jp1pc/***xxxx***/store[/*instance-name*]/backup/generation-number/database-ID.db

(see ***Note 1***, ***Note 2***, and ***Note 3***)

Note 1: *xxxx* indicates the service key of each Agent. For details on the service keys of each agent, see Appendix A.

Note 2: The generation number is given in order starting from 01. The maximum number of the generation number is the value specified for Backup Save in the jpcsto.ini file. The default maximum of the generation number is **05**.

Note 3: The following shows the database ID:

- PI: PI record type database
- PD: PD record type database
- PL: PL record type database

Do not stop the Master Manager service during backup processing. If the Master Manager service is stopped, the following message is displayed:

```
KAVE05266-E The backup command was interrupted because Master
Manager stopped.
```

Even when this message is displayed, backup processing continues uninterrupted. The backup processing is executed for the Master Store or Agent Store service that was specified in the argument of the jpcctrl backup command. In this case, monitoring of backup processing is terminated only for the jpcctrl backup command.

While backup processing is underway for the Master Store or Agent Store service, if the Master Manager service is restarted and the jpcctrl backup command is re-executed, the following message may be displayed:

```
KAVE05232-E Because backup or export was being processed, the
request was refused.
```

If this message is displayed, wait a while after backup processing is completed, and then re-execute the command.

To determine whether the backup processing is completed after canceling the command, check the BACKUP.LOG file that has been output under the Store database's backup destination directory with generation number 01. The message Ended normally. indicates that the processing was completed successfully.

5.2.2 Restoring the Store Database

To restore the Store database, execute the `jpcresto` command at the Tuning Manager host or Agent host where the database to be restored is backed up. For details on this command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

To restore the Store database:

1. Log in to the Agent host where the backup file is stored.
2. Execute the `jpcstop` command to stop the service of the Agent.
3. Confirm the storage location of the backup file.
4. Execute the `jpcresto` command.

   Example:

   To restore the Store database of `oracleA` (which is an instance of the Agent for Oracle) stored in `c:\Program Files\HiCommand\TuningManager\jp1pc\agto\store\oracleA\backup\01`, execute the following command:

   ```
   jpcresto agto "c:\Program Files\HiCommand\TuningManager\jp1pc\agto\store\oracleA\backup\01" inst=oracleA
   ```

5. Execute the `jpcstart` command to start the service of the Agent.

5.3 Disk Management for the Store Database

To avoid increasing the amount of data that is stored in the Store database of the Tuning Manager series programs, records are automatically summarized after a certain amount of time has passed, and upper limits are set for the number of maximum records that can be stored. However, if the disk that the Store database is using becomes full, data will be overwritten, starting with the oldest data. Therefore, we recommend periodic maintenance of the Store database disk.

This section describes how to delete the data stored in the Store database.

5.3.1 User Permission Prerequisites for Deletion

If you no longer need the data that is stored in the Store database, you can delete the data using the `jpcctrl clear` command. The `jpcctrl clear` command can only be used on a host where Tuning Manager is installed.

To execute the `jpcctrl clear` command, the following OS user permissions are required.

- Windows systems: Users with Administrator or Backup Operator permissions
- UNIX systems: Users with a root user permissions

For details about the `jpcctrl clear` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

5.3.2 Deleting Data from the Store Database

In Tuning Manager series programs, if the data stored in the Store database is no longer needed, you can delete the data by using the `jpcctrl clear` command. This command can only be used on a host where Tuning Manager is installed.

To delete the data in the Store database:

1. Log in to the host where Tuning Manager is installed.
2. Execute the `jpcctrl list` command to confirm that the Name Server service, Master Manager Service, and the Master Store service are running. Also, confirm that the Agent Store service is running on the host that contains the data to be deleted.
3. Execute the `jpcctrl clear` command.

   Example:

   To delete all performance data contained in the Store database of Agent for Platform (Windows) on `host02`, execute the following command:

   ```
   jpcctrl clear TS* host=host02 *
   ```

5.4 Precautions Regarding Temporary Files

5.4.1 Temporary File Output Directory

When Performance Reporter is used to display real-time or historical reports, or Tuning Manager is used to perform polling, the Agent Collector service and Agent Store service create temporary files in the following directories:

In Windows

The directory set by the `TMP` system environment variable.

In UNIX

The directory set by the `TMPDIR` system environment variable.

If `TMPDIR` is not set, the `/var/tmp` or `/tmp` directory is used.

If the disk containing the above directory does not have enough free space, the message `KAVE00105-E The disk capacity is insufficient.` might be output by the Agent Collector service or Agent Store service.

If additional free space cannot be secured for the directory for storing temporary files, the directory in which temporary files are created can be changed by specifying the path of the directory for outputting temporary files by using the `JPC_TMPDIR` environment variable.

Notes

On Windows, set the `JPC_TMPDIR` environment variable in the system environment variables.

When automatic start is set up in a UNIX environment through an automatic start script, perform the following to edit the automatic start script file:

a) Log in as the root user.

Open the `jpc_start` automatic start script file in a text editor such as `vi`, and find the following line:

```
export PATH SHLIB_PATH LD_LIBRARY_PATH LIBPATH HCCLIBCNF
```

Change the line found in the previous step as follows:

```
JPC_TMPDIR=temporary-file-output-directory
export PATH SHLIB_PATH LD_LIBRARY_PATH LIBPATH HCCLIBCNF
JPC_TMPDIR
```

Note: For *temporary-file-output-directory*, specify the path to a directory on a disk that has sufficient free space.

Restart the system.

5.4.2 Estimating the Size of Temporary Files

The following explains how to estimate the size of temporary files.

Temporary files created by displaying reports in Performance Reporter

Use the following formula to estimate the size of temporary files created when Performance Reporter is used to display real-time or historical reports.

total-specified-field-size * *instance-count* * *record-count* (units: bytes)

Note: *total-specified-field-size* indicates the total size of fields specified in reports.

Temporary files created by Tuning Manager polling

Estimate for each Agent the size of temporary files created when polling is performed in Tuning Manager. Table 5.3 lists the file sizes needed for each Agent.

Table 5.3 Estimating the Size of Temporary Files Created by Tuning Manager Polling

Agent	File Size
Agent for RAID	15 MB
Agent for RAID Map	6 * *instance-count-of-the-PD_FSC-record* (units: KB)
Agent for Platform (Windows)	1 * *instance-count-of-the-PI_PHYD-record* (units: KB)
Agent for Platform (UNIX)	1 * *instance-count-of-the-PI_DEVD-record* (units: KB)
Agent for SAN Switch	200 KB
Agent for Oracle	5 + 1 * *instance-count-of-the-PI_PIDF-record* + 0.5 * *instance-count-of-the-PD_PDTS-record* (units: KB)

Example calculation

The following shows an example calculation for the size of temporary files created when a historical report is displayed for a PI_LDS record in Agent for RAID.

The following table describes the conditions when a historical report is displayed for a PI_LDS record.

Table 5.4 Example Conditions for Displaying a Historical Report for a PI_LDS Record

Item	Description
Specified fields	`Date and Time` (`char(6)` type: 6 bytes, see ***Note 1***) `LDEV Number` (`string` type: 16 bytes) `Read I/O/sec` (`float` type: 4 bytes) `Write I/O/sec` (`float` type: 4 bytes) `Read Xfer/sec` (`float` type: 4 bytes) `Write Xfer/sec` (`float` type: 4 bytes) `Date` (`char(3)` type: 6 bytes, see ***Note 2***)
Number of instances collected every 5 minutes	2,000
Collection interval (number of displayed records)	Displays as a report the records per minute for one day. The number of displayed records is 12 records (1 hour) * 24 hours, or 288 records.

In this case, the calculation of the size of created temporary files is as follows (units: bytes).

(6 + 16 + 4 + 4 + 4 + 4 + 6) * 2,000 * 288 = 25,344,000

Note 1: When a real-time or historical report is displayed, the `Date and Time` field and common key field are always obtained, even though they are not displayed in the report.

Note 2: For the `Date` and `Time` fields automatically added to the Store database, calculate using 6 bytes for historical reports. For other fields, calculate using the size for the type as shown in the manual.

Because the number of instances changes during each record collection, the above file size is just a guide. For details on the data types for each field and size for each data type, see the *HiCommand Tuning Manager Hardware Reports Reference*, *HiCommand Tuning Manager Operating System Reports Reference*, or *HiCommand Tuning Manager Application Reports Reference*.

Note

The temporary file size is proportionate to the number of instances per record and the number of records, as specified for the report. As such, if the temporary file size seems likely to exceed 2 GB, the amount of data to be displayed can be adjusted in the `record-count` attribute specified for the Performance Reporter historical report, and in the `maximum-number-of-records` attribute in the `indication-settings` tag for the input file used by the `jpcrpt` command. For the `record-count` and `maximum-number-of-records` attribute, specify the value corresponding to *instance-count* * *record-count* in the above formula.

Precaution

In UNIX, the temporary file size limit may be less than 2 GB due to a file size limit from the `ulimit` command. Even when the temporary file size estimated above does not exceed 2 GB, if the error message KAVE00103-E `An unexpected exception has occurred (rc=27)` is output during report display, use the `ulimit` command to check the file size limit.

Chapter 6 Preparing for Failover in a Cluster System

This chapter describes the flow of processing when a Tuning Manager series program operates in a cluster system.

This chapter covers the following topics:

- Cluster System Overview (see section 6.1)
- About Failover (see section 6.2)
- Operations in a Cluster System (see section 6.3)

6.1 Cluster System Overview

This section provides an overview of a cluster system and describes the configuration when an Agent is run in a cluster system.

A cluster system allows you to link multiple server systems together and handle them as one system. An Agent can run in the following cluster systems:

- HA (High Availability) cluster systems
- Combined database servers (for Agent for Microsoft SQL Server)

In this chapter, the term cluster system refers to an **HA cluster system**. The term environment directory refers to a shared disk that is specified when you create a logical host.

6.1.1 HA Cluster System

This section provides an overview of the HA cluster system and describes the configuration when an Agent is run in an HA cluster system.

6.1.1.1 HA Cluster System Overview

An HA cluster system provides high system availability. The HA cluster system is designed to continue operating even if a failure occurs. If a failure occurs in a server executing tasks, a standby server immediately takes over and continues operation processes. This prevents operation interruption in case of a failure and thus ensures high availability.

In a cluster system, a server system that is executing system operations is called an executing node, while a server system that is on standby and is waiting to take over operations whenever a failure occurs in an executing system is called a standby node. A cluster system is also called a node switching system, because it switches servers from the executing node to standby node whenever necessary to continue operation if a failure occurs.

The software program that controls the entire HA cluster system is called *cluster software*. Cluster software monitors the system to ensure it is operating properly, and prevents operation interruption using the failover mechanism when it detects an abnormality.

To allow applications such as Tuning Manager series programs to fail over, you need to run the Tuning Manager series programs in a logical host. A *logical host* is a logical node that is controlled by the cluster software and used as the unit for failover. A logical host uses a *logical host name* as its host name, and has a *shared disk* and a *logical IP address* which are transferred from the executing node to an associated standby node in an event of a failover. Applications in the executing node can store their data in the shared disk and communicate with the standby node using the logical IP address. Thus, the applications can fail over independently of physical nodes.

While a logical node that is used as the unit for failover is called a logical host, a physical node is called a *physical host*. The host name used by a physical host (the host name displayed when the `hostname` command is executed) is called a *physical host name*, and the IP address corresponding to a physical host name is called a *physical IP address*. The disk used by a physical node is a *local disk*. These settings are specific to each node and cannot be inherited by another node.

6.1.1.2 Agent Configuration in an HA Cluster System

When you operate Agents in an HA cluster system, Agents operate on a physical host or a logical host, depending on their types.

For Agents that cannot run on logical hosts

The following Agents cannot run on logical hosts for HA cluster systems.

- Agent for RAID Map
- Agent for Platform
- Agent for Microsoft Exchange Server

Since Agent for Platform monitors OS performance, even in a cluster system it runs on a physical host to collect the performance data from each node. Likewise, Agent for RAID Map and Agent for Microsoft Exchange Server also collect configuration information and performance information for physical hosts in a cluster system. This is not possible for configurations in which execution and failover are performed on a logical host environment. Do not register this product to cluster software even if you use it in a cluster system.

For Agents that can run on logical hosts

The following Agents can run on physical hosts and logical hosts for HA cluster systems.

- Agent for RAID
- Agent for SAN Switch
- Agent for NAS
- Agent for Oracle

- Agent for Microsoft SQL Server
- Agent for DB2®

These Agents run on configurations combined with monitoring targets. When a monitoring target is run on a logical host, the user executes Agent on the same logical host as the monitoring target, and monitors the performance of the monitoring target.

For example, in the case of Agent for RAID, it runs in a logical host environment of a cluster configuration and monitors a storage subsystem. Agent for SAN Switch or Agent for NAS also runs in a logical host environment of a cluster configuration and monitors a switch or the NAS system of the monitoring target applications.

Figure 6.1 shows an example of a configuration when operating Agent for RAID in an HA cluster system.

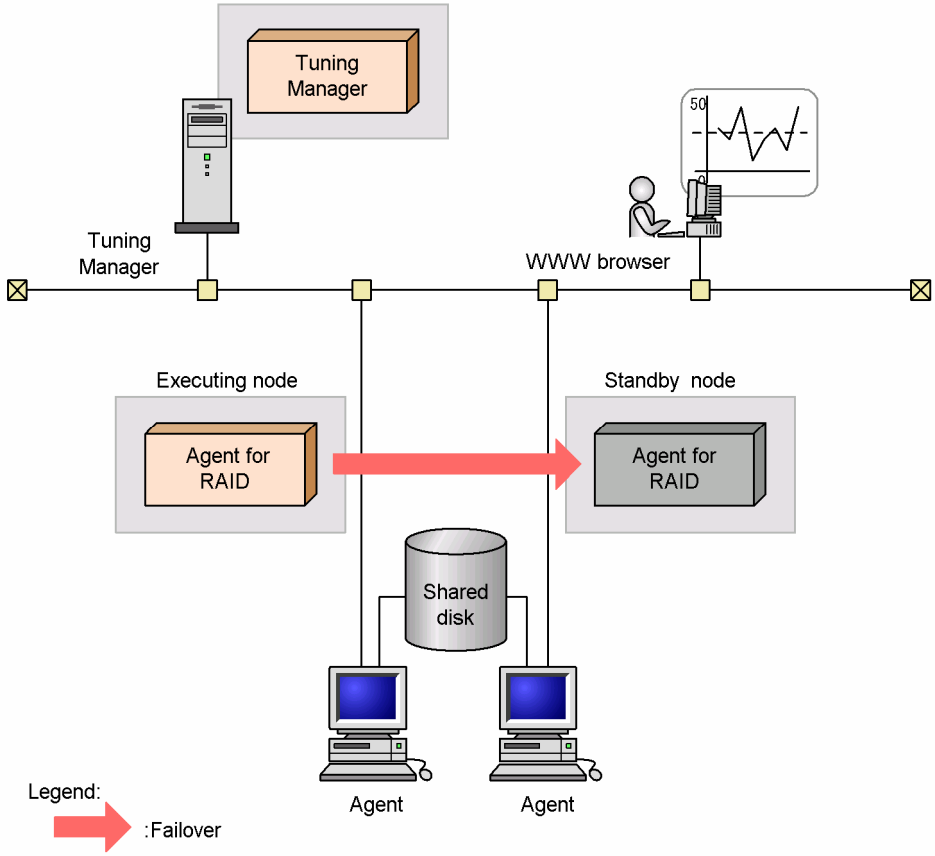


Figure 6.1 Example of a Configuration When Operating Agent for RAID in an HA Cluster System

In the case of Agent for Oracle, it runs in the same logical host environment as the cluster configuration of Oracle and monitors Oracle. Agent for Microsoft SQL Server or Agent for DB2 also runs in the same logical host environment as the database of a cluster configuration and monitors the database.

Figure 6.2 shows an example of a configuration when operating Agent for Oracle in an HA cluster system.

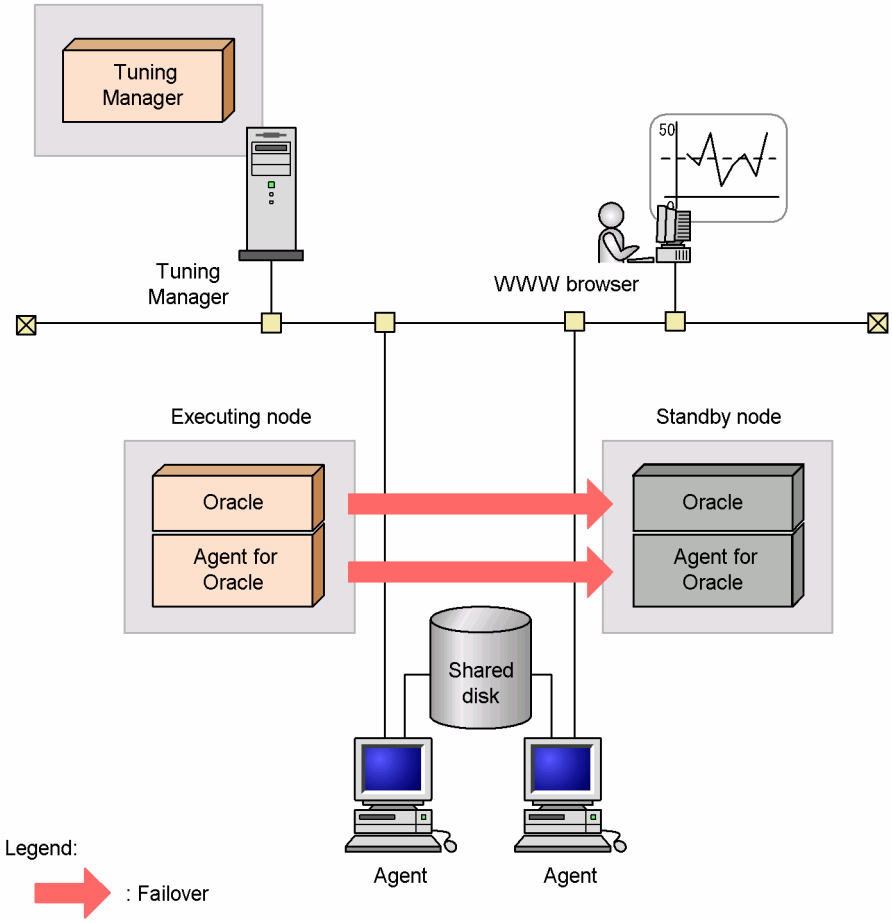


Figure 6.2 Example of a Configuration When Operating Agent for Oracle in an HA Cluster System

An Agent running in a logical host environment stores definition information and the Store database on a shared disk, which are inherited by the standby node in the event of failover. If there are multiple Tuning Manager series programs for a single logical host, all the Tuning Manager series programs use the same shared directory.

In the case of Agent for Oracle, Agent for Microsoft SQL Server, or Agent for DB2, multiple Agents can run on a single node. In the case of a configuration that contains multiple databases of a cluster configuration (active/active configuration), run an Agent in each logical host environment. Each Agent operates independently and can fail over respectively.

6.1.2 Combined Database Servers

This section gives an overview of the combined database servers and the configuration when you operate Agent for Microsoft SQL Server on a combined database server.

6.1.2.1 Combined Database Servers Overview

A combined database server allows tables that span multiple nodes to be divided into rows, to create a distributed partition view. This functionality allows linking and operation of node groups, to support large-scale Web sites and corporate data processing.

6.1.2.2 Configuring Agent for Microsoft SQL Server on a Combined Database Server

When Agent for Microsoft SQL Server monitors combined database server systems, it is run on each node of the combined database server. When running Agent for Microsoft SQL Server on a combined database server, use a configuration such as the one depicted in Figure 6.3.

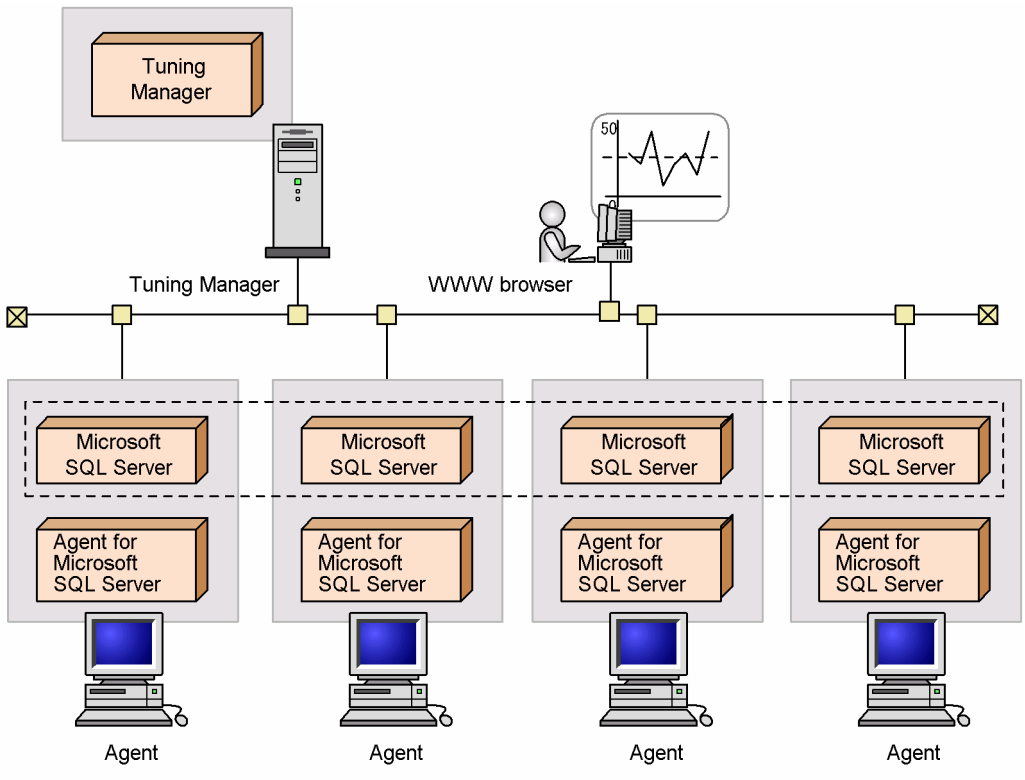


Figure 6.3 Configuring Agent for Microsoft SQL Server on a Combined Database Server

Microsoft SQL Server, which has a unique instance name, operates on each node. The Microsoft SQL Server instance on each node is monitored. As with stand-alone nodes, you need to set up Agent for Microsoft SQL Server on each node, and configure it so that the instance of Microsoft SQL Server on each node is monitored. Do not perform registration in the cluster software.

Note: When running Agent for Microsoft SQL Server on a combined database server to monitor the combined database server, you perform operations in the same way as for a system with many stand-alone nodes: Operate the system in the same way as a non-cluster system.

6.2 About Failover

When a failure occurs in the executing host, the system performs a failover and switches the processes to the standby host.

This section describes the process flow to be used when a failure occurs in Tuning Manager or in an Agent.

6.2.1 Failover When a Failure Occurs in Tuning Manager

The following figure shows the processing when a failover occurs for an executing Tuning Manager.

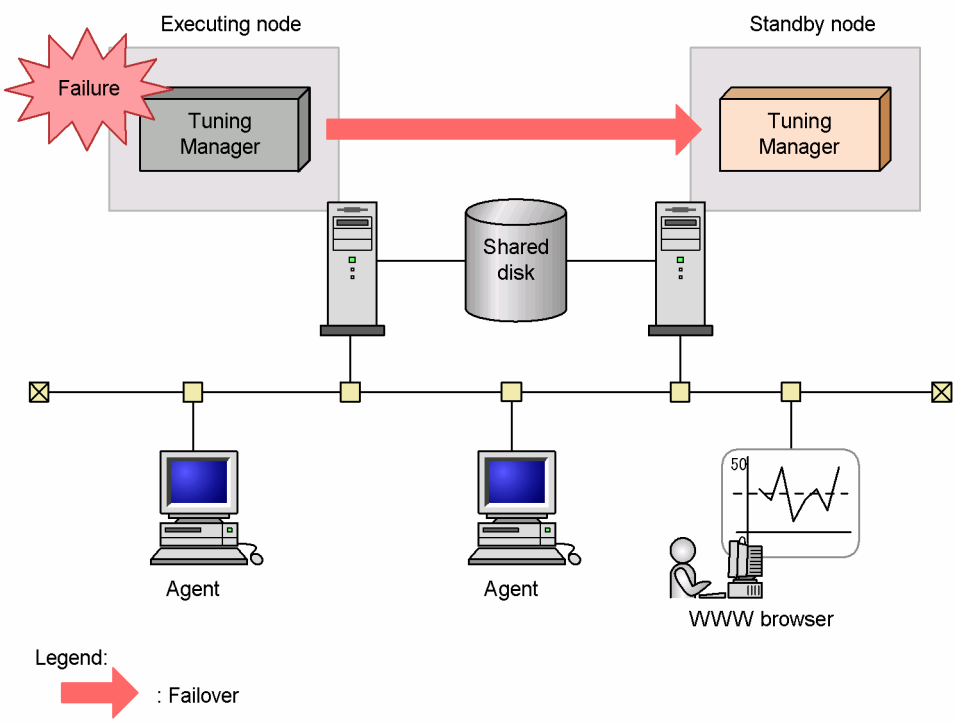


Figure 6.4 Processing When a Failover Occurs for Tuning Manager

The flow of system processing is as follows:

1. When the failover occurs, Tuning Manager is stopped in a forced termination.
2. The processing of the Tuning Manager service is taken over by the standby node.
3. The Tuning Manager in the standby node starts.

6.2.1.1 Agent Operates Normally During Failover of Tuning Manager

When a failover for an executing Tuning Manager occurs, no particular task needs to be performed for the Agent. During the failover of Tuning Manager, the Agent collects performance data without interruption.

6.2.1.2 How Tuning Manager Shutdown Affects Running Agents

A Tuning Manager shutdown affects running Agents because a Tuning Manager centrally administers the information about Agents operating on every node. The following shows the effects on Agents and what corrective actions to take when Tuning Manager shuts down:

Effects

- Agents continue to collect performance data.
- Attempting to shut down an Agent takes time because the shutdown cannot be reported to Tuning Manager.

Corrective Action

Start Tuning Manager. You can continue to operate running Agents without change.

In addition to shutdowns caused by failures or errors, you may occasionally need to shut down Tuning Manager for tasks such as maintenance or system configuration changes. We recommend that you carry out such maintenance work when it will have a minimum effect on operations.

6.2.2 Failover When a Failure Occurs in an Agent

When an error occurs on an Agent, an Agent fails over to continue monitoring the performance. The following figure shows the processing when a failover occurs on an Agent.

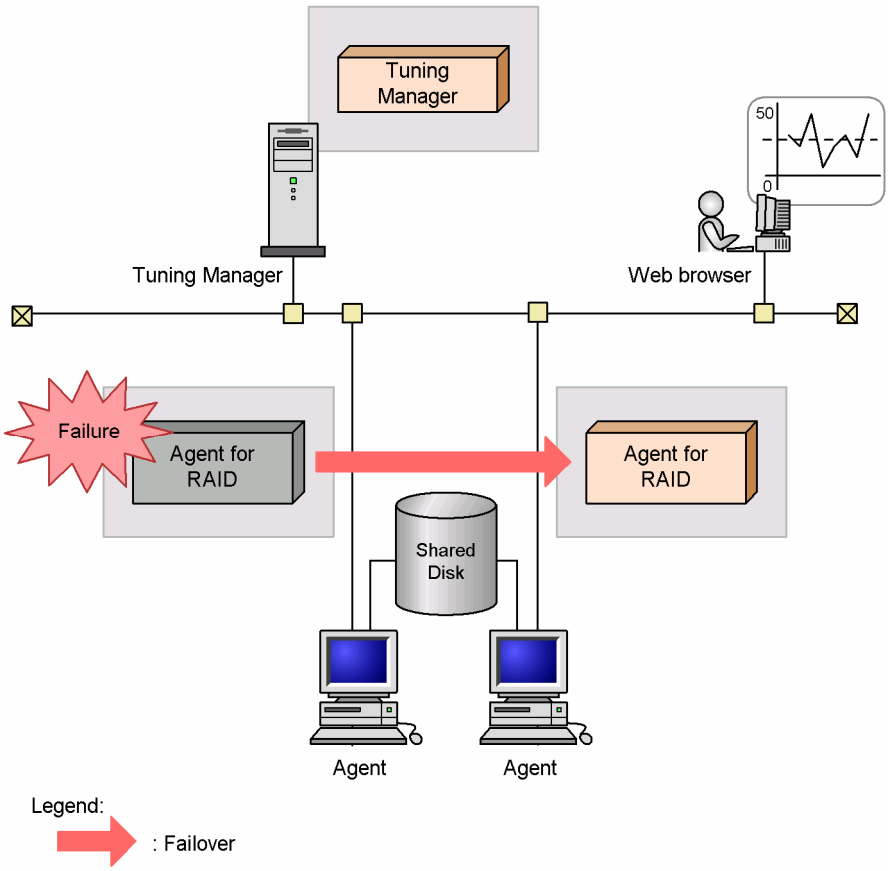


Figure 6.5 Processing When a Failover Occurs for an Agent

During a failover of an Agent, if you attempt to operate that Agent by using Performance Reporter, the message `There was no answer (-6)` is displayed. Wait for a while until failover finishes.

After failover occurs on an Agent, a Performance Reporter operation accesses the Agent that has been started at the failover target node.

6.2.3 Failover in a Mirroring Configuration

The following explains the processing for monitoring databases configured with the mirroring functionality for Microsoft SQL Server 2005, and when failover occurs for databases in a mirroring configuration.

6.2.3.1 Monitoring Databases in a Mirroring Configuration

For databases configured with mirroring functionality, Agent for Microsoft SQL Server collects information for non-mirrored databases (neither those configured for mirroring nor the principal database).

The following gives an overview of monitoring for databases configured with mirroring functionality.

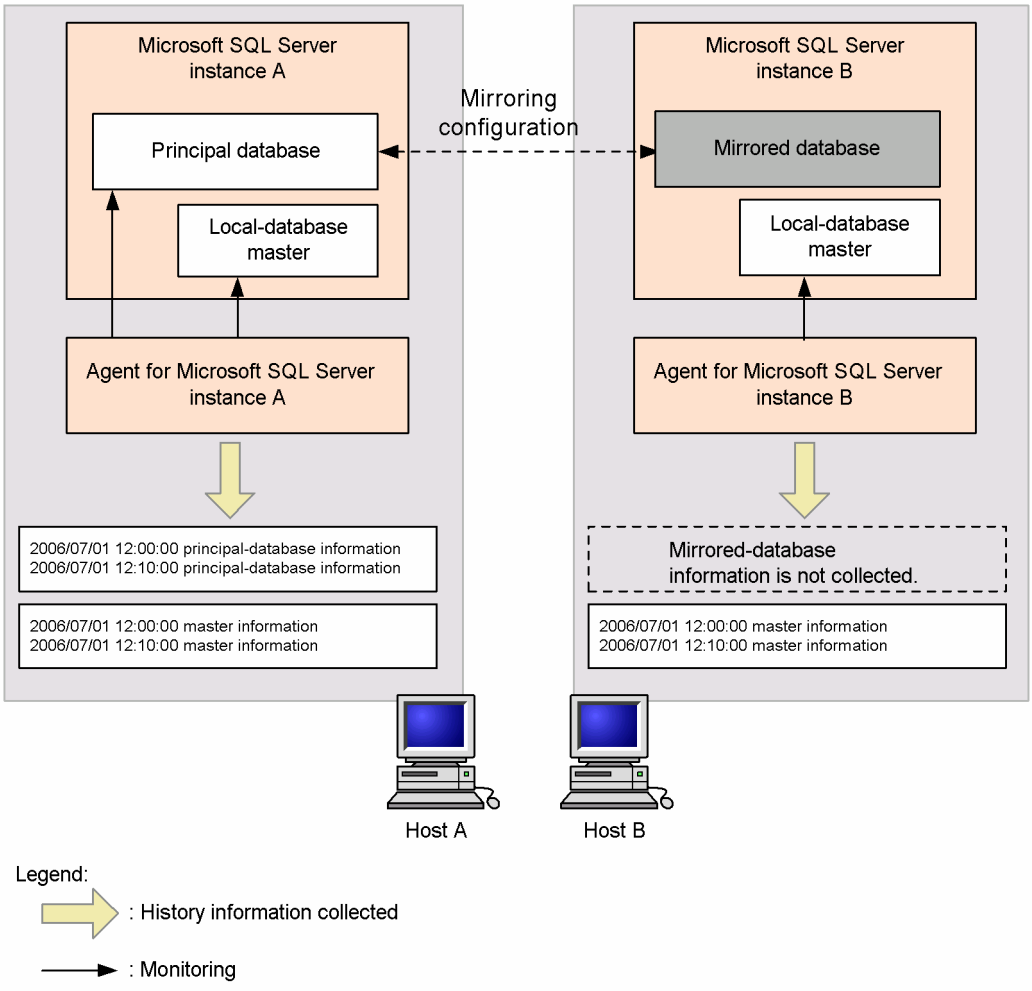


Figure 6.6 Monitoring for Mirroring Configurations

6.2.3.2 Monitoring during Failover in a Mirroring Configuration

Figure 6.7 shows the processing for Agent for Microsoft SQL Server when failover occurs for a database configured for mirroring.

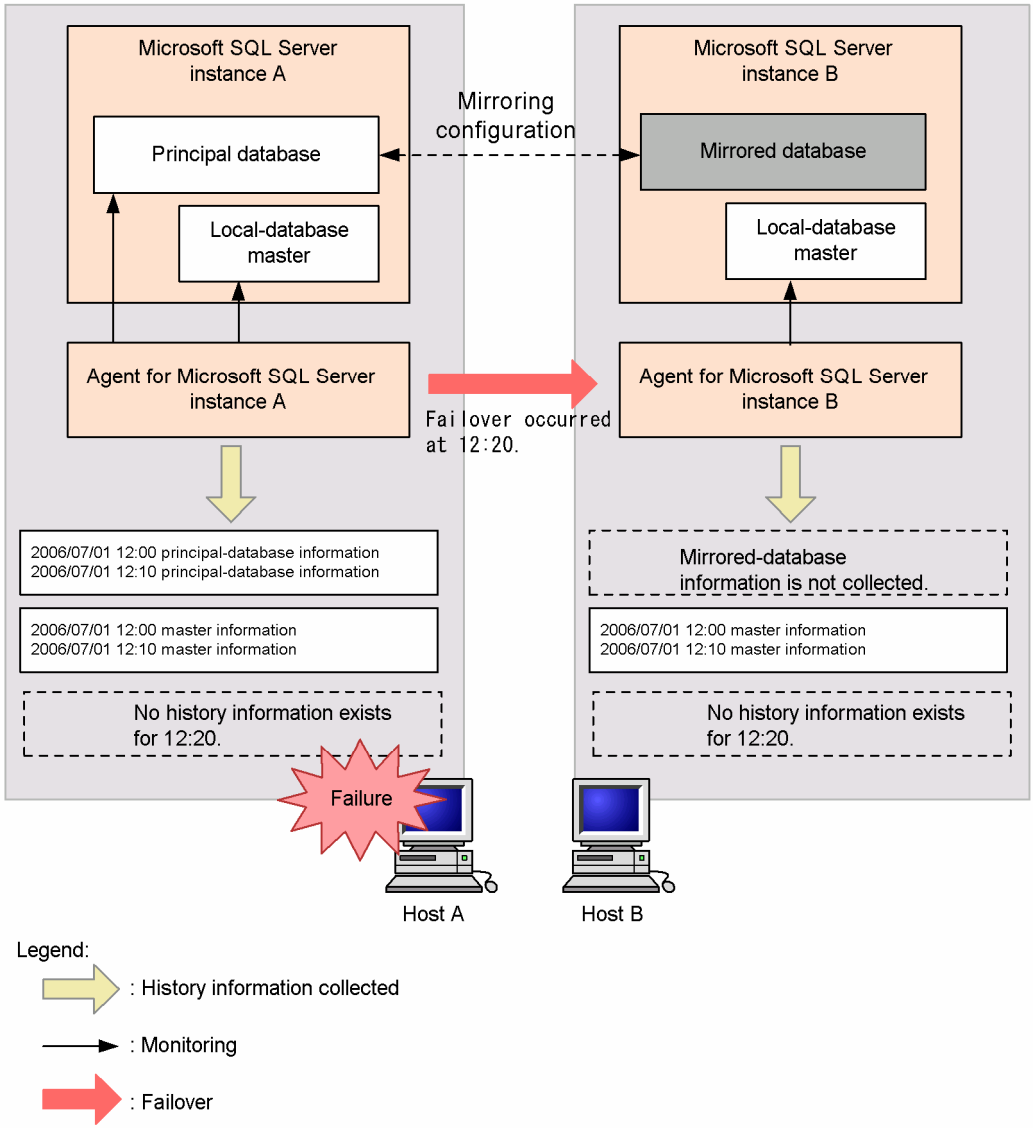


Figure 6.7 Monitoring during Failover

When failover occurs while information for a Microsoft SQL Server instance with databases configured for mirroring is collected, Agent for Microsoft SQL Server cannot collect information for each database instance (when failover occurs).

In this case, the KAVF21812-E message is output to the agent log, but no operational problem exists.

6.2.3.3 Monitoring after Failover in a Mirroring Configuration

Figure 6.8 shows the processing for Agent for Microsoft SQL Server once failover for a database configured for mirroring is completed.

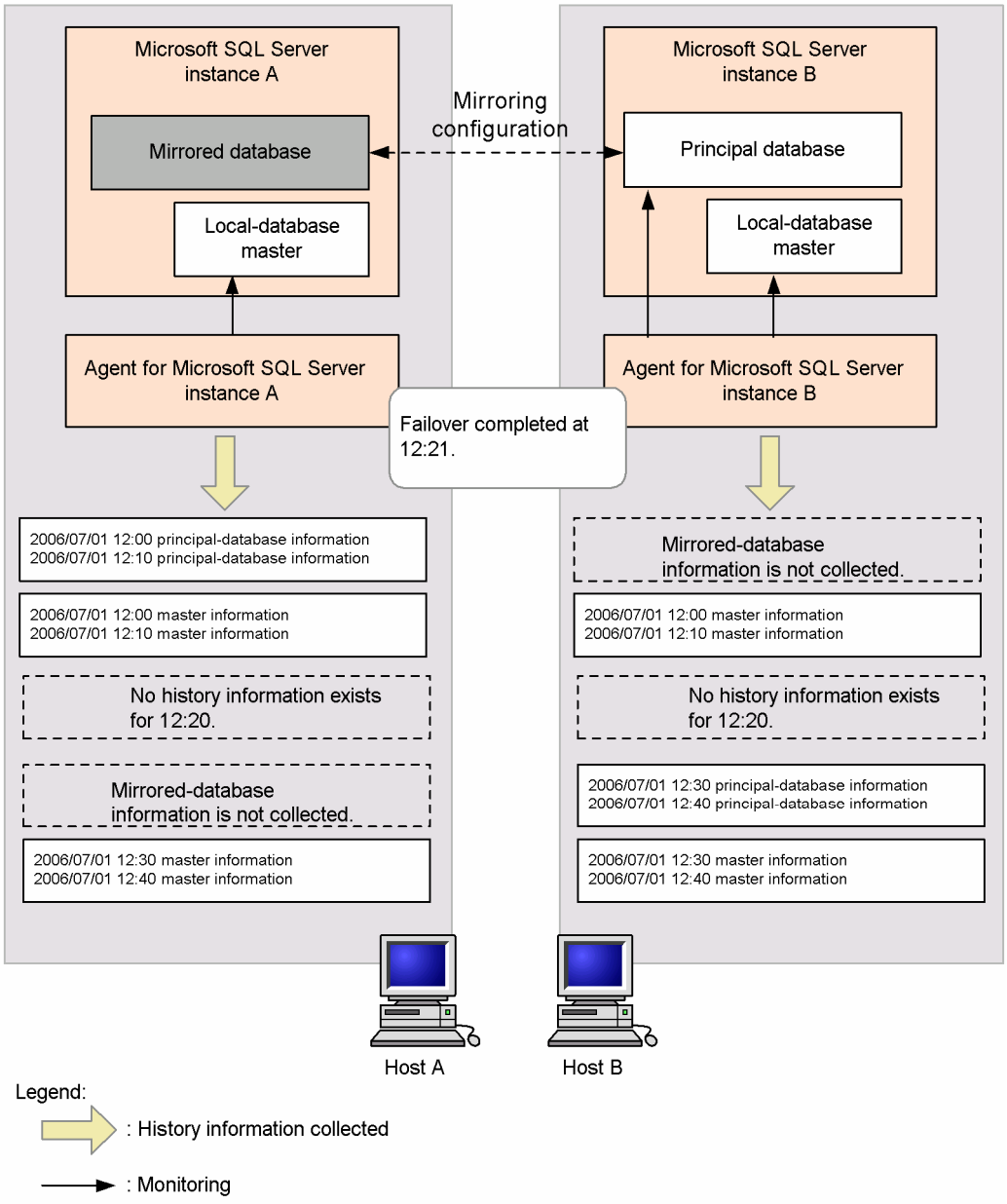


Figure 6.8 Monitoring after Failover

Once failover is complete, the principal database is transferred to host B, and the mirrored database is transferred to host A.

After failover, the Agent for Microsoft SQL Server instance (Agent for Microsoft SQL Server instance A in Figure 6.8) with the database transferred to the mirrored database collects information for non-mirrored databases.

Note that the instance (Agent for Microsoft SQL Server instance B in Figure 6.8) with the database transferred to the principal database collects information for the principal database and databases not configured for mirroring.

6.3 Operations in a Cluster System

6.3.1 Tuning Manager Service Names During Cluster System Operation

The Agent product and services in a logical host environment use the services (in Windows systems) and processes (in UNIX systems) shown in the tables below. In Table 6.1 and Table 6.2, *INST* indicates an instance name, whereas *LHOST* indicates a logical host name.

Table 6.1 Windows Service Names on a Physical Host and a Logical Host (Windows)

Tuning Manager Series Service Name	Windows Service Name on a Physical Host	Windows Service Name on a Logical Host
Name Server	PFM - Name Server	PFM - Name Server [*LHOST*]
Master Manager	PFM - Master Manager	PFM - Master Manager [*LHOST*]
Master Store	PFM - Master Store	PFM - Master Store [*LHOST*]
View Server	PFM - View Server	PFM - View Server [*LHOST*]
Correlator	PFM - Correlator	PFM - Correlator [*LHOST*]
Trap Generator	PFM - Trap Generator	PFM - Trap Generator [*LHOST*]
Action Handler	PFM - Action Handler	PFM - Action Handler [*LHOST*]
Agent Collector	Agent Collector for *xxxx instance-name* (see ***Note***)	Agent Collector for *xxxx INST* [*LHOST*] (see ***Note***)
Agent Store	Agent Store for *xxxx instance-name* (see ***Note***)	Agent Store for *xxxx INST* [*LHOST*] (see ***Note***)

Note: *xxxx* indicates the monitoring target of each Agent.

Table 6.2 Process Names on a Physical Host and a Logical Host (UNIX)

Tuning Manager Series Service Name	Process Name on a Physical Host	Process Name on a Logical Host
Name Server	jpcnsvr	jpcnsvr *LHOST*
Master Manager	jpcmm	jpcmm *LHOST*
Master Store	mgr/jpcsto	mgr/jpcsto *LHOST*
View Server	jpcvsvr	jpcvsvr *LHOST*
Correlator	jpcep	jpcep *LHOST*
Trap Generator	jpctrap	jpctrap *LHOST*
Action Handler	jpcah	jpcah *LHOST*
Agent Collector	jpcagt*X_instance-name* (see ***Note***)	jpcagt*X_INST LHOST* (see ***Note***)

Tuning Manager Series Service Name	Process Name on a Physical Host	Process Name on a Logical Host
Agent Store	agt*X*/jpcsto_*instance-name* (see ***Note***)	agt*X*/jpcsto_*INST LHOST* (see ***Note***)

Note: *X* indicates the product ID of each Agent.

6.3.2 Starting and Stopping in a Cluster System

This section describes how to start and stop the Tuning Manager series programs running on a logical host in a cluster system.

6.3.2.1 Starting Tuning Manager Programs in Logical Host Operation

This section describes how to start the Tuning Manager series programs in logical host use in a cluster system.

- **Starting sequence for Tuning Manager series programs**: The starting sequence for the Tuning Manager series programs in a cluster system is the same as the sequence for starting a non-cluster system. For details, see Chapter 1.
- **Starting the Tuning Manager series programs in logical host operation:** In order to start a Tuning Manager series program during use of logical host, use the cluster software to specify that the Tuning Manager series program is to start automatically at the time the logical host in which you have registered the Tuning Manager series program is started.

Note: If you execute the `jpcstart` command directly without using the cluster software to start the Tuning Manager series programs, the cluster software regards the status of the Tuning Manager series programs as different from the actual status and the cluster software may misinterpret this situation as a failure.

6.3.2.2 Stopping Tuning Manager Programs in Logical Host Operation

This section describes how to stop the Tuning Manager series programs in logical host operation in a cluster system.

- **Stopping sequence for Tuning Manager series programs**: The stopping sequence for the Tuning Manager series programs in a cluster system is the same as the sequence for stopping a non-cluster system. For details, see section 1.3.
- **Stopping the Tuning Manager series programs in logical host operation:** To stop the Tuning Manager series programs in logical host operation, use the cluster software so that the Tuning Manager series programs stop automatically when the logical host to which you have registered the Tuning Manager series programs is stopped.

Note: If you execute the `jpcstop` command directly without using the cluster software to stop the Tuning Manager series programs, the cluster software regards the status of the Tuning Manager series programs as different from the actual status and the cluster software may misinterpret this situation as a failure.

Note: If you want to stop the Tuning Manager series programs to make changes on the Tuning Manager series configuration without stopping other resources such as the shared disk or logical IP address, use the cluster software to stop only the Tuning Manager series programs. If the cluster software does not have a function that allows you to stop the Tuning Manager series programs only, temporarily stop performance monitoring by the Tuning Manager series and then manually stop the Tuning Manager series using the `jpcstop` command. To do this, you must set up a mechanism that allows you to stop performance monitoring in advance when you register the Tuning Manager series programs to the cluster system.

6.3.3 Backing up and Restoring in a Cluster System Environment

When you use the Tuning Manager series programs in logical host operations in a cluster system, you must back up the system periodically in case of a failure.

6.3.3.1 Backing up the Service Definition Information

Back up the service definition information periodically. Table 6.3 and

Table 6.4 show the definition information files to be backed up.

Note: Before you back up the definition information in this table, stop the Tuning Manager series program services on the local host.

Windows systems:

Table 6.3shows the definition information files to be backed up on a Windows platform.

Table 6.3 Definition Information Files to Be Backed Up (Windows)

Type	File Name	Description
Common	*installation-folder**.ini	Common settings file in the Tuning Manager series programs
	environment-directory\jp1pc*.ini	Common settings file in the Tuning Manager series programs
	environment-directory\jp1pc\bin\action*.ini	Settings file of the Action Handler service
	Installation-folder\bin\statsvr*.ini (see ***Note 1***)	Settings file of the Status Server service
Tuning Manager	*environment-directory*\jp1pc\mgr\clator*.ini	Settings file of the Correlator service
	environment-directory\jp1pc\mgr\manager*.ini	Settings file of the Master Manager service
	environment-directory\jp1pc\mgr\manager*.db	Database file of the Master Manager service
	environment-directory\jp1pc\mgr\store*.ini	Settings file of the Master Store service
	environment-directory\jp1pc\mgr\namesvr*.ini	Settings file of the Name Server service
	environment-directory\jp1pc\mgr\namesvr*.db	Database file of the Name Server service
	environment-directory\jp1pc\mgr\trapgen*.ini	Settings file of the Trap Generator service
	environment-directory\jp1pc\mgr\viewsvr*.ini	Settings file of the View Server service
	environment-directory\jp1pc\mgr\viewsvr\data*.lmk	User definition information file of the View Server service
	environment-directory\jp1pc\mgr\viewsvr\Reports*	Report definition information file of the View Server service
Agent	*environment-directory*\jp1pc*xxxx**instance-name*\agent*.ini (see ***Note 2*** and ***Note 3***)	Settings file of the Agent Collector service
	environment-directory\jp1pc*xxxx**instance-name*\store*.ini (see ***Note 2*** and ***Note 3***)	Settings file of the Agent Store service

Note 1: The settings file of the Status Server service only exists on the physical host even in logical host operation.

Note 2: *xxxx* indicates the service key of each Agent. For the service key of each Agent, see Appendix A.

Note 3: *instance-name* indicates a folder for operating in the instance environment. For an Agent that monitors an application program that can start a set of multiple services at the same host, the system creates a number of directories equal to the number of instances.

- **UNIX systems:** Table 6.4 shows the definition information files to be backed up on a UNIX platform.

Table 6.4 Definition Information Files to Be Backed Up (UNIX)

Type	File Name	Description
Common	/opt/jp1pc/*.ini	Common settings file in the Tuning Manager series programs
	environment-directory/jp1pc/*.ini	Common settings file in the Tuning Manager series programs
	environment-directory/jp1pc/bin/action/*.ini	Settings file of the Action Handler service
	/opt/jp1pc/bin/statsvr/*.ini (see ***Note 1***)	Settings file of the Status Server service
Tuning Manager	*environment-directory*/jp1pc/mgr/clator/*.ini	Settings file of the Correlator service
	environment-directory/jp1pc/mgr/manager/*.ini	Settings file of the Master Manager service
	environment-directory/jp1pc/mgr/manager/*.DB	Database file of the Master Manager service
	environment-directory/jp1pc/mgr/store/*.ini	Settings file of the Master Store service
	environment-directory/jp1pc/mgr/namesvr/*.ini	Settings file of the Name Server service
	environment-directory/jp1pc/mgr/namesvr/*.DB	Database file of the Name Server service
	environment-directory/jp1pc/mgr/trapgen/*.ini	Settings file of the Trap Generator service
	environment-directory/jp1pc/mgr/viewsvr/*.ini	Settings file of the View Server service
	environment-directory/jp1pc/mgr/viewsvr/data/*.lmk	User definition information file of the View Server service
	environment-directory/jp1pc/mgr/viewsvr/Reports/*	Report definition information file of the View Server service
Agent	*environment-directory*/jp1pc/*xxxx*/*instance-name*/agent/*.ini (see ***Note 2*** and ***Note 3***)	Settings file of the Agent Collector service
	environment-directory/jp1pc/*xxxx*/*instance-name*/store/*.ini (see ***Note 2*** and ***Note 3***)	Settings file of the Agent Store service

Note 1: The settings file of the Status Server service only exists on the physical host even in logical host operation.

Note 2: *xxxx* indicates the service key of each Agent. For the service key of each Agent, see Appendix A.

Note 3: These are the directories to be used in an instance environment operation. In an instance configuration, the system creates a number of directories equal to the number of instances.

6.3.3.2 Restoring the Service Definition Information

When you need to restore the service definition information, copy the definition information files you have saved in section 6.3.3.1 to the original directory.

6.3.4 Real-time Monitoring Using Alarms in a Cluster System

To notify users of a problem that has occurred in the monitored system, you need to set alarms. Note that, when a logical host operation is used in a cluster system, the alarm setting method differs from that for a non-cluster system.

Notes on the node on which actions are to be performed:

- In the alarm definition file, if `LOCAL` is set for the Action Handler label in the Action Definition Command subsection, an action is performed on the node where the Agent for monitoring alarms is running. For example, when the Agent is running on a logical host and an alarm occurs, an action will be executed on the executing node where the Agent is running.
- In the alarm definition file, if `LOCAL` is set for the Action Handler label in the Action Definition Command subsection, a command is executed on the node where the Tuning Manager series programs are running. Therefore, set up the environment so that commands can be executed in similar ways on both the executing and standby nodes.
- If the Action Handler service is running on a logical host, the current directory is as follows. *environment-directory* indicates the environment directory name specified by the `jpchasetup create` command.

  environment-directory`\jp1pc\bin\action`

For details on how to set alarms, see section 4.5.

6.3.5 Cluster System Considerations

Note the following when you operate Tuning Manager series programs in a cluster system.

- **Detecting failover occurrences:** It is difficult for the Tuning Manager series programs to detect a failover of a node running an Agent. To detect failover occurrences, use an administration tool of the cluster software, monitor SNMP traps issued by the cluster software, or monitor messages saved in log files.
- **Starting and stopping Tuning Manager series programs in logical host operation:** The Tuning Manager series programs that are on a logical host and are registered in the cluster software must be started or stopped from the cluster software. If you execute the `jpcstart` command or the `jpcstop` command to directly start or stop the Tuning Manager series programs without using the cluster software, the cluster software regards the status of Tuning Manager series programs as different from the actual status and the cluster software may misinterpret this situation as a failure.

- **Working with networks:** When the Tuning Manager series is run on a physical host, the physical IP address corresponding to the physical host name (the host name displayed when the `hostname` command is executed on a Windows system, or the `uname -n` command is executed on a UNIX system) needs to be set.
- **When running Agent for Oracle and Agent for Microsoft SQL Server in a logical host operation:** The performance data Agent for Oracle and Agent for Microsoft SQL Server collect includes records that contain fields related to the host name. When Agent for Oracle and Agent for Microsoft SQL Server are running on a logical host operation, the host name field may contain physical host names or logical host names. For Agent for Oracle, the physical host name is stored in the `Host` field of the Instance (`PD_PDI`) record as the host name of the connected instance. For Agent for Microsoft SQL Server, the physical or logical host name is stored in the `Host` field of the Process Detail (`PD_PDET`) record as the host name of the process executing on Microsoft SQL Server.
- **Reviewing dependencies between the Agent Store service and Microsoft SQL Server resources:** When operating Agent for Microsoft SQL Server in a cluster system, if dependencies are set between the Agent Store service and Microsoft SQL Server resources, Agent for Microsoft SQL Server stops and then Microsoft SQL Server stops. Therefore, if a failover occurs when records are being collected, it takes time to stop Microsoft SQL Server because the stop processing is performed after the record collection processing ends. If you want to consider the time required to stop Microsoft SQL Server, cancel the dependencies between Agent for Microsoft SQL Server and Microsoft SQL Server. By doing this, Microsoft SQL server will sometimes stop before Agent for Microsoft SQL Server. At this time, the records to be performed are not collected.

  Note that it might still take time to stop Microsoft SQL Server even if you canceled the dependencies. In this case, set the connection timeout value and use the pending functionality to adjust the timing. For details on how to set the connection timeout value and the pending functionality, see the *HiCommand Tuning Manager Installation Guide*.
- Log Output

  In a cluster configuration, when the dependency setting between Microsoft SQL Server and Agent for Microsoft SQL Server is cleared, and Microsoft SQL Server stops before Agent for Microsoft SQL Server, a record collection error message indicating that Microsoft SQL Server does not exist, such as the following example log information, is output to the log file.

  Even when this error message is output, no operational problem exists, so failover can be continued.

Example Log Information

jpclog (common message log) file (only when Agent is running)

```
2005/10/25 18:22:25 jpcagtq 00002140 00002124 PWBSqlCollector 4241
   KAVF21400-W A connection to SQL Server cannot be established.
```

`agtqerr01.log` (agent log) file (output for each collection interval for each record)

```
2005/10/25 18:24:23 jpcagtq 00002140 00002124 Sqlservado.cpp 0267
   E Error Code = 0x80004005, Error Description = [DBNETLIB]
   [ConnectionOpen (Connect()).]SQL Server either does not exist, or is
refusing access.
```

Chapter 7 Collecting Log Information

This chapter describes the setup procedures for Agent for Platform that enables you to collect log information and use Performance Reporter to monitor the collected log information.

- Overview of Collecting Log Information (see section 7.1)
- Performing the Setup Procedure for Collecting Log Information (see section 7.2)

7.1 Overview of Collecting Log Information

Agent for Platform can collect the following log information:

- UNIX log information
- Log information for applications running on UNIX
- Log information for databases running on UNIX

Notes:

- Log information can be collected from text-format incremental log files. Only single-byte characters can be collected.
- In Linux, log information cannot be collected because the Logged Messages (PL_MESS) record cannot be used.

By using a command, if specific log information is set (such as an error message) as the threshold for an alarm, you can notify users when the specified message is output.

Agent for Platform's log information collection program collects log information from log files according to the log file names and filter conditions set in an event file. The collected log information is collected by the Agent Collector service and managed as a Logged Messages (PL_MESS) record, which is a record of the PL record type. As with other records, the Logged Messages (PL_MESS) record can be used in displayed reports and can be monitored by alarms. Figure 7.1 provides an overview of log information monitoring.

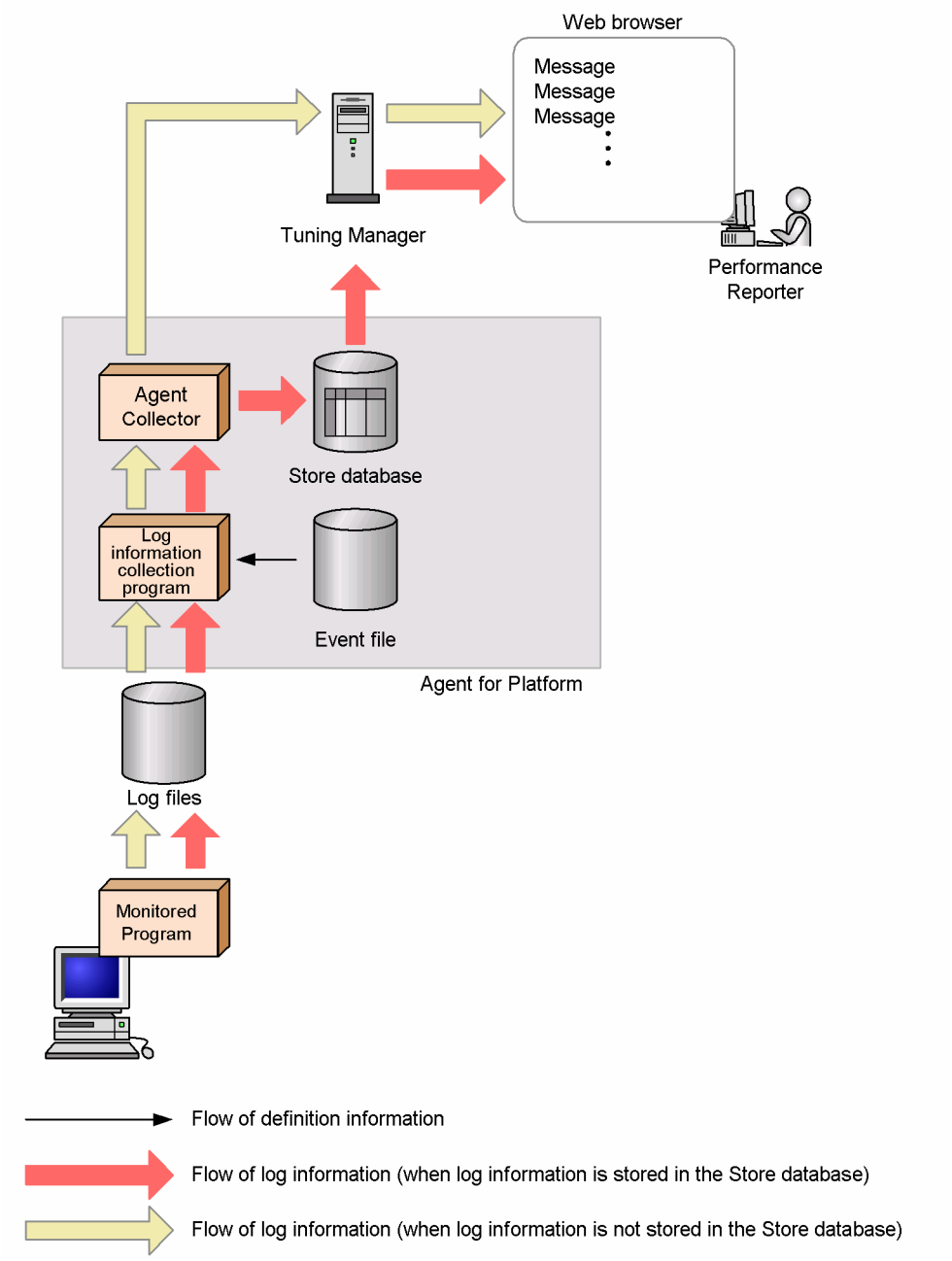


Figure 7.1 Overview of Log Information Monitoring

7.2 Performing the Setup Procedure for Collecting Log Information

Perform the following steps to set up log information collection by Agent for Platform and log information monitoring by Performance Reporter:

1. Set up the event file.
2. Use the Performance Reporter command to set up the system so that performance data for Logged Messages (PL_MESS) records is stored in the Store database.

   This step is necessary in order to use Performance Reporter to display historical reports.
3. Restart Agent for Platform.

Proceed to section 7.2.1 for instructions.

7.2.1 Setting Up the Event File

To collect log information, you must first set up the event file. The event file specifies such information as the name of the log file in which the log information to be collected is output and collection filtering conditions.

You can use the following event file only:

```
/opt/jp1pc/agtu/agent/evfile
```

The event file consists exclusively of comment lines (lines that begin with #). To set parameters, edit the event file directly. You can also create a copy of the file in the same directory, and then edit the copy.

To set up the event file:

1. Use a text editor to open the event file.
2. Add the following parameters to the event file:

   logfile=*file-name*

   [id=*identifier*]

   [regexp=*filter-condition*]

   The parameters are explained below:

   logfile=*file-name*

   Specify the full pathname of the log file in which the log information to be collected is output. Specify the log file name using alphanumeric characters. For details about the number of bytes that can be specified, see the operating system documentation.

   id=*identifier*

Specify the character string to be displayed as an identifier for the log information. You can specify up to 1,023 single-byte alphanumeric characters and symbols (excluding asterisks (*)). The value specified in this parameter becomes the character string that follows the jpcagtu character string in the Message Text (MESSAGE_TEXT) field of the Logged Messages (PL_MESS) record. If this parameter is omitted, the log file name (not including the directory name) is displayed.

regexp=*filter-condition*

Specify a filtering condition for the log information to be collected in the Logged Messages (PL_MESS) record. You can specify up to 2,040 single-byte alphanumeric characters and symbols (including line feed characters). To define a conditional expression, use an expanded normal expression. For details about expanded normal expressions, see the operating system documentation. If you specify multiple expressions, they are interpreted as being connected by OR statements.

You can also use the Portable Operating System Interface for UNIX (POSIX) to specify filter conditions. If you use the /i suffix, the system stores log information in the Logged Messages (PL_MESS) record without distinguishing between upper case and lower case characters.

Notes:

- Parameter names are not case-sensitive.
- When you add a parameter, you must not specify any blanks or tab characters before or after the equals sign.
- To write a comment, insert a line that begins with a hash mark (#), and write a comment after the hash mark.
- When you use regular expressions and specify the dot asterisk (.*) combination as the filter condition, depending on the OS performance, it might take time to collect records. In this case, replace the regular expression format with other characters, such as the caret asterisk (^*) combination.

3. To collect information from more than one log file, specify the parameters for each log file.
4. Save the event file with file name evfile, which is the default file name.

Note: To revert the setup information in the evfile file to the original contents at the time of system installation, copy evfile.model (the model file for evfile) into evfile.

7.2.1.1 Example of Specifying an Event File

To collect log information for the Sample Application that was output to `/opt/sampleapp/log`, and then store, in the Logged Messages (PL_MESS) record, only log information whose status is `warning`, `error`, or `fatal` (not case-sensitive), specify the event file as follows:'

```
logfile=/opt/sampleapp/log
id=SAMPLE
regexp=warning/i
regexp=error/i
regexp=fatal/i
```

7.2.2 Setting Up Performance Reporter

To display historical reports, you must set up Performance Reporter so that performance data in the Logged Messages (PL_MESS) records is stored in the Store database.

For details about setting up Performance Reporter, see Chapter 3.

7.2.3 Notes About Collecting Log Information

This section provides notes about using the Logged Messages (PL_MESS) record to monitor messages.

- A maximum of 511 bytes of characters stored in the Logged Messages (PL_MESS) record can be monitored using a conditional expression in the alarm definition. However, because the character string stored in a Logged Messages (PL_MESS) record includes header information such as an identifier (id), the actual message length that can be monitored is 511 bytes minus the length of the header information.
- To monitor a character string whose length exceeds 511 bytes, set the character string as a filter condition in the event file settings for Agent for Platform. In such a case, also set a desired identifier (id) for the message. By setting an identifier (id) in the conditional expression in the alarm definition, you can monitor messages containing the character string set as the filter condition.

  For example, to monitor messages containing the character string `ABC`, set `Console` as the identifier (id) in the event file settings for Agent for Platform and `ABC` as the filter condition:

  ```
  logfile=/tmp/console_log
  id=Console
  regexp=ABC
  ```

Then, in the alarm definition file, set `Console` as a threshold for the abnormal or warning value.

Based on these settings, a message that contains the character string `ABC` will be assigned `Console` as the id in the message header. An alarm is then generated for messages that contain the character string `Console`.

Chapter 8 Collecting Workgroup Information

This chapter describes the setup procedures that enable you to use Agent for Platform to collect workgroup information and to use Performance Reporter to monitor the collected workgroup information.

- Overview of Collecting Workgroup Information (see section 8.1)
- Setup Procedure for Collecting Workgroup Information (see section 8.2)

8.1 Overview of Collecting Workgroup Information

If multiple users are using UNIX system resources or operating as UNIX groups, Agent for Platform enables you to set up selected UNIX users and UNIX groups as *workgroups*. You can then collect information about the workgroups. You can set up workgroups to consist of the following:

- UNIX users
- UNIX groups
- Programs being executed by a process

The workgroup information collection program of Agent for Platform summarizes the performance data that relates to workgroups in Process Detail (PD) records. The program summarizes the data about a workgroup on the basis of the workgroup name and other information that is set in the workgroup file. The summarized performance data is managed as a Workgroup Summary (PI_WGRP) record. As with other records, Workgroup Summary (`PI_WGRP`) records can be used in displayed reports and can be monitored by alarms. Figure 8.1 shows the flow of data in workgroup information monitoring.

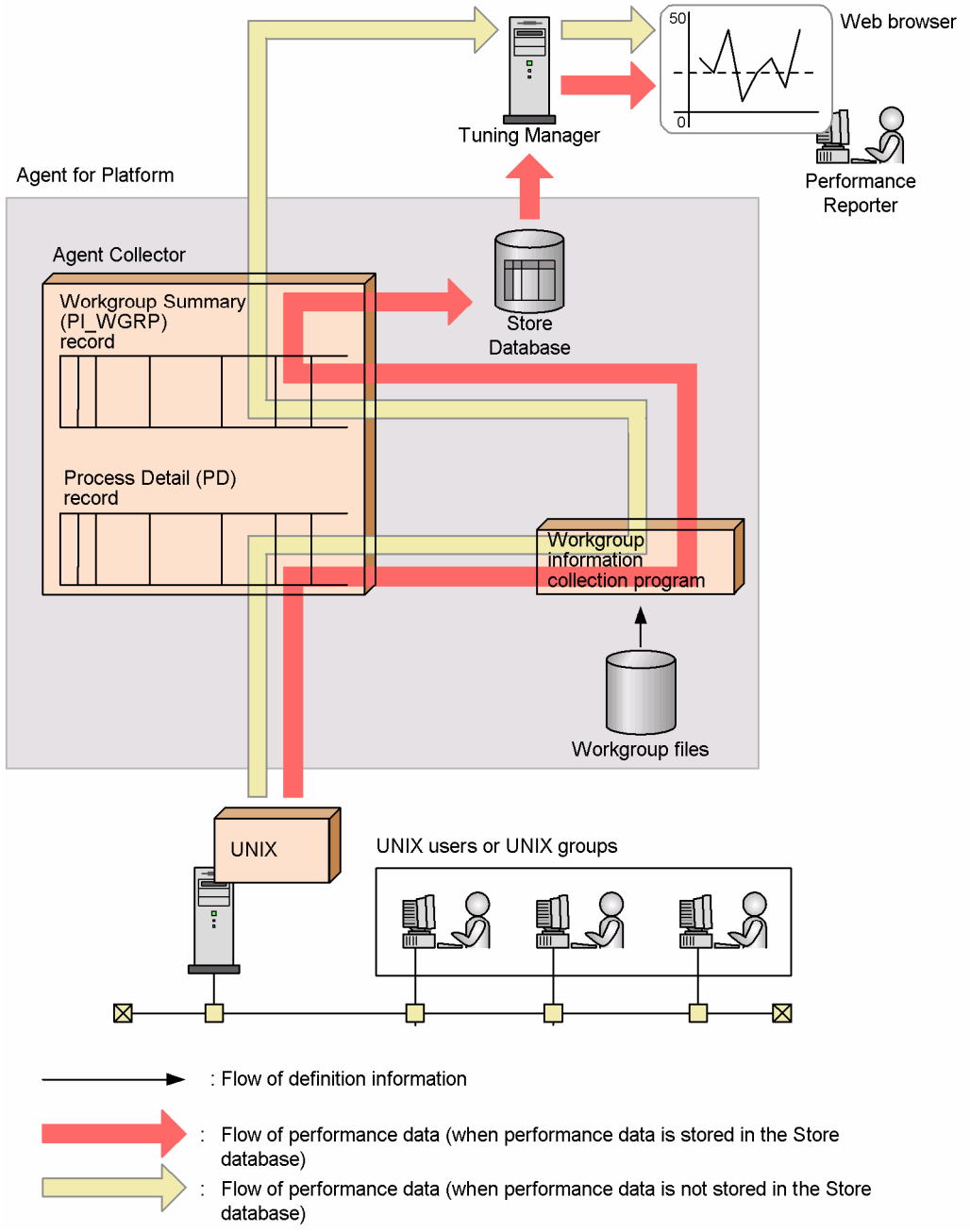


Figure 8.1 Flow of Data in Workgroup Information Monitoring

8.2 Setup Procedure for Collecting Workgroup Information

You perform the following steps to set up a system that enables Agent for Platform to collect workgroup information, and enables Performance Reporter to monitor workgroup information.

To set up the system:

1. Set up the workgroup file.
2. Set up Performance Reporter so that performance data for Workgroup Summary (PI_WGRP) records is stored to the Store database.

   This step is necessary in order to use Performance Reporter to display historical reports.
3. Restart Agent for Platform.

8.2.1 Setting Up the Workgroup File

To collect workgroup information, you must first set up the workgroup file. The workgroup file sets information such as the name for a workgroup.

You can use only one workgroup file; its name is as follows:

```
/opt/jp1pc/agtu/agent/wgfile
```

This workgroup file consists exclusively of comment lines (lines that begin with #). To set parameters, you can edit this workgroup file directly, or edit a copy of the workgroup file.

To set up the workgroup file:

1. Use a text editor to open the workgroup file.
2. Add the following parameters to the workgroup file:

   workgroup=*workgroup-name*

   [users=*UNIX-user-names*]

   [groups=*UNIX-group-names*]

   [programs=*program-names*]

   The parameters are described below:

   workgroup=*workgroup-name*

   Specify a name for the workgroup for which information is to be collected. You can specify the workgroup name by using up to 2,037 single-byte alphanumeric characters (including line feed characters); however, only the first 29 bytes are stored in the Store database. You must specify a workgroup name.

   users=*UNIX-user-names*

Specify the names of UNIX users that are to be set as the workgroup for which information is to be collected. Each UNIX user name can be up to 2,041 single-byte alphanumeric characters (including line feed characters); however, only the first 29 bytes are stored in the Store database, and the remaining characters become >.

You specify multiple UNIX user names by using at least one comma or space as the delimiter between the individual names.

In this parameter, specify the value to be stored in the Real User (REAL_USER_NAME) field of the Process Detail (PD) record.

`groups=`*`UNIX-group-names`*

Specify the names of UNIX groups that are to be set as the workgroup for which information is to be collected. You specify each UNIX group with the group name, not the ID. Each UNIX group name can be up to 2,040 single-byte alphanumeric characters (including line feed characters); however, only the first 29 bytes are stored in the Store database, and the remaining characters become >.

You specify multiple UNIX group names by using at least one comma or space as the delimiter between the individual names.

In this parameter, specify the value to be stored in the Real Group (REAL_GROUP_NAME) field of the Process Detail (PD) record.

`programs=`*`program-names`*

Specify the names of programs executed by a process that are to be set as the workgroup for which information is to be collected. Each program name can be up to 2,038 single-byte alphanumeric characters (including line feed characters); however, only the first 29 bytes are stored in the Store database, and the remaining characters become >.

You specify multiple program names by using at least one comma or space as the delimiter between the individual names.

In this parameter, specify the value to be stored in the Program (PROGRAM_NAME) field of the Process Detail (PD) record.

Notes:

- Parameter names are not case-sensitive.
- If a single parameter requires continuation lines, specify a comma (`,`) at the end of each line that is being continued.
- You must specify the workgroup parameter first. You can specify the other parameters in any sequence.
- You can use a regular expression to specify a parameter. For details about regular expressions, see the operating system documentation. For examples of using regular expressions, see Example 2, in section 8.2.1.1.
- If you use regular expressions and specify the dot asterisk (`.*`) combination as the filter condition, depending on the OS performance, it might take time to collect records. In such a case, replace the regular expression format with other characters, such as the caret asterisk (`^*`) combination.

- When you add a parameter, you must not specify any blanks or tab characters before or after the equals sign.
- Each inserted comment line must begin with a hash mark (#).

3. To collect information about multiple workgroups, specify separate sets of parameters for each workgroup for which information is to be collected.
4. Save the workgroup file with the default file name `wgfile`.

Note: To revert the setup information in the `wgfile` to the original contents at the time of system installation, copy `wgfile.model` (the model file of `wgfile`) into `wgfile`.

8.2.1.1 Examples of a Workgroup File

The following examples illustrate how to specify a workgroup file.

Example 1

This example specifies the following information:

- Workgroup name: `sysadmin`
- UNIX user name: `root`
- UNIX group names: `sys, user, system`
- Program names: `netscape, turkey`

```
workgroup=sysadmin
groups=sys,user,system
users=root
programs=netscape,turkey
```

Example 2

This example uses regular expressions to specify the user name, group name, and program names.

When you use a regular expression, you enclose each parameter and its value in braces ({ and }). You can also specify multiple regular expressions by separating them with a comma (,).

The example uses regular expressions to specify the following information:

- Workgroup name: `sysadmin`
- UNIX user name: `*adm??`
- UNIX group name: `adm*s`
- Program names: `jpcagt?, g*rd, [ef]grep`

```
workgroup=sysadmin
regexp={programs=jpcagt.,g.*rd,[ef]grep},{users=adm..},{groups=adm.*s}
```

Asterisks (*) and question marks (?) are wildcard characters, not regular expressions.

For details about regexp, see the operating system documentation.

Example 3

This example uses a regular expression to specify the following information:

- Workgroup name: `perfMonTools`
- Program names: `jpcagt?` (not case-sensitive), `*perfmon, top, monitor, vmstat, iostat, sar`

```
workgroup=perfMonTools
regexp={programs=jpcagt./i,
perfmon}
programs=top,monitor,vmstat,iostat,sar
```

Asterisks (`*`) and question marks (`?`) are wildcard characters, not regular expressions.

For details about regexp, see the operating system documentation.

8.2.2 Setting Up Performance Reporter

To display historical reports, you must set up Performance Reporter so that information in Workgroup Summary (PI_WGRP) records is collected.

For details about setting up Performance Reporter, see Chapter 3.

Chapter 9 Checking Service Status Using the Status Management Function

This chapter describes the status management function, which provides an accurate indication of the status of Collection Manager and Agent services during the Tuning Manager and Agent startup and shutdown or when the Tuning Manager and Agent have stopped due to a fault.

You can use this function when the version of the Tuning Manager series program you are using supports the status management function and it is enabled.

- Overview (see section 9.1)
- Setup (see section 9.2)
- Checking the Service Status (see section 9.3)
- Status Management During Cluster System Operation (see section 9.4)
- Status Management Function Errors (see section 9.5)

9.1 Overview

The status management function manages the status of Collection Manager and Agent services that operate as part of Tuning Manager and the Agent. The status management function allows the system administrator to make accurate assessments of the startup and shutdown status of all hosts, thereby enabling faster responses to faults as they occur.

Figure 9.1 provides an overview of checking the service status by using the status management function.

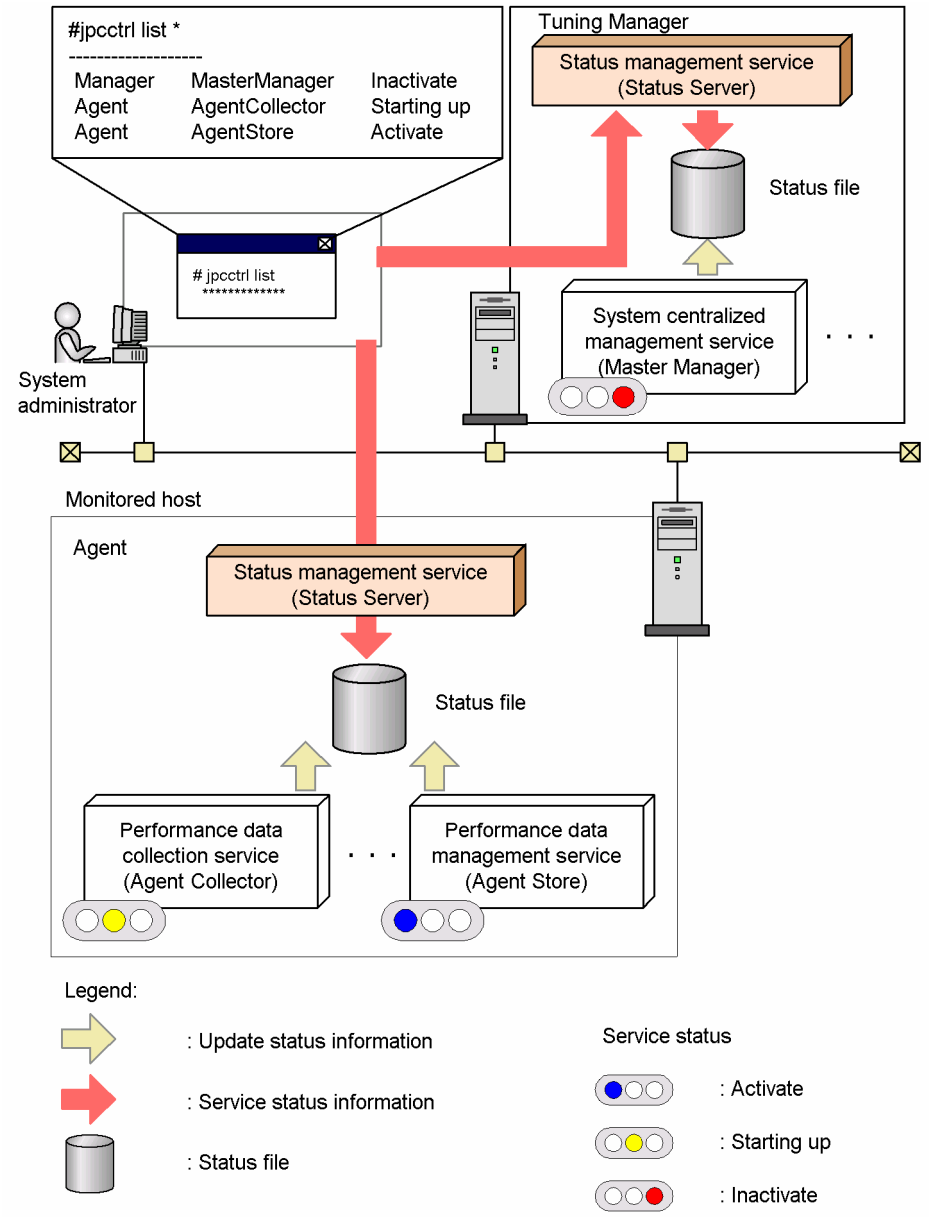


Figure 9.1 Overview of Checking the Service Status by Using the Status Management Function

When the status management function is available, the Collection Manager and Agent services register their own statuses in status files. The system administrator then uses the status management service (Status Server service) to assess statuses by reading the status files.

Note: When the status management function is unavailable, the service status is determined on the basis of whether Tuning Manager has sent a response to the Agent. Tuning Manager also coordinates network information such as Agent IP addresses and port numbers. Thus, service status cannot be checked if communication with Tuning Manager is disabled (for a reason such as a fault or the service is starting up) or if the Agent is operating in stand-alone mode.

When the status management function is unavailable, the `jpcctrl list` command can be used to check the status of services that are operating or that have stopped. However, it does not always work properly. If greater accuracy is required, the status management function must be used. Figure 9.2 shows how to check the Tuning Manager series status when the status management function is unavailable.

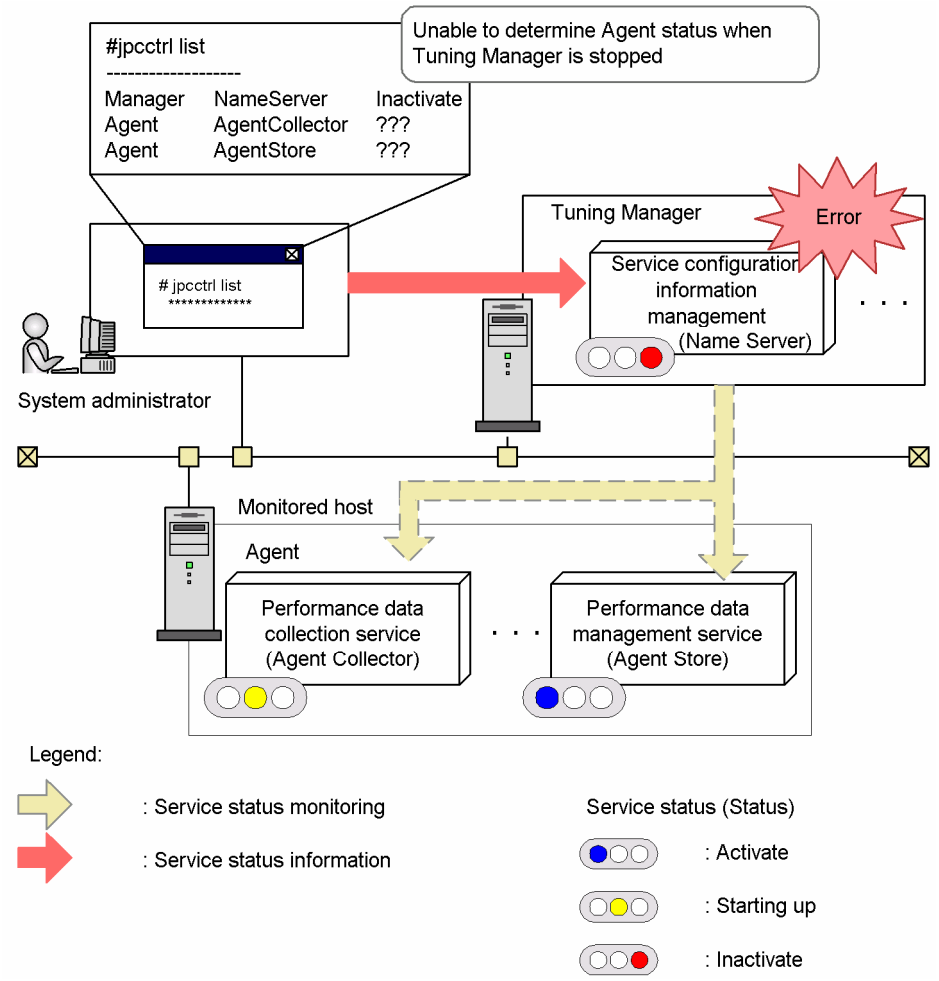


Figure 9.2 Checking the Tuning Manager Series Status When the Status Management Function Is Unavailable

For details on the `jpcctrl list` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

9.2 Setup

The status management function is one of the functions provided in Tuning Manager and Agent.

The default setting (enable or disable) specified after the installation of the status management function varies depending on the environment of the installation destination host. Use the `jpcstsetup display` command to check the settings.

This section explains the setup procedure for enabling and disabling the status management function.

Note: A fixed port number is allocated by default to the Status Server service associated with the status management function.

Enabling the Status Management Function

To enable the status management function:

1. Stop all Tuning Manager series program services.

   Use the following command to stop all Collection Manager and Agent services running on physical or logical hosts:

   ```
   jpcstop all
   ```

2. Execute the `jpcstsetup enable` command.

   The following command is used to enable the status management function:

   ```
   jpcstsetup enable
   ```

3. Check the status management function status.

   Use the following command to make sure that the status management function is `available`:

   ```
   jpcstsetup display
   ```

4. Start the Collection Manager and Agent services.

   Use the following command to start all Collection Manager and Agent services running on physical or logical hosts:

   ```
   jpcstart all
   ```

Disabling the Status Management Function

To disable the status management function:

1. Stop all Collection Manager and Agent services.

   Use the following command to stop all Collection Manager and Agent services running on physical or logical hosts:

```
jpcstop all
```

2. Execute the `jpcstsetup disable` command.

   The following command is used to disable the status management function:

```
jpcstsetup disable
```

3. Check the status management function status.

   Use the following command to make sure that the status management function is `unavailable`:

```
jpcstsetup display
```

4. Start the Collection Manager and Agent services.

   Use the following command to start all Collection Manager and Agent services running on physical or logical hosts:

```
jpcstart all
```

Note: The `jpchasetup list` command can be used to check system settings in a logical host environment. An example of the output generated by `jpchasetup list` is shown below:

```
Logical Host Name  Key    Environment Directory                [Instance Name]
------------------ ------ ------------------------------------ ----------------
lhost2             mgr    "t:¥lhost2¥jp1pc"
lhost2             agto   "t:¥lhost2¥jp1pc"                    inst1

KAVE05136-I The logical host startup information listing ended normally.
```

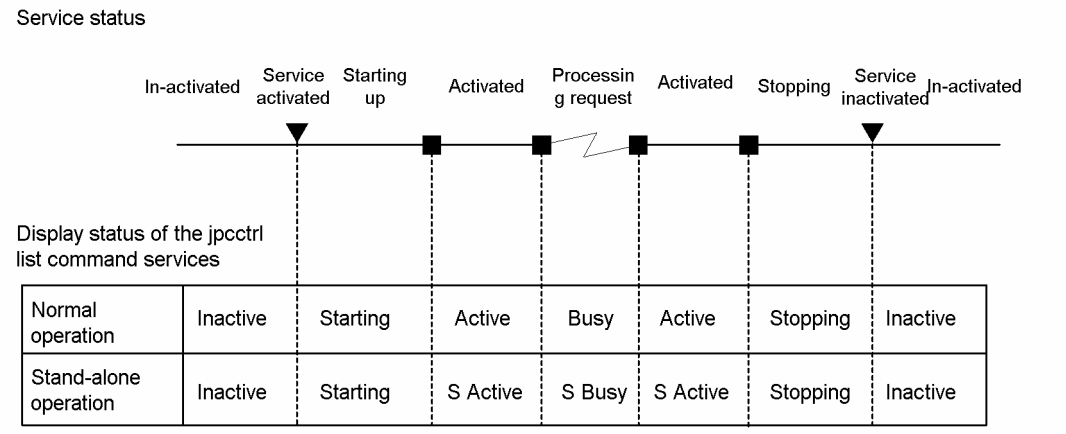
Note: The service keys that can be specified by the `jpcstart` and `jpcstop` commands differ depending on whether the status management function is available or unavailable. For details about these commands, see the *HiCommand Tuning Manager Command Line Interface Guide*.

9.3 Checking the Service Status

The types of service status information that can be accessed by using the `jpcctrl list` command differ depending on whether the status management function is available or unavailable, as explained below.

9.3.1 When the Status Management Function is Available

When the status management function is available, status details such as service startup and shutdown can be accessed by using the `jpcctrl list` command.



Normal operation	Inactive	Starting	Active	Busy	Active	Stopping	Inactive
Stand-alone operation	Inactive	Starting	S Active	S Busy	S Active	Stopping	Inactive

Figure 9.3 Status Information When the Status Management Function is Available

Figure 9.4 shows examples of the system configuration and output generated by the `jpcctrl list` command when the status management function is available for Tuning Manager and the Agent.

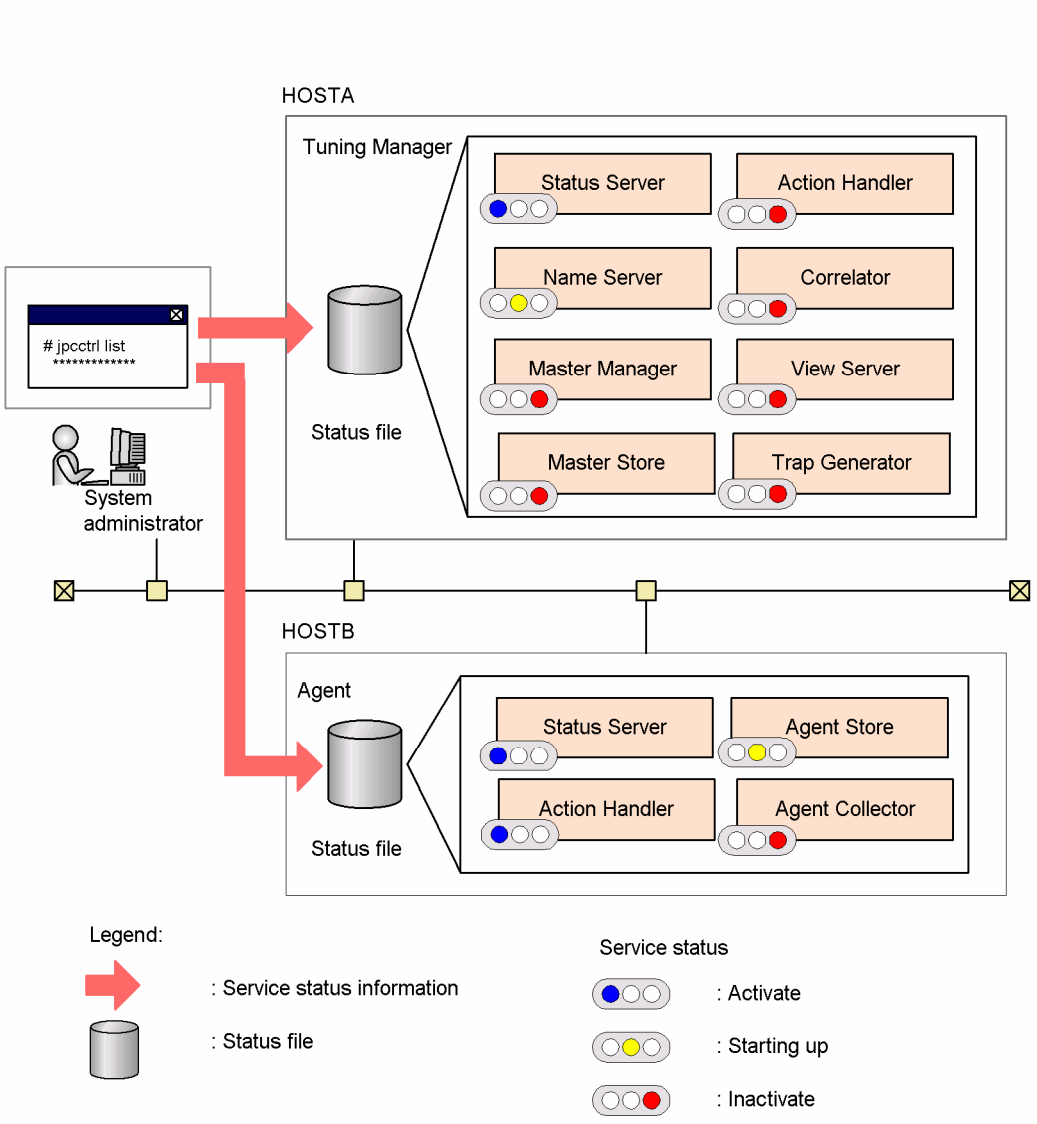


Figure 9.4 Example of a System Configuration When the Status Management Function is Available

```
# jpcctrl list * host=*
Host Name   ServiceID        Service Name     PID    Port     Status
---------   ---------        -------------    -----  -------- ---------

HOSTA       PT1HOSTA         Status Server     483   8206     Busy
HOSTA       PN1001           Name Server       6588           Starting
HOSTA       PM1001           Master Manager                   Inactive
HOSTA       PS1001           Master Store                     Inactive
HOSTA       PE1001           Correlator                       Inactive
HOSTA       PC3HOSTA         Trap Generator                   Inactive
HOSTA       PP1HOSTA         View Server                      Inactive
HOSTA       PH1HOSTA         Action Handler                   Inactive
HOSTB       PT1HOSTB         Status Server    9876   22291    Busy
HOSTB       PH1HOSTB         Action Handler    4872  1116     Active
HOSTB       OS1inst1[HOSTB]  Agent Store        4321           Starting
HOSTB       OA1inst1[HOSTB]  Agent Collector                   Inactive

KAVE06003-I List processing of the service information terminated normally.
```

Figure 9.5 Output Example of the jpcctrl list Command

9.3.2 When the Status Management Function is Unavailable

If the `jpcctrl list` command is executed on a host where the installed version does not support the status management function or where the status management function is unavailable, a message will be output stating that the status management function is not supported, and the following status information is displayed. Note that service status checking requires Tuning Manager to be running.

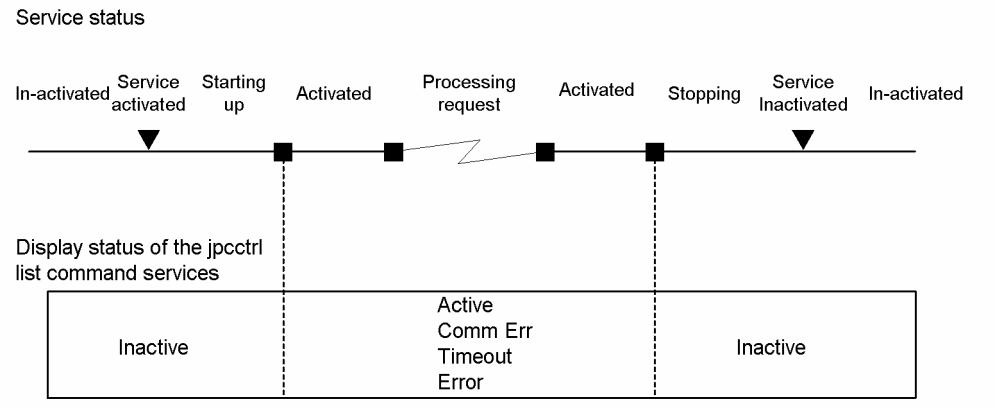


Figure 9.6 Status Information When the Status Management Function is Unavailable

9.4 Status Management During Cluster System Operation

Only one Status Server service operates on each host. During cluster system operation, a Status Server service on a physical host manages the service status on physical and logical hosts.

Figure 9.7 shows an overview of the status management function during cluster operation.

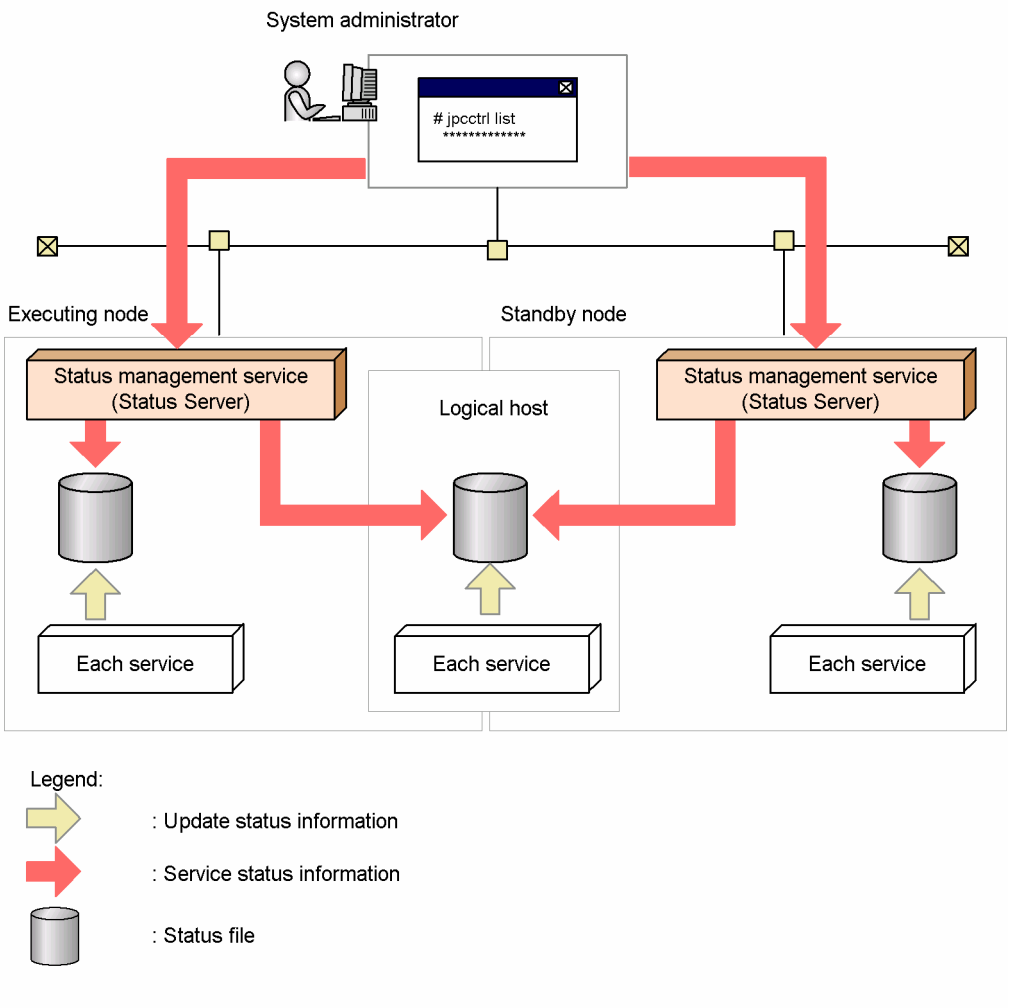


Figure 9.7 Overview of the Status Management Function During Cluster Operation

Note: If the cluster system has an active-active configuration, the Status Server service must be set to run normally or set to avoid failover.

9.5 Status Management Function Errors

This section explains how to check the service status in the event of an error in the status management function.

9.5.1 Abnormal Termination of Status Server Service

When the Status Server service stops, the KAVE00203-W message is output to the common message log. Although detailed service status information is not provided, the basic service status can be checked in the same way as it is checked when the status management function is unavailable, provided that Tuning Manager is operating.

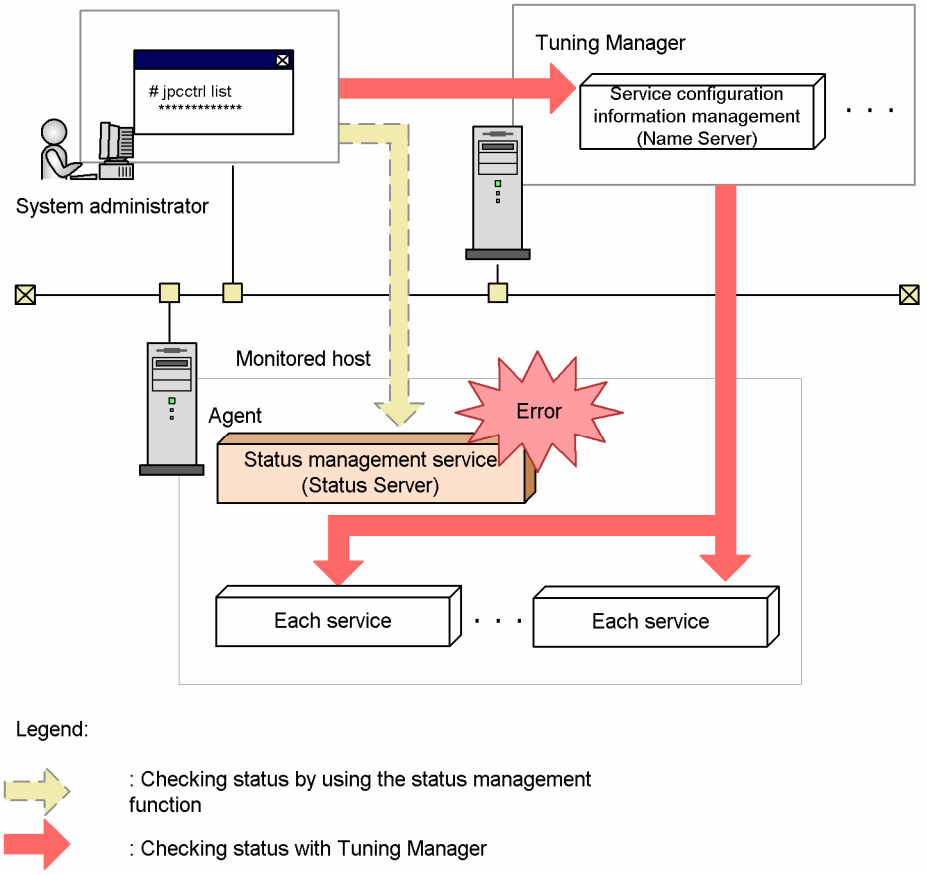


Figure 9.8 Example of an Abnormal Termination of the Status Server Service

9.5.2 Abnormal Termination of Other Services

If services other than Status Server services terminate abnormally, the system might not properly update statuses in the status files. If this happens, the status management function determines the service status from the status of the internal files generated by each service. Thus, detailed service status information is available in the same way as when the status management function is available.

Chapter 10 Error Handling Procedures

This chapter explains how to handle any errors that may occur while you are using Collection Manager and the Agent.

- Error Handling Procedures (see section 10.1)
- Troubleshooting (see section 10.2)
- Log Information (see section 10.3)
- Data Collected in the Event of an Error (see section 10.4)
- Data Collection Procedure (see section 10.5)
- Calling the Support Center (see section 10.6)

10.1 Error Handling Procedures

This section describes the procedures for handling errors that might occur while you are using Collection Manager and the Agent.

For details about the procedures for handling errors that may occur while you are using Main Console and Performance Reporter, see the *HiCommand Tuning Manager Server Administration Guide*, *HiCommand Tuning Manager Installation Guide*, or *HiCommand Tuning Manager User's Guide*.

- Checking the event: When an error occurs, you need to check the event. If an error message is issued, check its contents. For details about the messages, and how to respond to each message, see the *HiCommand Tuning Manager Messages Reference*. For details about the log information that is output by Collection Manager and the Agent, see section 10.3.
- Data to be collected: Collect data to determine the cause of an error. For details about collecting the necessary data, see sections 10.4 and 10.5.
- Determining the cause: Use the collected data to determine the cause and the extent of the error, as well as the range of its consequences.

10.2 Troubleshooting

This section explains how to conduct troubleshooting while you are using Collection Manager and the Agent. If an error occurs while you are using Collection Manager and the Agent, first check to see if any of the events described in this section have occurred.

Table 10.1 lists the principal errors that may occur while you are using Collection Manager and the Agent.

Table 10.1 Errors

Classification	Error	Reference
Problems starting the Agent service	▪ Agent service does not start ▪ Start-up errors with service control manager (Windows) ▪ Service start delay ▪ The Agent Collector service does not start automatically when the OS is restarted. (Windows) ▪ Communication not performed properly ▪ While `jpcstart` is executing, the following error message is output: "KAVE05033-E A service could not start" ▪ Disk capacity is insufficient ▪ The Oracle database does not stop	10.2.1
Problems related to executing commands	▪ `jpcctrl list` command issues ▪ `jpcctrl dump` command issues	10.2.2
Problems related to report definitions	No time period indicated on history report	10.2.3
Problems related to alarm definitions	The program defined in an action is not operating correctly.	10.2.4
Problems related to collecting and managing performance data	▪ Data storage time issues ▪ Common message log error output ▪ Performance data was not collected even though the Agent was started. ▪ Performance data collection is delayed temporarily.	10.2.5

10.2.1 Problems Starting the Agent Service

This section explains how to handle errors related to setup or service startup.

10.2.1.1 Agent Service Does Not Start

Possible causes and solutions:

- The same port number is set for multiple services of Tuning Manager series programs.

  When the same port number is set for multiple services of Tuning Manager series programs, none of the services of Tuning Manager series programs can start. By default, the system automatically assigns port numbers, so that there will be no duplicated port number. If you have specified a fixed port number for a service of the Tuning Manager series program during Tuning Manager setup, check the specified port number. If the same port number is set for more than one service of Tuning Manager series programs, you must make appropriate corrections in the port number settings. For details about setting port numbers, see the *HiCommand Tuning Manager Installation Guide*.

- There is an error in the setting for a storage directory of the Store database.

  If a directory that cannot be accessed or a directory that does not exist is set for any of the directories listed below, the Agent Store service cannot start. Check the setting and, if necessary, correct the directory name.

  - Installation directory of the Store database
  - Backup directory of the Store database
  - Export directory of the Store database

  Also, if one of these directories is set for multiple Agent Store services, the Agent Store service cannot start. Check the setting and, if necessary, correct the directory name.

- The host name of the machine was changed using a non-permitted procedure.

  For details about changing a machine's host name, see the *HiCommand Tuning Manager Installation Guide*. Under some circumstances when the host name is changed using a procedure other than those permitted, a service for a Tuning Manager series program may not start.

- The Oracle database has not been installed. (Agent for Oracle)

  If no Oracle database has been installed, you cannot start the Agent Collector service. Install the Oracle database on the Agent host.

- The Oracle database has not started. (Agent for Oracle)

  If the Oracle database has not started, you cannot start the Agent Collector service. Start the Oracle database and then start the Agent Collector service.

- There is an error in the settings specified when an instance environment was set up.

  Execute the `jpcinssetup` command for the items specified when an instance environment of Agent was set up, and change the settings by specifying valid values for each setup item for the instance environment. For details on setup items for an instance environment and available values for each item, see the *HiCommand Tuning Manager Installation Guide*. For details about the `jpcinssetup` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

10.2.1.2 KAVF29015-F Message Is Output and Agent Collector Service Does Not Start (Agent for DB2)

If the DB2 user specified in the instance information (`db2 user`) does not have SYSADM, SYSCTRL, or SYSMAINT privileges, the Agent Collector service cannot be started.

For details on the privileges for DB2 users, see the chapter that describes the Agent for DB2 setup in the *HiCommand Tuning Manager Installation Guide*.

10.2.1.3 KAVF29023-E Message Is Output and Agent Collector Service Does Not Start (Agent for DB2)

If the DB2 user specified in the instance information (`db2 user`) does not have the SYSADM privileges, the Agent Collector service cannot be started.

When you login as a DB2 user that has SYSCTRL or SYSMAINT privileges, make sure that you set the snapshot monitor switches to ON in advance.

For details on the snapshot monitor switches, see the chapter that describes the Agent for DB2 setup in the *HiCommand Tuning Manager Installation Guide*.

10.2.1.4 Startup Errors with Service Control Manager (Windows)

When executing the `jpcstart` command to start a Tuning Manager series service, the service may not start if another Windows service is activated concurrently. The following error message is output to the common message log:

```
KAVE05163-E An error occurred in the Windows service control manager.
(service=service-name, lhost=logical-host-name, inst=instance-name)
```

If the message appears, re-execute the `jpcstart` command.

If this error occurs repeatedly, change the retry interval and number of retries that are set for retrying to start the service when the `jpcstart` command is executed. To change these values, edit the `jpccomm.ini` file. Changing the retry interval and the number of retries can resolve failures on service startup caused by the service control manager.

To change the retry interval and the number of retries, edit the `jpccomm.ini` file directly. The table below describes the range of values that can be specified, with the corresponding section name and label name.

Table 10.2 Range of Values Specifiable in the jpccom.ini File (with the Corresponding Section and Label)

Section Name	Label Name	Value Range	Default Value	Description
[Tools Section]	StartService Retry Interval	30 to 600 see ***Note1***	45	Retry interval for service startup (in units of seconds)
	StartService Retry Count	0 to 120 see ***Note2***	3	Number of retries for service startup (number of times)

Note 1: When the specified value is less than 30, the system assumes that the minimum value of 30 was specified. When the specified value is greater than 600, the system assumes that the maximum value of 600 was specified.

Note 2: When the specified value is less than 0, the system assumes that the minimum value of 0 was specified. When the specified value is greater than 120, the system assumes that the maximum value of 120 was specified.

The `jpccomm.ini` file is stored in the *installation-folder*.

To change the retry interval and the number of retries:

1. Use a text editor to open the `jpccomm.ini` file.
2. Change the retry interval and the number of retries.

   Change the value of the following labels:

```
[Tools Section]
StartService Retry Interval=45
StartService Retry Count=3
```

3. Save and close the `jpccomm.ini` file.

Note: If a Tuning Manager series program service takes a long time to start when restarting the OS, the service control manager sometimes outputs the following message:

`Event ID: 7022`

`Type: Error`

`Description:` *Service Name* `service stopped during startup.`

If this happens, check for a service startup message in the common message log. If a service startup message has been output, the services have started normally.

10.2.1.5 Service Start Delay

In some cases it might take a long time for a service to actually start once you execute the `jpcstart` command or start a service by choosing the **Service** icon. If service startup takes a long time due to one of the following reasons, subsequent service startups should take less time.

- During initial startup after the Store database is restored, the indexes of the Store database must be rebuilt. This may slow startup of the service.
- During initial startup after a new Agent is added, the indexes of the database must be rebuilt. This may slow startup of the service.
- If the Store service cannot stop normally due to a problem such as disconnection of the power, the indexes of the Store database must be reconstructed at the restart. It might therefore take a long time to start the Store service.

10.2.1.6 Agent Collector Service Does Not Start Automatically When the OS Is Restarted (Windows)

If the Agent Store service did not end normally due to the power being disconnected or some other reason, the Agent Collector service might not automatically start when the OS restarts. If this problem occurs, the event IDs 7022 and 7001 will be output in the event log (system log). Make sure that the Agent Store service has started, and then use the `jpcstart` command to manually start the Agent Collector service.

10.2.1.7 Communication Not Performed Properly

Immediately after a service of a Tuning Manager series program stops, another program service that uses the same port that the stopped service was using might start. In this case, communication may not be performed properly. You can use either of the following techniques to avoid this problem:

- Use fixed port numbers for the Collection Manager and Agent services.

  Allocate a fixed port number to each service of Tuning Manager series programs. For details about setting port numbers, see the *HiCommand Tuning Manager Installation Guide*.

- Set the `TCP_TIMEWAIT` value.

  Use an OS command to set the connection wait time by specifying the `TCP_TIMEWAIT` value.

  For HP-UX or AIX, specify a connection wait time of at least 75 seconds, as shown below:

  - HP-UX: `tcp_time_wait_interval:240000`
  - AIX: `tcp_timewait:5`

  In Windows and Solaris, use the default connection wait time. The defaults values are:

  - Windows 2000 and Solaris: 4 minutes
  - Windows Server 2003 and Windows Server 2003 (IPF): 2 minutes

For Linux, the default connection wait time (60 seconds) cannot be altered. Avoid this problem by using a fixed service port number for the Tuning Manager series programs.

10.2.1.8 While jpcstart Is Executing, the Following Error Message Is Output: "KAVE05033-E A service could not start"

If a 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.

10.2.1.9 Disk Capacity Is Insufficient

If space is insufficient on the disk used by the Store database, the storing of data in the Store database terminates. If this happens, the message, "`The disk capacity is insufficient.`" is output, then the Master Store service or Agent Store service stops.

If this message appears, take one of the following actions:

- Ensure sufficient disk space.

  Estimate the disk space required for the Store database, and then change the storage destination for the Store database to a disk that has sufficient disk space. For details about how to estimate the required disk space for the Store database, see the system requirements described in the *HiCommand Tuning Manager Installation Guide*. For details about how to change the storage destination of the Store database, see the *HiCommand Tuning Manager Installation Guide*.

- Reduce the disk space required for the Store database.

  To reduce the disk space required for the Store database, first change the settings to reduce the maximum amount of data in the Store database. You can reduce the maximum amount of data by reducing the record retention period and the number of records to be stored (specified in the data retention conditions of the Store database) or by narrowing down the records to be collected by the Agent Collector service. For details about how to change the records collected by the Agent Collector service, see section 3.2. For details about how to change the data retention conditions of the Store database, see section 3.3.

  Note that you cannot reduce the disk space required for the Store database by simply specifying the maximum amount of data to be stored in the Store database. To reduce the disk space, perform the following procedure:

  a) Delete the performance data for unnecessary records in the Store database.

In the Store database, unnecessary records are deleted when performance data is stored for the records. If you reduce the records to be collected by the Agent Collector service, the reduced records are no longer collected. As a result, performance data for the records remain undeleted and the required disk space is not reduced. Therefore, you need to perform the following procedure to delete from the Store database the performance data for the records that are no longer collected. If you have not reduced the records to be collected by the Agent Collector service, this procedure is not required.

The following describes how to delete the data for the records that are no longer collected:

Example: When using Agent for Platform, the current collection settings are: PI_LOGD, PI_NIND, and PD_PD are `Yes`. The new collection settings are: PI_LOGD is `No`, PI_NIND is `Yes`, and PD_PD is `No`.

(1) Specify `Yes` for the records that will no longer be collected, and `No` for the other records. In this example, specify `Yes` for PI_LOGD and PD_PD, and `No` for the other records.

(2) Change the data retention conditions as follows:

- For the PD and PL record types, set the maximum number of records to 0.

- For the PI record type, set the record retention period to the shortest period of time corresponding to the aggregation period. For example, for performance data aggregated by the minute, set the retention period to `Minute`, and for performance data aggregated by the hour, set the retention period to `Hour`.

(3) Store the performance data into the Store database at least once.

Note 1: For details about the timing when performance data is stored in the Store database, see section 2.3.

Note 2: Performing steps (1) through (3) disables the performance data space for the records that exist in the Store database but are set to not be collected anymore (in the above example, PI_LOGD and PD_PD). By reconfiguring the Store database, you can delete the disabled space from the database file. Note that, for the PI record type records and the Process Detail(PD) records for Agent for Platform, you might not be able to completely disable the performance data space for the records. For details, see *Records that cannot be deleted from the Store database even if performance data is stored for the records*.

(4) Set all the record collection settings to `No`.

(5) Consider and specify the data retention conditions of the Store database.

(6) Consider and specify the collection settings.

Delete unnecessary performance data from the Store database.

When you reduce the number of records stored in the Store database, or shorten the retention period, the performance data stored based on the previous data retention conditions still exists in the Store database, which means the performance data that exceeds the current data retention conditions exists in the Store database. In this case, perform the following procedure to delete the performance data that does not meet the data retention conditions. If you have not reduced the number of records stored in the Store database or shortened the retention period, this procedure is not required.

(1) Consider and specify the data retention conditions of the Store database.

(2) Store performance data for the records whose data retention conditions are changed in the Store database, one or more times.

Notes:

- For details about the timing when performance data is stored in the Store database, see section 2.3.
- In the Store database, the space for the data that does not meet the data retention conditions is disabled at the timing when the records are stored and performance data increases in the Store database. By reconfiguring the Store database, you can delete the disabled space from the database file. Note that, for the PI record type records and the Process Detail(PD) records for Agent for Platform, you might not be able to completely disable the performance data space for the records. For details, see *Records that cannot be deleted from the Store database even if performance data is stored for the records*.

(3) Reconfigure the Store database.

Reconfigure the Store database to reduce the disk space required for the Store database. For details about how to reconfigure the Store database, see section 3.6.3.

Records that cannot be deleted from the Store database even if performance data is stored for the records.

In the Store database, the space for the data that does not meet the data retention conditions is disabled when the records are stored and performance data increases.The disabling action varies according to record type, however. The PI record and Process Detail (PD) record for Agent for Platform are affected differently.

For PI record type records, if the aggregation type of the records creates new performance data when the performance data is stored, the performance data for the records is deleted from the Store database. For other aggregation types, the data remains in the Store database. However, if the aggregation type is year, all the performance data for the records will remain.

For example, assume that you store performance data for PI_LOGD records at 10:00:00 on 2006/5/24 (Wed.), when all the performance data for the PI_LOGD records in the Store database was collected before 16:00:00 on 2006/5/23 (Tue.). In this case, the performance data whose aggregation type is year remains. If the aggregation type of the performance data is month, the stored performance data is aggregated into the performance data dated May 2006, and new performance data is not created. For this reason, the performance data for PI_LOGD records whose aggregation type is month remains in the Store database. Likewise, the performance data whose aggregation type is week remains in the Store database. For the performance data for the records whose aggregation type is day, new performance data dated 2006/5/24 is created and this disables all the performance data space for the PI_LOGD records whose aggregation type is day. The performance data space for the records whose aggregation type is hour or minute will also be disabled. Add, to the estimate for the disk space required for the Store database, the disk space for the performance data that is not freed in the above operations.

When the records are Process Detail (PD) records for Agent for Platform, if there is no difference between the previously collected performance data and the subsequently collected data, the performance data remains undeleted in the Store database. For details about Process Detail (PD) records for Agent for Platform, see the *HiCommand Tuning Manager Operating System Reports Reference*. Delete the performance data by generating a difference, or add the disk space required for the Process Detail (PD) record data to the disk space estimate for the Store database.

If the Master Store service or Agent Store service still does not start after the above actions are taken, then an unrecoverable logical conflict exists in the Store database. In such a case, restore the Store database from backup data, and then start either the Master Store service or Agent Store service as appropriate. If backup data is not available, initialize the Store database, and then start either the Master Store service or Agent Store service as appropriate. To initialize the Store database, delete the following files from the Store database's storage directory of the Store database:

- All files that have the extension `.DB`
- All files that have the extension `.IDX`

For details about the storage destination of the Store database, see the *HiCommand Tuning Manager Installation Guide*.

10.2.1.10 The Oracle Database Does Not Stop

Before Agent for Oracle stops, the Oracle database being monitored may not stop in the NORMAL shutdown mode. Use the IMMEDIATE shutdown mode to stop the Oracle database.

10.2.2 Problems Related to Executing Commands

This section explains how to handle errors related to executing Collection Manager and Agent commands.

10.2.2.1 jpcctrl list Command Issues

When the `jpcctrl list` command is executed, the names of services not operating are output. Possible causes and solutions:

- A Tuning Manager series program was uninstalled without its service information being deleted.

  Service information for a Tuning Manager series program remains in the database even after the program is uninstalled. Execute the `jpcctrl delete` command to delete the service information. For details about this command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

- The host name of the machine was changed without deleting the service information of the Tuning Manager series program.

  If you change the host name of a machine without deleting the service information of the Tuning Manager series programs, the service information corresponding to the service ID to which the previous host name was appended remains in the database that the Master Manager service manages. Execute the `jpcctrl delete` command to delete the service information. For details about this command, see the *HiCommand Tuning Manager Command Line Interface Guide*. For details about changing the machine's host name, see the *HiCommand Tuning Manager Installation Guide*.

10.2.2.2 jpcctrl dump Command Issues

When the `jpcctrl dump` command is executed, data other than the specified Store data is output. Specifying the same export file name for the same Store service in multiple executions of the `jpcctrl dump` command causes the initial output results to be overwritten with the subsequent output results. Each time you execute the `jpcctrl dump` command for the same Store service, specify a different export file name. For details about executing this command, see section 3.5.

10.2.3 Problems Related to Report Definitions

This section explains how to handle errors related to Tuning Manager series report definitions.

10.2.3.1 No Time Period Indicated on History Report

If the current time of the machine where the Agent is installed is moved forward in time, the history information from before the change to after the change will not be saved.

10.2.4 Problems Related to Alarm Definitions

This section explains how to handle errors related to alarm definitions in Tuning Manager series programs.

10.2.4.1 The Program Defined in an Action Is Not Operating Correctly

Possible causes and solutions:

- The environment of the login machine differs from the execution environment of the program defined in the action.

  If the environment of the login machine differs from the execution environment of the program defined in the action, Tuning Manager may not be able to execute the program. Check whether the defined program can be executed as a Tuning Manager series action. Table 10.3 shows the execution environments in which programs can be executed as a Tuning Manager series action.

  Table 10.3 Programs That Can Be Executed as a Tuning Manager Series Action

Execution Environment	Windows	UNIX
Account	System account	root user
Environment variables	System environment variables when services of Tuning Manager series programs are running	root user environment variables when Tuning Manager series programs are running
Current directory	Action Handler service folder	Action Handler service directory
Shell at runtime	Not applicable	Login shell of root user

- The user does not have execution permission to execute the program defined in the action.

  If any of the following programs is defined as a Tuning Manager series action, the program cannot be executed due to execution permission restrictions:

  - Any program in an NFS mount directory
  - Any program that references or updates files in an NFS mount directory

Check whether the defined program can be executed as a Tuning Manager series action. Table 10.3 shows the execution environments in which programs can be executed as a Tuning Manager series action.

- Either Tuning Manager or the Action Handler service of the Agent on the host that is executing the action is not running.

  Actions cannot be executed if either Tuning Manager or the Action Handler service of the Agent on the host that is executing the action is stopped. Before executing the action, start Tuning Manager and the Action Handler service of the Agent on the host that is executing the action.

10.2.5 Problems Related to Collecting and Managing Performance Data

This section explains how to handle errors related to collecting and managing performance data.

10.2.5.1 Data Storage Time Issues

If the file capacity of the Agent 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 Agent Store database, and then restore the database again.

For details on setting the data storage period, see the *HiCommand Tuning Manager User's Guide*. For details on backing up and restoring the Agent Store database, see section 5.2.

10.2.5.2 The Following Message Is Output to the Common Message Log: "Illegal Data was detected in the Store Database"

An unexpected service halt or machine shutdown may result in invalid data in the Store database. Recover from this problem as described below:

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

10.2.5.3 Performance Data Was Not Collected

If performance data is not being collected, use one of the following methods to recover:

- Check the startup status and settings of the Agent.

- Make sure that the IP address to be monitored, which was set when an instance environment of Agent for NAS was set up, is correct.
- If you create a table while Microsoft SQL Server is performing transactions, a sharing lock is applied to system tables until the table is created, which causes collection of performance data to fail. Finish creating the table, and then collect performance data.
- Check the status of the Oracle database, and if it has stopped, restart it.
- Check the status of Microsoft SQL Server, and if it stopped, restart it.
- Check the status of DB2, and if it stopped, restart it.
- If the `KAVF29018-E` message is output in the Agent for DB2 and the value in the `LANG` environment variable, which is set at the startup of the Agent for DB2, is different from the value in the database code page, records might not be collected. In this case, make sure that the value in the `LANG` environment variable is the same as that in the database code page (the `DB2CODEPAGE` registry variable).
- 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 *HiCommand Tuning Manager Command Line Interface Guide*.

10.2.5.4 Performance Data Collection Is Delayed Temporarily

If a conflict occurs with another operation request in a monitored system when performance data is collected, performance data collection is delayed temporarily. For Agent for NAS, there are two possible causes of this conflict:

- NAS Package is now being installed in a storage subsystem.
- A cluster configuration is now being set up in the NAS system.

10.2.6 Other Problems

Check the existing circumstances when an error occurs. Read any messages that are output. For details about the log information that is output by Collection Manager and the Agent, see section 10.3.

If you cannot resolve an error by taking any of the steps described from section 10.2.1 to section 10.2.5, or if an error occurs that is not described in these sections, collect the 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 section 10.4 and section 10.5.

10.3 Log Information

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

- System log
- Common message log
- Trace log
- Agent log

This section describes each log information.

10.3.1 Log Information Types

- System log: The *system log* contains log information that reports the system status and errors that occurred. This log information is output to the following log file:
  - In Windows: Event log file
  - In UNIX: `syslog` file

  For details about the output formats, see Chapter 11.
- Cluster software log: When operating logical hosts, cluster software logs are required for checking control of Collection Manager and the Agent.
- 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 10.3.2. For details about the output formats, see Chapter 11.

  When running Collection Manager and the Agent on a logical host, the common message logs are output to a log file on the shared disk. When a failover occurs, the log file on the shared disk, along with the system, are transferred, so messages are recorded to the same log file.
- 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 different log file for each service of Collection Manager or the Agent.

  When running Collection Manager and the Agent on a logical host, the trace logs are output to a log file on the shared disk. When a failover occurs, the log file on the shared disk, along with the system, are transferred, so messages are recorded to the same log file.

- Agent log: The *agent log* contains log information for the processing related to record collection. This log is output by Agent for Oracle and Agent for Microsoft SQL Server. When an error occurs, you can use this log to obtain detailed information for the above processing. The agent log outputs the normal log information and error log information to separate files. For details about the log output destinations, see section 10.3.2.3.

10.3.2 Log Files and Directories

This section describes the log information that is output from Collection Manager or the Agent.

10.3.2.1 Common Message Log

This section describes the common message log output from the Collection Manager or the Agent. The following tables list, for each OS, the name of the service or control that output the log, the log file name, and the required disk space.

Table 10.4 File Name of the Common Message Log (Windows)

Log Information Type	Output Source	File Name	Disk Space Used (KB) *(see Note 2)*
Common message log	Collection Manager and Agent	*installation-folder*\log\jpclog{01|02} *(see **Note 1**)*	2,048 (*2)
Common message log (when running logical hosts)	Collection Manager and Agent running on a logical host	*environment-directory* \jp1pc\log\jpclog{01|02} (see ***Note 1*** and ***Note 3***)	2,048 (*2)

Table 10.5 File Name of the Common Message Log (UNIX)

Log Information Type	Output Source	File Name	Disk Space Used (KB) *(see Note 2)*
Common message log	Collection Manager and Agent	/opt/jp1pc/log/jpclog{01|02}(see ***Note 1***)	2,048 (*2)
Common message log (when running logical hosts)	Collection Manager and Agent running on a logical host	*environment-directory* /jp1pc/log/jpclog{01|02} (see ***Note 1*** and ***Note 3***)	2,048 (*2)

(Notes apply to both Table 10.4 and Table 10.5)

Note 1: The system uses the following rules when creating log file names:

*Common message lo*g: 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 new 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.

Note 2: The value in parentheses is the number of log files that can be created for a single service. For example, `256(*2)` indicates that up to two log files, each using 256 KB of disk space, can be created. In this case, a total of 512 KB of disk space will be used.

Note 3: The *environment-directory* is the directory that was specified on a shared disk when the logical host was created.

10.3.2.2 Trace Log

Table 10.6 and Table 10.7 contain the trace log output from the Collection Manager or the Agent. The following tables list, for each OS, the names of the services or controls that output the log, and the name of the directories where the logs are output.

Table 10.6 Folders Where Trace Logs Are Output (Windows)

Type of Log Information	Output Source	Folder Name
Trace log	Name Server	*installation-folder*\mgr\namesvr\log\
	Master Manager	*installation-folder*\mgr\manager\log\
	View Server	*installation-folder*\mgr\viewsvr\log\
	Correlator	*installation-folder*\mgr\clator\log\
	Trap Generator	*installation-folder*\mgr\trapgen\log\
	Action Handler	*installation-folder*\bin\action\log\
	Manager Store	*installation-folder*\mgr\store\log\
	Status Server	*installation-folder*\bin\statsvr\log\
	Collection Manager and Agent command	*installation-folder*\tools\log\
	Agent Collector	*installation-folder**xxxx*\agent[*instance-name*]\log\ (see ***Note 1*** and ***Note 2***)
	Agent Store	*installation-folder**xxxx*\store[*instance-name*]\log\ (see ***Note 1*** and ***Note 2***)

Type of Log Information	Output Source	Folder Name
Trace log (when running logical hosts)	Name Server	*environment-directory*\jp1pc\mgr\namesvr\log\ (see ***Note 3***)
	Master Manager	*environment-directory*\jp1pc\mgr\manager\log\ (see ***Note 3***)
	View Server	*environment-directory*\jp1pc\mgr\viewsvr\log\ (see ***Note 3***)
	Correlator	*environment-directory*\jp1pc\mgr\clator\log\ (see ***Note 3***)
	Trap Generator	*environment-directory*\jp1pc\bin\trapgen\log\ (see ***Note 3***)
	Action Handler	*environment-directory*\jp1pc\mgr\action\log\ (see ***Note 3***)
	Manager Store	*environment-directory*\jp1pc\mgr\store\log\ (see ***Note 3***)
	Collection Manager and Agent Command	*environment-directory*\jp1pc\tools\log\ (see ***Note 3***)
	Agent Collector	*environment-directory*\jp1pc*xxxx*\agent[*instance- name*]\log\ (see ***Note 1***, ***Note 2*** and ***Note 3***)
	Agent Store	*environment-directory*\jp1pc*xxxx*\store [*instance-name*]\log\ (see ***Note 1***, ***Note 2*** and ***Note 3***)

Table 10.7 Directories Where Trace Logs Are Output (UNIX)

Type of Log Information	Output Source	Directory Name
Trace log	Name Server	/opt/jp1pc/mgr/namesvr/log/
	Master Manager	/opt/jp1pc/mgr/manager/log/
	View Server	/opt/jp1pc/mgr/viewsvr/log/
	Correlator	/opt/jp1pc/mgr/clator/log/
	Trap Generator	/opt/jp1pc/mgr/trapgen/log/
	Action Handler	/opt/jp1pc/bin/action/log/
	Manager Store	/opt/jp1pc/mgr/store/log/
	Status Server	/opt/jp1pc/bin/statsvr/log/
	Collection Manager and Agent command	/opt/jp1pc/tools/log/
	Agent Collector	/opt/jp1pc/*xxxx*/agent[/*instance-name*]/log/ (see ***Note 1*** and ***Note 2***)

Type of Log Information	Output Source	Directory Name
	Agent Store	/opt/jp1pc/*xxxx*/store[/*instance-name*]/log/ (see ***Note 1*** and ***Note 2***)
Trace log (for logical hosts)	Name Server	*environment-directory*/jp1pc/mgr/namesvr/log/ (see ***Note 3***)
	Master Manager	*environment-directory*/jp1pc/mgr/manager/log/ (see ***Note 3***)
	View Server	*environment-directory*/jp1pc/mgr/viewsvr/log/ (see ***Note 3***)
	Correlator	*environment-directory*/jp1pc/mgr/clator/log/ (see ***Note 3***)
	Trap Generator	*environment-directory*/jp1pc/mgr/trapgen/log/ (see ***Note 3***)
	Action Handler	*environment-directory*/jp1pc/bin/action/log/ (see ***Note 3***)
	Manager Store	*environment-directory*/jp1pc/mgr/store/log/ (see ***Note 3***)
	Collection Manager and Agent Command	*environment-directory*/jp1pc/tools/log/ (see ***Note 3***)
	Agent Collector	*environment-directory*/jp1pc/*xxxx*/agent[/*instance name*]/log/ (see ***Note 1***, ***Note 2*** and ***Note 3***)
	Agent Store	*environment-directory*/jp1pc/*xxxx*/store[/*instance name*]/log/ (see ***Note 1***, ***Note 2*** and ***Note 3***)

The following notes apply to both Table 10.6 and Table 10.7:

Note 1*: *xxxx indicates the service key of each Agent. For details on the service keys of each Agent, see Appendix A.

***Note 2*:** For an Agent that monitors multiple storage subsystems and an Agent that monitors an application program that can start a set of multiple services at the same host, there is an instance name directory.

***Note 3*:** *environment-directory* indicates a directory that was specified on a shared disk when the logical host was created.

10.3.2.3 Agent Log

The following tables list and describe, from among the agent log information for Agent for Oracle and Agent for Microsoft SQL Server, the name of the service or control that outputs the agent log information, the log file name, and the required disk space.

Table 10.8 Agent Log Files for Agent for Oracle

Log Information Type	Output Source	Default Output Destination *(see Note)*	File Name	Default Disk Space Used (MB) *(see Note)*
Normal log	PFM - Agent for Oracle	In Windows: *installation-folder*\agto\agent*instance-name*\log\ In UNIX: /opt/jp1pc/agto/agent/*instance-name*/log/	agtoinf{01|02} (see ***Note 2***)	16
Error log			agtoerr{01|02} (see ***Note 2***)	

Table 10.9 Agent Log Files for Agent for Microsoft SQL Server

Log Information Type	Output Source	Default Output Destination *(see Note 1)*	File Name	Default Disk Space Used (MB) *(see Note 1)*
Normal log	PFM - Agent for Microsoft SQL Server	*installation-folder*\agtq\agent*instance-name*\log\	agtqinf{01|02} (see ***Note 2***)	16
Error log			agtqerr{01|02} (see ***Note 2***)	

The following notes apply to both Table 10.8 and Table 10.9:

Note 1: You can use the jpcinssetup command to check and change the output destination and maximum file size of the agent log. For details about how to change them using the jpcinssetup command, see the *HiCommand Tuning Manager Installation Guide*.

***Note 2*:** The agent log is output to two files in wrap-around style. The number 01 or 02 is appended to the file name. Each number has the following meaning:

- 01: current file
- 02: backup file

When files are output in wrap-around style, log information is first output to the log file whose name ends with 01. When the maximum log file size is reached, a new log file whose name ends with 02 is created. Log information is then output to the file whose name ends in 02. If a log file with a name ending in 02 already exists, all data in the log file is deleted and then log information is output to the file from the first line. The two files are then used interchangeably as the log output file.

10.4 Data Collected in the Event of an Error

If the appropriate action described in section 10.2 is not successful in correcting the error, collect the 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 the Agent provide a command for collecting the needed data. Use the `jpcras` command to collect Collection Manager and 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 command options and collectable data, see the *HiCommand Tuning Manager Command Line Interface Guide*.

Consider the following when collecting data to troubleshoot a logical host:

- Logs of Collection Manager and the Agent are stored on a shared disk. When the common disk is placed online (Windows) or mounted (UNIX), all the logs stored on the common disk can be collected by executing the `jpcras` command.
- To investigate failovers, collect data before and after the failover from both the executing node and the standby node.
- To investigate Collection Manager and the Agent running on a logical host, you will need data on the cluster software. Since cluster software controls starting and stopping of Collection Manager and the Agent running on a logical host, comparing the operations of these three is necessary for the investigation.

10.4.1 Information Collected for Windows Systems

Table 10.10 contains log information collected for Windows systems.

Table 10.10 Log Information (Windows)

Type of Information	Overview	Default File Name	Collected by jpcras Command
System log	Windows event log	N/A	Yes
Process information	List of processes	N/A	Yes
System file	`hosts` file	*system-folder*`\system32\drivers\etc\hosts`	Yes
	`services` file	*system-folder*`\system32\drivers\etc\services`	Yes
OS information	System information	N/A	Yes

Type of Information	Overview	Default File Name	Collected by jpcras Command
	Network status	N/A	Yes
	Host name	N/A	Yes
Dump information	Dr. Watson log file	▪ In Windows 2000: *system-drive*\Documents and Settings\All Users\Documents\DrWatson\drwtsn32.log *system-drive*\Documents and Settings\All Users\Documents\DrWatson\user.dump ▪ In Windows Server 2003 and Windows Server 2003 (IPF): *system-drive*\Documents and Settings\All Users\Application Data\Microsoft\Dr Watson\drwtsn32.log *system-drive*\Documents and Settings\All Users\Application Data\Microsoft\Dr Watson\user.dump ***Note***: If your setup outputs log files to a different folder, be sure to collect data from the correct folder.	Yes

10.4.1.1 Information About Collection Manager or the Agent

Collect the information about Collection Manager or the Agent that is listed in Table 10.11. In case of a network error, also collect applicable files from the connection-target host.

Table 10.11 Collection Manager/Agent Information (Windows)

Type of Information	Overview	Default File Name	Collected by jpcras Command
Common message log	Message log output from Collection Manager or the Agent	*installation-folder*\log\jpclog{01|02}	Yes
Configuration information	Each configuration information file	Not applicable	Yes
	Output results of the jpcctrl list command	Not applicable	Yes
Version information	Product version	Not applicable	Yes
	Historical information	Not applicable	Yes

Type of Information	Overview	Default File Name	Collected by jpcras Command
Database information	Name Server	*installation-folder*\mgr\namesvr*.DB *installation-folder*\mgr\namesvr*.IDX	Yes
	Master Manager	*installation-folder*\mgr\manager*.DB *installation-folder*\mgr\manager*.IDX	Yes
	Master Store	*installation-folder*\mgr\store*.DB *installation-folder*\mgr\store*.IDX	Yes
	View Server	*installation-folder*\mgr\viewsvr\data* *installation-folder*\mgr\viewsvr\Reports*	Yes
	Agent Store	*installation-folder**xxxx*\store[*instance-name*]*.DB (see ***Note 1*** and ***Note 2***) *installation-folder**xxxx*\store[*instance-name*]*.IDX (see ***Note 1*** and ***Note 2***)	Yes
Trace log	Trace information for each service of Collection Manager or the Agent	Not applicable (see ***Note 3***)	Yes
Agent log	Normal log for processing related to record collection of Agent for Oracle	*installation-folder*\agto\agent*instance-name*\log\agtoinf{01|02} (see ***Note 4***)	Yes (see ***Note 5***)
	Error log for processing related to record collection of Agent for Oracle	*installation-folder*\agto\agent*instance-name*\log\agtoerr{01|02} (see ***Note 4***)	Yes (see ***Note 5***)
	Normal log for processing related to record collection of Agent for Microsoft SQL Server	*installation-folder*\agtq\agent*instance-name*\log\agtqinf{01|02} (see ***Note 4***)	Yes (see ***Note 5***)
	Error log for the processing related to record collection of Agent for Microsoft SQL Server	*installation-folder*\agtq\agent*instance-name*\log\agtqerr{01|02} (see ***Note 4***)	Yes (see ***Note 5***)

Type of Information	Overview	Default File Name	Collected by jpcras Command
Installation log (see ***Note 6***)	Message log generated during installation (when the OS is Windows 2000 or Windows Server 2003)	*%TEMP%*\pfm_inst.log	No
Monitored storage information (see ***Note 7***)	utlprm.inf file (see ***Note 8***)	*installation-folder*\agtd\agent*instance-name*\utlprm.inf	Yes
	Microcode version of the monitored storage subsystem	Not applicable	Yes
	Execution result of API for collecting performance information of Agent for RAID	Not applicable	Yes

Note 1*: *xxxx indicates the service key of each Agent. For details on the service keys of each Agent, see Appendix A.

***Note 2*:** For an Agent that monitors multiple storage subsystems and an Agent that monitors an application program that can start a set of multiple services at the same host, there is an instance name folder.

***Note 3*:** For details about the installation folder for the trace log, see section 10.3.2.2

***Note 4*:** For details about the output format of the agent logs and how to change the folder in which the agent logs are stored, see 10.3.2.3

***Note 5*:** The jpcras command collects agent log information only from the log output destination folder that is currently specified. If you change the output destination folder for the agent log, manually collect the agent log files that were output to the previous output destination folder.

***Note 6*:** Collect this log if installation failed.

***Note 7*:** Collect this information when using Agent for RAID.

***Note 8*:** You do not need to collect this file if the monitored storage subsystem is TagmaStore USP, the Lightning 9900 Series, or the Lightning 9900V Series.

10.4.1.2 Operation Information

Collect the following information about the operation being performed when the error occurred:

- Details of the operation

- Time the error occurred
- Machine configuration (such as, the OS version, host name, and the Tuning Manager and Agent configuration)
- Whether the error is replicable

10.4.1.3 Error Information on Screen Displays

Obtain printouts of the following:

- The active screen when the application error occurred
- The error message dialog box (including the contents of detailed information, if displayed)
- The Command Prompt window, if the error occurred during command execution

10.4.1.4 Information Related to Performance Data (Agent for Platform (Windows))

Collect the following information related to performance data for Agent for Platform (Windows). If an error occurs (for example, in a network connection), you also need to collect the files on the machine, command execution results, and registry information. The following table lists the information related to the performance data that is collected uniquely by Agent for Platform (Windows) in an environment where Agent for Platform (Windows) is installed.

Table 10.12 Performance Data Information Collected by Agent for Platform (Windows)

Type of Information	Overview	File Name, Windows Command Name, and Registry Definition Location	Collected by jpcras Command
Performance definition information	Counter definition file	*system-folder*\system32\perfc009.dat	Yes
		system-folder\system32\perfh009.dat	Yes
	Location of counter definition (registry)	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib	Yes
		HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services	Yes
	Counter information during startup of the Agent Collector service	*installation-folder*\agtt\agent\map.log	Yes
OS information (Windows command name) (see ***Note***)	Host name	hostname	Yes
	Version	ver	Yes

Type of Information	Overview	File Name, Windows Command Name, and Registry Definition Location	Collected by jpcras Command
	Cluster	cluster	Yes
	Device	mode	Yes
	Disk counter	diskperf	Yes
	Disk volume	mountvol	Yes
		vol	Yes
	Virtual drive	subst	Yes
	TCP/IP	jpconfig	Yes
	IPX	ipxroute config	Yes
	Network status	nbtstat	Yes
	Network name	net name	Yes
	Client	net view	Yes
	Service	net start	Yes
	Server service	net config server	Yes
	Workstation service	net config workstation	Yes
	Session	net session	Yes
	Shared resource	net share	Yes
		net use	Yes
	User accounts	net user	Yes
		net accounts	Yes
	Local group	net local group	Yes

***Note*:** For details about the commands, see Windows Help.

10.4.1.5 Information Related to Performance Data (Agent for Oracle)

Collect the following information related to performance data for Agent for Oracle. If an error occurs (for example, in a network connection), you also need to collect the files on the machine, command execution results, and registry information. The following table lists the information related to the performance data that is collected uniquely by Agent for Oracle in an environment where Agent for Oracle is installed.

Table 10.13 Performance Data Information Collected by Agent for Oracle

Type of Information	Overview	File Name	Collected by jpcras Command
PDNL record information	Output result of `lsnrctl` (Oracle command)	*installation-folder*\agto\agent*instance-name*\pdnl.out	Yes
PDNL record information	Output result of `lsnrctl` (Oracle command)	*installation-folder*\agto\agent*instance-name*\lsnrctl.status.pdnl.out.err.txt	Yes
PDNL record information	Execution result of `hostname`	*installation-folder*\agto\agent*instance-name*\sqlnet.tmp	Yes
PDLS record information	Output result of `lsnrctl` (Oracle command)	*installation-folder*\agto\agent*instance-name*\pdls.out	Yes
PDLS record information	Output result of `lsnrctl` (Oracle command)	*installation-folder*\agto\agent*instance-name*\lsnrctl.status.pdls.out.err.txt	Yes

10.4.1.6 Other Information

Collect the following additional information:

- The contents of **System Information** under **Computer Management** or the contents of **System Information**, which is displayed by choosing **Accessories** and then **System Tools** (in Windows 2000)
- The contents of **System Information**, which is displayed by clicking **Accessories** and then **System Tools** (in Windows Server 2003, Windows Server 2003 (IPF))
- The command arguments that were specified, if the error occurred during command execution
- When Brocade switches are being monitored, the following contents about the switches that make up the proxy switches and fabric:
  - If the firmware of the monitored switch is version 4.4.0b or earlier, the contents of log file collected with the `supportShow` command
  - If the firmware of the monitored switch is version 4.4.0c or later, the contents of the log file collected with the `supportSave` command
- The contents of the following EFCM log files (when monitoring McDATA switches):
  - `Audit Log`
  - `Event Log`
  - `Fabric Log`
  - `Hardware Log`

- Link Incident Log
- Product Status Log
- Threshold Alert Log

10.4.2 Information Collected for UNIX Systems

Table 10.14 contains log information collected for UNIX systems.

Table 10.14 Log Information (UNIX)

Type of Information	Overview	Default File Name	Collected by jpcras Command
System log	syslog	▪ In HP-UX /var/adm/syslog/syslog.log ▪ In Solaris /var/adm/messages ▪ In AIX -- ▪ In Linux /var/log/messages	Yes ***Note 1***: The command only collects log files with the default name. For others, collect manually.
Process information	List of processes	Not applicable	Yes
System file	hosts file	/etc/hosts	Yes
	services file	/etc/services	Yes
OS information	Patch information	Not applicable	Yes
	Kernel information	Not applicable	Yes
	Version information	Not applicable	Yes
	Network status	Not applicable	Yes
	Environmental variables	Not applicable	Yes
	Host name	Not applicable	Yes
	Device information (see ***Note 2***)	Not applicable	Yes
	File system information (see ***Note 2***)	Not applicable	Yes
	Disk group information (see ***Note 2***)	Not applicable	Yes
Dump information	core file	Not applicable	Yes

Note 2: Collect this information when using Agent for RAID Map.

10.4.2.1 Information About Collection Manager or the Agent

Collect the information about Collection Manager or the Agent that is listed below. In case of a network error, also collect applicable files from the connection-target host.

Table 10.15 Collection Manager/Agent Information (UNIX)

Type of Information	Overview	Default File Name	Collected by jpcras Command
Common message log	Message log output from Collection Manager or the Agent	`/opt/jp1pc/log/jpclog{01|02}`	Yes
Configuration information	Each configuration information file	Not applicable	Yes
	Output results of the `jpcctrl list` command	Not applicable	Yes
Version information	Product version	Not applicable	Yes
	Historical information	Not applicable	Yes
Database information	Name Server	`/opt/jp1pc/mgr/namesvr/*.DB` `/opt/jp1pc/mgr/namesvr/*.IDX`	Yes
	Master Manager	`/opt/jp1pc/mgr/manager/*.DB` `/opt/jp1pc/mgr/manager/*.IDX`	Yes
	Master Store	`/opt/jp1pc/mgr/store/*.DB` `/opt/jp1pc/mgr/store/*.IDX`	Yes
	View Server	`/opt/jp1pc/mgr/viewsvr/data/*` `/opt/jp1pc/mgr/viewsvr/Reports/*`	Yes

	Agent Store	`/opt/jp1pc/`*xxxx*`/store[/`*instance-name*`]/*.DB`
		`/opt/jp1pc/`*xxxx*`/store[/`*instance-name*`]/*.IDX`
		(see ***Note 1*** and ***Note 2***)
			Yes

Type of Information	Overview	Default File Name	Collected by jpcras Command
Trace log	Trace information for each service of Collection Manager or the Agent	Not applicable (see ***Note 3***)	Yes
Agent log	Normal log for processing related to record collection of Agent for Oracle	`/opt/jp1pc/agto/agent/`*instance-name*`/log/agtoinf{01|02}` (see ***Note 4***)	Yes (see ***Note 5***)
	Error log for processing related to record collection of Agent for Oracle	`/opt/jp1pc/agto/agent/`*instance-name*`/log/agtoerr{01|02}` (see ***Note 4***)	Yes (see ***Note 5***)
Monitored storage information (see ***Note 6***)	utlprm.inf file (see ***Note 7***)	`/opt/jp1pc/agtd/agent/`*instance-name*`/utlprm.inf`	Yes
	Microcode version of the monitored storage subsystem	-	Yes
	Execution result of API for collecting performance information of Agent for RAID	-	Yes

Note 1*: *xxxx indicates the service key of each Agent. For details on the service keys of each Agent, see Appendix A.

***Note 2*:** For an Agent that monitors multiple storage subsystems and an Agent that monitors an application program that can start a set of multiple services at the same host, there is an instance name directory.

***Note 3*:** For details about the directories storing the trace log, see section 10.3.2.2

***Note 4*:** For details about the output format of the agent logs and how to change the directory in which the agent logs are stored, see 10.3.2.3.

***Note 5*:** The `jpcras` command collects agent log information only from the log output destination directory that is currently specified. If you change the output destination directory for the agent log, manually collect the agent log files that were output to the previous output destination directory.

***Note 6*:** Collect this information when using Agent for RAID.

***Note 7*:** You do not need to collect this file if the monitored storage subsystem is TagmaStore USP, or the Lightning 9900 Series or Lightning 9900V Series.

10.4.2.2 Operation Information

Collect the following information about the operation being performed when the error occurred:

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

10.4.2.3 Error Information

Obtain the following error information:

Messages output to the console, if the error occurred during command execution

10.4.2.4 Information Related to Performance Data (Agent for Platform (UNIX))

Collect the following information related to the performance data. The following table lists the information related to the performance data that is collected uniquely by Agent for Platform (UNIX) in an environment where Agent for Platform (UNIX) is installed.

Table 10.16 Performance Data Information Collected by Agent for Platform (UNIX)

Collected Item	Command				Collected by jpcras Command
	AIX	HP-UX	Solaris	Linux	
Processor information	`lsdev -Cc processor` `lsattr -El proc#`	`ioscan -fnC processor`	`psrinfo -v`	`/proc/cpuinfo`	Yes

System paging space information	`lsps -a`	`swapinfo –a`	`swap -l`	`/proc/meminfo`
				`free`
				`/proc/swaps`
				`procinfo`
					Yes

I/O statistical information	`lsdev -Cc disk`
	`iostat`
	`lscfg`
		`lsdev -C disk`
		`iostat`
		`/etc/netconfig`
			`iostat`
			`prtconf -D`
				`lsdev`
				`iostat`
				`procinfo`
					Yes

Collected Item	Command				Collected by jpcras Command
	AIX	HP-UX	Solaris	Linux	
Status information for the interprocess communication facilities	ipcs -a	ipcs -a	ipcs -a	ipcs –a ipcs –at ipcs –ap ipcs –ac ipcs –al ipcs –au	Yes
Static information for the file system	/etc/filesystems	/etc/fstab	/etc/vstab	/etc/fstab	Yes
Network status information (interface)	netstat –in lsattr -E -l inet0 /etc/inetd.conf	netstat -in ioscan -fnC lan /etc/rc.config.d/netconf /etc/inetd.conf	netstat -in /etc/inet/* /etc/defaultrouter /etc/inetd.conf	netstat –ni /etc/xinetd.d	Yes
Display of LAN device configuration and status	ifconfig -a	lanscan -v	ifconfig -a	ifconfig -a	Yes
NIS configuration file information	/etc/netsvc.conf	/etc/nsswitch.conf	/etc/inet.hosts	/etc/yp.conf /etc/nsswitch.conf	Yes
Network status information (protocol)	netstat -s	netstat -s	netstat -s	netstat –s	Yes
Summary information of file system allocation	repquota -a	repquota -a	repquota -a	repquota -a	Yes
Storage information collected by the Agent	agtu/agent/storage.d	agtu/agent/storage.d	agtu/agent/storage.d	agtu/agent/storage.d	Yes
Kernel bits information	bootinfo -K	getconf KERNEL_BITS	isainfo -kv	getconf LONG_BIT	Yes
Physical memory	bootinfo -r	grep -i Physical /var/adm/syslog/syslog.log	prtconf	/proc/meminfo free	Yes
System information (OS, version, and model)	uname -a oslevel	uname –a	uname –a	uname –a /etc/redhat-release	Yes
Runlevel information	who –r	who –r	who –r	runlevel	Yes
System date and time	date	date	date	date	Yes
Time zone	/etc/environment	/etc/TIMEZONE	/etc/TIMEZONE /etc/default/init	/etc/sysconfig/clock	Yes

Collected Item	Command				Collected by jpcras Command
	AIX	HP-UX	Solaris	Linux	
Directory information exported to the NFS client	/etc/exports/etc/xtab	/etc/exports/etc/xtab	/etc/dfs/dfstab /etc/dfs/sharetab etc/rmtab	/etc/exports /var/lib/nfs/xtab	Yes
Workgroup monitor configuration information	agtu/agent/wgfile	agtu/agent/wgfile	agtu/agent/wgfile	agtu/agent/wgfile	Yes
Event file monitor configuration information	agtu/agent/evfile	agtu/agent/evfile	agtu/agent/evfile	agtu/agent/evfile	Yes
Installed software information	lslpp –L all	swlist	pkginfo	rpm –qai	Yes
OS patch information	instfix –a	swlist –l patch what /stand/vmunix	showrev –a	rpm –qai	Yes
OS setup information	ls -l /unix ls -la /usr/lib/boot	N/A	N/A	N/A	Yes
List of processes	N/A	N/A	/bin/ps –elfZ (Solaris 10 only)	N/A	Yes
Solaris zone information	N/A	N/A	/usr/sbin/zoneadm list –cv (Solaris 10 only)	N/A	Yes
Network file system statistics	N/A	N/A	N/A	nfsstat	Yes
Host name information	N/A	N/A	N/A	/etc/hosts	Yes

10.4.2.5 Information Related to Performance Data (Agent for Oracle)

Collect the following information related to performance data for Agent for Oracle. If an error occurs (for example, in a network connection), you also need to collect the files on the machine, command execution results, and registry information. The following table lists the information related to the performance data that is collected uniquely by Agent for Oracle in an environment where Agent for Oracle is installed.

Table 10.17 Performance Data Information Collected by Agent for Oracle

Type of Information	Overview	File Name	Collected by jpcras Command
PDNL record information	Output result of `lsnrctl` (Oracle command)	`\opt\jp1pc\agto\agent\`*instance-name*`\pdnl.out`	Yes
PDNL record information	Output result of `lsnrctl` (Oracle command)	`\opt\jp1pc\agto\agent\`*instance-name*`\lsnrctl.status.pdnl.out.err.txt`	Yes
PDNL record information	Execution result of `uname`	`\opt\jp1pc\agto\agent\`*instance-name*`\sqlnet.tmp`	Yes
PDLS record information	Output result of `lsnrctl` (Oracle command)	`\opt\jp1pc\agto\agent\`*instance-name*`\pdls.out`	Yes
PDLS record information	Output result of `lsnrctl` (Oracle command)	`\opt\jp1pc\agto\agent\`*instance-name*`\lsnrctl.status.pdls.out.err.txt`	Yes

10.4.2.6 Other Information

Collect the following additional information:

- The command arguments that were specified, if the error occurred during command execution
- When Brocade switches are being monitored, the following contents about the switches that make up the proxy switches and fabric:
  - If the firmware of the monitored switch is version 4.4.0b or earlier, the contents of log file collected with the `supportShow` command
  - If the firmware of the monitored switch is version 4.4.0c or later, the contents of the log file collected with the `supportSave` command
- The contents of the following EFCM log files (when monitoring McDATA switches):
  - `Audit Log`
  - `Event Log`
  - `Fabric Log`
  - `Hardware Log`
  - `Link Incident Log`
  - `Product Status Log`
  - `Threshold Alert Log`

10.5 Data Collection Procedure

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

10.5.1 Windows Systems - Executing the Data Collection Command

Use the `jpcras` command to collect the data needed to determine the cause of an error.

Note: An OS user with the Administrators permissions must execute the procedure described below.

To execute the data collection command:

1. Log in to the host where the service subject to this data collection is installed.
2. At the command prompt, execute the following command to enable the command extension function of the command interpreter:

```
cmd /E:ON
```

3. In the `jpcras` command, specify the data to be collected and the storage folder for the data, and then execute the command.

   The following shows an example of specifying the command when all information that can be obtained by the `jpcras` command is to be stored in the `c:\tmp\jpc\mgr` folder:

```
jpcras c:\tmp\jpc\mgr all all
```

For details about the `jpcras` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

To execute the data collection command when running a logical host:

Follow these steps to collect data when running Collection Manager and the Agent on a logical host:

1. Make sure that the shared disk is online.

   Data on the logical host is stored on the shared disk. When you are operating an executing node, make sure that the shared disk is placed online before data is collected.

2. Execute the `jpcras` command, specifying the following for both the executing and standby nodes:
   - Directories that have data to be collected
   - Directories that will store collected data

   ***Example*:** To store all the available information in the `c:\tmp\jpc\mgr` directory:

```
jpcras c:\tmp\jpc\mgr all all
```

To collect all the data from Collection Managers and the Agents running on the physical and logical hosts of both the executing node and standby node, execute the `jpcras` command without specifying any arguments. If the Collection Managers and the Agents are running in a logical host environment, the log file on the shared disk is retrieved.

If the `jpcras` command is executed on a node where the shared disk is offline, files on the shared disk will not be obtained but the command will terminate normally without an error.

For details about the `jpcras` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

Note: To collect data, execute the data collection command at both the executing and standby nodes. To investigate the status before and after failover, you need to collect data from both the executing and standby nodes.

3. Collect data on cluster software.

   Examine data from the cluster software to determine if the problem comes from either the cluster software or from Collection Manager and the Agent. Examine the control requests made for operations such as starting or stopping Collection Manager and the Agent, and the results of the requests.

To check information about the operation:

Check and save the following information about the operation being performed when the error occurred:

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

To collect error information on screen displays:

Obtain printouts of the following:

- The Web browser
- The active screen when the application error occurred
- The error message dialog box

  Also print a copy of any detailed information.
- The Command Prompt window, if the error occurred during command execution

  To obtain printouts of the Command Prompt window, in the Properties window of the command prompt, do the following: Right-click the title bar of the Command Prompt window. The menu appears. From the menu, click **Properties** to display the Command Prompt Properties window.
  - **Options** tab

Select the **Quick Edit Mode** checkbox.

– **Layout** tab

Under **Screen Buffer Size**, set **Height** to `500`.

To display the Command Prompt window, do one of the following:

– Select **Start**, **Program**, **Accessories**, and then **Command Prompt**.

– Select **Start**, and then **Run**. Enter `cmd`, and then click the **OK** button.

Additional information to be collected:

Also, collect the following information:

- The contents of **System Information** under **Computer Management** or the contents of **System Information**, which is displayed by choosing **Accessories** and then **System Tools** (in Windows 2000)
- The contents of **System Information** which is displayed by clicking **Accessories** and then **System Tools** (in Windows Server 2003, Windows Server 2003 (IPF))
- The command arguments that were specified, if the error occurred during command execution

10.5.2 UNIX Systems - Executing the Data Collection Command

Use the `jpcras` command to collect the data needed to determine the cause of an error. Note that the user that executes the procedure described below must have root permissions.

To execute the data collection command:

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

   The following shows an example of specifying the command when all information that can be obtained by the `jpcras` command is to be stored in the `/tmp/jpc/mgr` directory:

   ```
   jpcras /tmp/jpc/mgr all all
   ```

   The data collected by the data collection command is stored in the specified directory in a compressed format by using either the `tar` or the `compress` command. The file name for data collected by the `jpcras` command is as follows:

   `jpcras`*YYMMDD*`.tar.Z`

   The date is added to the location indicated by *YYMMDD*.

For details about the `jpcras` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

To execute the data collection command when running a logical host:

Follow these steps to collect data when running Collection Manager and the Agent on a logical host:

1. Mount the shared disk.

   Data on the logical host is stored on the shared disk. When operating an executing node, make sure that the shared disk is mounted before data is collected.

2. Execute the `jpcras` command, specifying the following for both the executing and standby nodes:

   - Directories that have data to be collected
   - Directories that will store collected data

   ***Example*:** To store all the available information in the `/tmp/jpc/mgr` directory:
   `jpcras /tmp/jpc/mgr all all`

   You can store the collected data in a compressed format in a specified directory by using the `tar` command or the compress command. The compressed file name is as follows:

   Data collected by the `jpcras` command: jpcras YYMMDD.tar.Z

   The date of the file creation will be automatically added in the form YYMMDD.

   To collect all the data from Collection Managers and the Agents running on the physical and logical hosts of both the executing node and standby node, execute the `jpcras` command without specifying any arguments. If the Collection Managers and the Agents are running in a logical host environment, the log file on the shared disk is retrieved.

   If the `jpcras` command is executed on a node where the shared disk is not mounted, files on the shared disk will not be obtained but the command will end normally without an error.

   For details about the `jpcras` command, see the *HiCommand Tuning Manager Command Line Interface Guide*.

   Note: To collect data, execute the data collection command at both the executing and standby nodes. To investigate the status before and after failover, you need to collect data from both the executing and standby nodes.

3. Collect data on cluster software.

   Examine data from the cluster software to determine if the problem comes from either the cluster software or from Collection Manager and the Agent. Examine the control requests made for operations such as starting or stopping Collection Manager and the Agent, and the results of the requests.

To check information about the operation:

Check and save the following information about the operation being performed when the error occurred:

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

To collect error information:

Obtain the following error information:

- Messages output to the console, if the error occurred during command execution

Other information to be collected:

Also collect the following information:

- The command arguments that were specified, if the error occurred during command execution

10.6 Calling the 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, including the circumstances surrounding the error or failure, the exact content of any error messages displayed, and any troubleshooting data collected.

The worldwide Hitachi Data Systems Support Centers are:

- Hitachi Data Systems North America/Latin America

  San Diego, California, USA

  1 800 446-0744

- Hitachi Data Systems Europe

  Contact Hitachi Data Systems Local Support

- Hitachi Data Systems Asia Pacific

  North Ryde, Australia

  61 2 9325 3300

Chapter 11 Log Information

This chapter explains the log information output by the Tuning Manager series programs, as well as the format of the output information.

This chapter covers the following topics:

- About Log Information Output by Tuning Manager Products (see section11.1)
- Windows Event Log (see section 11.2)
- Syslog (UNIX Only) (see section 11.3)
- Common Message Log (see section 11.4)
- Agent log (Agent for Oracle and Agent for Microsoft SQL Server only) (see section 11.5)

11.1 Reviewing the Types of Log Files

This section explains the following types of log information output by the Tuning Manager series programs, as well as their output formats:

- Windows event log (Windows systems only)
- syslog (UNIX systems only)
- Common message log
- Agent log (Agent for Oracle and Agent for Microsoft SQL Server only)

The following sections explain each of these types of log information, as well as their output formats.

11.2 Windows Event Log

The Windows event log contains log information that notifies the user of statuses and problems in Windows systems.

Events output by the Tuning Manager series programs are displayed in the Application Log area of the Windows Event Viewer window. To distinguish events issued by Tuning Manager products, an identifier is displayed under Source in the Event Viewer window. For information about event identifiers issued by Tuning Manager series programs, see the *HiCommand Tuning Manager Messages Reference*.

This section explains the format of information output to the Windows event log by the Tuning Manager series programs. For details about the Windows event log, see the appropriate Windows documentation.

- Format of output information:

```
message-ID message
```

- Explanation of items (Table 11.1):

Table 11.1 Message Descriptions (Windows)

Item	Description	Length (bytes)
Message ID	The message ID is output here.	11
Message	The message is output here.	1 to 128

- Example of output:

```
KAVE00001-I Name Server has started. (host=host01, service=PN1001)
```

11.3 Syslog (UNIX only)

syslog contains log information that notifies users of statuses and problems in UNIX.

This section explains the format of information output to syslog by the Tuning Manager series programs. For details about syslog, see the UNIX documentation.

- Format of output information:

```
logged-date-and-time host-name program-name [process-ID]: message-ID message
```

- Explanation of items (Table 11.2):

Table 11.2 Message Descriptions (UNIX)

Item	Description	Length (bytes)
Log output date and time	The local date and time the log was output, in `month DD hh:mm:ss` format, is output here: - `month`: Indicates the month. - `DD`: Indicates the day. - `hh`: Indicates the hour. - `mm`: Indicates the minute. - `ss`: Indicates the second.	15
Host name	The name of the host is output here.	1 to 255
Program name	The name of the program is output here.	1 to 15
Process ID	The process ID, in decimal format, is output here.	1 to 10
Message ID	The message ID is output here.	11
Message	The message is output here.	1 to 128

- Example of output:

```
 Aug 25 09:21:07 host01 jpcnsvr[1123]: KAVE00001-I Name Server has started.
(host=host01, service=PN1001)
```

11.4 Common Message Log

The common message log contains log information that notifies the user of system statuses and of problems.

The language of the text in the common message log differs depending on the LANG environment variable set when the service was started or the command was executed. Therefore, if multiple language encodings are used, the user may encounter character strings from different languages.

This section explains the format of information that Tuning Manager outputs to the common message log.

- Format of output information:

```
logged-date-and-time program-name process-ID thread-ID source-file-name
line-number message-ID message
```

- Explanation of items (Table 11.3):

Table 11.3 Message Descriptions (Common)

Item	Description	Length (bytes)
Log output date and time	The local date and time the log was output, in `YYYY/MM/DD hh:mm:ss` format, is output here: ▪ `YYYY`: Indicates the year (4 digits). ▪ `MM`: Indicates the month. ▪ `DD`: Indicates the day. ▪ `hh`: Indicates the hour. ▪ `mm`: Indicates the minute. ▪ `ss`: Indicates the second.	19
Program name	The name of the program is output here.	8
Process ID	The process ID is output as a right-justified decimal value. For example, a process ID of **1000** is displayed as **00001000**.	8
Thread ID	The thread ID is output as a right-justified decimal value. For example, a thread ID of **1000** is displayed as **00001000**.	8
Source file name	The source file name is output here. The 16th and subsequent characters of any source file names having 16 or more characters are not output.	15

Item	Description	Length (bytes)
Line number	A right justified decimal value. For example, a line value of **100** is displayed as **0100**.	4
Message ID	The message ID is output here.	11
Message	The message is output here.	1 to 128

- Example of output:

```
2002/01/20 22:20:10 jpcnsvr  00000153 00000004 natext. cpp 0230 KAVE00001-I
Name Server has started. (host=host01, service=PN1001)
```

11.5 Agent log (Agent for Oracle and Agent for Microsoft SQL Server only)

The *agent log* contains log information for the processing related to record collection. This log is output by Agent for Oracle and Agent for Microsoft SQL Server. When an error occurs, you can use this log to obtain detailed information for the above processing.

This section describes the formats of the information of Agent for Oracle and Agent for Microsoft SQL Server that is output to the agent log.

- Format of output information:

  For Agent for Oracle:

  yyyy/mm/dd hh:mm:ss.sss `agto` *PID inf1 inf2 inf3 MessageID Message*

  For Agent for Microsoft SQL Server:

  yyyy/mm/dd hh:mm:ss.sss `agtq` *PID inf1 inf2 inf3 MessageID Message*

- Explanation of items (Table 11.4):

Table 11.4 Items Output to the Agent Log

Item	Description
yyyy/mm/dd	Date at which the log information is output (*yyyy*: year, *mm*: month, *dd*: day)
hh:mm:ss.sss	Local time at which the log information is output (*hh*: hour, *mm*: minute, *ss*: second, *sss*: millisecond)
`agto`	Name of the process that output the log information (`agto` for Agent for Oracle)
`agtq`	Name of the process that output the log information (`agtq` for Agent for Microsoft SQL Server)
PID	ID of the process that output the log information
inf1 to *inf3*	Maintenance information
MessageID	Message ID (see ***Note 1***)
Message	Message (see ***Note 1***)

Note 1: For details on the message contents, see the *HiCommand Tuning Manager Message Reference*.

***Notes*:**

- Do not change the time of the Agent host or the update time of the agent log file. If you do, the agent log file might not be output correctly because agent log information is output based on the time at which the log file was last updated.
- When operating Tuning Manager series programs on a logical host, make sure that you specify a path on the shared disk so that the agent log is output to the same destination for both the executing and standby nodes.

Appendix A List of Identifiers

This appendix describes the identifiers used with Tuning Manager and Agents.

When operating Tuning Manager and an Agent, and extracting performance data from the Store database of the Agent, sometimes an identifier is needed to indicate Tuning Manager and an Agent. Table A.1 shows the identifiers for Tuning Manager and Agents.

Table A.1 Identifiers for Tuning Manager and Agents

Main Use	Name	Identifier	Description
Commands	Product ID	P	Product ID of Tuning Manager
		D	Product ID of Agent for RAID
		E	Product ID of Agent for RAID Map
		T	Product ID of Agent for Platform (Windows)
		U	Product ID of Agent for Platform (UNIX)
		W	Product ID of Agent for SAN Switch
		N	Product ID of Agent for NAS
		O	Product ID of Agent for Oracle
		Q	Product ID of Agent for Microsoft SQL Server
		Z	Product ID of Agent for Microsoft Exchange Server
		R	Product ID of Agent for DB2
	Service key	all	All Collection Manager and Agent services
		mgr	Collection Manager services
		act	Action Handler service
		stat	Status Server service
		agtd	Service key for Agent for RAID
		agte	Service key for Agent for RAID Map and HTM Agent (see ***Note***)
		agtt	Service key for Agent for Platform (Windows)
		agtu	Service key for Agent for Platform (UNIX)
		agtw	Service key for Agent for SAN Switch
		agtn	Service key for Agent for NAS

Main Use	Name	Identifier	Description
		agto	Service key for Agent for Oracle
		agtq	Service key for Agent for Microsoft SQL Server
		agtz	Service key for Agent for Microsoft Exchange Server
		agtr	Service key for Agent for DB2

Note: You can specify `agte` as the service key for HTM Agent only by using the `jpctminfo` command.

The product ID is a part of the service ID. The service ID is needed when using a command to check the system configuration of Tuning Manager products, to back up performance data, and for other operations.

The service key is needed when using commands to start and stop Collection Manager and Agent, and for other operations.

For details about the service ID and the service key, see section 1.2.1.

Appendix B List of Processes

B.1 List of Collection Manager Processes

Table B.1 shows a list of Collection Manager processes. Note that the value in parentheses following the process name is the number of instances of that process that can be active at any one time.

Table B.1 Collection Manager Processes

Process Name (Process Count)		
Windows	**UNIX**	**Function**
jpcah.exe(1)	jpcah(1)	Action Handler service process
jpcep.exe(1)	jpcep(1)	Correlator service process
jpcmm.exe(1)	jpcmm(1)	Master Manager service process
jpcsto.exe(1)	jpcsto(1)	Master Store service process
jpcstatsvr.exe(1)	jpcstatsvr(1)	Status Server service process
jpcnsvr.exe(1)	jpcnsvr(1)	Name Server service process
jpctrap.exe(1)	jpctrap(1)	Trap Generator service process
jpctraps.exe(1)	jpctraps(1)	Trap Generator service trap transmission process ***Note***: Child process of the `jpctrap` process.
jpcvsvr.exe(1)	jpcvsvr(1)	View Server service process
stpqlpr.exe(1)	stpqlpr(1)	Store database backup or export process ***Note***: Child process of the `jpcsto` process.

B.2 List of Agent Processes

This section lists Agent processes.

Table B.2 shows a list of Agent processes. The value in parentheses following each process name is the number of instances of that process that can be active at the same time. The processes that can run and the number of processes are the same for an Agent that runs on a logical host.

Table B.2 Agent Processes

Process Name (Number of Processes)		
Windows	**UNIX**	**Function**
jpcagtx.exe(*n*) ***Note***: x indicates the product ID of each Agent.	jpcagtx(*n*) ***Note***: x indicates the product ID of each Agent.	The process of the Agent Collector service. One `jpcagtd` process is started for each Agent for RAID. ***Note***: This process is specific to each Agent.
jpcah.exe(1)	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.
jpcstatsvr.exe(1)	jpcstatsvr(1)	The process of the Status Server service. Only one `jpcstatsvr` process is started for each host even if Tuning Manager and multiple Agents are installed on the same host
jpcsto.exe(*n*)	agtx/jpcsto(*n*) ***Note***: x indicates the product ID of each Agent.	The process of the Agent Store service. One process is started for each Agent instance.
stpqlpr.exe(1)	stpqlpr(1)	The process for backup and export of the Store database. ***Note***: This is a child process of the `jpcagto` process.
jpcOcollect.exe(*n*)	jpcOcollect(*n*)	The process for collecting performance data. One process is started for each instance. ***Note***: This process is a child process of the `jpcagto` process.
-	jpc_hostutil (1)	The program for obtaining host information. ***Note***: This is a child process of the `jpcagtu` process.

Process Name (Number of Processes)		
Windows	**UNIX**	**Function**
-	jpc_process(1)	The program for obtaining 64-bit process information ***Note***: This is a child process of the `jpcagtu` process, and is not installed in HP-UX, HP-UX (IPF), AIX, and Linux.
-	jpc_ufss (1)	The process for collecting performance data for User File System Storage (PD_UFSS) records. ***Note***: This is a child process of the `jpcagtu` process.

Appendix C List of Port Numbers

This appendix lists the port numbers used for Collection Manager and the Agent.

You can change the port numbers used for Collection Manager and Agent to suit your environment. For details about changing a port number, see the *HiCommand Tuning Manager Installation Guide*. Note that the TCP/IP protocol is used.

Note: Collection Manager and the Agent support static NAT (Basic NAT) that performs one-to-one address translations do not support dynamic NAT or NAPT (IP Masquerade, NAT+) that include a port translation function.

C.1 Port Numbers Used for Collection Manager and the Agent

Table C.1 lists the port numbers used for Collection Manager and the Agent. In the default settings, each time a service other than the Name Server service, the View Server service (between Performance Reporter and the View Server service), or the Status Server service is started, the service automatically uses a port number not being used by the system. The `jpcnsconfig port` command can be used to fix the port numbers used by the system.

Table C.1 Port Numbers Used for Collection Manager and the Agent

Service Name	Parameter	Port Number	Purpose
Name Server service	jp1pcnsvr	22285	Used when a service establishes a communications connection to the Name Server.
Master Manager service	jp1pcmm	20271 See ***Note 1***	Used when a service establishes a communications connection to the Master Manager.
Master Store service	jp1pcsto	20272 See ***Note 1***	Used when an Agent records an event.
Correlator service	jp1pcep	20273 See ***Note 1***	Used when an Agent sends an event to Tuning Manager.
Trap Generator service	jp1pctrap	20274 See ***Note 1***	Used when the Trap Generator service sends an SNMP trap
View Server service	jp1pcvsvr2	20276 See ***Note 1***	Used when an Agent sends an event and a report to Performance Reporter.
View Server service (Between Performance Reporter and View Server Service)	jp1pcvsvr	22286	Used when a user logs into Tuning Manager, or runs Performance Reporter.
Status Server service	jp1pcstatsvr	22350	Used when a user checks the status of a service.

Service Name	Parameter	Port Number	Purpose
Action Handler service	jp1pcah	20275 See ***Note 1***	Used when the Action Handler service executes an action.
Agent Store service	jp1pcstox[*nnn*] See ***Note 2*** and ***Note 3***	Not Applicable See ***Note 4***	Used when the Agent Store service records performance data, or obtains a historical report.
Agent Collector service	jp1pcagtx[*nnn*] See ***Note 2*** and ***Note 3***		Used when the Agent Store service binds an alarm, or obtains a real-time report.

Note 1: The `jpcnsconfig port` command displays this port number as a default value. You can change this default setting to a desired number.

Note 2: x indicates the product ID of the Agent.

Note 3: 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.

Note 4: When you set a port number by executing the `jpcnsconfig port` command for the first time after an instance registration, a port number not being used by the system is displayed. You can change the displayed port number to a desired number. Once you set a port number by executing the `jpcnsconfig port` command, this number will be maintained unless changed.

C.2 Direction of Transmission Through a Firewall

If you want to establish a firewall between Collection Manager and Agent, you must set fixed port numbers for all Collection Manager and Agent services. To enable communications to pass through the firewall, port numbers that you set must comply with the directions shown in the following table 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 C.2 Directions of Transmission Through a Firewall Between Collection Manager and the Agent

Service Name	Parameter	Passage Directions
Name Server service	jp1pcnsvr	Tuning Manager host `<--` Agent host
Master Manager service	jp1pcmm	
Master Store service	jp1pcsto	
Correlator service	jp1pcep	
Trap Generator service	jp1pctrap	
View Server service	jp1pcvsvr2	
Status Server service	jp1pcstatsvr	Tuning Manager host `<-->` Agent

Service Name	Parameter	Passage Directions
Action Handler service	jp1pcah	Tuning Manager host <--> Agent host
Agent Store service	jp1pcsto*x* [*nnn*] See ***Note 1*** and ***Note 2***	Agent host <-- Tuning Manager host`
Agent Collector service	jp1pcagt*x*[*nnn*] See ***Note 1*** and ***Note 2***	

Note 1: ***x*** indicates the product ID of the Agent.

Note 2: When creating multiple instances, an ascending number (*nnn*) will be appended to the instances created after the first time. No number will be appended to the instance created first.

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

- Specifying the `proxy` option of the `jpcctrl dump` command or the `jpcctrl list` command so that communications are performed through Tuning Manager. For details about the `proxy` option of the `jpcctrl dump` and `jpcctrl list` commands, see the *HiCommand Tuning Manager Command Line Interface Guide*.
- Setting port numbers so that connections can be initiated between Agent Hosts (Tuning Agent host <--> Agent host), and communications can be passed through the firewall:
  - Action Handler service
    (jp1pcah parameter)
  - Agent Store service
    (jp1pcsto*x* parameter)[see *Note*]
  - Agent Collector service
    (jp1pcagt*x* parameter)[see *Note*]

  Note: ***x*** indicates the product ID of the Agent.

  For Agent Store and Agent Collector: 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.

C.3 Port Numbers of Ports Between Agent for SAN Switch and Proxy Switch or EFCM

Agent for SAN Switch, Proxy Switch, and EFCM use port numbers that are not being used by the system.

C.4 Port Numbers of Ports Between Agent for NAS and the NAS System

Agent for NAS uses port numbers that are not being used by the NAS system.

The port number `20265` is reserved in the NAS system. You cannot change this port number.

C.5 Transmission Through a Firewall Between Agent for SAN Switch and Proxy Switch or EFCM

If a firewall is set up between Agent for SAN Switch and Proxy Switch or EFCM, you cannot use a port number for transmission through the firewall. Make sure that you set an IP address for such transmissions.

C.6 Transmission Through a Firewall Between Agent for NAS and the NAS System

If a firewall is set up between Agent for NAS and the NAS system, you cannot use a port number for transmission through the firewall on the Agent side. Make sure that you set an IP address for such transmissions.

On the NAS system side, specify the port number `20265`.

C.7 Setting for NIC in an Environment Where Multiple NICs Are Used

If an Agent host has multiple IP addresses in an environment where multiple NICs are used, the Agent cannot specify a specific NIC when an Agent is connected to a NAS system, Proxy Switch, or EFCM.

C.8 Precautions for Windows Server 2003 and Windows Server 2003 (IPF) Service Pack 1

C.8.1 Collection Manager

When using Collection Manager in an environment where Windows Firewall is enabled, the port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.3 Services of Collection Manager

Service Name	Parameter
The Name Server service	jp1pcnsvr
The Master Manager service	jp1pcmm
The Master Store service	jp1pcsto
The Correlator service	jp1pcep
The Trap Generator service	jp1pctrap
The View Server service	jp1pcvsvr2
The View Server service (between Performance Reporter and the View Server service)	jp1pcvsvr
The Action Handler service	jp1pcah
The Status Server service	jp1pcstatsvr (see ***Note***)

Note: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers to the Windows Firewall exceptions list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers to the Windows Firewall exceptions list.

```
netsh firewall add portopening protocol=TCP
      port=22285 (port-number-corresponding-to-parameter-jp1pcnsvr) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20271 (port-number-corresponding-to-parameter-jp1pcmm) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20272 (port-number-corresponding-to-parameter-jp1pcsto) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20273 (port-number-corresponding-to-parameter-jp1pcep) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20274 (port-number-corresponding-to-parameter-jp1pctrap) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20276 (port-number-corresponding-to-parameter-jp1pcvsvr2) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=22286 (port-number-corresponding-to-parameter-jp1pcvsvr) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20275 (port-number-corresponding-to-parameter-jp1pcah) (see Note)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
```

```
port=22350 (port-number-corresponding-to-parameter-jp1pcstatsvr)(see Note)
name="HiCommand Tuning Manager" mode=ENABLE
```

Note: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Tuning Manager, remove all **HiCommand Tuning Manager** entries displayed in **Programs and Services**.

C.8.2 Agent for RAID

When using Agent for RAID in an environment where Windows Firewall is enabled, the port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.4 Services of Agent for RAID

Service Name	Parameter
The Action Handler service	`jp1pcah`
The Agent Store service	`jp1pcstod`[*nnn*] (see ***Note 1***)
The Agent Collector service	`jp1pcagtd`[*nnn*] (see ***Note 1***)
The Status Server service	`jp1pcstatsvr` (see ***Note 2***)

Note 1: A port number must be added for each created instance. If multiple instances have been created, a sequence number (*nnn*) is appended to the second and later instances. A sequence number is not appended to the first created instance.

Note 2: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers to the Windows Firewall exceptions list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers to the Windows Firewall exceptions list.

```
netsh firewall add portopening protocol=TCP
      port=20281 (port-number-corresponding-to-parameter-jp1pcstod[nnn])(see Note 1)
      name="HiCommand Tuning Manager - Agent for RAID" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20282 (port-number-corresponding-to-parameter-jp1pcagtd[nnn])(see Note 1)
      name="HiCommand Tuning Manager - Agent for RAID" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=20275 (port-number-corresponding-to-parameter-jp1pcah)(see Note 2)
      name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=22350 (port-number-corresponding-to-parameter-jp1pcstatsvr)(see Note 2)
      name="HiCommand Tuning Manager" mode=ENABLE
```

Note 1: The port number might differ depending on the environment. Add the port number that was checked in step 1.

Note 2: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for RAID, remove all **HiCommand Tuning Manager - Agent for RAID** entries displayed in **Programs and Services**. If you are uninstalling all of the HTM products from the same machine, remove the **HiCommand Tuning Manager** entry as well. If an HTM product is to remain on the machine, do not remove the **HiCommand Tuning Manager** entry.

C.8.3 Agent for RAID Map and Agent for Platform (Windows)

When using Agent for RAID Map and Agent for Platform (Windows) in an environment where Windows Firewall is enabled, the port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.5 Services of Agent for RAID Map and Agent for Platform (Windows)

Service Name	Parameter
The Action Handler service	jp1pcah
The Agent Store service	jp1pcstot
	jp1pcstoe
The Agent Collector service	jp1pcagtt
	jp1pcagte
The Status Server service	jp1pcstatsvr (see ***Note***)

Note: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers to the Windows Firewall exceptions list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers to the Windows Firewall exceptions list:

```
netsh firewall add portopening protocol=TCP
    port=20282 (the port number that corresponds to the parameter jp1pcagte)(see Note
1)
    name=" HiCommand Tuning Manager - Agent" mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20281 (the port number that corresponds to the parameter jp1pcstoe)(see Note
1)
    name=" HiCommand Tuning Manager - Agent " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20280 (the port number that corresponds to the parameter jp1pcagtt)(see Note
2)
    name=" HiCommand Tuning Manager - Agent for Platform " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port= 20279 (the port number that corresponds to the parameter jp1pcstot)(see Note
2)
    name=" HiCommand Tuning Manager - Agent for Platform " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20275 (the port number that corresponds to the parameter jp1pcah)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=22350 (the port number that corresponds to the parameter jp1pcstatsvr)(see
Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
```

Note 1: The port number might differ depending on the environment. Add the port number that was checked in step 1.

Note 2: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for RAID Map and Agent for Platform (Windows), remove all **HiCommand Tuning Manager - Agent** and **HiCommand Tuning Manager - Agent for Platform** entries displayed in **Programs and Services**. If you are uninstalling all of the HTM products from the same machine, remove the **HiCommand Tuning Manager** entry as well. If an HTM product is to remain on the machine, do not remove the **HiCommand Tuning Manager** entry. If Agent for Microsoft Exchange Server has been installed, remove the service information of Agent for Microsoft Exchange Server registered in the exceptions list.

C.8.5 Agent for SAN Switch

When using Agent for SAN Switch in an environment where Windows Firewall is enabled, the service programs and port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.6 Services of Agent for SAN Switch

Service Name	Parameter	Program
The Action Handler service	jp1pcah	--
The Agent Store service	jp1pcstow[*nnn*] (see ***Note 1***)	--
The Agent Collector service	jp1pcagtw[*nnn*] (see ***Note 1***)	jpcagtw.exe
The Status Server service	jp1pcstatsvr (see ***Note 2***)	--

Note 1: A port number must be added for each created instance. If multiple instances have been created, a sequence number (*nnn*) is appended to the second and later instances. A sequence number is not appended to the first created instance.

Note 2: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers and service programs to the Windows Firewall exception list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers and service programs to the Windows Firewall exceptions list.

```
netsh firewall add allowedprogram
    program="installation-folder\jp1pc\agtw\agent\jpcagtw.exe"
    name=" HiCommand Tuning Manager - Agent for SAN Switch " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20282 (the port number that corresponds to the parameter jp1pcagtw[nnn])(see
Note 1)
    name="HiCommand Tuning Manager - Agent for SAN Switch" mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20281 (the port number that corresponds to the parameter jp1pcstow[nnn])(see
Note 1)
    name="HiCommand Tuning Manager - Agent for SAN Switch" mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20275 (the port number that corresponds to the parameter jp1pcah)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=22350 (port-number-corresponding-to-parameter-jp1pcstatsvr)(see Note 2)
      name="HiCommand Tuning Manager" mode=ENABLE
```

Note 1: The port number might differ depending on the environment. Add the port number that was checked in step 1.

Note 2: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for SAN Switch, remove all **HiCommand Tuning Manager - Agent for SAN Switch** entries displayed in **Programs and Services**. If you are uninstalling all of the HTM products from the same machine, remove the **HiCommand Tuning Manager** entry as well. If an HTM product is to remain on the machine, do not remove the **HiCommand Tuning Manager** entry.

C.8.6 Agent for NAS

When using Agent for NAS in an environment where Windows Firewall is enabled, the service programs and port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.7 Services of Agent for NAS

Service Name	Parameter	Program
The Action Handler service	`jp1pcah`	-
The Agent Store service	`jp1pcston[`*nnn*`]` (see ***Note 1***)	-
The Agent Collector service	`jp1pcagtn[`*nnn*`]` (see ***Note 1***)	`jpcagtn.exe`
The Status Server service	`jp1pcstatsvr` (see ***Note 2***)	-

Note 1: A port number must be added for each created instance. If multiple instances have been created, a sequence number (*nnn*) is appended to the second and later instances. A sequence number is not appended to the first created instance.

Note 2: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers and service programs to the Windows Firewall exception list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers and service programs to the Windows Firewall exceptions list.

```
netsh firewall add allowedprogram
    program="installation-folder\jp1pc\agtn\agent\jpcagtn.exe "
    name=" HiCommand Tuning Manager - Agent for Network Attached Storage "
    mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20282(port-number-corresponding-to-the-parameter-jp1pcagtn[nnn])(see Note 1)
    name=" HiCommand Tuning Manager - Agent for Network Attached Storage "
    mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20281(port-number-corresponding-to-the-parameter-jp1pcston[nnn])(see Note 1)
    name=" HiCommand Tuning Manager - Agent for Network Attached Storage "
    mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20275(port-number-corresponding-to-the-parameter-jp1pcah)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=22350 (port-number-corresponding-to-parameter-jp1pcstatsvr)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
```

   Note 1: The port number might differ depending on the environment. Add the port number that was checked in step 1.

   Note 2: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for NAS, remove all **HiCommand Tuning Manager - Agent for Network Attached Storage** entries displayed in **Programs and Services**. If you are uninstalling all of the HTM products from the same machine, remove the **HiCommand Tuning Manager** entry as well. If an HTM product is to remain on the machine, do not remove the **HiCommand Tuning Manager** entry.

C.8.7 Agent for Oracle

When using Agent for Oracle in an environment where Windows Firewall is enabled, the port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.8 Services of Agent for Oracle

Service Name	Parameter
The Action Handler service	jp1pcah
The Agent Store service	jp1pcstoo[*nnn*] (see ***Note 1***)
The Agent Collector service	jp1pcagto[*nnn*] (see ***Note 1***)
The Status Server service	jp1pcstatsvr(see ***Note 2***)

Note 1: A port number must be added for each created instance. If multiple instances have been created, a sequence number (*nnn*) is appended to the second and later instances. A sequence number is not appended to the first created instance.

Note 2: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers to the Windows Firewall exceptions list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers to the Windows Firewall exceptions list.

```
netsh firewall add portopening protocol=TCP
    port=20281 (port number for parameter jp1pcagto[nnn])(see Note 1)
    name="HiCommand Tuning Manager - Agent for Oracle " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20282 (port number for parameter jp1pcstoo[nnn])(see Note 1)
    name="HiCommand Tuning Manager - Agent for Oracle " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20275 (port number for parameter jp1pcah)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=22350 (port-number-corresponding-to-parameter-jp1pcstatsvr)(see Note 2)
      name="HiCommand Tuning Manager" mode=ENABLE
```

   Note 1: The port number might differ depending on the environment. Add the port number that was checked in step 1.

   Note 2: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for Oracle, remove all **HiCommand Tuning Manager - Agent for Oracle** entries displayed in **Programs and Services**. If you are uninstalling all of the HTM products from the same machine, remove the **HiCommand Tuning Manager** entry as well. If an HTM product is to remain on the machine, do not remove the **HiCommand Tuning Manager** entry.

C.8.8 Agent for Microsoft SQL Server

When using Agent for Microsoft SQL Server in an environment where Windows Firewall is enabled, the port numbers used by the services indicated in the following table must be added to the Windows Firewall exceptions list.

Table C.9 Services of Agent for Microsoft SQL Server

Service Name	Parameter
The Action Handler service	`jp1pcah`
The Agent Store service	`jp1pcstoq`[*nnn*] (see ***Note 1***)
The Agent Collector service	`jp1pcagtq`[*nnn*] (see ***Note 1***)
The Status Server service	`jp1pcstatsvr` (see ***Note 2***)

Note 1: A port number must be added for each created instance. If multiple instances have been created, a sequence number (*nnn*) is appended to the second and later instances. A sequence number is not appended to the first created instance.

Note 2: If the status management function is enabled, the port number used by this service must be added to the Windows Firewall exceptions list.

To add port numbers to the Windows Firewall exceptions list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   After setting the port numbers, execute the `jpcnsconfig port list` command to make sure that the port numbers have been set correctly. For details about how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to add the port numbers to the Windows Firewall exceptions list.

```
netsh firewall add portopening protocol=TCP
    port=20281 (port-number-corresponding-to-the-parameter-jp1pcagtq[nnn])(see Note 1)
    name="HiCommand Tuning Manager - Agent for Microsoft(R) SQL Server "
    mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20282 (port-number-corresponding-to-the-parameter-jp1pcstoq[nnn])(see Note 1)
    name="HiCommand Tuning Manager - Agent for Microsoft(R) SQL Server "
    mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20275 (port-number-corresponding-to-the-parameter-jp1pcah)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=22350 (port-number-corresponding-to-parameter-jp1pcstatsvr)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
```

Note 1: The port number might differ depending on the environment. Add the port number that was checked in step 1.

Note 2: If a port number is changed by using the `jpcnsconfig port` command, the port number will be different from the port number indicated above. Add the port number that was checked in step 1.

3. Check the contents of the Windows Firewall exceptions list.

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for Microsoft SQL Server, remove all **HiCommand Tuning Manager - Agent for Microsoft(R) SQL Server** entries displayed in **Programs and Services**. If you are uninstalling all of the HTM products from the same machine, remove the **HiCommand Tuning Manager** entry as well. If an HTM product is to remain on the machine, do not remove the **HiCommand Tuning Manager** entry.

C.8.9 Agent for Microsoft Exchange Server

When Agent for Microsoft Exchange Server is used in an environment in which Windows Firewall is enabled, the port numbers used by the services listed in Table C.10 need to be registered in the Windows Firewall exceptions list.

Table C.10 Services of Agent for Microsoft Exchange Server

Service Name	Parameter
Action Handler service	jp1pcah
Agent Store service	jp1pcstoz
Agent Collector service	jp1pcagtz
Status Server service	jp1pcstatsvr (see ***Note***)

Note: This needs to be registered when the status management function is enabled.

To register the port numbers in the Windows Firewall exceptions list:

1. Execute the `jpcnsconfig port define` command to set the port numbers used by the above services.

   Once the port numbers are set, execute the `jpcnsconfig port list` command to check that they were set correctly. For details on how to set and check port numbers, see the *HiCommand Tuning Manager Installation Guide*.

2. Execute the `netsh` command to register the numbers in the Windows Firewall exceptions list.

```
netsh firewall add portopening protocol=TCP
    port=20282 (port number for the jp1pcagtz parameter)(see Note 1)
    name=" HiCommand Tuning Manager - Agent for MSExchange " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20281 (port number for the jp1pcstoz parameter)(see Note 1)
    name=" HiCommand Tuning Manager - Agent for MSExchange " mode=ENABLE
netsh firewall add portopening protocol=TCP
    port=20275 (port number for the jp1pcah parameter)(see Note 2)
    name="HiCommand Tuning Manager" mode=ENABLE
netsh firewall add portopening protocol=TCP
      port=22350 (port number for the jp1pcstatsvr parameter)(see Note 2)
      name="HiCommand Tuning Manager" mode=ENABLE
```

   Note 1: Port numbers differ depending on the environment used. Register the port numbers checked in step 1.

   Note 2: The port number will differ from those shown here when changed during execution of the `jpcnsconfig port` command. Register the port numbers checked in step 1.

3. Check the items registered in the Windows Firewall exceptions list..

   From the Control Panel, select **Windows Firewall** and then the **Exceptions** tab. The names of services added to the exceptions list are displayed in **Programs and Services**, and those whose corresponding check boxes are selected are currently enabled exceptions.

   To remove an item from the exceptions list, select **Windows Firewall** and then the **Exceptions** tab. From the items displayed in **Programs and Services**, select the item to be deleted, and then select **Delete**.

   To temporarily disable an exception, in **Programs and Services**, clear the check box corresponding to that exception.

Note: When uninstalling Agent for Microsoft Exchange Server, delete everything displayed for **HiCommand Tuning Manager - Agent for MSExchange** in **Programs and Services**. When uninstalling all HTM products on the same machine, also delete everything for **HiCommand Tuning Manager**. If other HTM products are still installed, do not delete **HiCommand Tuning Manager**.

Appendix D Agent Service Properties

This appendix lists the Agent Store service properties and the Agent Collector service properties. Both properties are output by using Performance Reporter commands.

D.1 Agent Store Service Properties

The following table lists the Agent Store service properties. These properties are output by executing `jpcaspsv output` commands of Performance Reporter.

Table D.1 Properties of Agent Store Service

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: ▪ 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: ▪ hour ▪ day ▪ 2 days ▪ 3 days ▪ 4 days ▪ 5 days ▪ 6 days ▪ week ▪ month ▪ year

Parameter Name	Element Name		Description
Service (continued)	product-interval	day-drawer	Outputs the duration for which daily records of the PI record type are stored. The following values are output: ▪ 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: ▪ week ▪ month ▪ year
		month-drawer	Outputs the duration for which monthly records of the PI record type are stored. The following values are output: ▪ 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 ***Note.***
	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 ***Note***. This cannot be used for Agents other than Agent for Platform (UNIX).

Note: For details about the record ID of each record, see the chapter that describes records in the following manuals.

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

D.2 Agent Collector Service Properties

Table D.2 lists the properties of the Agent Collector service. These properties are output by executing `jpcasrec output` commands of Performance Reporter.

Table D.2 Properties of the Agent Collector Service

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 ***Note***.
		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 the chapter that describes records in the following manuals.

- *HiCommand Tuning Manager Hardware Reports Reference*
- *HiCommand Tuning Manager Operating System Reports Reference*
- *HiCommand Tuning Manager Application Reports Reference*

Appendix E List of Files and Directories

This appendix lists files and directories of the Tuning Manager series programs for each OS.

The following shows the installation directory of the Tuning Manager series programs for each OS:

- In Windows:

  You may specify any folder as the installation folder of Tuning Manager series programs. The default is the following:

  The default installation folder for Windows (other than Windows Server 2003 x64):

  system-drive\Program Files\HiCommand\TuningManager\jp1pc\

  The default installation folder for Windows Server 2003 x64:

  system-drive\Program Files (x86)\HiCommand\TuningManager\jp1pc

- In UNIX:

  The installation directory of the Tuning Manager series programs is /opt/jp1pc/.

Note: For a file of the Tuning Manager series in UNIX, a file to which the extension .lck is added might be created in the same directory. For example, if the file is /opt/jp1pc/jpcns.ini, /opt/jp1pc/jpcns.ini.lck might be created. Do not change or delete this file because, in UNIX, the Tuning Manager series uses this file internally.

E.1 List of Files and Directories Shared by Collection Manager and Agent

E.1.1 Windows Systems

Table E.1 provides a list of files and folders shared by the Windows versions of Collection Manager and Agent.

Table E.1 Files and Folders Shared by Collection Manager and Agents (Windows)

Files and Folders Shared by Collection Manager and Agents (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\	—	HTM common root folder
	jpccomm.ini	Common start information file for each component
	jpccomm.ini.model	Sample of a common start information file for each component
	jpchosts	Host information setup file
	jpchosts.model	Sample of a host information setup file
	jpcns.ini	Service configuration information file
	jpcns.ini.model	Sample of a service configuration information file
	jpcns_backup.ini (see ***Note***)	Backup of a service configuration information file
	jpcplist.ini	Product definition file
	jpcplist.ini.model	Sample of the product definition file
installation-folder\bin\	—	Folder for storing commands
	agtsolmlt.bat	Internal command
	agtsolrm.bat	
	getpinfo.exe	
	jpccvtmdl.exe	
	jpciniupdate.exe	
	jpcmkindex.exe	
	mdlcvt_bkup.bat	
	mdlcvt_exec.bat	
	nscfg.exe	
	pfm_inst.bat	

Files and Folders Shared by Collection Manager and Agents (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\bin\action\	—	Root folder for the Action Handler service
	jpcah.exe	Program for executing the Action Handler service
	jpcah.ini	Action Handler service startup information file
	jpcah.ini.model	Sample of an Action Handler service startup information file
installation-folder\bin\action\log\	—	Folder for storing internal log files of the Action Handler service
	msglog01	Log file
	msglog02	Log file (alternative file)
	nslog01	Log file
	nslog02	Log file (alternative file)
installation-folder\bin\statsvr\	—	Root folder for the Status Server service
	jpcstatsvr	Program for executing the Status Server service
	jpcstat.ini	The startup information file for the Status Server service
	jpcstat.ini.model	Sample of a startup information file for the Status Server service
installation-folder\bin\statsvr\log\	—	Folder for storing internal log files of the Status Server service
installation-folder\lib\	—	Folder for storing message catalogs
	jpcmgrmsg.dll	Internal file
installation-folder\log\	—	Folder for storing log files
	jpclog01	Common message log file
	jpclog02	Common message log file (alternative file)
installation-folder\patch_files\	—	Folder for storing patch files
installation-folder\pid\	—	Folder for storing pid lock files
installation-folder\tools\	—	Folder for storing commands
	jpcagtsetup.exe	Command for adding and setting up the Agent
	jpcctrl.exe	Command for managing services
	jpchasetup.exe	Command for setting up a logical host environment
	jpcimevt.exe	Internal command

Files and Folders Shared by Collection Manager and Agents (Windows)		
Folder Name	**File Name**	**Description**
	jpcinslist.exe	Command for listing an instance environment
	jpcinssetup.exe	Command for setting up an instance environment
	jpcinsunsetup.exe	Command for performing an unsetup of an instance environment
	jpcnsconfig.exe	Command for defining a service configuration
	jpcnshostname.exe	Command for setting up a host name
	jpcras.bat	Command for collecting maintenance information
	jpcresto.exe	Command for restoring a store database
	jpcstart.exe	Command for issuing a startup request for a service
	jpcstop.exe	Command for issuing a stop request for a service
	jpcstsetup.exe	Command for controlling statuses
	jpctminfo.exe	Command for displaying product information
installation-folder\tools\log\	—	Folder for storing command log files
	msglog01	Log file
	msglog02	Log file (alternative file)
	nslog01	Log file
	nslog02	Log file (alternative file)

Note: This file is used internally by the Tuning Manager series, and should not be changed or deleted.

E.1.2 UNIX Systems

Table E.2 provides a list of files and directories shared by the UNIX versions of Collection Manager and Agent.

Table E.2 Files and Directories Shared by Collection Manager and Agents (UNIX)

Files and Directories Shared by Collection Manager and Agents (UNIX)		
Folder Name	**File Name**	**Description**
/opt/jp1pc/	—	HTM common root directory
	jpc_start	Service automatic start script file (see ***Note 1***)
	jpc_start.model	Sample of a service automatic start script file
	jpc_stop	Service automatic stop script file (see ***Note 1***)
	jpc_stop.model	Sample of a service automatic stop script file
	jpccomm.ini	Common start information file for each component
	jpccomm.ini.model	Sample of a common start information file for each component
	jpchosts	Host information setup file
	jpchosts.model	Sample of a host information setup file
	jpcns.ini	Service configuration information file
	jpcns.ini.model	Sample of a service configuration information file
	jpcns_backup.ini (see ***Note 2***)	Backup of a service configuration information file
	jpcplist.ini	Product definition file
	jpcplist.ini.model	Sample of the product definition file
/opt/jp1pc/bin/	—	Directory for storing commands
	agtsolmlt	Internal command
	agtsolrm	
	getpinfo	
	iniedit	
	jpccvtmdl	
	jpciniupdate	
	jpcmkindex	
	mdlcvt_bkup	

Files and Directories Shared by Collection Manager and Agents (UNIX)		
Folder Name	**File Name**	**Description**
	mdlcvt_exec	Internal command
	nscfg	
	pfm_inst	
/opt/jp1pc/bin/action/	—	Root directory for the Action Handler service
	jpcah	Program for executing the Action Handler service
	jpcah.ini	Action Handler service startup information file
	jpcah.ini.model	Sample of an Action Handler service startup information file
/opt/jp1pc/bin/action/log/	—	Directory for storing internal log files of the Action Handler service
	msglog01	Log file
	msglog02	Log file (alternative file)
	nslog01	Log file
	nslog02	Log file (alternative file)
/opt/jp1pc/bin/statsvr/	—	Root directory for the Status Server service
	jpcstatsvr	Program for executing the Status Server service
	jpcstat.ini	The startup information file for the Status Server service
	jpcstat.ini.model	Sample of a startup information file for the Status Server service
/opt/jp1pc/bin/statsvr/log/	—	Directory for storing internal log files of the Status Server service
/opt/jp1pc/lib/	—	Common library storage directory
/opt/jp1pc/log/	—	Directory for storing log files
	jpclog01	Common message log file
	jpclog02	Common message log file (alternative file)
/opt/jp1pc/nls/$LANG$/	—	Directory for storing message catalogs
	jpcmgrmsg.cat	Internal file
/opt/jp1pc/patch_files/	—	Directory for storing patch files
/opt/jp1pc/pid/	—	Directory for storing pid lock files

Files and Directories Shared by Collection Manager and Agents (UNIX)		
Folder Name	**File Name**	**Description**
/opt/jp1pc/tools/	—	Directory for storing commands
	jpcagtsetup	Command for adding and setting up the Agent
	jpcctrl	Command for managing services
	jpchasetup	Command for setting up a logical host environment
	jpcimevt	Internal command
	jpcinslist	Command for listing an instance environment
	jpcinssetup	Command for setting up an instance environment
	jpcinsunsetup	Command for unsetting up an instance environment
	jpcnsconfig	Command for defining a service configuration
	jpcnshostname	Command for setting up a host name
	jpcras	Command for collecting maintenance information
	jpcresto	Command for restoring a Store database
	jpcstart	Command for issuing a startup request for a service
	jpcstop	Command for issuing a stop request for a service
	jpcstsetup	Command for controlling statuses
	jpctminfo	Command for displaying product information
/opt/jp1pc/tools/log/	—	Directory for storing command log files
	msglog01	Log file
	msglog02	Log file (alternative file)
	nslog01	Log file
	nslog02	Log file (alternative file)

Note 1: The user creates this file by copying the sample (model) file.

Note 2: This file is used internally by the Tuning Manager series, and should not be changed or deleted.

E.2 List of Collection Manager Files and Directories

E.2.1 Windows Systems

Table E.3 provides a list of files and folders used by the Windows versions of Collection Manager.

Table E.3 Files and Folders of Collection Manager (Windows)

Files and Folders of Collection Manager (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\mgr\	--	Root folder for Collection Manager
	PATCHLOG.TXT	Internal file
installation-folder\mgr\clator\	--	Root folder for Correlator service
	jpcep.exe	Program that executes the Correlator service
	jpcep.ini	Startup initialization file for the Correlator service
	jpcep.ini.model	Sample of a startup initialization file for the Correlator service
installation-folder\mgr\clator\log\	--	Folder for storing Correlator service internal log files
	msglog01	Log file
	nslog01	Log file
installation-folder\mgr\manager\	--	Root folder for Master Manager service
	*.DB	Data files of service management information, etc.
	*.IDX	Index files for data files of service management information, etc.
	*.LCK	Lock files for data files of service management information, etc.
	jpcmm.exe	Program that executes the Master Manager service
	jpcmm.ini	Startup initialization file for the Master Manager service
	jpcmm.ini.model	Sample of a startup initialization file for the Master Manager service
	*.DAT	Data model definition files
installation-folder\mgr\manager\log\	--	Folder for storing Master Manager service internal log files
	msglog01	Log file
	nslog01	Log file

Files and Folders of Collection Manager (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\mgr\namesvr\	--	Root folder for Name Server service
	*.DB	Service information data files
	*.IDX	Index files for service information data files
	*.LCK	Lock files for service information data files
	jpcnsvr.exe	Program that executes the Name Server service
	jpcnsvr.ini	Startup initialization file for the Name Server service
	jpcnsvr.ini.model	Sample of a startup initialization file for the Name Server service
installation-folder\mgr\namesvr\log\	--	Folder for storing Name Server service internal log files
	msglog01	Log file
	nslog01	Log file
installation-folder\mgr\store\	--	Root folder for Master Store service
	*.DB	Management information data files
	*.IDX	Index files for management information data files
	*.LCK	Lock files for management information data files
	jpcsto.exe	Program that executes the Master Store service
	jpcsto.ini	Startup initialization file for the Master Store service
	jpcsto.ini.model	Sample of a startup initialization file for the Master Store service
	*.DAT	Data model definition file
installation-folder\mgr\store\backup\	--	Default folder for backing up databases
	datadir.lck	Internal file
installation-folder\mgr\store\dump\	--	Default folder for exporting databases
	datadir.lck	Internal file
installation-folder\mgr\store\log\	--	Folder for storing Master Store service internal log files
	msglog01	Log file
	nslog01	Log file
installation-folder\mgr\trapgen\	--	Root folder for Trap Generator service
	jpctrap.exe	Program that executes the Trap Generator service
	jpctrap.ini	Startup initialization file for the Trap Generator service

Files and Folders of Collection Manager (Windows)		
Folder Name	**File Name**	**Description**
	jpctrap.ini.model	Sample of a startup initialization file for the Trap Generator service
	jpctraps.exe	Internal command
installation-folder\mgr\trapgen\log\	--	Folder for storing Trap Generator service internal log files
	msglog01	Log file
	nslog01	Log file
installation-folder\mgr\viewsvr\	--	Root folder for View Server service
	jpcvsvr.exe	Program that executes the View Server service
	jpcvsvr.ini	Startup initialization file for the View Server service
	jpcvsvr.ini.model	Sample of a startup initialization file for the View Server service
installation-folder\mgr\viewsvr\data\	--	Folder for storing definition data
installation-folder\mgr\viewsvr\log\	--	Folder for storing View Server service internal log files
	msglog01	Log file
	nslog01	Log file
installation-folder\mgr\viewsvr\Reports\	--	Folder for storing report definition information files
installation-folder\setup\	--	Folder for storing setup files
	jpcagt?u.Z	Internal archive file
	jpcagt?w.exe	Internal archive file
installation-folder\setup\extract\	--	Internal work folder
installation-folder\setup\alarm\	--	Folder for storing alarm definition files
installation-folder\tools\	--	Folder for storing commands
	jpcalarm.exe	Command for defining alarms
	jpcimsetup.exe	Internal command

E.2.2 UNIX Systems

Table E.4 provides a list of files and directories used by the UNIX version of Collection Manager.

Table E.4 Files and Directories of Collection Manager (UNIX)

Files and Directories of Collection Manager (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/mgr/	--	Root directory for Collection Manager
/opt/jp1pc/mgr/clator/	--	Root directory for Correlator service
	jpcep	Program that executes the Correlator service
	jpcep.ini	Startup initialization file for Correlator service
	jpcep.ini.model	Sample of a startup initialization file for the Correlator service
/opt/jp1pc/mgr/clator/log/	--	Directory for storing Correlator service internal log files
	msglog01	Log file
	nslog01	Log file
/opt/jp1pc/mgr/manager/	--	Root directory for Master Manager service
	*.DB	Data files of service management information, etc.
	*.IDX	Index files for data files of service management information, etc.
	*.LCK	Lock files for data files of service management information, etc.
	jpcmm	Program that executes the Master Manager service
	jpcmm.ini	Startup initialization file for the Master Manager service
	jpcmm.ini.model	Sample of a startup initialization file for the Master Manager service
	*.DAT	Data model definition files
/opt/jp1pc/mgr/manager/log/	--	Directory for storing Master Manager service internal log files
	msglog01	Log file
	nslog01	Log file
/opt/jp1pc/mgr/namesvr/	--	Root directory for Name Server service
	*.DB	Service information data files
	*.IDX	Index files for service information data files
	*.LCK	Lock files for service information data files
	jpcnsvr	Program that executes the Name Server service
	jpcnsvr.ini	Startup initialization file for the Name Server service
	jpcnsvr.ini.model	Sample of a startup initialization file for the Name Server service

Files and Directories of Collection Manager (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/mgr/namesvr/log/	--	Directory for storing Name Server service internal log files
	msglog01	Log file
	nslog01	Log file
/opt/jp1pc/mgr/store/	--	Root directory for Master Store service
	*.DB	Management information data files
	*.IDX	Index files for management information data files
	*.LCK	Lock files for management information data files
	jpcsto	Program that executes the Master Store service
	jpcsto.ini	Startup initialization file for the Master Store service
	jpcsto.ini.model	Sample of a startup initialization file for the Master Store service
	*.DAT	Data model definition file
/opt/jp1pc/mgr/store/backup/	--	Default directory for backing up databases
	datadir.lck	Internal file
/opt/jp1pc/mgr/store/dump/	--	Default directory for exporting databases
	datadir.lck	Internal file
/opt/jp1pc/mgr/store/log/	--	Directory for storing Master Store service internal log files
	msglog01	Log file
	nslog01	Log file
/opt/jp1pc/mgr/trapgen/	--	Root directory for Trap Generator service
	jpctrap	Program that executes the Trap Generator service
	jpctrap.ini	Startup initialization file for the Trap Generator service
	jpctrap.ini.model	Sample of a startup initialization file for the Trap Generator service
	jpctraps	Internal command
/opt/jp1pc/mgr/trapgen/log/	--	Directory for storing Trap Generator service internal log files
	msglog01	Log file
	nslog01	Log file
/opt/jp1pc/mgr/viewsvr/	--	Root directory for View Server service
	jpcvsvr	Program that executes the View Server service
	jpcvsvr.ini	Startup initialization file for the View Server service

Files and Directories of Collection Manager (UNIX)		
Directory Name	**File Name**	**Description**
	jpcvsvr.ini.model	Sample of a startup initialization file for the View Server service
/opt/jp1pc/mgr/viewsvr/data/	--	Directory for storing definition data
/opt/jp1pc/mgr/viewsvr/log/	--	Directory for storing View Server service internal log files
	msglog01	Log file
	nslog01	Log file
/opt/jp1pc/mgr/viewsvr/Reports/	--	Directory for storing report definition information files
/opt/jp1pc/setup/	--	Directory for storing setup files
	jpcagt?u.Z	Internal archive file
	jpcagt?w.exe	Internal archive file
/opt/jp1pc/setup/extract/	--	Internal work directory
/opt/jp1pc/setup/alarm/	--	Directory for storing alarm definition files
/opt/jp1pc/tools/	--	Directory for storing commands
	jpcalarm	Command for defining alarms
	jpcimsetup	Internal command

E.3 List of Agent for RAID Files and Directories

E.3.1 Windows Systems

The following table lists the files and folders for the Windows version of the Agent for RAID.

Table E.5 Files and Folders of Agent for RAID (Windows)

Files and Folders of Agent for RAID (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtd\	—	Root folder for Agent for RAID
	insrules.dat	Definition file for instance startup environment rules
	jpcagtras.bat	Internal command
	PATCHLOG.TXT	Internal file
installation-folder\agtd\agent\	—	Root folder for the Agent Collector service
	hpmrl1kdump.exe	Internal command
	hpmrldump.exe	Internal command
	jpcagtd.exe	Program for executing the Agent Collector service
	jpcagtha.ini	Cluster definition file See ***Note 1***
	jpcagt.ini.instmpl	Internal file
	PortMap*.dat	Internal definition file
	ProductMap.dat	Internal definition file
	*.dll	Common library for Agent for RAID
installation-folder\agtd\agent\lib\	—	Folder for storing Agent for RAID
installation-folder\agtd\agent*instance-name*\	—	Root folder for Agent Collector service (for each instance) See ***Note 2***
	jpcagt.ini	Agent Collector service startup information file (for each instance) See ***Note 2***
	jpcagt.ini.model	Sample of an Agent Collector service startup information file (for each instance) See ***Note 2***

Files and Folders of Agent for RAID (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtd\agent*instance-name*\log\	—	Folder for storing internal log files of the Agent Collector service (for each instance) See ***Note 2***
	msglog01	Log file
	msglog02	Log file (alternative file)
	msglog03	
	msglog04	
	nslog01	Log file
	nslog02	Log file (alternative file)
installation-folder\agtd\lib\	—	Folder for storing message catalogs
	jpcagtdmsg.dll	Internal file
installation-folder\agtd\store\	—	Root folder for the Agent Store service
	jpcsto.exe	Program for executing the Agent Store service
	stpqlpr.exe	Program for executing backup and export of the Store database
	*.DAT	Data model definition file
installation-folder\agtd\store*instance-name*\	—	Root folder for the Agent Store service (for each instance) See ***Note 2***
	jpcsto.ini	Agent Store service startup information file (for each instance) See ***Note 2***
	jpcsto.ini.model	Sample of an Agent Store service startup information file (for each instance) See ***Note 2***
	*.DAT	Data model definition file (for each instance) See ***Note 2***
	*.DB	Performance data file (for each instance) See ***Note 3***
	*.IDX	Index file for performance data file (for each instance) See ***Note 3***

Files and Folders of Agent for RAID (Windows)		
Folder Name	**File Name**	**Description**
	*.LCK	Lock file for performance data file (for each instance) See ***Note 3***
installation-folder\agtd\store*instance-name*\backup\	—	Folder for default database backup destination (for each instance) See ***Note 2***
	datadir.lck	Internal file
installation-folder\agtd\store*instance-name*\dump\	—	Folder for default database export destination (for each instance) See ***Note 2***
	datadir.lck	Internal file
installation-folder\agtd\store*instance-name*\log\	—	Folder for storing internal log files of the Agent Store service (for each instance) See ***Note 2***
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	—	Folder for storing setup files
	jpcagtdu.Z	Internal archive file (UNIX)
	jpcagtdw.EXE	Internal archive file (Windows)
installation-folder\tools\	—	Folder for storing commands
	jpctdchkinst.bat	Command for checking instance settings
	jpctdinssetup.bat	Internal command
	jpctdlistraid.bat	Command for detecting command devices.

Note 1: Created by the user.

Note 2: Created by executing the `jpcinssetup` command.

Note 3: Created when the Agent Store service starts.

E.3.2 UNIX Systems

The following table lists the files and directories for the UNIX version of Agent for RAID.

Table E.6 Files and Directories of Agent for RAID (UNIX)

Files and Directories of Agent for RAID (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtd/	—	Root directory for Agent for RAID
	insrules.dat	Definition file for instance startup environment rules
	jpcagtras	Internal command
	PATCHLOG.TXT	Internal file
	patch_history	Internal file
/opt/jp1pc/agtd/agent/	—	Root directory for the Agent Collector service
	hpmrldump	Internal command
	jpcagtd	Program for executing the Agent Collector service
	jpcagtha.ini	Cluster definition file See ***Note 1***
	jpcagt.ini.instmpl	Internal file
	PortMap*.dat	Internal definition file
	ProductMap.dat	Internal definition file
/opt/jp1pc/agtd/agent/lib/	—	Directory for common library for Agent for RAID
	lib*[.sl/.so/.o]	Common library for Agent for RAID
/opt/jp1pc/agtd/agent/*instance-name*/	—	Root directory for the Agent Collector service (for each instance) See ***Note 2***
	jpcagt.ini	Agent Collector service startup information file (for each instance) See ***Note 2***
	jpcagt.ini.model	Sample of an Agent Collector service startup information file (for each instance) See ***Note 2***
/opt/jp1pc/agtd/agent/*instance-name*/log/	—	Directory for storing internal log files of the Agent Collector service (for each instance) See ***Note 2***
	msglog01	Log file
	msglog02	Log file (alternative file)

Files and Directories of Agent for RAID (UNIX)		
Directory Name	**File Name**	**Description**
	msglog03	
	msglog04	
	nslog01	Log file
	nslog02	Log file (alternative file)
/opt/jp1pc/agtd/nls/$LANG/	—	Directory for storing message catalogs
	jpcagtdmsg.cat	Internal file
/opt/jp1pc/agtd/store/	—	Root directory for the Agent Store service
	jpcsto	Program for executing the Agent Store service
	stpqlpr	Program for executing backup and export of the Store database
	*.DAT	Data model definition file
/opt/jp1pc/agtd/store/*instance-name*/	—	Root directory for the Agent Store service (for each instance) See ***Note 2***
	jpcsto.ini	Agent Store service startup information file (for each instance) See ***Note 2***
	jpcsto.ini.model	Sample of an Agent Store service startup information file (for each instance) See ***Note 2***
	*.DAT	Data model definition file (for each instance) See ***Note 2***
	*.DB	Performance data file (for each instance) See ***Note 3***
	*.IDX	Index file for performance data file (for each instance) See ***Note 3***
	*.LCK	Lock file for performance data file (for each instance) See ***Note 3***
/opt/jp1pc/agtd/store/*instance-name*/backup/	—	Directory for default database backup destination (for each instance) See ***Note 2***

Files and Directories of Agent for RAID (UNIX)		
Directory Name	**File Name**	**Description**
	datadir.lck	Internal file
/opt/jp1pc/agtd/store/*instance-name*/dump/	—	Directory for default database export destination (for each instance) See ***Note 2***
	datadir.lck	Internal file
/opt/jp1pc/agtd/store/*instance-name*/log/	—	Directory for storing internal log files of the Agent Store service (for each instance) See ***Note 2***
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/setup/	—	Directory for storing setup files
	jpcagtdu.Z	Internal archive file (UNIX)
	jpcagtdw.EXE	Internal archive file (Windows)
/opt/jp1pc/tools/	—	Folder for storing commands
	jpctdchkinst	Command for checking instance settings
	jpctdinssetup	Internal command
	jpctdlistraid	Command for detecting command devices.

Note 1: Created by the user.

Note 2: Created by executing the `jpcinssetup` command.

Note 3: Created when the Agent Store service starts.

E.4 List of Agent for RAID Map Files and Directories

E.4.1 Windows Systems

The following table lists the files and folders for the Windows version of the Agent for RAID Map.

Table E.7 Files and Folders of Agent for RAID Map (Windows)

Files and Folders of Agent for RAID Map (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agte\	—	Base folder of Agent for RAID Map
	jpcagteparm.ini	Settings file for IP addresses See ***Note***
	jpcagtras.bat	Internal command
	PATCHLOG.TXT	Internal file
installation-folder\agte\agent\	—	Base folder of the Agent Collector service
	HpmHL.dll	Shared libraries of Agent for RAID Map
	hpmhlinquiry.exe	Internal command
	jpcagte.exe	Program for executing the Agent Collector service
	jpcagt.ini	Service startup initialization file of Agent Collector
	jpcagt.ini.model	Sample of a service startup initialization file of Agent Collector
	PortMap*.dat	Internal definition file
	ProductMap.dat	Internal definition file
installation-folder\agte\agent\hldutility\bin	—	Folder for Agent Collector service subcommands
	hldutil.conf	Settings file for Agent Collector service subcommands
	hldutil.exe	Agent Collector service subcommands
	message.txt	Message file for Agent Collector service subcommands
installation-folder\agte\agent\hldutility\log	—	Folder for storing log files of Agent Collector service subcommands
	*.log	Subcommand log file
installation-folder\agte\agent\hldutility\bin\sub\windows\	—	Folder for storing platform-dependent files

Files and Folders of Agent for RAID Map (Windows)		
Folder Name	**File Name**	**Description**
	hldu_*.exe	Agent Collector service subcommands
installation-folder\agte\agent\log\	—	Folder for storing internal log files of Agent Collector service
	msglog01	Log file
	msglog02	Log file (alternate file)
	msglog03	
	msglog04	
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\agte\lib\	—	Message catalog installation folder
	jpcagtemsg.dll	Internal file
installation-folder\agte\store\	—	Base folder of the Agent Store service
	jpcsto.exe	Program for executing the Agent Store service
	jpcsto.ini	Service startup initialization file of Agent Store
	jpcsto.ini.model	Sample of a service startup initialization file of Agent Store
	stpqlpr.exe	Process for backup and export of the Store database
	*.DAT	Definition file for a data model
	*.DB	Performance data file
	*.IDX	Index file for a performance data file
	*.LCK	Lock file for a performance data file
installation-folder\agte\store\backup\	—	Default database backup folder
	datadir.lck	Internal file
installation-folder\agte\store\dump\	—	Default database export folder
	datadir.lck	Internal file
installation-folder\agte\store\log\	—	Folder for storing log files of the Agent Store service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	—	Folder for storing setup files
	jpcagteu.Z	Internal archive file (in UNIX)

Files and Folders of Agent for RAID Map (Windows)		
Folder Name	**File Name**	**Description**
	jpcagtew.EXE	Internal archive file (in Windows)

Note: The user must create this file.

E.4.2 UNIX Systems

The following table lists the files and directories for the UNIX version of Agent for RAID Map.

Table E.8 Files and Directories of Agent for RAID Map (UNIX)

Files and Directories of Agent for RAID Map (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agte/	—	Root directory for Agent for RAID Map
	jpcagteparm.ini	Settings file for IP addresses See ***Note***
	jpcagtras	Internal command
	PATCHLOG.TXT	Internal file
	patch_history	Internal file
/opt/jp1pc/agte/agent/	—	Base directory of the Agent Collector service
	hpmhlinquiry	Internal command
	jpcagte	Program for executing the Agent Collector service
	jpcagt.ini	Service startup initialization file of Agent Collector
	jpcagt.ini.model	Sample of a service startup initialization file of Agent Collector
/opt/jp1pc/agte/agent/cmd	—	Directory for storing internal commands
	getdevinf	Internal command ***Note***: This file is installed only in a Solaris environment.
	getip	
	getip.awk	
	getosname	Internal command
/opt/jp1pc/agte/agent/HLDUtility/bin/	—	Directory for Agent Collector service subcommands

Files and Directories of Agent for RAID Map (UNIX)		
Directory Name	**File Name**	**Description**
	hldutil.conf	Settings file for Agent Collector service subcommands
	hldutil	Agent Collector service subcommands
	message.txt	Message file for Agent Collector service subcommands
/opt/jp1pc/agte/agent/HLDUtility/bin/sub/platform-name/	—	Directory for storing platform-dependent files (*platform-name*: hpux for HP-UX, sun for Solaris, aix for AIX, linux for Linux)
	hldu_*	Agent Collector service subcommand
/opt/jp1pc/agte/agent/HLDUtility/log/	—	Directory for storing subcommand log files of the Agent Collector service
	*.log	Subcommand log file
/opt/jp1pc/agte/agent/lib/	—	Directory for storing shared libraries of Agent for RAID Map
	libhpmhl[.sl/.so/.o]	Shared libraries of Agent for RAID Map
	PortMap*.dat	Internal definition file
	ProductMap.dat	Internal definition file
/opt/jp1pc/agte/agent/log/	—	Directory for storing log files of the Agent Collector service
	msglog01	Log file
	msglog02	Log file (alternate file)
	msglog03	
	msglog04	
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/agte/nls/$LANG/	—	Message catalog installation directory
	jpcagtemsg.cat	Internal file

Files and Directories of Agent for RAID Map (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agte/store/	—	Base directory of the Agent Store service
	jpcsto	Program for executing the Agent Store service
	jpcsto.ini	Service startup initialization file of Agent Store
	jpcsto.ini.model	Sample of a service startup initialization file of Agent Store
	stpqlpr	Process for backup and export of the Store database
	*.DAT	Definition file of a data model
	*.DB	Performance data file
	*.IDX	Index file for a performance data file
	*.LCK	Lock file for a performance data file
/opt/jp1pc/agte/store/backup/	—	Default database backup directory
	datadir.lck	Internal file
/opt/jp1pc/agte/store/dump/	—	Default database export directory
	datadir.lck	Internal file
/opt/jp1pc/agte/store/log/	—	Directory for storing log files of Agent Store service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/setup/	—	Directory for storing setup files
	jpcagteu.Z	Internal archive file (in UNIX)
	jpcagtew.EXE	Internal archive file (in Windows)

Note: The user must create this file.

E.5 List of Agent for Platform (Windows) Files and Folders

The following table lists the files and folders of Agent for Platform (Windows).

Table E.9 Files and Folders of Agent for Platform (Windows)

Files and Folders of Agent for Platform (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtt\	-	Base folder of Agent for Platform
	PATCHLOG.TXT	Internal file
	jpcagtras.bat	Internal command
installation-folder\agtt\agent\	-	Base folder of the Agent Collector service
	jpcagte.exe	Service executing program of Agent Collector
	jpcagt.ini	Configuration file for the Agent Collector service
	jpcagt.ini.model	Sample of a configuration file for the Agent Collector service
installation-folder\agtt\agent\log\	-	Storage folder for internal log files of the Agent Collector service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\agtt\lib\	-	Message catalog installation folder
	jpcagttmsg.dll	Internal file
installation-folder\agtt\store\	-	Base folder of the Agent Store service
	jpcsto.exe	Service executing program of Agent Store
	stpqlpr.exe	Internal command
	*.DAT	Definition file for a data model
	*.DB	Performance data file
	*.IDX	Index file for a performance data file
	*.LCK	Lock file for a performance data file
	jpcsto.ini	Configuration file for the Agent Store service
	jpcsto.ini.model	Sample of a configuration file for the Agent Store service
	*.DAT	Definition file of a data model

Files and Folders of Agent for Platform (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtt\store\backup\	-	Default database backup folder
	datadir.lck	Internal file
installation-folder\agtt\store\dump\	-	Default database export folder
	datadir.lck	Internal file
installation-folder\agtt\store\log\	-	Storage folder for log files of the Agent Store service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	-	Storage folder for setup files
	jpcagttu.Z	Archive file for setting up Agent (UNIX)
	jpcagttw.EXE	Archive file for setting up Agent (Windows)

E.6 List of Agent for Platform (UNIX) Files and Directories

Table E.10 lists the files and directories of Agent for Platform (UNIX).

Table E.10 Files and Directories of Agent for Platform (UNIX)

Files and Directories of Agent for Platform (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtu/	—	The base directory of Agent for Platform
	PATCHLOG.TXT	Internal file
	jpcagtras	Internal command
/opt/jp1pc/agtu/agent/	—	The base directory of the Agent Collector service
	jpc_process	The program for obtaining 64-bit process information See ***Note***
	jpcagte	The executable program of the Agent Collector service
	jpcagt.ini	The startup-information file of the Agent Collector service
	jpcagt.ini.model	The model file for the startup-information file of the Agent Collector service
/opt/jp1pc/agtu/agent/log/	—	The storage directory for internal log files of the Agent Collector service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/agtu/nls/$LANG/	—	The storage directory for message catalogs
	jpcagtemsg.cat	Internal file

Files and Directories of Agent for Platform (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtu/store/	—	The base directory of the Agent Store service
	jpcsto	The executable program of the Agent Store service
	stpqlpr	Internal command
	*.DAT	The definition file for a data model
	*.DB	The performance data file
	*.IDX	The index file for a performance data file
	*.LCK	The lock file for a performance data file
	jpcsto.ini	The startup-information file of the Agent Store service
	jpcsto.ini.model	The model file for the startup-information file of the Agent Store service
	*.DAT	The definition file for a data model
/opt/jp1pc/agtu/store/backup/	—	The default database backup directory
	datadir.lck	Internal file
/opt/jp1pc/agtu/store/dump/	—	The default database export directory
	datadir.lck	Internal file
/opt/jp1pc/agtu/store/log/	—	The storage directory for internal log files of the Agent Store service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/setup/	—	The storage directory for setup files
	jpcagtuu.Z	Internal archive file (UNIX)
	jpcagtuw.EXE	Internal archive file (Windows)

Note: Not installed in HP-UX, AIX, and Linux.

E.7 List of Agent for SAN Switch Files and Directories

E.7.1 Windows Systems

The following shows a list of files and folders for the Windows edition of Agent for SAN Switch.

Table E.11 Files and Folders of Agent for SAN Switch (Windows)

Files and Folders of Agent for SAN Switch (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtw\	-	Base folder of Agent for SAN Switch
	insrules.dat	Definition file for instance startup environment rules
	PATCHLOG.TXT	Internal file
installation-folder\agtw\agent\	-	Base folder of the Agent Collector service
	*.DLL	DLL for collecting switch information
	jpcagt.ini.instmpl	Template file of the Agent Collector service startup-information file
	jpcagtw.exe	Executable program of the Agent Collector service
installation-folder\agtw\agent*instance-name*\	-	Base folder of the Agent Collector service (for each instance) (see ***Note 1***)
	jpcagt.ini	Startup-information file of the Agent Collector service (for each instance) (see ***Note 1***)
	jpcagt.ini.model	Sample of a startup-information file of the Agent Collector service (for each instance) (see ***Note 1***)
installation-folder\agtw\agent*instance-name*\log\	-	Folder for storing internal log files of the Agent Collector service (for each instance) (see ***Note 1***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	msglog03	
	msglog04	
	nslog01	Log file (alternate file)
	nslog02	Log file

Files and Folders of Agent for SAN Switch (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtw\lib\	-	Folder for storing message catalogs
	jpcagtwmsg.dll	Message catalog file
installation-folder\agtw\store\	-	Base folder of the Agent Store service
	*.DAT	Definition file for a data model
	jpcsto.exe	Executable program of the Agent Store service
	jpcsto.ini.instmpl	Template file for the startup-information file of the Agent Store service
	stpqlpr.exe	Program for executing backup and export of the Store database
installation-folder\agtw\store*instance-name*\	-	Base folder of the Agent Store service (for each instance) (see ***Note 1***)
	*.DAT	Definition file for a data model (for each instance) (see ***Note 1***)
	*.DB	Performance data file (for each instance) (see ***Note 2***)
	*.IDX	Index file for a performance data file (for each instance) (see ***Note 2***)
	*.LCK	Lock file for a performance data file (for each instance) (see ***Note 2***)
	jpcsto.ini	Startup-information file of the Agent Store service (for each instance) (see ***Note 1***)
	jpcsto.ini.model	Sample of a startup-information file of the Agent Store service (for each instance) (see ***Note 1***)
installation-folder\agtw\store*instance-name*\backup\	-	Default database backup folder (for each instance) (see ***Note 1***)
	datadir.lck	Lock file of the default database backup folder
installation-folder\agtw\store*instance-name*\dump\	-	Default database export folder (for each instance) (see ***Note 1***)

Files and Folders of Agent for SAN Switch (Windows)		
Folder Name	**File Name**	**Description**
	datadir.lck	Lock file of the default database backup folder
installation-folder\agtw\store*instance-name*\log\	-	Folder for storing internal log files of the Agent Store service (for each instance) (see ***Note 1***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	-	Folder for storing setup files
	jpcagtwu.Z	Archive file for setup (UNIX)
	jpcagtww.EXE	Archive file for setup (Windows)

Note 1: Created by executing the `jpcinssetup` command.

Note 2: Created when the Agent Store service starts.

E.7.2 UNIX Systems

The following shows a list of files and directories for the UNIX edition of Agent for SAN Switch.

Table E.12 Files and Directories of Agent for SAN Switch (UNIX)

Files and Directories of Agent for SAN Switch (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtw/	-	Base directory of Agent for SAN Switch
	insrules.dat	Definition file for instance startup environment rules
	PATCHLOG.TXT	Internal file
	patch_history	Internal file
/opt/jp1pc/agtw/agent/	-	Base directory of the Agent Collector service
	jpcagtw	Executable program of the Agent Collector service
	jpcagtw.ini.instmpl	Template file for the startup-information file of the Agent Collector service
	lib*	Library for collecting switch information

Files and Directories of Agent for SAN Switch (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtw/agent/*instance-name*/	-	Base directory of the Agent Collector service (for each instance) (see ***Note 1***)
	jpcagt.ini	Startup-information file of the Agent Collector service (for each instance) (see ***Note 1***)
	jpcagt.ini.model	Sample of a startup-information file of the Agent Collector service (for each instance) (see ***Note 1***)
/opt/jp1pc/agtw/agent/*instance-name*/log/	-	Directory for storing internal log files of the Agent Collector service (for each instance) (see ***Note 1***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	msglog03	
	msglog04	
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/agtw/nls/$LANG/	-	Directory for storing message catalogs
	jpcagtwmsg.cat	Message catalog
/opt/jp1pc/agtw/store/	-	Base directory of the Agent Store service
	*.DAT	Definition file for a data model
	jpcsto	Executable file of the Agent Store service
	jpcsto.ini.instmpl	Template file for the startup-information file of the Agent Store service
	stpqlpr	Program for executing backup and export of the Store database
/opt/jp1pc/agtw/store/*instance-name*/	-	Base directory of the Agent Store service (for each instance) (see ***Note 1***)
	*.DAT	Definition file for a data model (for each instance) (see ***Note 1***)
	*.DB	Performance data file (for each instance) (see ***Note 2***)
	*.IDX	Index file for a performance data file (for each instance) (see ***Note 2***)

Files and Directories of Agent for SAN Switch (UNIX)		
Directory Name	**File Name**	**Description**
	*.LCK	Lock file for a performance data file (for each instance) (see ***Note 2***)
	jpcsto.ini	Startup-information file of the Agent Store service (for each instance) (see ***Note 1***)
	jpcsto.ini.model	Sample of a startup-information file of the Agent Store service (for each instance) (see ***Note 1***)
/opt/jp1pc/agtw/store/*instance-name*/backup/	-	Default database backup directory (for each instance) (see ***Note 1***)
	datadir.lck	Lock file of the default database backup directory
/opt/jp1pc/agtw/store/*instance-name*/dump/	-	Default database export directory (for each instance) (see ***Note 1***)
	datadir.lck	Lock file of the default database backup directory
/opt/jp1pc/agtw/store/*instance-name*/log/	-	Directory for storing internal log files of the Agent Store service (for each instance) (see ***Note 1***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/setup/	-	Directory for storing setup files
	jpcagtwu.Z	Archive file for setup (UNIX)
	jpcagtww.EXE	Archive file for setup (Windows)

Note 1: Created by executing the `jpcinssetup` command.

Note 2: Created when the Agent Store service starts.

E.8 List of Agent for NAS Files and Directories

E.8.1 Windows Systems

Table E.13 lists the files and folders of Agent for NAS.

Table E.13 Files and Folders of Agent for NAS (Windows)

Files and Folders of Agent for NAS (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agtn\	-	Root folder for Agent for NAS
	insrules.dat	Definition file for instance startup environment rules
	PATCHLOG.TXT	Internal file
installation-folder\agtn\agent\	-	Root folder for the Agent Collector service
	jpcagtn.exe	Program for executing the Agent Collector service
	jpcagtnenas.dll	Execution file for collecting the Agent Collector service information
	jpcenas.dll	API execution file for collecting the Agent Collector service information
	PortMap.dat	Internal definition file
	ProductMap.dat	Internal definition file
	jpcagt.ini.instmpl	Internal file
Installation-folder\agtn\agent*instance-name*\	-	Root folder for the Agent Collector service (for each instance) See ***Note*** 1
	jpcagt.ini	Agent Collector service startup information file (for each instance)
	jpcagt.ini.model	Sample of Agent Collector service startup information file (for each instance)
installation-folder\agtn\agent*instance-name*\log\	-	Folder for storing internal log files of the Agent Collector service (for each instance) See ***Note*** 1
	msgdat01	Log file
	msgdat02	Log file (alternative file)
	msgdat03	Log file (alternative file)
	msgdat04	Log file (alternative file)
	msglog01	Log file
	msglog02	Log file (alternative file)
	msglog03	Log file (alternative file)
	msglog04	Log file (alternative file)

Files and Folders of Agent for NAS (Windows)		
Folder Name	**File Name**	**Description**
	nslog01	Log file
	nslog02	Log file (alternative file)
installation-folder\agtn\lib\	-	Folder for storing the Agent for NAS message catalogs
	jpcagtnmsg.dll	Internal file
installation-folder\agtn\store\	-	Root folder for the Agent Store service (default folder for storing a Store database)
	jpcsto.exe	Program for executing the Agent Store service
	jpcsto.ini.instmpl	Internal file
	stpqlpr.exe	Program for executing backup and export of the Store database
	*.DAT	Data model definition file
installation-folder\agtn\store*instance-name*\	-	Root folder for the Agent Store service (for each instance) See ***Note*** 1
	jpcsto.ini	Agent Store service startup information file (for each instance)
	jpcsto.ini.model	Sample of Agent Store service startup information file (for each instance)
	*.DAT	Data model definition file (for each instance)
	*.DB	Performance data file (for each instance) See ***Note*** 2
	*.IDX	Index file for performance data file (for each instance) See ***Note*** 2
	*.LCK	Lock file for performance data file (for each instance) See ***Note*** 2
installation-folder\agtn\store*instance-name*\backup\	-	Folder for default database backup destination (for each instance) See ***Note*** 1
	datadir.lck	Internal file
installation-folder\agtn\store*instance-name*\dump\	-	Folder for default database export destination (for each instance) See ***Note*** 1
	datadir.lck	Internal file
installation-folder\agtn\store*instance-name*\log\	-	Folder for storing internal log files of the Agent Store service (for each instance) See ***Note*** 1
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file

Files and Folders of Agent for NAS (Windows)		
Folder Name	**File Name**	**Description**
	nslog02	Log file (alternate file)
installation-folder\setup\	-	Folder for storing setup files
	jpcagtnu.Z	Setup file for UNIX
	jpcagtnw.EXE	Setup file for Windows

Note 1: Created by executing the jpcinssetup command.

Note 2: Created when the Agent Store service starts.

E.8.2 UNIX Systems

Table E.14 lists the files and directories of Agent for NAS.

Table E.14 Files and Directories of Agent for NAS (UNIX)

Files and Directories of Agent for NAS (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/	instagtn.ini	Agent for NAS version information
/opt/jp1pc/agtn/	-	Directory for storing Agent for NAS components
	insrules.dat	Definition file for instance startup environment rules
	PATCHLOG.TXT	Internal file
	patch-history	Internal file
/opt/jp1pc/agtn/agent/	-	Root directory for the Agent Collector service
	jpcagt.ini.instmpl	Internal file
	jpcagtn	Program for executing the Agent Collector service
	jpcagtnenas	Execution file for collecting the Agent Collector service information
	jpcenas	API execution file for collecting the Agent Collector service information
	PortMap.dat	Internal definition file
	ProductMap.dat	Internal definition file
/opt/jp1pc/agtn/agent/*instance-name*/	-	Root directory for the Agent Collector service (for each instance) See ***Note*** 1
	jpcagt.ini	Agent Collector service startup information file (for each instance)
	jpcagt.ini.model	Directory for storing internal log files of the Agent Collector service (for each instance) See ***Note*** 1

Files and Directories of Agent for NAS (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtn/agent/*instance-name*/log/	-	Directory for storing log files of the Agent Collector service (for each instance) See ***Note*** 1
	msgdat01	Log file
	msgdat02	Log file (alternative file)
	msgdat03	Log file (alternative file)
	msgdat04	Log file (alternative file)
	msglog01	Log file
	msglog02	Log file (alternative file)
	msglog03	Log file (alternative file)
	msglog04	Log file (alternative file)
	nslog01	Log file
	nslog02	Log file (alternative file)
/opt/jp1pc/agtn/nls/$LANG/	-	Directory for storing Agent for NAS message catalogs
	jpcagtnmsg.cat	Internal file
/opt/jp1pc/agtn/store/	-	Root directory for the Agent Store service (default directory for storing a Store database)
	jpcsto	Program for executing the Agent Store service
	jpcsto.ini.instmpl	Internal file
	stpqlpr	Program for executing backup and export of the Store database
	*.DAT	Data model definition file
/opt/jp1pc/agtn/store/*instance-name*/	-	Root directory for the Agent Store service (for each instance) See ***Note*** 1
	jpcsto.ini	Agent Store service startup information file (for each instance)
	jpcsto.ini.model	Sample of Agent Store service startup information file (for each instance)
	*.DAT	Data model definition file
	*.DB	Performance data file (for each instance) See ***Note*** 2
	*.IDX	Index file for a database file (for each instance) See ***Note*** 2
	*.LCK	Lock file for a database file (for each instance) See ***Note*** 2

Files and Directories of Agent for NAS (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtn/store/*instance-name*/backup/	-	Directory for default database backup destination (for each instance) See ***Note*** 1
	datadir.lck	Internal file
/opt/jp1pc/agtn/store/*instance-name*/dump/	-	Directory for default database export destination (for each instance) See ***Note*** 1
	datadir.lck	Internal file
/opt/jp1pc/agtn/store/*instance-name*/log/	-	Directory for storing internal log files of the Agent Store service (for each instance) See ***Note*** 1
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/setup/	-	Directory for storing setup files
	jpcagtnu.Z	Setup file for UNIX
	jpcagtnw.EXE	Setup file for Windows

Notes 1: Created by executing the jpcinssetup command.

Notes 2: Created when the Agent Store service starts.

E.9 List of Agent for Oracle Files and Directories

E.9.1 Windows Systems

The following shows a list of files and folders for the Windows edition of Agent for Oracle.

Table E.15 Files and Folders of Agent for Oracle (Windows)

Files and Folders of Agent for Oracle (Windows)		
Folder Name	**File Name**	**Description**
installation-folder\agto\	—	The base folder of Agent for Oracle
	PATCHLOG.TXT	Internal file
	insrules.dat	The definition file for instance startup environment rules
	jpcagtras.bat	Programs for collecting maintenance information
	jpcagtras.exe	
installation-folder\agto\agent\	—	The base folder of Agent Collector
	jpcagto.exe	The executable program of the Agent Collector service
	jpcagt.ini.instmpl	Internal file
	inssetup.bat.instmpl	Internal file
	jpcOcollect.exe	The program for collecting Agent Collector service performance data (for Oracle8 and Oracle8i) (see ***Note 1***)
	jpcOcollect_9.exe	The program for collecting Agent Collector service performance data (for Oracle9i and Oracle 10g)
installation-folder\agto\agent\sql\	—	The storage folder for SQL scripts
	sp_drop.sql	The SQL script file for deleting objects from Oracle Database
	sp_inst.sql	The SQL script file for registering objects from Oracle Database
	mk_user.spl	Script file for creating Oracle accounts used in Agent for Oracle (for Oracle9i or later)
	mk_user8.spl	Script file for creating Oracle accounts used in Agent for Oracle (for Oracle8i)
installation-folder\agto\agent*instance-name*\	—	Base folder of Agent Collector (for each instance) (see ***Note 1*** and ***Note 2***)
	jpcagt.ini	The service startup initialization file of Agent Collector (for each instance) (see ***Note 2***)

Files and Folders of Agent for Oracle (Windows)		
Folder Name	**File Name**	**Description**
	jpcagt.ini.model	The sample of a service startup initialization file of Agent Collector (for each instance) (see ***Note 2***)
	jpcOcollect.exe	The program for collecting Agent Collector service performance data (for each instance) (see ***Note 2***)
installation-folder\agto\agent*instance-name*\log\	—	The folder for storing internal log files of Agent Collector (for each instance) (see ***Note 2***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\agto\lib\	—	The storage folder for message catalogs
	jpcagtomsg.dll	Internal file
installation-folder\agto\store\	—	The root folder of Agent Store
	jpcsto.exe	The executable program of the Agent Store service
	stpqlpr.exe	The program for executing backup and export of the Store database
	jpcsto.ini.instmpl	Internal file
	*.DAT	The definition file for a data model
installation-folder\agto\store*instance-name*\	—	The base folder of Agent Store (for each instance) (see ***Note 2***)
	*.DB	The performance data file (for each instance) (see ***Note 3***)
	*.IDX	The index file for a performance data file (for each instance) (see ***Note 3***)
	*.LCK	The lock file for a performance data file (for each instance) (see ***Note 3***)
	jpcsto.ini	The service startup initialization file of Agent Store (for each instance) (see ***Note 2***)
	jpcsto.ini.model	The sample of a service startup initialization file of Agent Store (for each instance) (see ***Note 2***)
	*.DAT	The definition file for a data model (for each instance) (see ***Note 2***)
installation-folder\agto\store*instance-name*\backup\	—	The default database backup folder (for each instance) (see ***Note 2***)

Files and Folders of Agent for Oracle (Windows)		
Folder Name	**File Name**	**Description**
	datadir.lck	Internal file
installation-folder\agto\store*instance-name*\dump\	—	The default database export folder (for each instance) (see ***Note 2***)
	datadir.lck	Internal file
installation-folder\agto\store*instance-name*\log\	—	The folder for storing internal log files of Agent Store (for each instance) (see ***Note 2***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	—	The storage folder for setup files
	jpcagtou.Z	The internal archive file (UNIX)
	jpcagtow.EXE	The internal archive file (Windows)

Note 1: The user creates these files and folders.

Note 2: Created by executing the jpcinssetup command.

Note 3: Created when the Agent Store service starts.

E.9.2 UNIX Systems

The following shows a list of files and directories for the UNIX edition of Agent for Oracle.

Table E.16 Files and Directories of Agent for Oracle (UNIX)

Files and Directories of Agent for Oracle (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agto/	—	The base directory of Agent for Oracle
	PATCHLOG.TXT	Internal file
	insrules.dat	The definition file for instance startup environment rules
	jpcagtras	Program for collecting maintenance information
/opt/jp1pc/agto/agent/	—	The base directory of Agent Collector
	jpcagto	The execution program of the Agent Collector service
	jpcagt.ini.instmpl	Internal file

Files and Directories of Agent for Oracle (UNIX)		
Directory Name	**File Name**	**Description**
	inssetup.instmpl	Internal file
	jpcOcollect	The Agent Collector service performance data collector program (for Oracle8 and Oracle8i) (see ***Note 3***)
	jpcOcollect_9	The Agent Collector service performance data collector program (for Oracle9i)
	jpcOcollect_10	The Agent Collector service performance data collector program (for Oracle9i and Oracle 10g)
/opt/jp1pc/agto/agent/sql/	—	The storage directory for SQL scripts
	mk_user.spl	Script file for creating Oracle accounts used in Agent for Oracle (for Oracle9i or later)
	mk_user8.spl	Script file for creating Oracle accounts used in Agent for Oracle (for Oracle8i)
	sp_drop.sql	The SQL script file for deleting objects from Oracle Database
	sp_inst.sql	The SQL script file for registering objects from Oracle Database
/opt/jp1pc/agto/agent/*instance-name*	—	Base directory of Agent Collector (for each instance) (see ***Note 1*** and ***Note 2***)
	jpcagt.ini	The service startup initialization file of Agent Collector (for each instance) (see ***Note 2***)
	jpcagt.ini.model	Sample of a service startup initialization file of Agent Collector (for each instance) (see ***Note 2***)
	jpcOcollect	The program for collecting Agent Collector service performance data (for each instance) (see ***Note 1***)
/opt/jp1pc/agto/agent/*instance-name*/log/	—	The directory for storing internal log files of Agent Collector (for each instance) (see ***Note 2***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/opt/jp1pc/agto/nls/$LANG/	—	The storage directory for message catalogs
	jpcagtomsg.cat	The message catalog
/opt/jp1pc/agto/store/	—	The base directory of Agent Store
	jpcsto	The executable program of the Agent Store service

Files and Directories of Agent for Oracle (UNIX)		
Directory Name	**File Name**	**Description**
	stpqlpr	Internal command
	jpcsto.ini.instmpl	Internal file
	*.DAT	The definition file for a data model
/opt/jp1pc/agto/store/*instance-name*/	—	The base directory of Agent Store (for each instance) (see ***Note 2***)
	*.DB	The performance data file (for each instance) (see ***Note 3***)
	*.IDX	The index file for a performance data file (for each instance) (see ***Note 3***)
	*.LCK	The lock file for a performance data file (for each instance) (see ***Note 3***)
	jpcsto.ini	The service startup initialization file of Agent Store (for each instance) (see ***Note 2***)
	jpcsto.ini.model	Model file for the startup-information file of the Agent Store service (for each instance) (see ***Note 2***)
	*.DAT	The definition file for a data model (for each instance)
/opt/jp1pc/agto/store/*instance-name*/backup/	—	The default database backup directory (for each instance) (see ***Note 2***)
	datadir.lck	Internal file
/opt/jp1pc/agto/store/*instance-name*/dump/	-	The default database export directory (for each instance) (see ***Note 2***)
	datadir.lck	Internal file
/opt/jp1pc/agto/store/log/	—	The directory for storing internal log files of Agent Store (for each instance) (see ***Note 2***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)

Files and Directories of Agent for Oracle (UNIX)		
Directory Name	**File Name**	**Description**
/opt/jp1pc/setup/	—	The storage directory for setup files
	jpcagtou.Z	The internal archive file (UNIX)
	jpcagtow.EXE	The internal archive file (Windows)

Note 1: The user creates these files and directories

Note 2: Created by executing the `jpcinssetup` command.

Note 3: Created when the Agent Store service starts.

E.10 List of Agent for Microsoft SQL Server Files and Folders

The following table lists the files and folders of Agent for Microsoft SQL Server.

Table E.17 Files and Folders of Agent for Microsoft SQL Server

Files and Folders of Agent for Microsoft SQL Server		
Folder Name	**File Name**	**Description**
installation-folder\agtq\	-	Base folder of Agent for Microsoft SQL Server
	PATCHLOG.TXT	Internal file
	insrules.dat	Definition file for instance startup environment rules
	jpcagtras.bat	Program for collecting maintenance information
	jpcagtras.exe	Program for collecting maintenance information
installation-folder\agtq\agent\	-	Base folder of Agent Collector
	jpcagtq.exe	Service executing program for Agent Collector
	jpcagt.ini.instmpl	Internal file
	inssetup.bat.instmpl	Internal file
installation-folder\agtq\agent*instance-name*\	--	Base folder for Agent Collector (for each instance) (see ***Note 1***)
	jpcagt.ini	Service startup initialization file of Agent Collector (for each instance) (see ***Note 1***)
	jpcagt.ini.model	Sample of a service startup initialization file of Agent Collector (for each instance) (see ***Note 1***)
installation-folder\agtq\agent*instance-name*\log\	--	Folder for storing internal log files for Agent Collector (for each instance) (see ***Note 1***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\agtq\lib\	--	Message catalog installation folder
	jpcagtqmsg.dll	Internal file

Files and Folders of Agent for Microsoft SQL Server		
Folder Name	**File Name**	**Description**
installation-folder\agtq\sql\	--	Folder for storing scripts
	sp_drop.sql	SQL script for deleting stored procedures for Microsoft SQL Server
	sp_inst.sql	SQL script for registering stored procedures from Microsoft SQL Server
installation-folder\agtq\store\	--	Base folder for Agent Store
	jpcsto.exe	Service executing program for Agent Store
	stpqlpr.exe	Program for executing backup and export of the Store database
	*.DAT	Data model definition file
installation-folder\agtq\store*instance-name*\	--	Base folder for Agent Store (for each instance) (see ***Note 1***)
	*.DB	Performance data file (for each instance) (see ***Note 2***)
	*.IDX	Index file for a performance data file (for each instance) (see ***Note 2***)
	*.LCK	Lock file for a performance data file (for each instance) (see ***Note 2***)
	jpcsto.ini	Service startup initialization file of Agent Store (for each instance) (see ***Note 1***)
	jpcsto.ini.model	Sample of a service startup initialization file of Agent Store (for each instance) (see ***Note 1***)
	*.DAT	Data model definition file (for each instance) (see ***Note 1***)
installation-folder\agtq\store*instance-name*\backup\	--	Default database backup folder (for each instance) (see ***Note 1***)
	datadir.lck	Internal file
installation-folder\agtq\store*instance-name*\dump\	--	Default database export folder (for each instance) (see ***Note 1***)
	datadir.lck	Internal file

Files and Folders of Agent for Microsoft SQL Server		
Folder Name	**File Name**	**Description**
installation-folder\agtq\store*instance-name*\log\	--	Folder for storing internal log files of Agent Store (for each instance) (see ***Note 1***)
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	--	Storage folder for setting file
	jpcagtqu.Z	Internal archive file (UNIX)
	jpcagtqw.EXE	Internal archive file (Windows)

Note 1: These are created by executing the `jpcinssetup` command.

Note 2: Created when the Agent Store service starts.

E.11 List of Agent for Microsoft Exchange Server Files and Folders

The following table lists the files and folders for Agent for Microsoft Exchange Server.

Table E.18 Files and Folders of Agent for Microsoft Exchange Server

Files and Folders of Agent for Microsoft Exchange Server		
Folder Name	**File Name**	**Explanation**
installation-folder\agtz\	--	Root folder for Agent for Microsoft Exchange Server
	PATCHLOG.TXT	Internal file
installation-folder \agtz\agent\	--	Root folder for the Agent Collector service
	jpcagtz.exe	Executable program for the Agent Collector service
	jpcagt.ini	Settings file for the Agent Collector service
	jpcagt.ini.model	Model file for the settings file for the Agent Collector service
installation-folder \agtz\agent\log\	--	Folder for storing internal log files for the Agent Collector service
	msglog01	Log file
	msglog02	Log file (alternate file)
	msglog03	Log file (alternate file)
	msglog04	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder \agtz\lib\	--	Folder for storing message catalogs
	jpcagtzmsg.dll	Internal file
installation-folder \agtz\store\	--	Root folder for the Agent Store service
	jpcsto.exe	Executable program for the Agent Store service
	stpqlpr.exe	Internal command
	*.DAT	Data model definition file
	*.DB	Performance data file
	*.IDX	Index file for performance data files
	*.LCK	Lock file for performance data files
	jpcsto.ini	Settings file for the Agent Store service
	jpcsto.ini.model	Model file for the settings file for the Agent Store service

Files and Folders of Agent for Microsoft Exchange Server		
Folder Name	**File Name**	**Explanation**
installation-folder \agtz\store\backup\	--	Standard database backup folder
	datadir.lck	Internal file
installation-folder \agtz\store\dump\	--	Standard database export folder
	datadir.lck	Internal file
installation-folder \agtz\store\log\	--	Folder for storing internal log files for the Agent Store service
	msglog01	Log file
	msglog02	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
installation-folder\setup\	--	Folder for storing setup files
	jpcagtzu.Z	Archive file for Agent setup (UNIX)
	jpcagtzw.EXE	Archive file for Agent setup (Windows)

E.12 List of Agent for DB2 Files and Directories

Table E.19 lists the files and directories of Agent for DB2.

Table E.19 Files and Directories of Agent for DB2

Files and Directories of Agent for DB2		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtr/	—	The base directory of Agent for DB2
	PATCHLOG.TXT	Internal file
	insrules.dat	Definition file for instance startup environment rules
/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/*instance-name*	-	The root directory of the Agent Collector service (for each instance) See ***Note 1***
	jpcagt.ini	The service startup initialization file of Agent Collector (for each instance) See ***Note 1***
	jpcagt.ini.model	The model file for a service startup initialization file of the Agent Collector service (for each instance) See ***Note 1***

Files and Directories of Agent for DB2		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtr/agent/*instance-name*/log/	—	The directory for storing internal log files of the Agent Collector service (for each instance) See ***Note 1***
	msglog01	Log file
	msglog02	Log file (alternate file)
	msglog03	Log file (alternate file)
	msglog04	Log file (alternate file)
	nslog01	Log file
	nslog02	Log file (alternate file)
/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/*instance-name*/	—	The base directory of the Agent Store service (for each instance) See ***Note 1***
	*.DB	Performance data file (for each instance) See ***Note 2***
	*.IDX	Index file for a performance data file (for each instance) See ***Note 2***
	*.LCK	Lock file for a performance data file (for each instance) See ***Note 2***
	jpcsto.ini	The service startup initialization file of the Agent Store service (for each instance) See ***Note 1***
	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/jp1pc/agtr/store/*instance-name*/backup/	—	The default database backup directory (for each instance) See ***Note 1***
/opt/jp1pc/agtr/store/*instance-name*/dump/	—	The default database export directory (for each instance) See ***Note 1***

Files and Directories of Agent for DB2		
Directory Name	**File Name**	**Description**
/opt/jp1pc/agtr/store/*instance-name*/log/	—	The directory for storing internal log files of the Agent Store service (for each instance) See ***Note 1***
/opt/jp1pc/setup/	—	The storage directory for setup files
	jpcagtru.Z	Internal archive file (UNIX)
	jpcagtrw.EXE	Internal archive file (Windows)

Notes 1: Created by executing the `jpcinssetup` command.

Notes 2: Created when the Agent Store service starts.

Appendix F Program Version Compatibility with the Data Model Version

In addition to the product version, an Agent includes a data model version.

When you upgrade an Agent, the data model might also be upgraded. However, because upward compatibility of the data model versions is maintained, newer version data models can use report and alarm definitions created in older versions.

The following table lists the correspondence between agent versions, data model versions, and alarm table versions:

Table F.1 Correspondence Between Agent Versions, Data Model Versions, and Alarm Table Versions

Agent Name	Agent Version	Data Model Version	Alarm Table Version in a Solution Set
Agent for RAID	5.5	7.0	7.00
	5.1	7.0	7.00
	5.0	7.0	-
	4.1	6.0	-
	4.0	6.0	-
	3.5	5.0	-
	3.3	-	-
	3.2	4.0	-
	3.1	4.0	-
	3.0	4.0	-
Agent for RAID Map	5.5	4.0	-
	5.1	4.0	-
	5.0	4.0	-
	4.1	4.0	-
	4.0	4.0	-
	3.5	4.0	-
	3.3	-	-
	3.2	4.0	-
	3.0	4.0	-
Agent for Platform Windows	5.5	5.0	7.50
	5.1	5.0	7.50
	5.0	5.0	-
	4.1	4.0	-

Agent Name	Agent Version	Data Model Version	Alarm Table Version in a Solution Set
	4.0	4.0	-
	3.5	4.0	-
	3.3	-	-
	3.2	4.0	-
	3.0	4.0	-
Agent for Platform UNIX	5.5	5.1	7.50
	5.1	5.1	7.50
	5.0	5.1	-
		5.0	-
	4.1	5.0	-
	4.0	4.0	-
	3.5	4.0	-
	3.3	-	-
	3.2	4.0	-
	3.0	4.0	-
Agent for SAN Switch	5.5	5.0	7.00
	5.1	5.0	7.00
	5.0	5.0	-
	4.1	4.0	-
	4.0	4.0	-
	3.5	4.0	-
	3.3	-	-
	3.2	4.0	-
	3.0	4.0	-
Agent for NAS	5.0	5.0	7.00
	4.1	-	-
	4.0	5.0	-
	3.5	5.0	-
	3.3	-	-
	3.2	-	-
	3.0	4.0	-
Agent for Oracle	5.5	5.0	8.00
	5.1	4.1	7.50

Agent Name	Agent Version	Data Model Version	Alarm Table Version in a Solution Set
	5.0	4.0	-
	4.1	4.0	-
	4.0	4.0	-
	3.5	4.0	-
	3.3	-	-
	3.2	-	-
	3.0	4.0	-
Agent for Microsoft SQL Server	5.5	4.0	8.00
	5.1	3.1	7.50
	5.0	3.0	-
	4.1	3.0	-
	4.0	-	-
	3.5	3.0	-
	3.3	3.0	-
Agent for Microsoft Exchange Server	5.5	3.0	8.00
Agent for DB2	5.5	5.0	8.00
	5.1	5.0	8.00
	5.0	4.1	-
	4.1	3.0	-

The following uses examples where data model versions 5.0 and 6.0 coexist to describe version compatibility:

F.1 Displaying a Report

Reports that are defined with data model version 5.0 can be displayed from Agents defined with data model version 5.0 or 6.0. Reports that are defined with data model 6.0 can be displayed only with Agents defined with data model version 6.0.

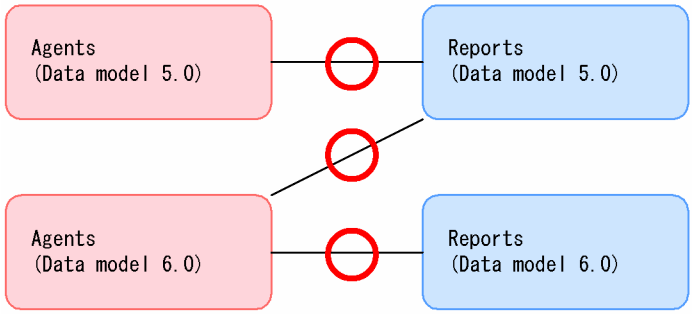


Figure F.1 Data Model Version Compatibility (When Displaying a Report)

F.2 Binding an Alarm Table

Alarm tables that are defined with data model version 5.0 can be bound to Agents defined with data model version 5.0 or 6.0. Alarm tables that are defined with data model 6.0 can be bound only to Agents defined with data model version 6.0.

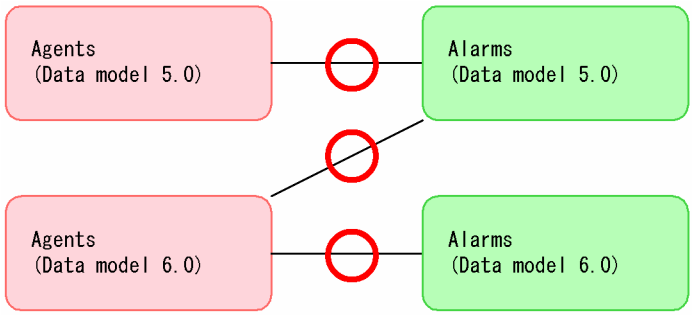


Figure F.2 Data Model Version Compatibility (When Binding an Alarm Table)

F.3 Associating a Drill-down Report with a Report

Drill-down reports that are defined with data model version 5.0 can be associated with reports defined with data model version 5.0 or 6.0. Drill-down reports that are defined with data model 6.0 can be associated only with reports defined with data model version 6.0.

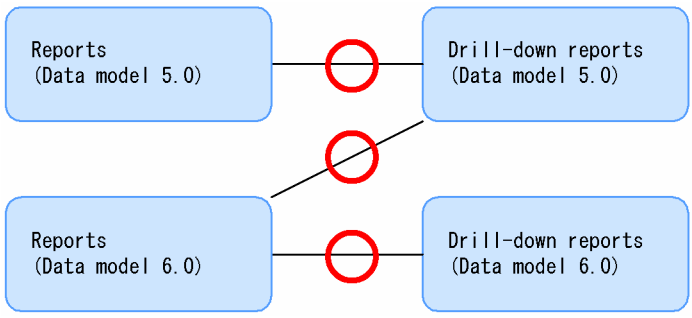


Figure F.3 Data Model Version Compatibility (When Associating a Drill-down Report with a Report)

Appendix G Structure of MIB Objects

This appendix describes Management Information Base (MIB) objects for SNMP traps used by Tuning Manager.

MIB files (`HTM-ALRM-MIB.txt`) are stored in the following directories:

In Windows:

Tuning-Manager-installation-folder`\docs`

In UNIX:

`/opt/jp1pc/docs/`

The following shows the structure of MIB objects.

```
iso(1).org(3).dod(6).internet(1).private(4).enterprise(1)
  |
hitachi(116)
  |
systemAP(7)
  |
hiCommandMgr(17)
  |
PerformanceManagement(12)
|
.
PerformanceManagementTrap(2)
```

The following table shows the contents of the MIB object in Tuning Manager:

Table G.1 Contents of MIB Object

Object ID	Description
.1.3.6.1.4.1.116.5.17.12.2.1	Agent's instance number (ASCII code for the instance number)
.1.3.6.1.4.1.116.5.17.12.2.3	Report information defined during the alarm definition *host-name-of-Tuning-Manager@report-ID*
.1.3.6.1.4.1.116.5.17.12.2.6	Name of the Tuning Manager category and the condition expression of the alarm
.1.3.6.1.4.1.116.5.17.12.2.7	Alarm value See ***Note 1***
.1.3.6.1.4.1.116.5.17.12.2.8	Alarm status: ▪ `NORMAL: Normal` ▪ `WARNING: Warning` ▪ `CRITICAL: Abnormal`
.1.3.6.1.4.1.116.5.17.12.2.9	Product type identifier of the Agent For example, this object is `Windows` for Agent for Platform (Windows), `ORACLE` for Agent for Oracle
.1.3.6.1.4.1.116.5.17.12.2.10	Management unit that caused the alarm See ***Note 2***

Object ID	Description
.1.3.6.1.4.1.116.5.17.12.2.12	Agent's instance name
.1.3.6.1.4.1.116.5.17.12.2.13	Contents of the message text that was specified during the alarm definition See ***Note 3***
.1.3.6.1.4.1.116.5.17.12.2.14	Alarm table name
.1.3.6.1.4.1.116.5.17.12.2.15	Alarm name
.1.3.6.1.4.1.116.5.17.12.2.16	Agent host name
.1.3.6.1.4.1.11.2.17.2.2.0	Agent host name (fully qualified domain name)

Note 1:.For multiple instance records, the following action occurs:

- If any critical or warning level value is detected in the target instances:

  The user-defined message specified during the alarm definition is displayed in the message text (MIB object ID: 1.3.6.1.4.1.116.5.17.12.2.13). In this case, the data element value that triggered the event is set to the instance value where the exceeded threshold value was first detected.

- When the status changes from Abnormal or Warning to Normal:

  Although a normal event is issued, nothing is set for the message text because all instance values are within the normal range, and the value that causes an event is not determined. In this case, the data element value that triggered the event is set to `<OK>`.

Note 2: Do not use this object ID during operation because the ID is used internally by Tuning Manager.

Note 3: The following values are displayed in the message text:

- 1: Alarm updated/deleted: The alarm definition was updated or deleted.
- 2: Alarm deactivated: The alarm status changes to inactive.
- 3: Alarm cleared: The alarm binding is cleared.
- 4: Alarm expired: The current time is outside the alarm evaluation time period
- 5: User-defined message: The alarm status changes from Normal to Abnormal or Warning. Alternatively, in a single instance record, the alarm status changes from Abnormal to Warning or Normal.
- 6: None: In a multiple instance record, the alarm status changes from Abnormal to Warning or Normal.

For the above 1 to 4, the data element value that triggered the event is set to `(N/A)`.

Acronyms and Abbreviations

API	application program interface
ASCII	American Standard Code for Information Interchange
CLPR	Cache Logical PaRtition
CPU	Central Processing Unit
DLL	Dynamic Linking Library
EFCM	Enterprise Fabric Connectivity Manager
HA	High Availability
HTM	HiCommand Tuning Manager
I/O	Input/Output
ID	identifier, identification
IP	Internet Protocol
MIB	Management Information Base
NAS	Network Attached Storage
NFS	Network File System
NIC	Network Interface Card
OS	Operating System
POSIX	Portable Operating System Interface for UNIX
RAID	redundant array of inexpensive disks
SAN	Storage Area Network
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
TCP	transmission control protocol
WWN	World Wide Name
WWW	World Wide Web

Index

S

T