HITACHI
Inspire the Next

JP1 Version 12

JP1/Performance Management - Remote Monitor for Virtual Machine Description, User's Guide and Reference

3021-3-D80-20(E)

Notices

■ Relevant program products

For details about the applicable OS versions, and the service packs and patches required for JP1/Performance Management - Remote Monitor for Virtual Machine, see the Release Notes.

JP1/Performance Management - Manager (For Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019):

P-2A2C-AACL JP1/Performance Management - Manager version 12-50

The above product includes the following:

P-CC2A2C-5ACL JP1/Performance Management - Manager version 12-50

P-CC2A2C-5RCL JP1/Performance Management - Web Console version 12-50

JP1/Performance Management - Manager (For CentOS 6 (x64), CentOS 7, CentOS 8, Linux 6 (x64), Linux 7, Linux 8, Oracle Linux 6 (x64), Oracle Linux 7, Oracle Linux 8, SUSE Linux 12 and SUSE Linux 15):

P-812C-AACL JP1/Performance Management - Manager version 12-50

The above product includes the following:

P-CC812C-5ACL JP1/Performance Management - Manager version 12-50

P-CC812C-5RCL JP1/Performance Management - Web Console version 12-50

JP1/Performance Management - Remote Monitor for Virtual Machine (For Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019):

P-2A2C-GVCL JP1/Performance Management - Remote Monitor for Virtual Machine version 12-50

The above product includes the following:

P-CC2A2C-5VCL JP1/Performance Management - Remote Monitor for Virtual Machine version 12-50

P-CC2A2C-AJCL JP1/Performance Management - Base version 12-00

These products include the parts developed under licenses from other companies.

■ Export restrictions

If you export this product, please check all restrictions (for example, Japan's Foreign Exchange and Foreign Trade Law, and USA export control laws and regulations), and carry out all required procedures.

If you require more information or clarification, please contact your Hitachi sales representative.

■ Trademarks

HITACHI, Cosminexus, HiRDB, JP1, OpenTP1, ServerConductor, uCosminexus are either trademarks or registered trademarks of Hitachi, Ltd. in Japan and other countries.

Docker is a trademark or registered trademarks of Docker, Inc. in the United States and/or other countries.

Hyper-V is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Internet Explorer is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

■ Microsoft product screen shots

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

■ Microsoft product name abbreviations

This manual uses the following abbreviations for Microsoft product names.

Abbreviation		Full name or meaning
Hyper-V	2012 Hyper-V	Microsoft(R) Windows Server(R) 2012 Hyper-V(R)
	2012 R2 Hyper-V	Microsoft(R) Windows Server(R) 2012 R2 Hyper-V(R)
	2016 Hyper-V	Microsoft(R) Windows Server(R) 2016 Hyper-V(R)
	2019 Hyper-V	Microsoft(R) Windows Server(R) 2019 Hyper-V(R)
Internet Explorer		Windows(R) Internet Explorer(R)
Windows Server 2012	Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Standard
		Microsoft(R) Windows Server(R) 2012 Datacenter
	Windows Server 2012 R2	Microsoft(R) Windows Server(R) 2012 R2 Standard
		Microsoft(R) Windows Server(R) 2012 R2 Datacenter
Windows Server 2016		Microsoft(R) Windows Server(R) 2016 Datacenter
		Microsoft(R) Windows Server(R) 2016 Standard
Windows Server 2019		Microsoft(R) Windows Server(R) 2019 Datacenter
		Microsoft(R) Windows Server(R) 2019 Standard
WSFC		Microsoft(R) Windows Server(R) Failover Cluster

Windows is sometimes used generically, referring to Windows Server 2012, Windows Server 2016 and Windows Server 2019.

■ Restrictions

Information in this document is subject to change without notice and does not represent a commitment on the part of Hitachi. The software described in this manual is furnished according to a license agreement with Hitachi. The license

■ Issued

Jan. 2021: 3021-3-D80-20(E)

■ Copyright

Summary of amendments

The following table lists changes in this manual (3021-3-D80-20(E)) and product changes related to this manual.

Changes	Location
Added the Podman environment of Red Hat Enterprise Linux(R) Server 8 to the targets whose resources are to be monitored.	*1.1*, *1.8*, *2.1.1(4)(b)*, *2.1.4*, *2.2.2*, *2.4.2*, *2.4.3*, *2.5.5*, *2.6.5*, *3.3.4*, *3.4.2*, *Chapter 4*, *Chapter 5*, *7.2.1(1)(e)*, *Appendix C*, *D.2(6)*, *E.2*, *E.3*, *F.1*, *L.5*
Red Hat Enterprise Linux(R) Server 8 is now supported when KVM is to be monitored.	*2.1.1(4)(b)*
Added `SSH_Type` to the instance information.	*2.4.2*, *2.6*, *7.2.1(1)(c)*
Added Windows Server 2019 standard OpenSSH as an SSH client when KVM is to be monitored.	*2.5.7*
Changed the version of the data model to 9.0.	*Chapter 4*
Changed the version of the alarm table for monitoring templates to 12.50.	*Chapter 4*, *Appendix H*
With the change of data model, the versions of the following reports that use fields for which the type of their data model was changed in version 9.0 or that refer to such reports were changed: • Host CPU Used Status • Host Disk Used • Host Disk Used Status • VM CPU Allocation Value • VM CPU Insufficient • VM CPU Trend • VM CPU Used • VM CPU Used Status • VM Disk Used • VM Disk Used Status	*Chapter 4*
Added the next item to records: • POD Status Detail • POD Status Interval • POD Container Status Interval	*Chapter 5*
Added the fields for the following records: • Host Logical Disk Status (PI_HLDI) • Host Status (PI) • VM CPU Status (PI_VCI) • VM Logical Disk Status (PI_VLDI) • VM Status Detail (PD_VM) • VM Status (PI_VI)	*Chapter 5*
Added the descriptions or actions for when the following messages are output: • KAVL20014-W • KAVL20201-W • KAVL20517-W	*6.4*

Changes	Location
• KAVL20524-W • KAVL20526-W	*6.4*

In addition to the above changes, minor editorial corrections were made.

Preface

This manual describes the functions of JP1/Performance Management - Remote Monitor for Virtual Machine and the records it collects.

■ Intended readers

This manual is intended for the following readers:

- Users who wish to design or construct an operation monitoring system.
- Users who wish to define conditions for collecting performance data.
- Users who wish to define reports and alarms.
- Users who wish to use collected performance data to monitor a system.
- Users who wish to use monitoring results to consider what actions should be taken or delegated to improve the system.

Also, this manual assumes that the reader is familiar with operations and virtual environments of monitored systems and that the reader is knowledgeable about the OS.

For details about how to configure and operate a system that uses JP1/Performance Management, also see the following manuals:

- *JP1/Performance Management Planning and Configuration Guide*
- *JP1/Performance Management User's Guide*
- *JP1/Performance Management Reference*

■ Organization of this manual

This manual is organized into the following parts, and is a common reference for the following supported OSs: Windows Server 2012, Windows Server 2012 R2, Windows Server 2016 and Windows Server 2019. Any platform-dependent differences are noted separately in the manual.

PART 1: Overview

Part 1 provides an overview of JP1/Performance Management - Remote Monitor for Virtual Machine.

PART 2: Setup and Operation

Part 2 explains how to install and set up JP1/Performance Management - Remote Monitor for Virtual Machine, and how to operate it in a cluster system.

PART 3: Reference

Part 3 explains the monitoring templates, records, and messages of JP1/Performance Management - Remote Monitor for Virtual Machine.

PART 4: Troubleshooting

Part 4 explains the actions to take when problems occur in JP1/Performance Management - Remote Monitor for Virtual Machine.

■ Document conventions

The Hitachi Virtualization Manager (HVM) name has been changed to Hitachi logical partitioning manager (LPAR manager, or LP). If you are using HVM based logical partitioning feature, substitute references to Hitachi logical partitioning manager (LPAR manager, or LP) with HVM.

■ Conventions: Fonts and symbols

The following table explains the text formatting conventions used in this manual:

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

The following table explains the symbols used in this manual:

Symbol	Convention
|	In syntax explanations, a vertical bar separates multiple items, and has the meaning of OR. For example: `A|B|C` means A, or B, or C.
{ }	In syntax explanations, curly brackets indicate that only one of the enclosed items is to be selected. For example: `{A|B|C}` means only one of A, or B, or C.
[ ]	In syntax explanations, square brackets indicate that the enclosed item or items are optional. For example: `[A]` means that you can specify `A` or nothing. `[B|C]` means that you can specify `B`, or `C`, or nothing.
...	In coding, an ellipsis (...) indicates that one or more lines of coding have been omitted. In syntax explanations, an ellipsis indicates that the immediately preceding item can be repeated as many times as necessary. For example: `A, B, B, ...` means that, after you specify `A, B,` you can specify B as many times as necessary.
x	Multiplication sign

Symbol	Convention
/	Division sign

■ Conventions: Version numbers

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

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

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

Contents

Index 739

1 Overview of PFM - RM for Virtual Machine

This chapter provides an overview of PFM - RM for Virtual Machine.

1.1 Purpose of performance monitoring using PFM - RM for Virtual Machine

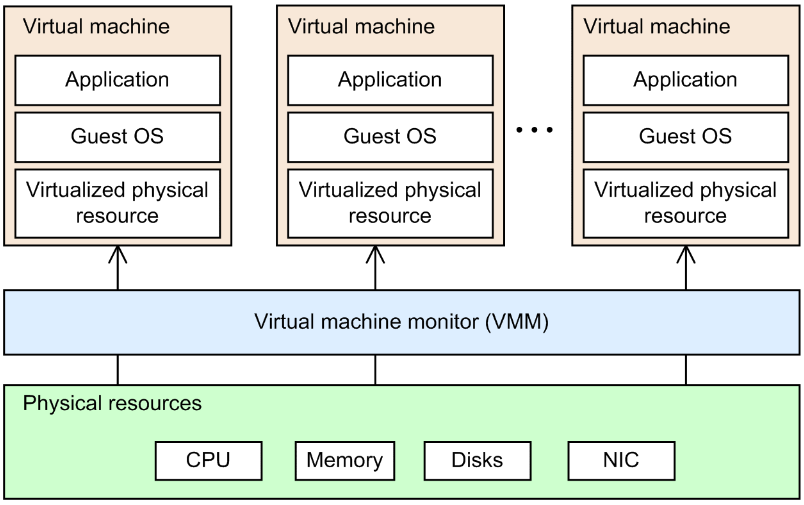
Server performance has improved in recent years to the point that it is often not possible for a single system to make full use of the resources of a single server. As a way to resolve this problem, more attention is being given to virtual environments because they can utilize system resources more effectively. Virtual environments also offer the following benefits:

- OSs and devices can be shared and centrally managed.
- Since fewer servers are used, power and air conditioning costs can be reduced.
- Multiple old servers can be replaced with a virtual machine on a single new server.

Reducing power and air conditioning costs by reducing the number of servers can also lead to a greener IT environment.

The following figure shows a typical structure of a virtual environment in which multiple virtual machines are configured on a single physical server.

Figure 1–1: Typical structure of a virtual environment



Each virtual machine runs on virtualized physical resources. A virtual environment reaches the highest possible efficiency level when its physical resources are utilized fully, and when there is no resource insufficiency. If physical resources are excessively consumed or are hardly being utilized, a virtual environment is not being used effectively. Therefore, it is important to decide what amount of physical resources should be allocated to each virtual machine.

PFM - RM for Virtual Machine is a program that monitors the performance of a virtual environment, and collects and manages performance data.

PFM - RM for Virtual Machine enables you to monitor the resource consumption status in various types of virtual environments. PFM - RM for Virtual Machine can monitor the following types of virtual environments:

- VMware vSphere ESXi
- Hyper-V
- Logical partitioning feature, which is logical partitioning for Hitachi compute products#
- KVM
- Docker environment
- Podman environment

#

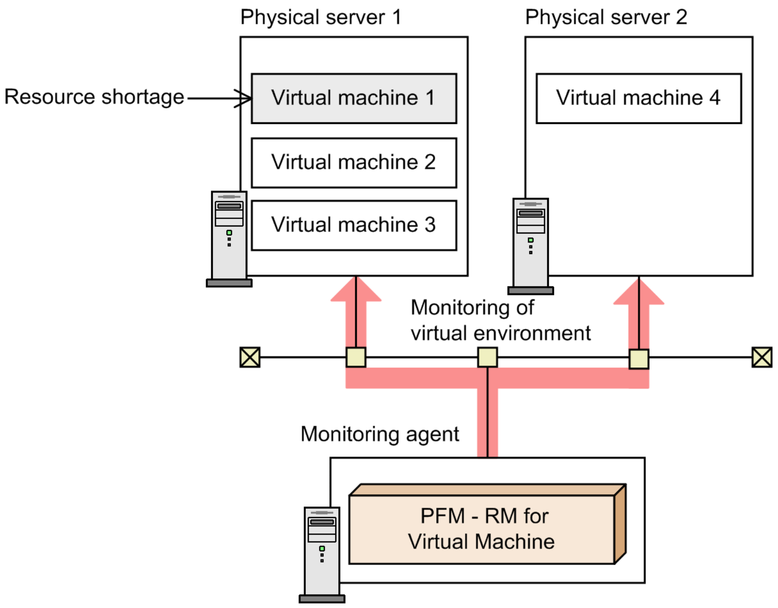
Subsequently, it is referred to as *logical partitioning feature* in this manual.

Resource monitoring is an important aspect of effective utilization of a virtual environment. Using PFM - RM for Virtual Machine, you can determine the degree to which a virtual environment is configured effectively by monitoring the consumption of resources in the virtual environment. PFM - RM for Virtual Machine can monitor the following resources in a virtual environment:

- CPU
- Memory
- Disks
- Network

The following figure shows an example of using PFM - RM for Virtual Machine to monitor a VMware system.

Figure 1–2: Monitoring a virtual environment with PFM - RM for Virtual Machine (monitoring a VMware system)

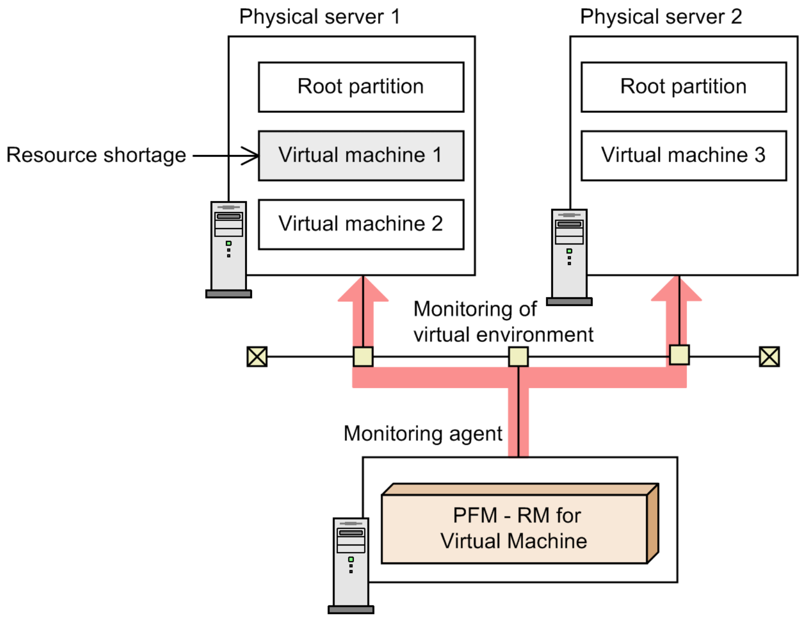


In this example, PFM - RM for Virtual Machine has detected a resource shortage in Virtual machine 1 on Physical server 1. In this case, the following corrective measures are possible:

- Increase the amount of Physical server 1 resources allocated to Virtual machine 1.
- Add resources to Physical server 1.
- Move Virtual machine 1 from Physical server 1 to Physical server 2, which has available resources.

The following figure shows an example of using PFM - RM for Virtual Machine to monitor a Hyper-V system.

Figure 1–3: Monitoring a virtual environment with PFM - RM for Virtual Machine (monitoring a Hyper-V system)

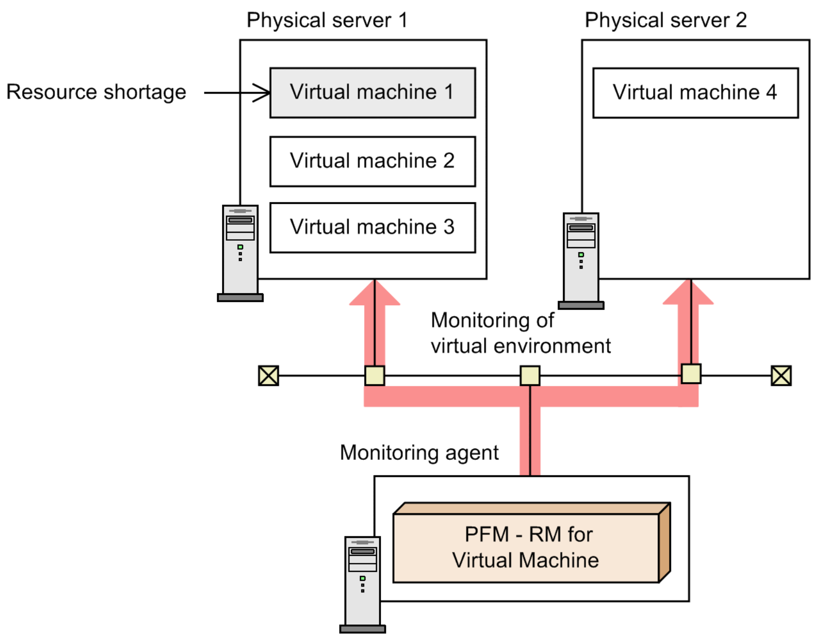


In this example, PFM - RM for Virtual Machine has detected a resource shortage in Virtual machine 1 on Physical server 1. In this case, the following corrective measures are possible:

- Increase the amount of Physical server 1 resources allocated to Virtual machine 1.
- Add resources to Physical server 1.
- Move Virtual machine 1 from Physical server 1 to Physical server 2, which has available resources.

The following figure shows an example of using PFM - RM for Virtual Machine to monitor a KVM system.

Figure 1–4: Monitoring a virtual environment with PFM - RM for Virtual Machine (monitoring a KVM system)

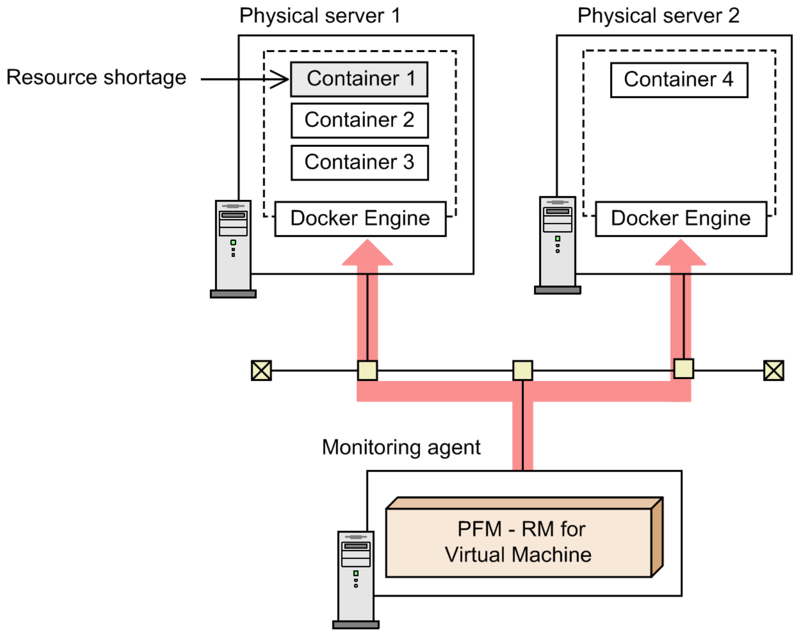


In this example, PFM - RM for Virtual Machine has detected a resource shortage in Virtual machine 1 on Physical server 1. In this case, the following corrective measures are possible:

- Increase the amount of Physical server 1 resources allocated to Virtual machine 1.
- Add resources to Physical server 1.
- Move Virtual machine 1 from Physical server 1 to Physical server 2, which has available resources.

The following figure shows an example of using PFM - RM for Virtual Machine to monitor a Docker environment.

Figure 1–5: Monitoring a virtual environment with PFM - RM for Virtual Machine (monitoring a Docker environment)



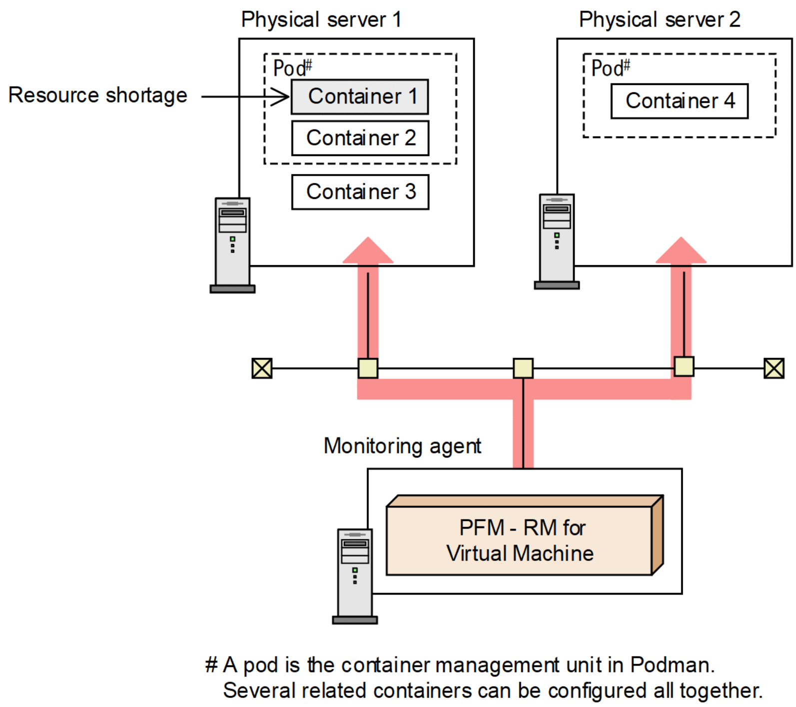
In this example, PFM - RM for Virtual Machine has detected a resource shortage in Container 1 on Physical server 1. In this case, the following corrective measures are possible:

- Increase the amount of Physical server 1 resources allocated to Container 1.
- Add resources to Physical server 1.
- Move Container 1 from Physical server 1 to Physical server 2, which has available resources.

By using PFM - RM for Virtual Machine to monitor a virtual environment, you can take appropriate corrective actions, making it possible to maintain an ideal virtual environment in which resource shortages and excesses do not occur.

The following figure shows an example of using PFM - RM for Virtual Machine to monitor a Podman environment.

Figure 1–6: Monitoring a virtual environment with PFM - RM for Virtual Machine (monitoring a Podman environment)



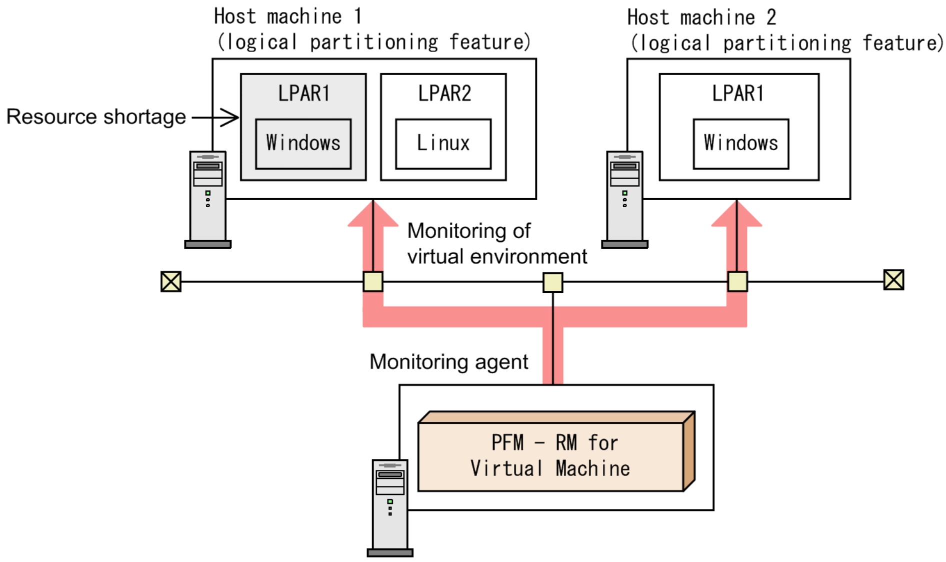
In this example, PFM - RM for Virtual Machine has detected a resource shortage in Container 1 on Physical server 1. In this case, the following corrective measures are possible:

- Increase the amount of Physical server 1 resources allocated to Container 1.
- Add resources to Physical server 1.
- Move Container 1 from Physical server 1 to Physical server 2, which has available resources.

By using PFM - RM for Virtual Machine to monitor a virtual environment, you can take appropriate corrective actions, making it possible to maintain an ideal virtual environment in which resource shortages and excesses do not occur.

The following figure shows an example of using PFM - RM for Virtual Machine to monitor a system with logical partitioning feature.

Figure 1–7: Monitoring a virtual environment with PFM - RM for Virtual Machine (monitoring a system with logical partitioning feature)



In this example, PFM - RM for Virtual Machine has detected a resource shortage in LPAR1 on the host machine 1. In this case, the following corrective measures are possible:

- Increase the amount of resources on the host machine 1 allocated to LPAR1.
- Add resources to host machine 1.
- Move LPAR1 from host machine 1 to host machine 2, which has available resources.

1.2 Features of PFM - RM for Virtual Machine

PFM - RM for Virtual Machine is a program that monitors the performance of a virtual environment, and collects and manages its performance data.

This subsection describes the features of PFM - RM for Virtual Machine.

- Analysis of the operation status of a virtual environment

  PFM - RM for Virtual Machine collects and summarizes performance data, such as CPU usage and memory usage, from the monitored virtual environment, enabling the user to easily analyze the operation status of the virtual environment by graphically displaying its trends and history.

  A single instance of PFM - RM for Virtual Machine can monitor multiple virtual environments.
- Early detection of operational problems in a virtual environment and reporting of troubleshooting information

  When a problem such as a resource shortage occurs in the monitored virtual environment, PFM - RM for Virtual Machine can quickly alert the user via e-mail, thus ensuring early problem detection. Furthermore, by graphically displaying the information related to the problem, PFM - RM for Virtual Machine provides the necessary troubleshooting data.

To use PFM - RM for Virtual Machine, you need PFM - Manager, PFM - Base, and PFM - Web Console. However, PFM - Base is not necessary if you install both PFM - Manager and PFM - RM for Virtual Machine on the same host.

Note

In addition to PFM - RM for Virtual Machine, PFM - RM for Platform is a Performance Management product that can analyze the operation status of multiple hosts. Like PFM - RM for Virtual Machine, PFM - RM for Platform need not be installed on the host to be monitored. However, the types of performance data that PFM - RM for Platform can collect differ from those that can be collected by PFM - RM for Virtual Machine.

Performance data that can be collected by PFM - RM for Virtual Machine

- Operation status of physical servers
- Operation status of virtual machines

Performance data that can be collected by PFM - RM for Platform

- Operation status of Windows
- Operation status of UNIX

To collect and manage the operation status of an OS, consider installing PFM - RM for Platform.

The following subsections explain the tasks you can perform using PFM - RM for Virtual Machine.

1.2.1 Collecting performance data from a virtual environment

Using PFM - RM for Virtual Machine, you can collect performance data, such as the statistical information of a virtual environment that is running on the target physical server.

You can use the collected performance data for the following purposes:

- Graphically displaying the operation status of the virtual environment

You can use PFM - Web Console to process the performance data in a graphical format called a *report*, and display it. A report makes it easier to understand and analyze the operation status of the virtual environment.

The following types of reports are available:

- Real-time report

  This report shows the current status of the virtual environment being monitored. It is primarily used for checking the current state of the system and any problems that have occurred. Performance data is used immediately, at the time of its collection, to display a real-time report.

- Historical report

  This report shows the history of the virtual environment being monitored. It is primarily used for analyzing trends in the operation status of the system. Performance data stored in the database of PFM - RM for Virtual Machine is used to display a historical report.

- Using the collected performance data to determine whether a problem has occurred

  You can configure the system to take an action, such as notifying the user, if the collected performance data shows some sort of abnormality.

1.2.2 Collecting performance data according to its characteristics

Performance data is collected in the form of *records*. Each record is divided into smaller units called *fields*. Records and fields are collectively referred to as *data models*.

Records are grouped into two record types according to their characteristics. The type of performance data that is collected in each record type is defined by PFM - RM for Virtual Machine. You use PFM - Web Console to select the performance data records to collect.

The record types of PFM - RM for Virtual Machine are as follows.

- Product Interval (PI) record type

  For a record of the PI record type, performance data is collected for a given time interval, such as the CPU resources of a virtual machine used each minute. The PI record type is used to analyze changes and trends in the system status over time.

- Product Detail (PD) record type

  For a record of the PD record type, performance data is collected that indicates the system status at a given point in time, such as configuration information data about a virtual machine. The PD record type is used to determine the system status at a given point in time.

For details about each record, see *5. Records*.

1.2.3 Saving the performance data

By storing the collected performance data in PFM - RM for Virtual Machine's database called the *Store database*, you can save performance data up to the present time and analyze historical trends in the operation status of the virtual environment. To analyze trends, you use a historical report.

Use PFM - Web Console to select which performance data records to store in the Store database. For details about how to use PFM - Web Console to select records, see the chapter that explains management of operation monitoring data in the *JP1/Performance Management User's Guide*.

1.2.4 Alerting the user about an operational problem in the virtual environment

Besides being used to display the performance of the virtual environment in the form of a report, performance data is used to alert the user when an operational problem or error occurs.

For example, when physical CPU usage rate exceeds 90%, an e-mail notification can be sent to the user. To send this notification, you set *usage of the physical CPU exceeds 90%* as the threshold for an abnormal condition, and you specify that an e-mail notification be sent to the user when this threshold is reached. The action that is taken when the threshold is reached is called an *action*. The following types of actions can be taken:

- E-mail notification
- Command execution
- SNMP trap issuance
- JP1 event issuance

An entity that defines a threshold and an action is called an *alarm*. A table that lists more than one alarm is called an *alarm table*. After you define an alarm table, you link it to PFM - RM for Virtual Machine. The act of linking an alarm table to PFM - RM for Virtual Machine is called *binding*. Once an alarm table is bound to PFM - RM for Virtual Machine, the user is notified when the performance data being collected by PFM - RM for Virtual Machine reaches the *threshold* defined by an alarm.

By defining alarms and actions in this way, you can discover operational problems in the virtual environment early, and take the necessary corrective actions.

For details about how to set alarms and actions, see the chapter that explains operation monitoring using alarms in the *JP1/Performance Management User's Guide*.

1.2.5 Defining reports and alarms easily

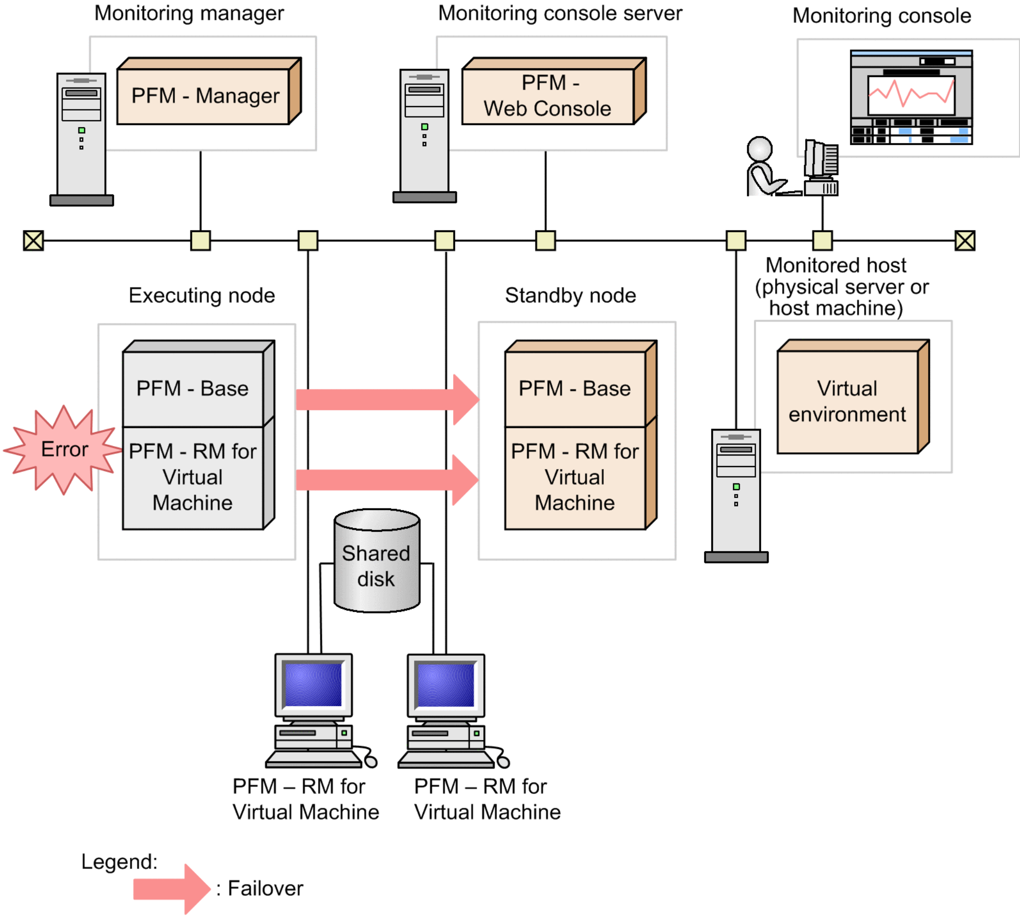
PFM - RM for Virtual Machine provides reports and alarms that predefine the necessary information, called *monitoring templates*. By using a monitoring template, you can easily prepare to monitor the operation status of the virtual environment without having to set up complicated definitions. A monitoring template can also be customized to fit the user's environment. For details about how to use a monitoring template, see the chapter that explains how to create reports for operational analysis or operation monitoring using alarms in the *JP1/Performance Management User's Guide*. For details about monitoring templates, see *4. Monitoring Template*.

1.2.6 Operations in a cluster system

When a cluster system is used, you can configure a highly reliable system that can continue to operate even when an error occurs in it. This allows you to operate and monitor Performance Management around the clock.

The figure below shows an example of operations when an error occurs at the monitored host in a cluster system.

Figure 1–8: Example of cluster system operations



Two environments with identical settings are configured, and the host that normally runs is defined as the *executing node*, while the host that is used when an error occurs is defined as the *standby node*.

For details about Performance Management operations in a cluster system, see *3. Operations in a Cluster System*.

1.3 Overview of collecting and managing performance data

The procedure for collecting and managing performance data differs depending on the type of record in which performance data is stored. Records in PFM - RM for Virtual Machine are classified according to the following two types:

- PI record type
- PD record type

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

- Performance data collection method
  PFM - RM for Virtual Machine collects performance data from monitoring targets.
  For the performance data collection method, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.
  For details about the values of collected performance data, see *5. Records*.
- Performance data management method
  PFM - RM for Virtual Machine manages performance data collected from monitoring targets.
  For the performance data management method, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.

You use PFM - Web Console to select which performance data to use out of the records collected and managed by PFM - RM. For the selection method, see the chapter that explains management of operation monitoring data in the *JP1/Performance Management User's Guide*.

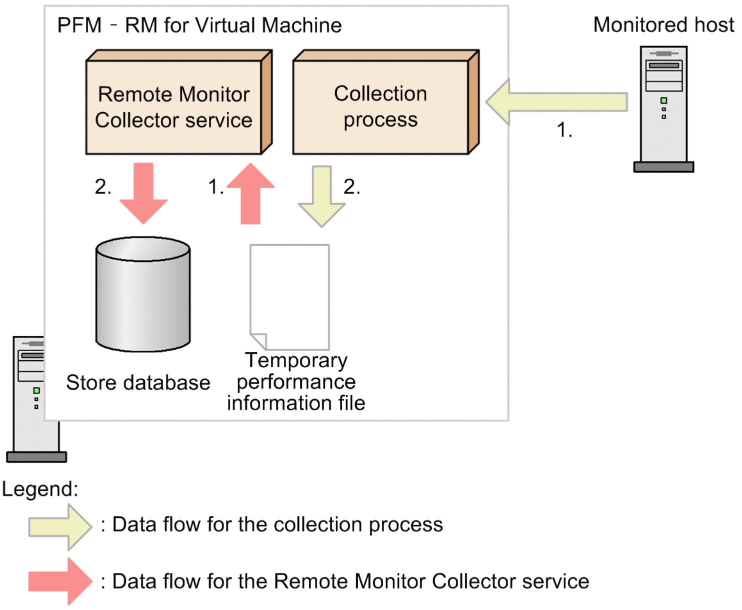
1.3.1 General procedure for performance data collection tasks

This subsection describes the flow of data and the general procedure for collecting performance data.

(1) Data flow during performance data collection

The following figure shows data flow that occurs when performance data is collected.

Figure 1–9: Data flow occurring during collection of performance data



- Data flow for the collection process
  1. The collection process connects to the monitored host, and collects performance data from the host.
  2. The collection process outputs the collected performance data to a temporary performance information file.
- Data flow for the Remote Monitor Collector service
  1. The Remote Monitor Collector service reads in the temporary performance information file that was output by the collection process.
  2. The Remote Monitor Collector service saves the performance data as records in the Store database.

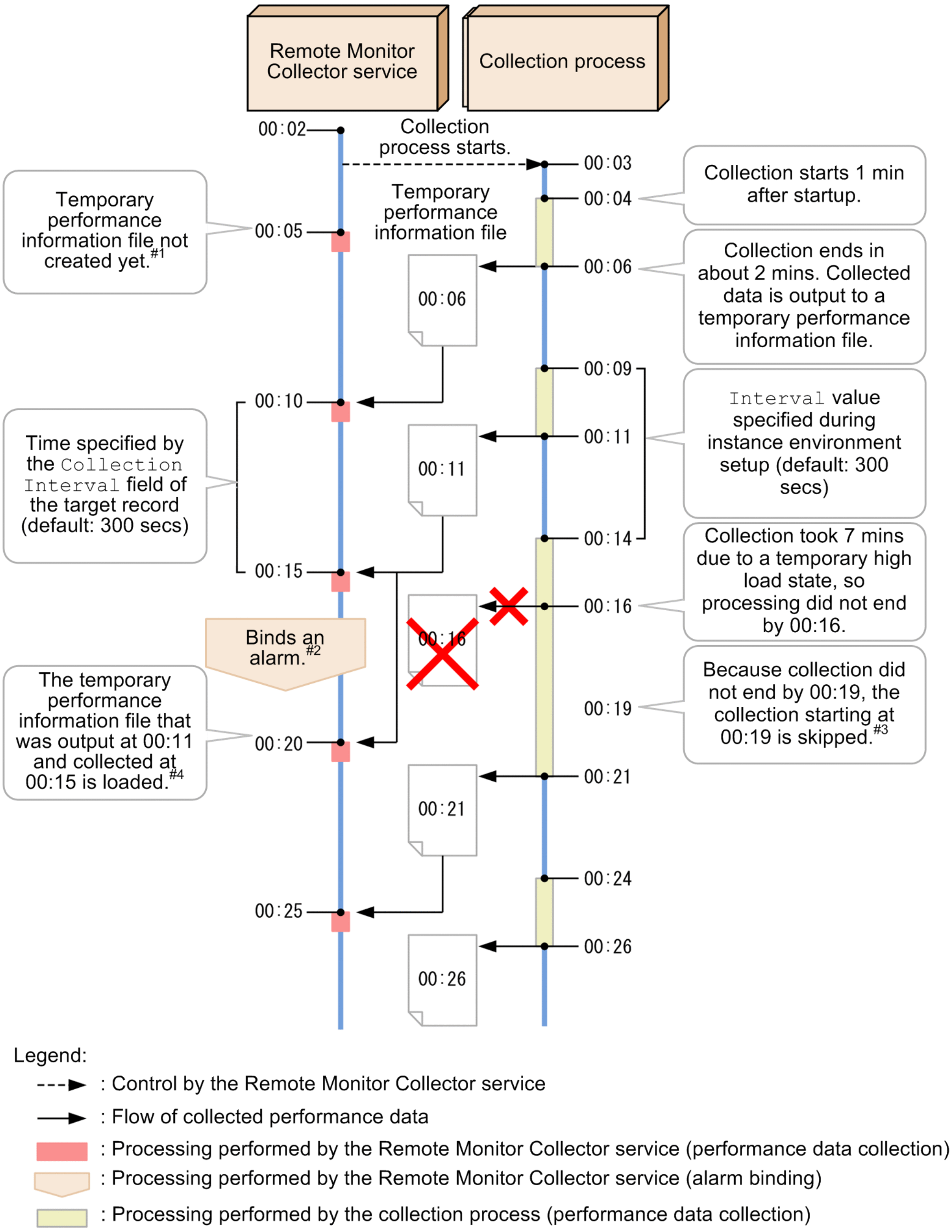
The Remote Monitor Collector service and the collection process operate asynchronously.

(2) General procedure for collecting performance data

The Remote Monitor Collector service of PFM - RM for Virtual Machine collects records as specified by the `Collection Interval` setting of each record. The collection process operates based on the `Interval` setting specified during setup of the instance environment.

The following figure shows an example of the procedure for collecting performance data.

Figure 1–10: Example of the procedure for collecting performance data



#1

Although performance data collection is to start at 00:05, the processing does not start and a KAVL20517-W message is output, because a temporary performance information file has not been created.

#2

If an alarm is bound in the period between collections at 00:15 and 00:20, whether to evaluate the alarm at 00:20 (first collection after the binding) depends on whether log data is collected by the record used by that alarm.

- If log data is collected by the record that is used by the bound alarm
  The alarm is not evaluated.
- If log data is not collected by the record that is used by the bound alarm
  The alarm is evaluated.

#3

Because the collection that started at 00:14 did not end by 00:19, the collection that was to start at 00:19 is skipped.

#4

During the collection at 00:20, in the same way as for the collection at 00:15, the temporary performance information file that was output at 00:15 is loaded.

If log data has been collected or the alarm is evaluated, no performance data is collected. At this time, a KAVL20516-W message is output.

For collection of data for a real-time report, the contents of the temporary performance information file that was output at 00:11 are displayed in the same way as for collection at 00:15.

Important

Depending on the time when the PFM - RM for Virtual Machine service starts, the message KAVL20516-W can be output repeatedly even though a problem (for example, the system is temporarily under high load or takes a long time to communicate with the monitoring target) has not occurred. In that case, you need to change the settings. For details, see *Appendix N. What to Do When the Message KAVL20516-W Is Output Repeatedly*.

Note

- When performance data is stored in the Store database as log data, the data that was collected during the interval before the storage time is stored. The maximum time that is required from the time performance data collection starts until the data is stored in the Store database is approximately the `Interval` time specified during instance environment setup (the default is 300 seconds).
- When a real-time report is displayed, the contents of the temporary performance information file existing at that time are displayed. When a displayed real-time report is refreshed from PFM - Web Console, the display is updated to show the current contents of the temporary performance information file.
- Collection processing might be delayed depending on the number of monitored hosts and the load status during processing.
- Alarm evaluation is performed with performance data that was collected during the interval before the time at which evaluation is performed. The maximum time that is required from the time performance data collection starts until alarm evaluation ends is approximately the `Interval` time specified during instance environment setup (the default is 300 seconds).

  Note that performance data collected during an earlier interval might be used for evaluation, depending on the alarm-binding time or collection delay, if the bound alarm does not use a record for which no log data is collected.

1.3.2 Differences in performance data collection from PFM - Agent for Virtual Machine

Before a PFM - RM for Virtual Machine instance can collect performance data, you must specify the `Interval` setting (collection process execution interval) during instance environment setup. Note that this setting does not exist in PFM - Agent for Virtual Machine.

PFM - Agent for Virtual Machine collects performance data at the interval specified by the `Collection Interval` property. PFM - RM for Virtual Machine executes the collection process at the interval specified by the `Interval` property (the default is 300 seconds) to create a temporary performance information file. PFM - RM for Virtual Machine then collects performance data from that file at the interval specified by the `Collection Interval` property (the default is 300 seconds). For this reason, the `Collection Interval` value must be larger than the `Interval` value.

If the value of the `Collection Interval` property is smaller than the value of the `Interval` property, performance data will be collected at the interval specified by the `Interval` property.

1.4 Example of performance monitoring operation using PFM - RM for Virtual Machine (for VMware)

To ensure the stability of operations in a system, it is important to monitor the system's performance and assess its status. This section explains how to use PFM - RM for Virtual Machine to monitor the performance of a VMware system.

1.4.1 System resources that are important for performance monitoring in a VMware system

The following are the system resources that are important when you use PFM - RM for Virtual Machine to monitor the performance of a VMware system.

- CPU resource (see *1.4.3*)
  This is the CPU resource installed on the physical server. It is important for performance monitoring of CPU usage and other items.
- Memory resource (see *1.4.4*)
  This is the memory resource installed on the physical server. It is important for performance monitoring of memory usage, swap usage, and other items.
- Disk resource (see *1.4.5*)
  This is the disk resource installed on the physical server. It is important for performance monitoring of disk usage, disk I/O status, and other items.
- Network resource (see *1.4.6*)
  This is the NIC resource installed on the physical server. It is important for performance monitoring of data send/receive speed and other items.

To set up definitions to monitor these important items, PFM - RM for Virtual Machine provides monitoring templates. Therefore, this section mainly explains monitoring methods that use monitoring templates. It also gives examples of compound report definition, which simplifies monitoring.

1.4.2 Selecting a baseline

Before you use a PFM product to monitor system performance, you must select a baseline.

To do this, you select a performance value based on the performance measurement results that is expected to not cause any problems in system operations.

A PFM product uses the baseline value as a threshold when it monitors the system performance. Therefore, selecting the baseline is an important task in performance monitoring.

When you select a baseline, consider the following points:

- We recommend that you measure the system's peak state during a situation such as a heavy workload test of the operating environment.
- The threshold varies greatly depending on the system configuration. Therefore, after you have modified the system resources or the operating environment, we recommend that you select a new baseline.

1.4.3 Monitoring the CPU resource

This subsection explains how to monitor the CPU resource of a VMware system.

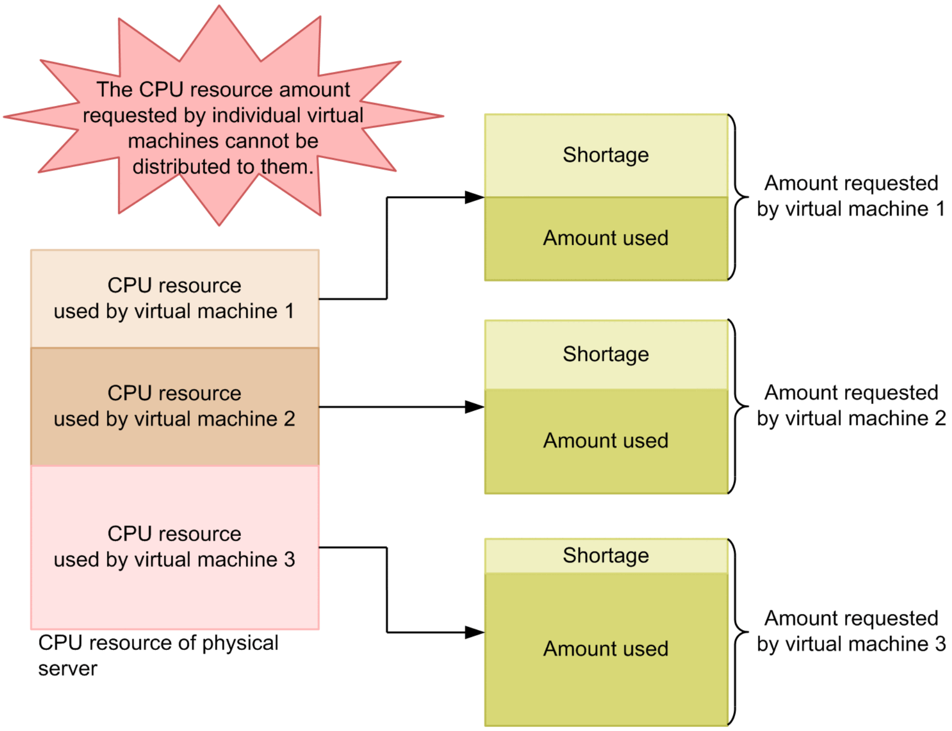
(1) Overview

In a VMware system, multiple virtual machines share the physical server's CPU. The CPU resource allocated to each virtual machine is called a *virtual CPU*. The OS running on a virtual machine recognizes a virtual CPU as a normal physical CPU.

The CPU resource of the physical server is distributed according to each virtual machine's CPU resource requirement. However, if the combined total of the CPU resource requirements for the individual virtual machines exceeds the CPU resource on the physical server, the required amount of CPU resource cannot be distributed, resulting in a virtual CPU resource shortage. This adversely affects the performance of the virtual machine.

The schematic diagram below shows a state in which a virtual CPU resource shortage has occurred.

Figure 1–11: Schematic diagram of a virtual CPU resource shortage



By monitoring CPU performance data, you can detect such performance deterioration in the virtual machines, and thus you can take appropriate corrective action.

Furthermore, in a virtual environment all physical devices are virtualized, such as memory, disks, and network interfaces. The CPU performs this virtualization of physical devices. Therefore, the CPU resource is an important resource that also affects the performance of other virtual devices.

The following four records are used to monitor the CPU resource. For details about records, see *5. Records*.

1. `PI` record

   This record is used to monitor the performance data of the physical server's CPU.

2. `PI_HCI` record

   This record is used to monitor the performance data of each core of the physical CPU.

3. PI_VI record

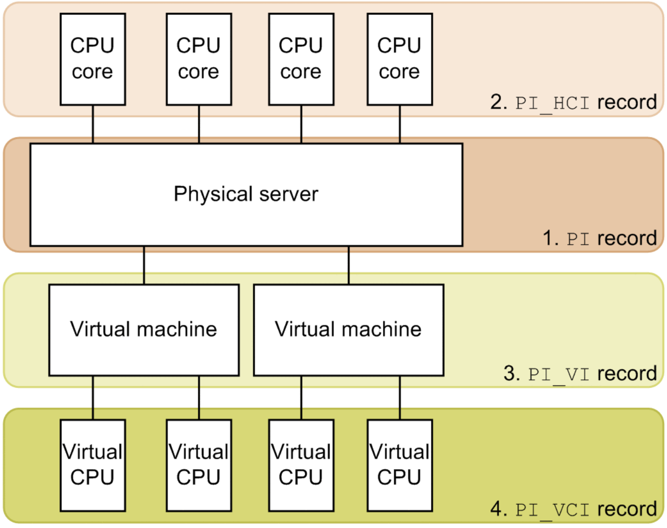
This record is used to monitor the performance data of the CPU that is being used by each virtual machine.

4. PI_VCI record

This record is used to monitor the performance data of each virtual CPU.

The following figure shows the range of performance data collected in each record.

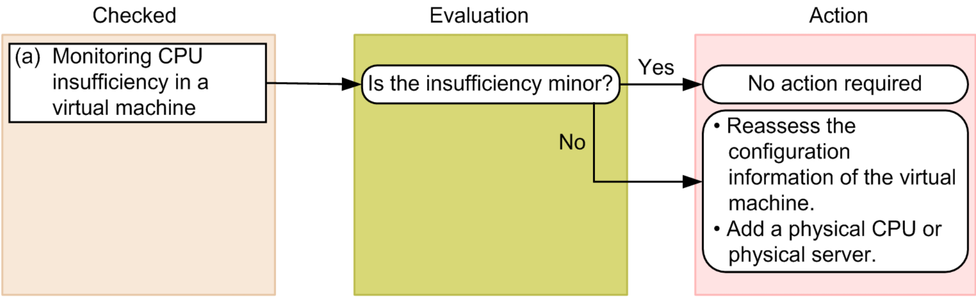
Figure 1–12: Range of performance data collected in each record



(2) Monitoring examples

Using CPU resource monitoring on virtual machines vhost1 and vhost2 as an example, this subsection explains the factors that cause insufficient CPU resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–13: Monitored items and flow of actions

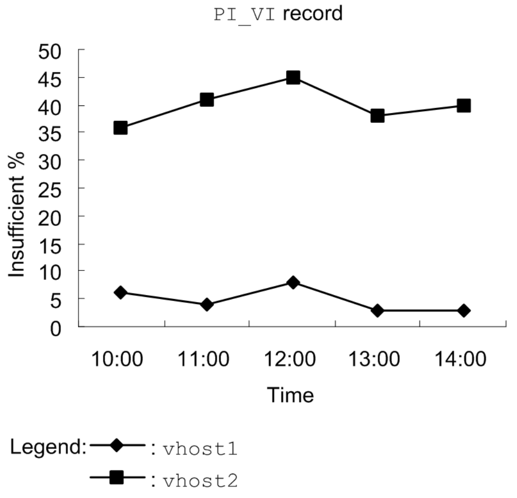


(a) Example of monitoring CPU insufficiency in a virtual machine

You can monitor the virtual machine's CPU insufficiency in the Insufficient % field of the PI_VI record. If a sufficient amount of CPU resources have been allocated to the virtual machine, the CPU insufficiency approaches 0%. Note that you can monitor this item with an alarm provided in a monitoring template.

The figure below shows an example of monitoring CPU insufficiency in a virtual machine.

Figure 1–14: CPU insufficiency monitoring example



Monitoring template report to be checked

VM CPU Insufficient

Monitoring template alarm to be checked

VM CPU Insufficient

In this example, there appears to be severe insufficiency in the CPU resource of vhost2.

In this case, reassess the configuration information of the virtual machine. If the CPU insufficiency is still high even after the configuration information has been reassessed, consider either adding a CPU to the physical server or adding a physical server.

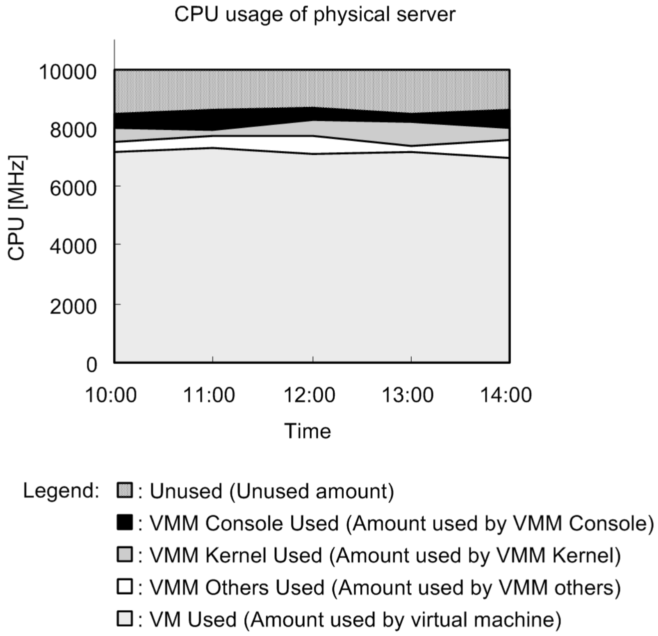
(3) Other monitoring examples

Monitoring examples follow that use monitoring templates other than those explained in *1.4.3(2) Monitoring examples*.

(a) Checking the physical server's CPU usage status

■ Report that displays a physical server's CPU usage status

Figure 1–15: Example of monitoring a physical server's CPU usage status



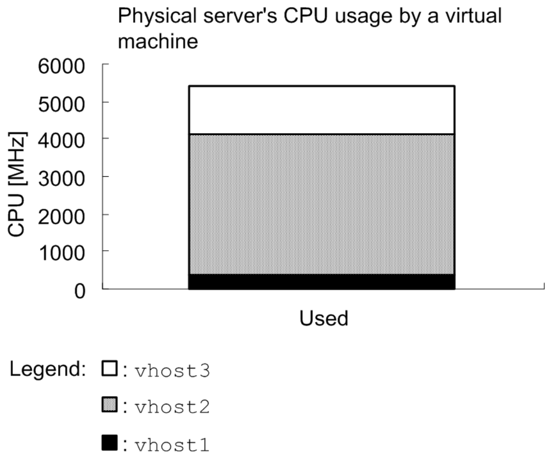
Monitoring template report to be checked

Host CPU Used Status

In this report, a drill-down feature is provided on the reports stored in the `Troubleshooting/Recent Past` folder. Clicking a virtual machine's usage plane on the graph displays the next report.

■ Report that displays a virtual machine's usage of a physical server's CPU

Figure 1–16: Example of monitoring a virtual machine's usage of a physical server's CPU



Monitoring template report to be checked

VM CPU Used Status

1.4.4 Monitoring the memory resource

This subsection explains how to monitor the memory resource of a VMware system.

(1) Overview

In a VMware system, multiple virtual machines share the memory of the physical server. The memory resource allocated to each virtual machine is called *virtual memory*. The OS running on a virtual machine recognizes virtual memory as normal physical memory.

The memory resource of the physical server is distributed according to each virtual machine's memory resource requirement. However, if the combined total of the memory resource requirements of the individual virtual machines exceeds the memory resources on the physical server, the required amount of memory resources cannot be distributed, resulting in a virtual memory resource shortage. This adversely affects the performance of the virtual machine. By monitoring the memory performance data, you can detect such performance deterioration of the virtual machine.

Furthermore, in the virtual environment, *swapping* is used. Swapping means using some disk space as memory. Disk space that can be swapped is called a *swap*. Through swapping, you can use a memory resource amount that is larger than the amount of memory installed in the physical server.

Since disk access speed is slower than physical memory, using a swap adversely affects the performance of the virtual machine. Therefore, we recommend that when you monitor memory resources, you also check the swapping status.

The following two records are used to monitor memory resources. For details about records, see *5. Records*.

1. `PI_VMI` record

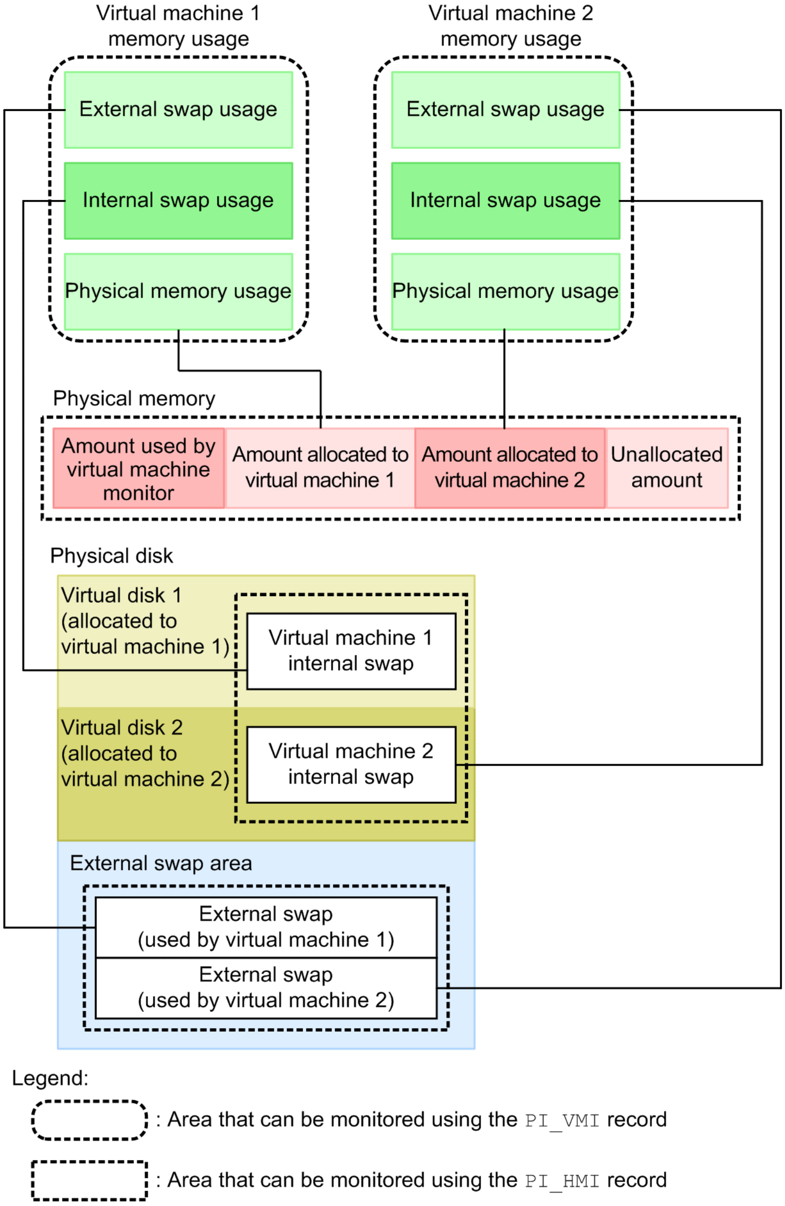
   This record is used to monitor each virtual machine's memory usage and the number of external and internal swaps used.

2. `PI_HMI` record

   This record is used to monitor the usage of physical memory by the virtual machine monitor and by each virtual machine, and the status of internal and external swaps used by the entire physical server.

The following figure shows the range of performance data collected in each record.

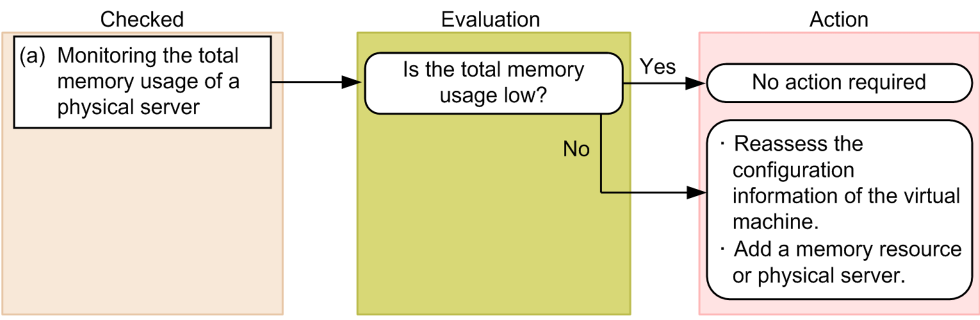
Figure 1–17: Correspondence between records and data collection ranges



(2) Monitoring examples

Using monitoring of a physical server on which the virtual environment is running as an example, this subsection explains the factors that cause insufficient memory resources and how to solve these problems. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–18: Monitored items and flow of actions

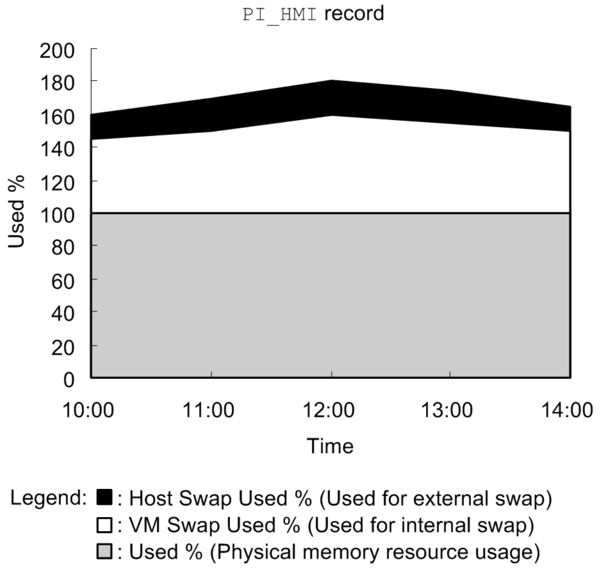


(a) Example of monitoring the total memory usage of a physical server

You can monitor the total memory usage of a physical server in the Total Used % field of the `PI_HMI` record. The total memory usage indicates the usage of all memory resources (physical memory resources, internally swapped resources, and externally swapped resources) provided on the physical server. If this value is large, the memory resources of the physical server are considered insufficient.

The figure below shows an example of monitoring the total memory usage of a physical server.

Figure 1–19: Example of monitoring the total memory usage



Monitoring template report to be checked

Host Memory Used

Monitoring template alarm to be checked

Host Memory Usage

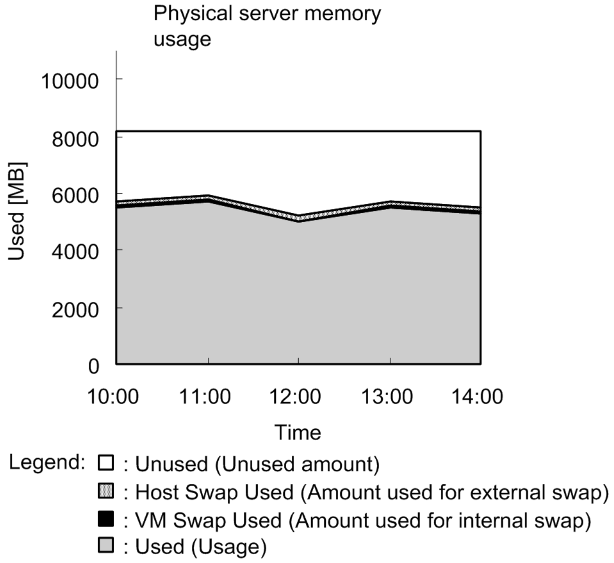
In this example, the value of Total Used %, which indicates the combined total of Used %, VM Swap Used %, and Host Swap Used %, exceeds 100%. Therefore, the memory resources of the physical server are considered insufficient.

In this case, reassess the configuration information of the virtual machine. If the total memory usage value does not improve even after the configuration information has been reassessed, consider either adding memory resources to the physical server or adding a physical server.

(3) Other monitoring examples

(a) Report for evaluating the memory usage of a physical server

Figure 1–20: Example of monitoring the memory usage of a physical server

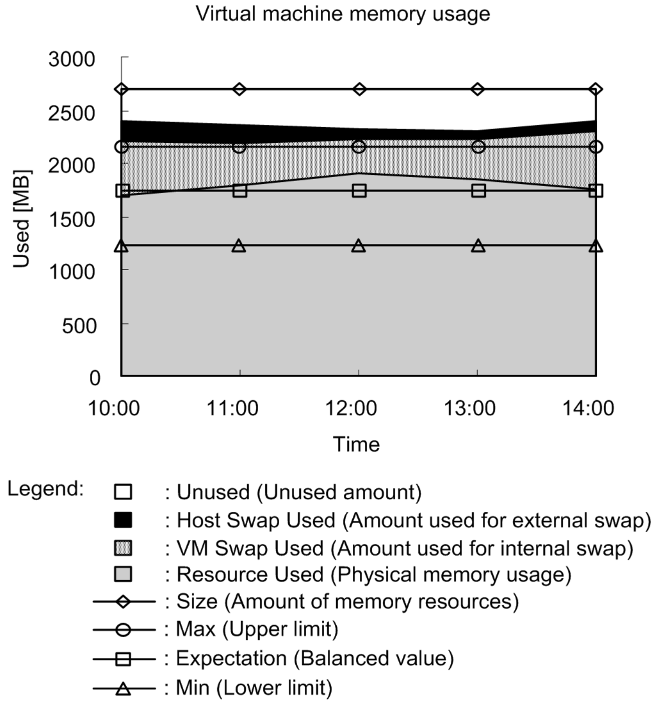


Monitoring template report to be checked

Host Memory Used Status

(b) Report for evaluating the memory usage of a virtual machine

Figure 1–21: Example of monitoring the memory usage of a virtual machine



Compound report to be checked (see *1.10*)

Monitoring of the upper limit of memory allocation in a virtual machine

1.4.5 Monitoring disk resources

This subsection explains how to monitor the disk resources of a VMware system.

(1) Overview

With a VMware system, you can create a datastore on a physical disk and share it with multiple virtual machines. A virtual disk created on the datastore is allocated to each virtual machine. The OS running on each virtual machine recognizes the allocated virtual disk as a normal physical disk and uses it as logically partitioned logical disks.

The following two types of disk resources are available:

- Disk I/O resource
  This is a resource related to disk access statuses.
- Disk space resource
  This is a resource related to disk space.

Since the physical disk is shared in a virtual environment, it is accessed by the individual virtual machines simultaneously. If multiple disk commands are issued simultaneously, the disk I/O resource is distributed according to share ratios that have been set up by the virtual environment software.

Disk commands from a virtual machine for which a large share ratio is set are executed at higher priority. Consequently, disk commands from a virtual machine with a small share ratio may be discarded without being executed. Discarded disk commands are re-executed after a time.

If the rate at which commands are discarded becomes high, disk access is delayed, adversely affecting the performance of the virtual machine. By monitoring the performance data of the disk I/O resource, you can detect such performance deterioration of the virtual machine, and you can thus take the necessary corrective action.

Furthermore, you can check whether there is a disk space insufficiency by monitoring the performance data of the disk space resource.

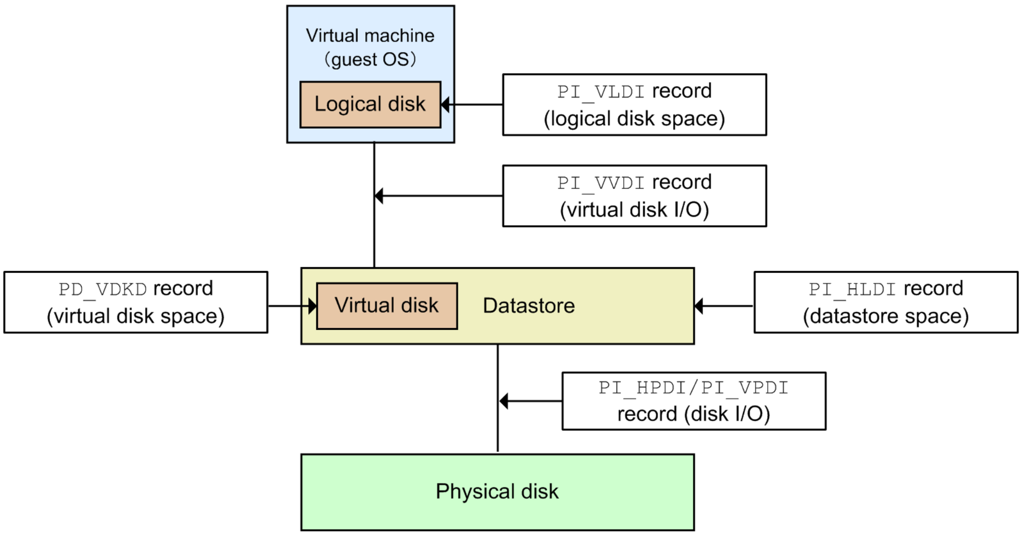
The six records described below are used to monitor the disk resource. For details about records, see *5. Records*.

1. `PI_HPDI` record
   This record is used to monitor the performance data of the physical disk. It shows the disk I/O resource as viewed from the physical server.
2. `PI_VPDI` record
   This record is used to monitor the performance data of the physical disk being used by the virtual machine. It shows the disk I/O resource as viewed from the virtual machine.
3. `PI_HLDI` record
   This record is used to monitor the performance data of the physical server's datastore. It shows the datastore space resource as viewed from the physical server.
4. `PI_VLDI` record
   This record is used to monitor the performance data of the guest OS's logical disk. It shows the disk space resource as viewed from the guest OS.
5. `PI_VVDI` record
   This record is used to monitor performance data of the virtual disk used by the virtual machine. It shows the virtual disk I/O resource as viewed from the virtual machine.
6. `PD_VDKD` record

This record is used to monitor performance data of the virtual disk used by the virtual machine. It shows the allocated virtual disk space resource as viewed from the physical server.

The figure below shows the range of performance data collected in each record.

Figure 1–22: Correspondence between records and data collection ranges



Important

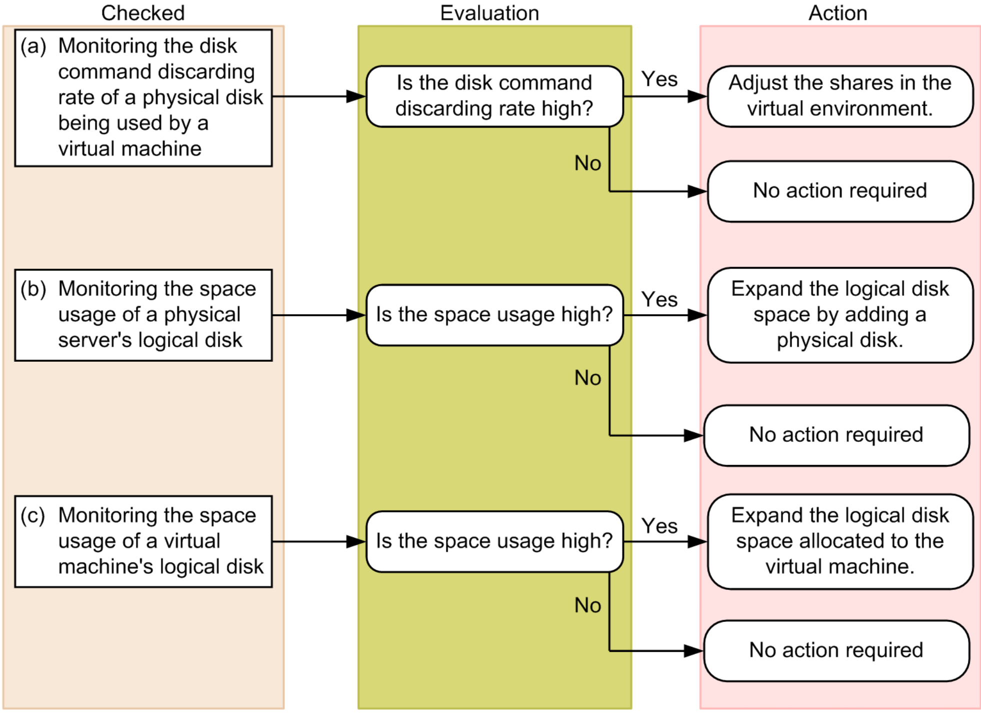
The *free space* of the VMware datastore is not updated regularly by default. Use the `Last Update` field of the `PI_HLDI` record to make sure that the *last update time* of the VMware datastore is periodically updated.

For details, see *2.5.1(5) Confirmation of Host Logical Disk Status (PI_HLDI) record*.

(2) Monitoring examples

Using monitoring of disk resources for `vhost1` and `vhost2` running on a physical server called `host1` as an example, this subsection explains the problems that might occur in the disk resources and how to solve them. The following figure shows the items monitored here, and the flow of actions to take.

Figure 1–23: Monitored items and flow of actions

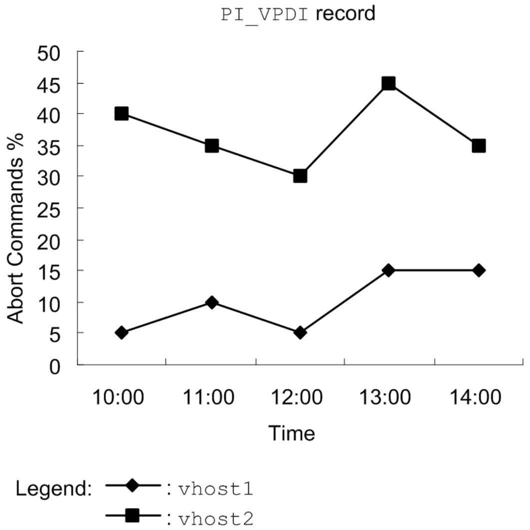


(a) Example of monitoring the disk command discarding rate of a physical disk being used by a virtual machine

You can monitor the disk command discarding rate of a physical disk being used by a virtual machine in the Abort Commands % field of the `PI_VPDI` record. Note that you can monitor this item with an alarm provided in a monitoring template.

The following figure shows an example of monitoring the disk command discarding rate of a physical disk.

Figure 1–24: Example of monitoring the disk command discarding rate



Monitoring template report to be checked

VM Disk Abort Commands

Monitoring template alarm to be checked

VM Disk Abort Cmds

If there is a virtual machine with a high disk command discarding rate, adjust its share of the physical disk. In this example, the disk command discarding rate for the physical disk being used by `vhost2` is high, and increasing `vhost2`'s share can solve this problem. For details about how to adjust the share, see the documentation for the virtual environment software.

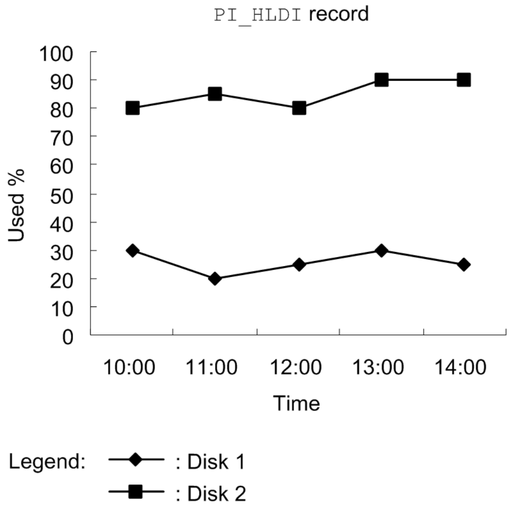
(b) Example of monitoring the space usage of a physical server's logical disk

You can check whether a physical server's logical disk has sufficient free space based on the space usage. Note that you can monitor this item with an alarm provided in a monitoring template.

You can check the space usage in the Used % field of the `PI_HLDI` record.

The following figure shows an example of monitoring the space usage of a physical server's logical disk.

Figure 1–25: Example of monitoring the space usage of a physical server's logical disk



Monitoring template report to be checked

Host Disk Used

Monitoring template alarm to be checked

Host Disk Usage

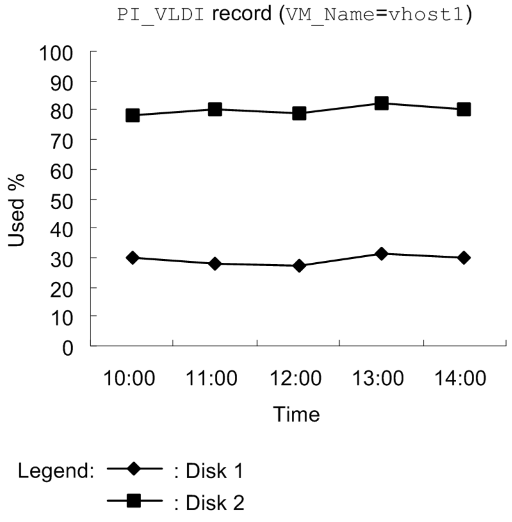
In this example, the space usage of Disk 2 is high. If the space usage is high, you can solve this problem by expanding the logical disk space through the addition of a physical disk. For details about how to expand the disk space, see the documentation for the virtual environment software.

(c) Example of monitoring the space usage of a virtual machine's logical disk

You can check whether a virtual machine's logical disk has sufficient free space based on the space usage. You can check the space usage in the Used % field of the `PI_VLDI` record. Note that you can monitor this item with an alarm provided in a monitoring template.

The following figure shows an example of monitoring the space usage of a virtual machine's logical disk.

Figure 1–26: Example of monitoring the space usage of a virtual machine's logical disk



Monitoring template report to be checked

VM Disk Used

Monitoring template alarm to be checked

VM Disk Usage

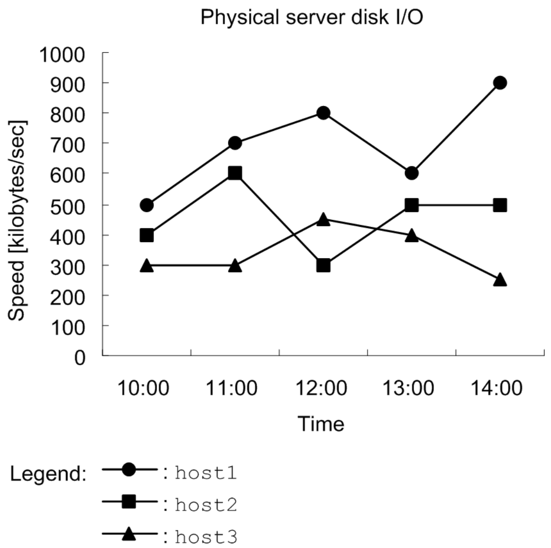
In this example, the space usage of Disk 2 being utilized by `vhost1` is high. If the space usage is high, you can solve this problem by expanding the logical disk space allocated to the virtual machine. For details about how to expand the disk space, see the documentation for the virtual environment software.

(3) Other monitoring examples

Monitoring examples follow that use monitoring templates other than those explained in *(2) Monitoring examples*.

(a) Report that displays the disk I/O state of a physical server

Figure 1–27: Example of monitoring a physical server's disk I/O state

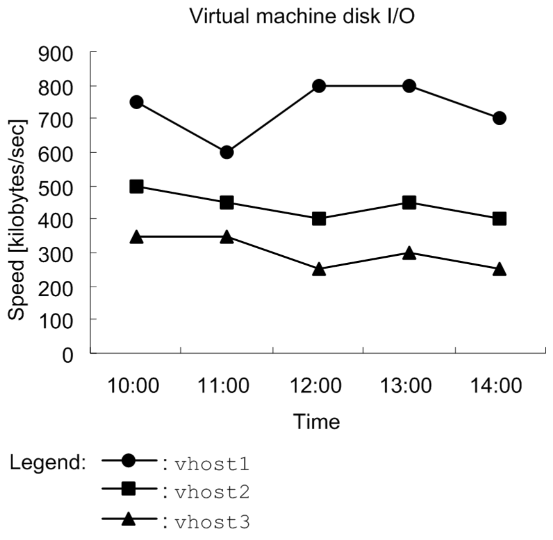


Monitoring template report to be checked

Host Disk I/O

(b) Report that displays the disk I/O state of a virtual machine

Figure 1–28: Example of monitoring a virtual machine's disk I/O state

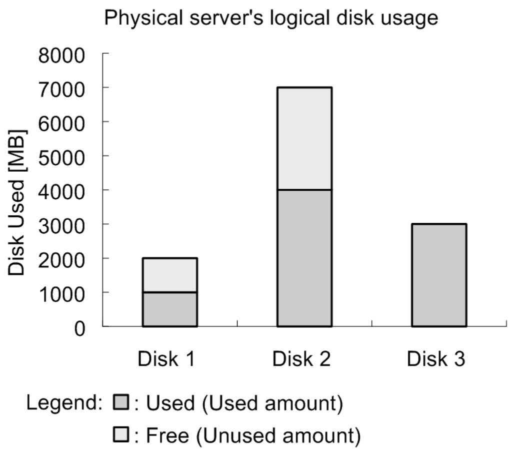


Monitoring template report to be checked

VM Disk I/O

(c) Report that displays the usage status of a physical server's logical disk

Figure 1–29: Example of monitoring the usage status of a physical server's logical disk

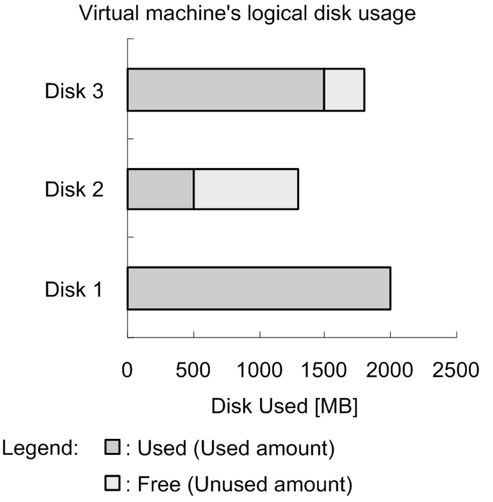


Monitoring template report to be checked

Host Disk Used Status

(d) Report that displays the usage status of a virtual machine's logical disk

Figure 1–30: Example of monitoring the usage status of a virtual machine's logical disk

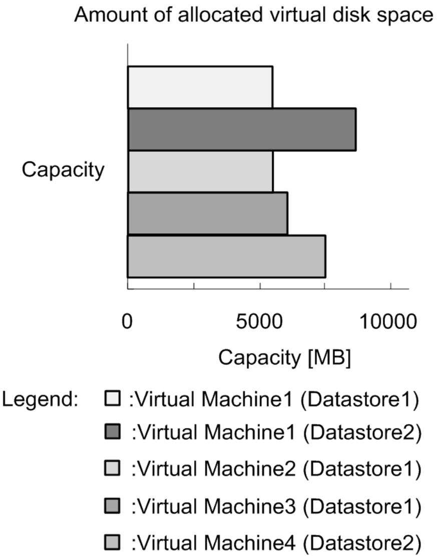


Monitoring template report to be checked

VM Disk Used Status

(e) Report that displays the capacity of the virtual disk allocated to a virtual machine

Figure 1–31: Example of monitoring the capacity of the virtual disk allocated to a virtual machine



Monitoring template report to be checked

VM Virtual Disk Allocation Value

By outputting this report to a CSV file and summing up the latest values, you can identify the capacity of the virtual disk allocated to a datastore.

When a datastore is shared by multiple physical servers, information regarding each one of these physical servers is acquired and data belonging to the same `Datastore ID` or `Datastore Name` field is summed up. This allows you to identify the capacity of the virtual disk allocated across the entire system.

Furthermore, if ESXi 6.5 or later is monitored, you can identify the virtual disk shared by multiple virtual machines by using the `Disk UUID` field.

1.4.6 Monitoring the network resources

This subsection explains how to monitor the network resources of a VMware system.

(1) Overview

In a VMware system, multiple virtual machines share the NIC of the physical server. The NIC allocated to each virtual machine is called the *virtual NIC*. The OS running on a virtual machine recognizes a virtual NIC as a normal NIC.

In a virtual environment, virtual machines use the physical NIC at the same time, and as a result, the network bandwidth that can be used by each virtual machine becomes narrow. Consequently, each virtual machine's network data send/receive speed may decrease.

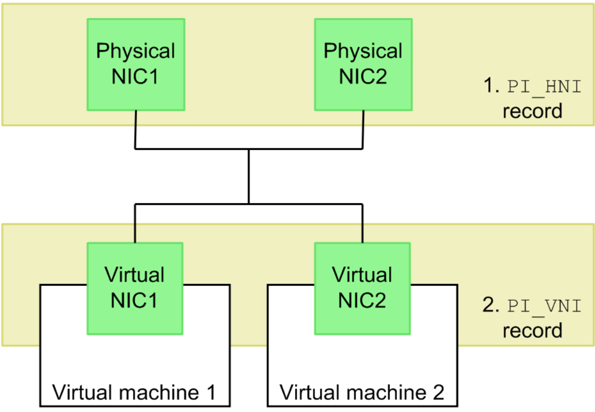
By monitoring the network's performance data, you can detect such performance deterioration of the virtual machine, and thus you can take the necessary corrective action.

The two records described below are used for monitoring the network resources. For details about the records, see *5. Records*.

1. `PI_HNI` record
   This record is used to monitor the performance data of the physical NIC.
2. `PI_VNI` record
   This record is used to monitor the performance data of the virtual NIC.

The figure below shows the range of performance data collected in each record.

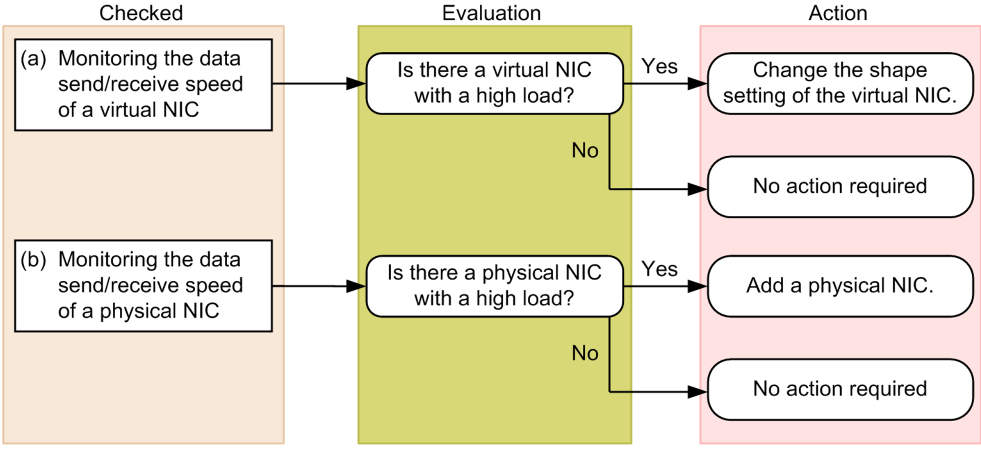
Figure 1–32: Correspondence between records and data collection ranges



(2) Monitoring examples

Using resource monitoring of virtual NIC1 and NIC2 and physical NIC1 and NIC2 as an example, this subsection explains problems that may occur with network resources, and how to solve them. The figure below shows the items monitored here and the flow of actions to take.

Figure 1–33: Monitored items and flow of actions

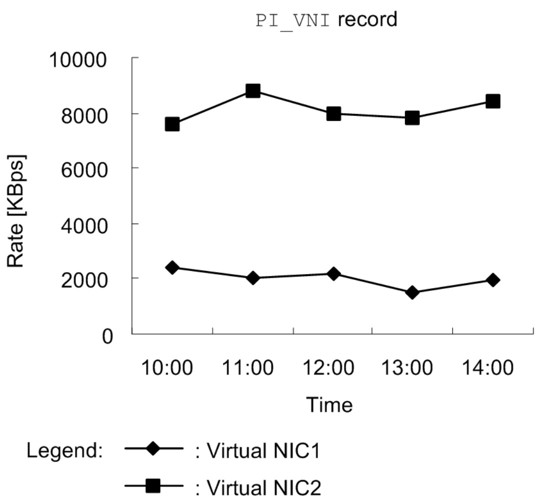


(a) Example of monitoring the data send/receive speed of a virtual NIC

You can monitor the load applied to a virtual NIC based on the data send/receive speed of the virtual NIC. You can check the data send/receive speed of the virtual NIC in the Rate field of the `PI_VNI` record.

The figure below shows an example of monitoring the data send/receive speed of a virtual NIC.

Figure 1–34: Example of monitoring the data send/receive speed of a virtual NIC



Monitoring template report to be checked

VM Network Data

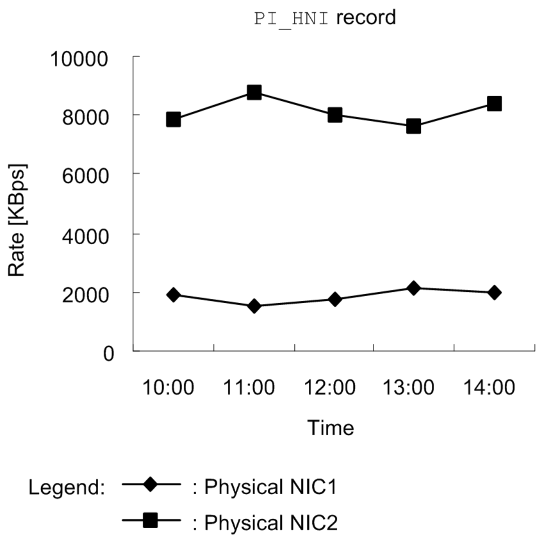
In this example, the data send/receive speed of virtual NIC2 is high, indicating a high load level. If there is a virtual NIC with a high load, you can solve this problem by changing the shape settings of a lower-priority virtual NIC. For details about how to change the shape settings, see the documentation for the virtual environment software.

(b) Example of monitoring the data send/receive speed of a physical NIC

You can monitor the load applied to a physical NIC based on the data send/receive speed of the physical NIC. You can check the data send/receive speed of the physical NIC in the Rate field of the `PI_HNI` record.

The figure below shows an example of monitoring the data send/receive speed of a physical NIC.

Figure 1–35: Example of monitoring the data send/receive speed of a physical NIC



Monitoring template report to be checked

Host Network Data

In this example, the data send/receive speed of physical NIC2 is high, indicating a high load level. If there is a physical NIC with a high load, consider adding a physical NIC.

1.5 Example of performance monitoring operation using PFM - RM for Virtual Machine (for Hyper-V)

To ensure the stability of operations in a system, it is important to monitor the system's performance and assess its status. This section explains how to use PFM - RM for Virtual Machine to monitor the performance of a Hyper-V system.

1.5.1 System resources that are important for performance monitoring in a Hyper-V system

The following are the system resources that are important when you use PFM - RM for Virtual Machine to monitor the performance of a Hyper-V system.

- CPU resource (see *1.5.3*)
  This is the CPU resource installed on the physical server. It is important for performance monitoring of CPU usage and other items.
- Memory resource (see *1.5.4*)
  This is the memory resource installed on the physical server. It is important for performance monitoring of memory usage and other items.
- Disk resource (see *1.5.5*)
  This is the disk resource installed on the physical server. It is important for performance monitoring of disk usage, disk I/O status, and other items.
- Network resource (see *1.5.6*)
  This is the NIC resource installed on the physical server. It is important for performance monitoring of data send/receive speed and other items.

To set up definitions to monitor these important items, PFM - RM for Virtual Machine provides monitoring templates. Therefore, this section mainly explains monitoring methods that use monitoring templates.

1.5.2 Selecting a baseline

See *1.4.2 Selecting a baseline*.

1.5.3 Monitoring the CPU resource

This subsection explains how to monitor the CPU resource of a Hyper-V system.

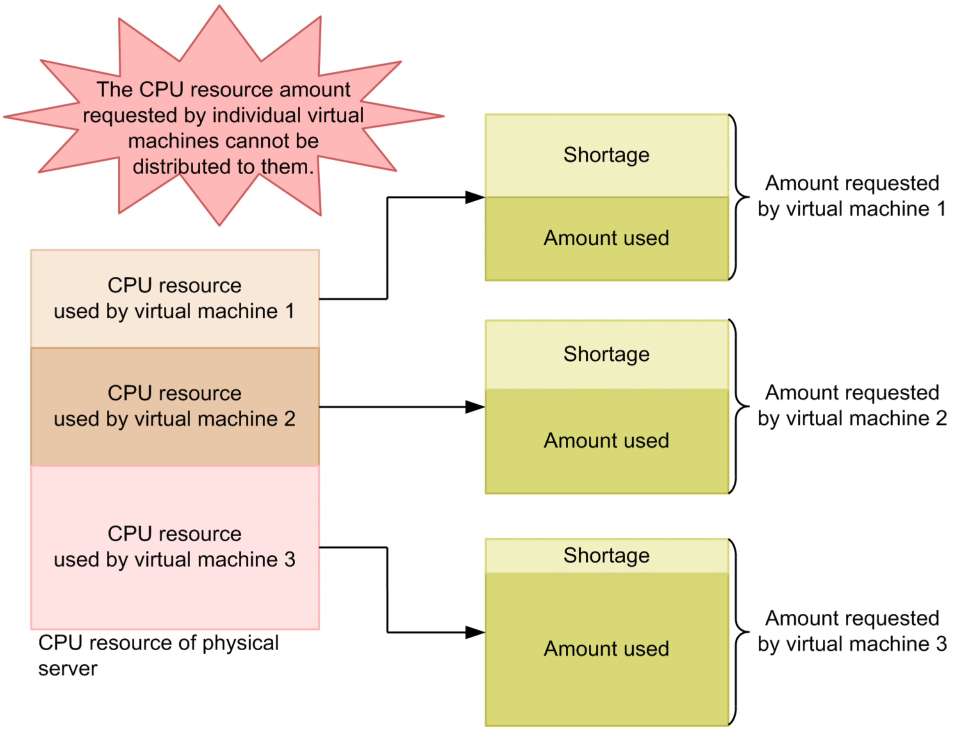
(1) Overview

In a Hyper-V system, multiple virtual machines share the physical server's CPU. The CPU resource allocated to each virtual machine is called a *virtual CPU*. The OS running on a virtual machine recognizes a virtual CPU as a normal physical CPU.

The CPU resource of the physical server is distributed according to each virtual machine's CPU resource requirement. However, if the combined total of the CPU resource requirements for the individual virtual machines exceeds the CPU resource on the physical server, the required amount of CPU resource cannot be distributed, resulting in a virtual CPU resource shortage. This adversely affects the performance of the virtual machine.

The schematic diagram below shows a state in which a virtual CPU resource shortage has occurred.

Figure 1–36: Schematic diagram of a virtual CPU resource shortage



By monitoring CPU performance data, you can detect such performance deterioration in the virtual machines, and thus you can take appropriate corrective action.

Furthermore, in a virtual environment all physical devices are virtualized, such as memory, disks, and network interfaces. The CPU performs this virtualization of physical devices. Therefore, the CPU resource is an important resource that also affects the performance of other virtual devices.

The following four records are used to monitor the CPU resource. For details about records, see *5. Records*.

1. `PI` record

   This record is used to monitor the performance data of the physical server's CPU.

2. `PI_HCI` record

   This record is used to monitor the performance data of each core of the physical CPU.

3. `PI_VI` record

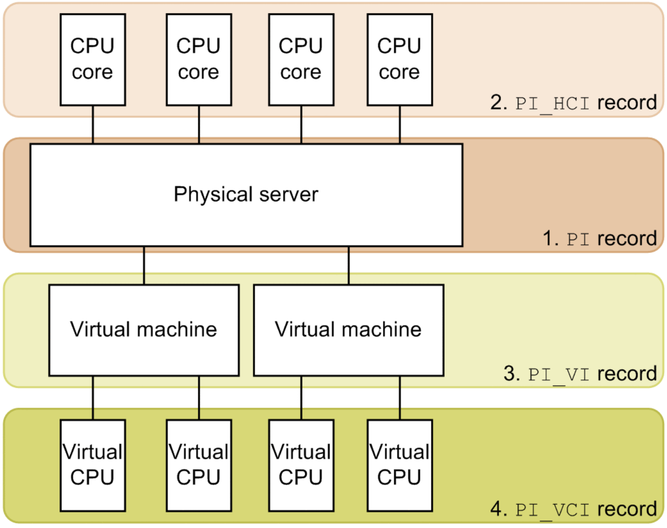
   This record is used to monitor the performance data of the CPU that is being used by each virtual machine.

4. `PI_VCI` record

   This record is used to monitor the performance data of each virtual CPU.

The following figure shows the range of performance data collected in each record.

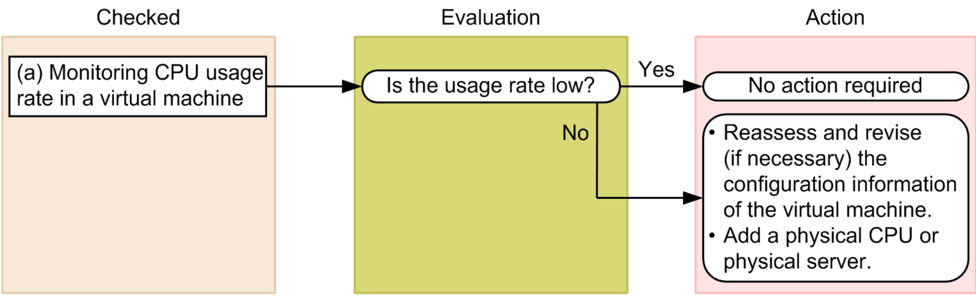
Figure 1–37: Range of performance data collected in each record



(2) Monitoring examples

Using CPU resource monitoring on virtual machines `vhost1` through `vhost3` as an example, this subsection explains the factors that cause insufficient CPU resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–38: Monitored items and flow of actions

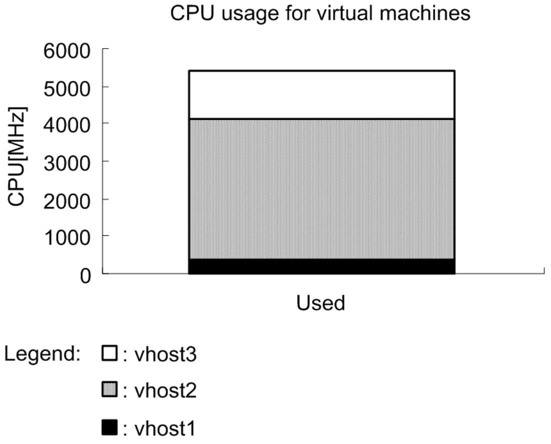


(a) Example of monitoring the CPU usage rate for virtual machines

You can check the CPU usage rate for virtual machines in the Used % field of the `PI_VI` record.

An example of monitoring is shown below. This example is based on using the monitoring template to display a `VM CPU Used Status` report. The chart in this report displays the Used field that indicates the CPU usage for the virtual machines. To monitor the Used % field, check the table that is displayed under the chart.

Figure 1–39: Example of monitoring the CPU usage for virtual machines



Monitoring template report to be checked

VM CPU Used Status

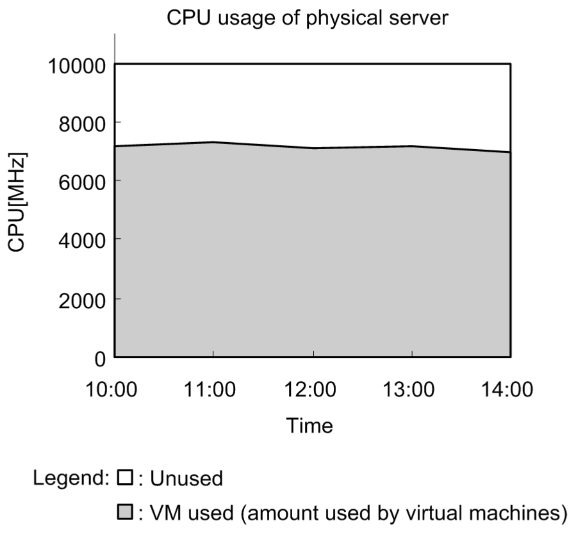
In this example, the CPU usage rate of `vhost2` has increased. If the CPU usage rate is high and the virtual machine's performance is poor, you should revise configuration information for the virtual machines, such as by resetting the maximum CPU allocation size. If the problem persists, consider either adding a physical CPU or adding a physical server.

(3) Other monitoring examples

Monitoring examples follow that use monitoring templates other than those explained in *(2) Monitoring examples*.

(a) Monitoring the physical server's CPU usage status

Figure 1–40: Example of monitoring a physical server's CPU usage status



Monitoring template report to be checked

Host CPU Used Status

In this report, a drill-down feature is provided on the reports stored in the `Troubleshooting/Recent Past` folder. Clicking a virtual machine's usage plane on the graph displays the VM CPU Used Status report shown in Figure 1-35.

1.5.4 Monitoring the memory resource

This subsection explains how to monitor the memory resource of a Hyper-V system.

(1) Overview

In a Hyper-V system, the physical server's memory resource is allocated to multiple virtual machines. The OS running on a virtual machine recognizes its allocated memory resource as a normal physical memory.

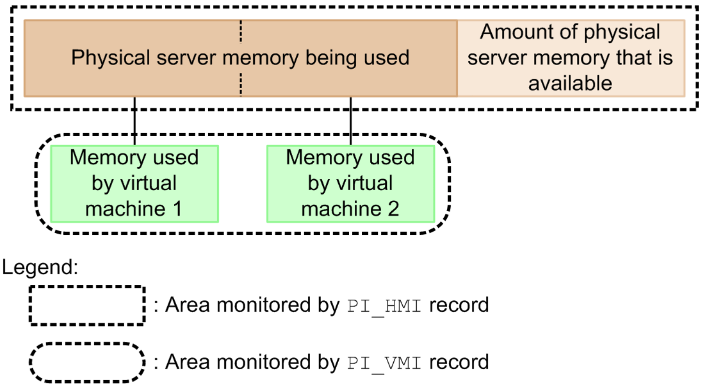
If the combined total of the memory resources required by all the virtual machines exceeds the memory resources on the physical server, performance of the virtual machines will be affected adversely. Monitoring memory performance enables you to detect such performance deteriorations in the virtual machines.

The following two records are used to monitor memory resources. For details about records, see *5. Records*.

1. `PI_HMI` record
   This record is used to monitor the usage of physical memory by the virtual machine monitor and by each virtual machine, and the status of internal and external swaps used by the entire physical server.
2. `PI_VMI` record
   This record is used to monitor the amount of memory resources allocated to each virtual machine.
   The amount of the physical server's memory resources available to each virtual machine is determined when each virtual machine is created. Therefore, in a Hyper-V system, the memory required for operations cannot exceed the size of the real memory that is installed on the physical server. Normally, there is no need to monitor the `PI_VMI` record because the memory allocation does not change during system operation.

The following figure shows the range of performance data collected in the `PI_HMI` and `PI_VMI` records.

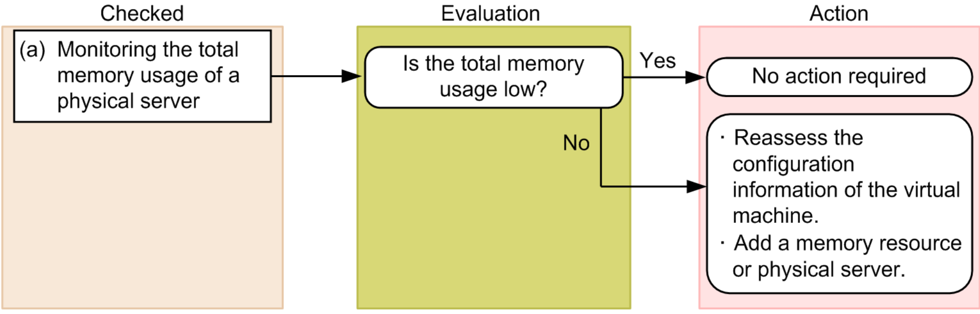
Figure 1–41: Correspondence between records and data collection ranges



(2) Monitoring examples

This subsection uses an example of monitoring a physical server on which a virtual environment is running to explain the factors that cause insufficient memory resources and how to solve such problems. The following figure shows the items monitored here and the flow of actions to take.

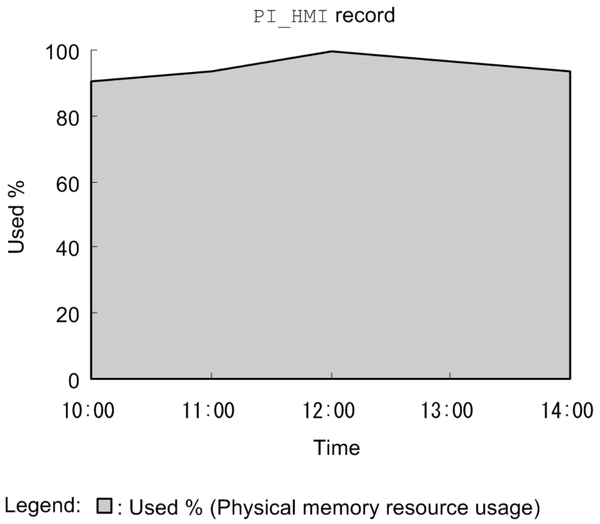
Figure 1–42: Monitored items and flow of actions



(a) Example of monitoring the total memory usage of a physical server

You can monitor the total memory usage of a physical server in the Total Used % field of the `PI_HMI` record. If this value is large, the memory resources of the physical server are considered insufficient. Note that you can monitor this item by using an alarm that is provided in the monitoring template. The figure below shows a monitoring example.

Figure 1–43: Example of monitoring the total memory usage



Monitoring template report to be checked

Host Memory Used

Monitoring template alarm to be checked

Host Memory Usage

In this example, the memory resources of the physical server are considered insufficient because the value remains large over the entire period. In such a case, revise the configuration information for the virtual machines. If the total memory usage value does not improve even after the configuration information has been revised, consider either adding memory resources to the physical server or adding a physical server.

1.5.5 Monitoring disk resources

This subsection explains how to monitor the disk resources of a Hyper-V system.

(1) Overview

In a Hyper-V system, one of the following methods is used to provide disk resources to a virtual machine:

- Allocation to a virtual machine of a virtual hard disk file created on the physical disk
- Allocation of a physical disk to a virtual machine

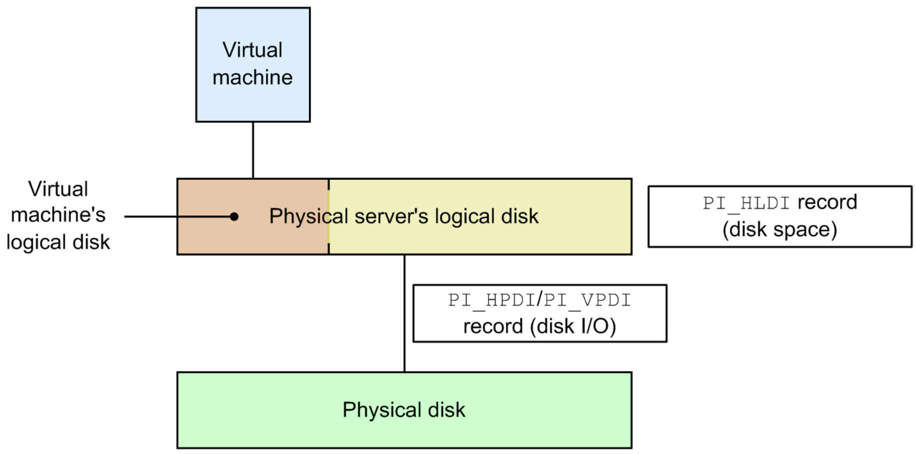
The OS running on a virtual machine recognizes the allocated disk resources as a normal physical disk.

Information on disk resources can be classified as the disk I/O resource (which includes information such as the disk transfer rate) and the disk space resource (which includes information such as the disk capacity). PFM - RM for Virtual Machine provides records for these two types of disk resources. When you monitor a Hyper-V system, you can use the records listed below to monitor the disk resources. Note that the `PI_VLDI` record that indicates a virtual machine's disk space resource is not supported.

1. `PI_HPDI` record
   This record is used to monitor the performance data of the physical disk. It shows the disk I/O resource as viewed from the physical server.
2. `PI_VPDI` record
   This record is used to monitor the performance data of the physical disk being used by a virtual machine. It shows the disk I/O resource as viewed from the virtual machine.
3. `PI_HLDI` record
   This record is used to monitor the performance data of the physical server's logical disk. It shows the disk space resource as viewed from the physical server.

The following figure shows the range of performance data collected in each record.

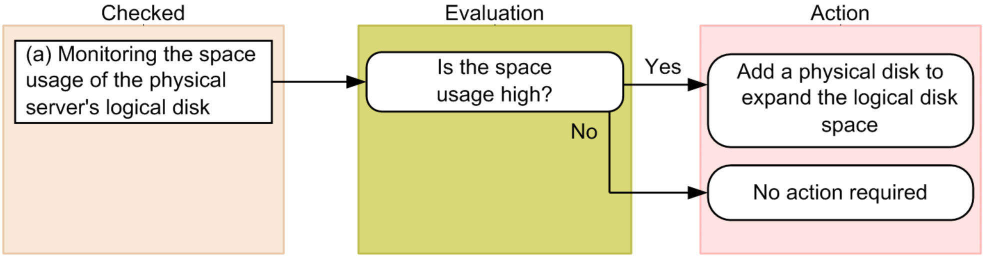
Figure 1–44: Correspondence between records and data collection ranges



(2) Monitoring examples

This subsection uses an example of monitoring `disk 1` and `disk 2` installed on the physical server to explain the problems that might occur in disk resources and how to solve them. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–45: Monitored items and flow of actions

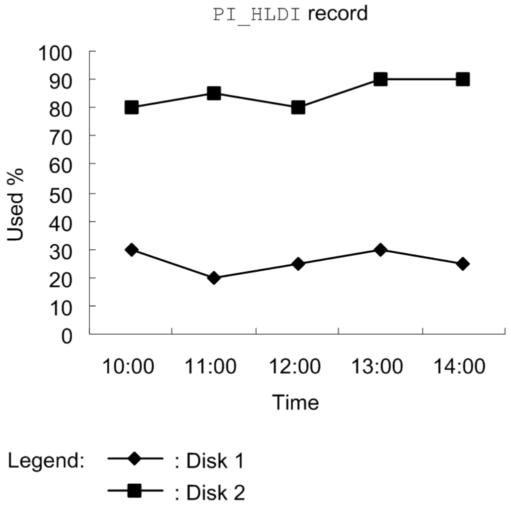


(a) Example of monitoring the space usage of a physical server's logical disk

You can check whether a physical server's logical disk has sufficient free space based on the space usage.

You can check the space usage in the Used % field of the `PI_HLDI` record. Note that you can monitor this field using an alarm that is provided in the monitoring template. The figure below shows a monitoring example.

Figure 1–46: Example of monitoring the space usage of a physical server's logical disk



Monitoring template report to be checked

Host Disk Used

Monitoring template alarm to be checked

Host Disk Usage

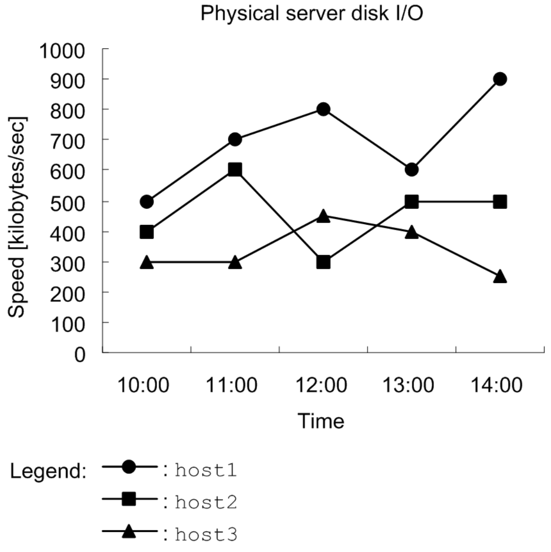
In this example, the space usage of Disk 2 is high. If the space usage is high, you can solve this problem by expanding the logical disk space through the addition of a physical disk or by reducing the logical disk space that is allocated to each virtual machine.

(3) Other monitoring examples

Monitoring examples follow that use monitoring templates other than those explained in *(2) Monitoring examples*.

(a) Report that displays the disk I/O state of a physical server

Figure 1–47: Example of monitoring a physical server's disk I/O state

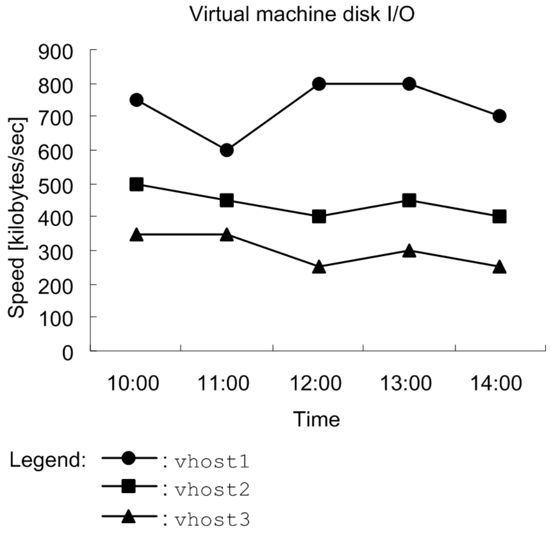


Monitoring template report to be checked

Host Disk I/O

(b) Report that displays the disk I/O state of a virtual machine

Figure 1–48: Example of monitoring a virtual machine's disk I/O state

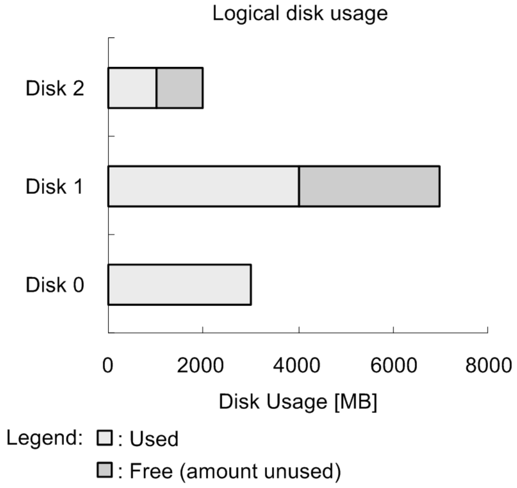


Monitoring template report to be checked

VM Disk I/O

(c) Report that displays the usage status of a virtual machine's logical disk

Figure 1–49: Example of monitoring the usage status of a virtual machine's logical disk



Monitoring template report to be checked

VM Disk Used Status

1.5.6 Monitoring the network resources

This subsection explains how to monitor the network resources of a Hyper-V system.

(1) Overview

In a Hyper-V system, multiple virtual machines share the NIC of the physical server. The NIC allocated to each virtual machine is called the *virtual NIC*.

The OS running on a virtual machine recognizes a virtual NIC as a normal NIC.

In a virtual environment, multiple virtual machines may use the physical NIC at the same time. As a result, the network bandwidth available to each virtual machine narrows, and each virtual machine's network data send/receive speed may decrease.

By monitoring the network's performance data, you can detect such performance deterioration in the virtual machines, enabling you to take appropriate corrective action.

The two records described below are used for monitoring the network resources. For details about the records, see *5. Records*.

1. `PI_HNI` record

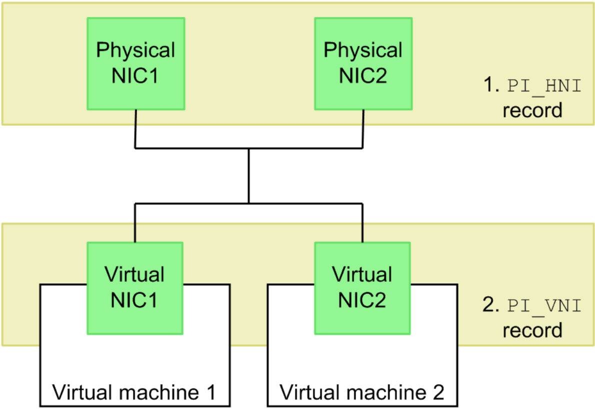
   This record is used to monitor the performance data of the physical NIC.

2. `PI_VNI` record

   This record is used to monitor the performance data of the virtual NIC.

The following figure shows the range of performance data collected in each record.

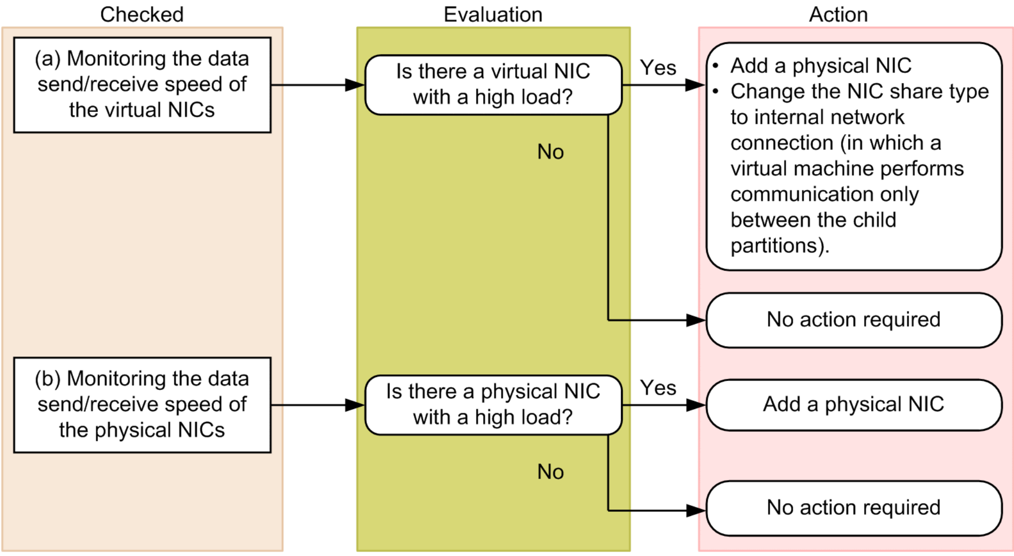
Figure 1–50: Correspondence between records and data collection ranges



(2) Monitoring examples

Using resource monitoring of virtual NIC1 and NIC2 and physical NIC1 and NIC2 as an example, this subsection explains problems that may occur with network resources, and how to solve them. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–51: Monitored items and flow of actions

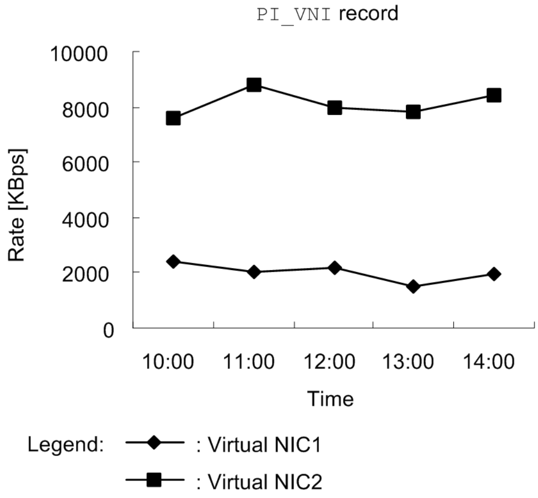


(a) Example of monitoring the data send/receive speed of a virtual NIC

You can monitor the load applied to a virtual NIC based on the data send/receive speed of the virtual NIC. You can check the data send/receive speed of the virtual NIC in the Rate field of the PI_VNI record.

The figure below shows an example of monitoring the data send/receive speed of a virtual NIC.

Figure 1–52: Example of monitoring the data send/receive speed of a virtual NIC



Monitoring template report to be checked

VM Network Data

In this example, the data send/receive speed of virtual NIC2 is high, indicating a high load level. When there is a virtual NIC with a high load, take the following action according to the virtual machine's communication status:

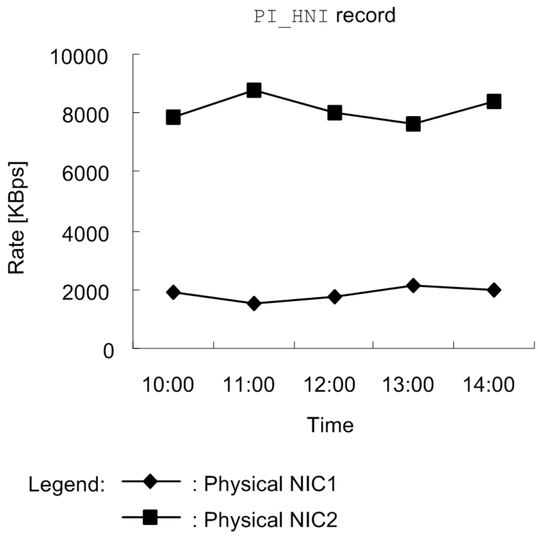
- If the virtual machine communicates with an external network
  Add a physical NIC to distribute the load of the virtual NIC.
- If the virtual machine communicates only with another virtual machine (child partition)
  Change the virtual NIC type to internal network connection.

(b) Example of monitoring the data send/receive speed of a physical NIC

You can monitor the load applied to a physical NIC based on the data send/receive speed of the physical NIC. You can check the data send/receive speed of the physical NIC in the Rate field of the PI_HNI record.

The figure below shows an example of monitoring the data send/receive speed of a physical NIC.

Figure 1–53: Example of monitoring the data send/receive speed of a physical NIC



Monitoring template report to be checked

Host Network Data

In this example, the data send/receive speed of physical NIC2 is high, indicating a high load level. If there is a physical NIC with a high load, consider adding a physical NIC.

1.6 Example of performance monitoring operation using PFM - RM for Virtual Machine (for KVM)

To ensure the stability of operations in a system, it is important to monitor the system's performance and assess its status. This section explains how to use PFM - RM for Virtual Machine to monitor the performance of a KVM system.

1.6.1 System resources that are important for performance monitoring in a KVM system

The following are the system resources that are important when you use PFM - RM for Virtual Machine to monitor the performance of a KVM system.

- CPU resource (see *1.6.3*)
  This is the CPU resource installed on the physical server. It is important for performance monitoring of CPU usage and other items.
- Memory resource (see *1.6.4*)
  This is the memory resource installed on the physical server. It is important for performance monitoring of memory usage and other items.
- Disk resource (see *1.6.5*)
  This is the disk resource installed on the physical server. It is important for performance monitoring of disk usage, disk I/O status, and other items.
- Network resource (see *1.6.6*)
  This is the NIC resource installed on the physical server. It is important for performance monitoring of data send/receive speed and other items.

To set up definitions to monitor these important items, PFM - RM for Virtual Machine provides monitoring templates. Therefore, this section mainly explains monitoring methods that use monitoring templates.

1.6.2 Selecting a baseline

See *1.4.2 Selecting a baseline*.

1.6.3 Monitoring the CPU resource

This subsection explains how to monitor the CPU resource of a KVM system.

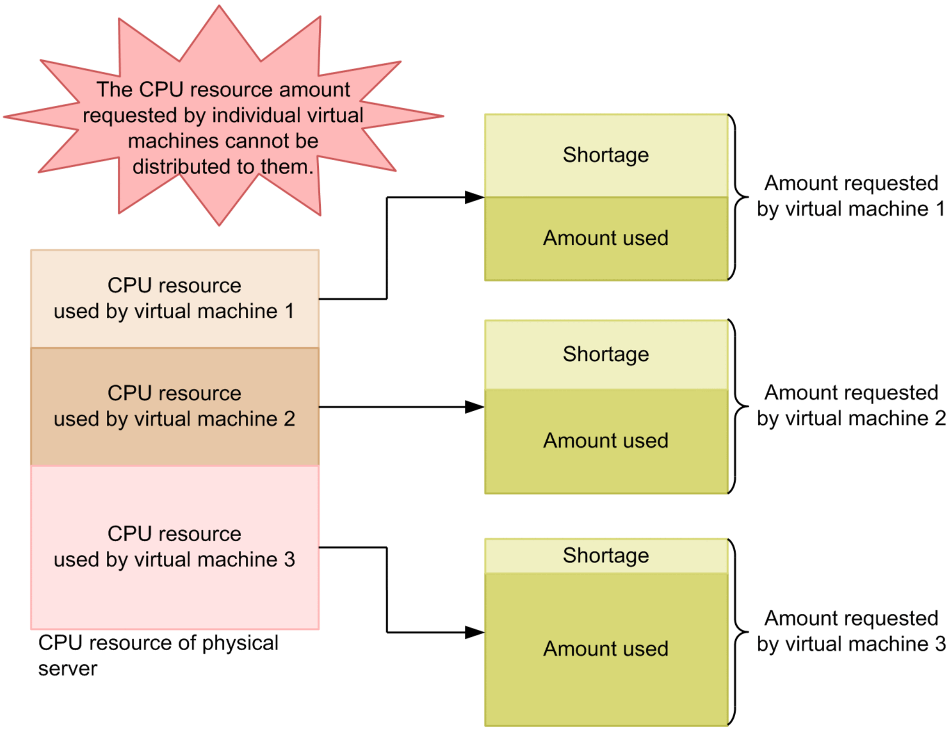
(1) Overview

In a KVM system, multiple virtual machines share the physical server's CPU. The CPU resource allocated to each virtual machine is called a *virtual CPU*. The OS running on a virtual machine recognizes a virtual CPU as a normal physical CPU.

The CPU resource of the physical server is distributed according to each virtual machine's CPU resource requirement. However, if the combined total of the CPU resource requirements for the individual virtual machines exceeds the CPU resource on the physical server, the required amount of CPU resource cannot be distributed, resulting in a virtual CPU resource shortage. This adversely affects the performance of the virtual machine.

The schematic diagram below shows a state in which a virtual CPU resource shortage has occurred.

Figure 1–54: Schematic diagram of a virtual CPU resource shortage



By monitoring CPU performance data, you can detect such performance deterioration in the virtual machines, and thus you can take appropriate corrective action.

Furthermore, in a virtual environment all physical devices are virtualized, such as memory, disks, and network interfaces. The CPU performs this virtualization of physical devices. Therefore, the CPU resource is an important resource that also affects the performance of other virtual devices.

The following four records are used to monitor the CPU resource. For details about records, see *5. Records*.

1. `PI` record

   This record is used to monitor the performance data of the physical server's CPU.

2. `PI_HCI` record

   This record is used to monitor the performance data of each core of the physical CPU.

3. `PI_VI` record

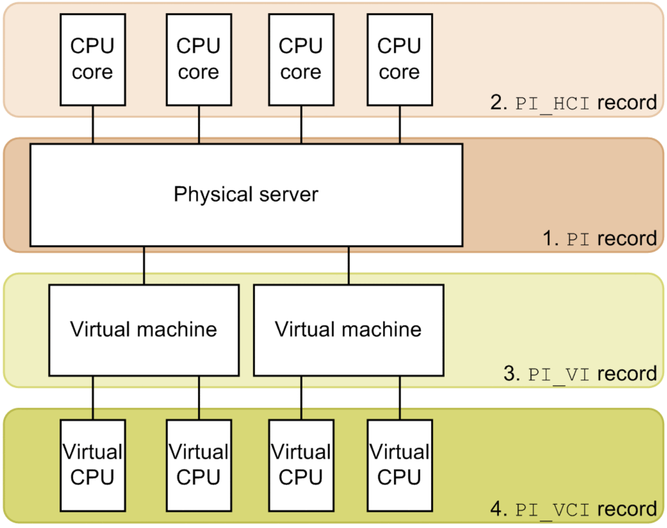
   This record is used to monitor the performance data of the CPU that is being used by each virtual machine.

4. `PI_VCI` record

   This record is used to monitor the performance data of each virtual CPU.

The following figure shows the range of performance data collected in each record.

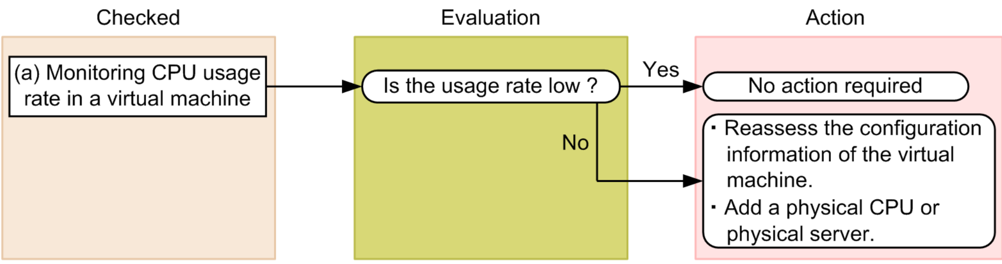
Figure 1–55: Range of performance data collected in each record



(2) Monitoring examples

Using CPU resource monitoring on virtual machines `vhost1` through `vhost3` as an example, this subsection explains the factors that cause insufficient CPU resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–56: Monitored items and flow of actions

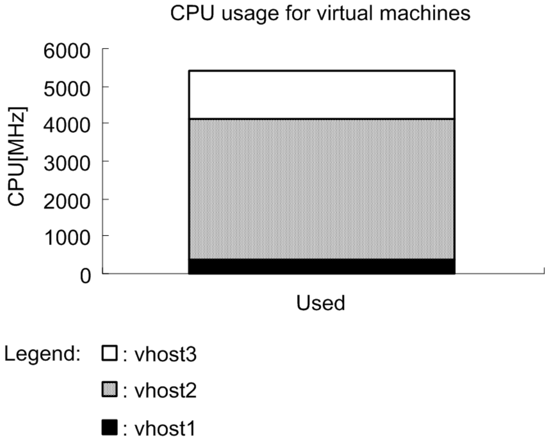


(a) Example of monitoring the CPU usage rate for virtual machines

You can check the CPU usage rate for virtual machines in the Used % field of the `PI_VI` record.

An example of monitoring is shown below. This example is based on using the monitoring template to display a `VM CPU Used Status` report. The chart in this report displays the Used field that indicates the CPU usage for the virtual machines. To monitor the Used % field, check the table that is displayed under the chart.

Figure 1–57: Example of monitoring the CPU usage for virtual machines



Monitoring template report to be checked

VM CPU Used Status

In this example, the CPU usage of `vhost2` is high. If the performance of a virtual machine has degraded due to high CPU usage, review the number of allocated CPUs and other configuration information of the virtual environment. If the problem still cannot be solved, consider whether you can add physical CPUs or physical servers.

1.6.4 Monitoring the memory resource

This subsection explains how to monitor the memory resource of a KVM system.

(1) Overview

In a KVM system, the physical server's memory resource is allocated to multiple virtual machines. The OS running on a virtual machine recognizes its allocated memory resource as a normal physical memory.

If the combined total of the memory resources required by all the virtual machines exceeds the memory resources on the physical server, performance of the virtual machines will be affected adversely. Monitoring memory performance enables you to detect such performance deteriorations in the virtual machines.

The following two records are used to monitor memory resources. For details about records, see *5. Records*.

1. `PI_HMI` record

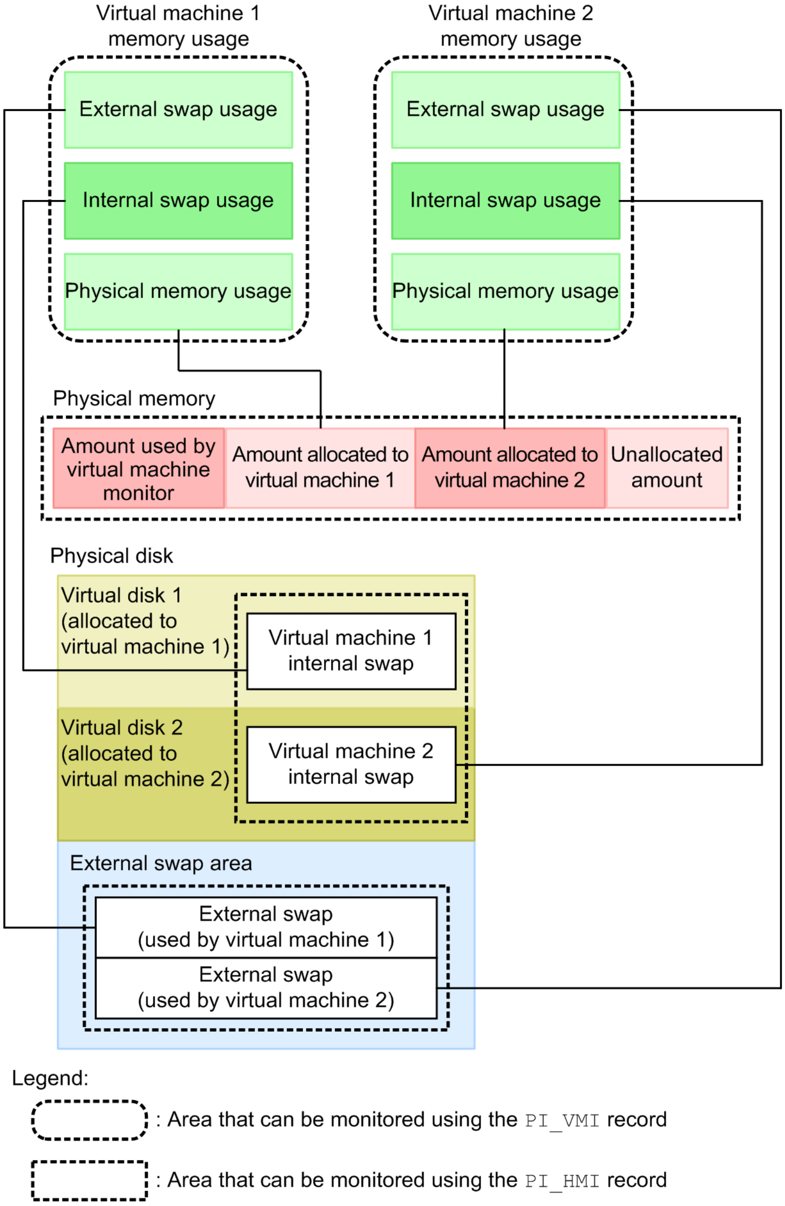
   This record is used to monitor the usage of physical memory by the virtual machine monitor and by each virtual machine.

2. `PI_VMI` record

   This record is used to monitor the amount of memory resources allocated to each virtual machine.

The following figure shows the range of performance data collected in the `PI_HMI` and `PI_VMI` records.

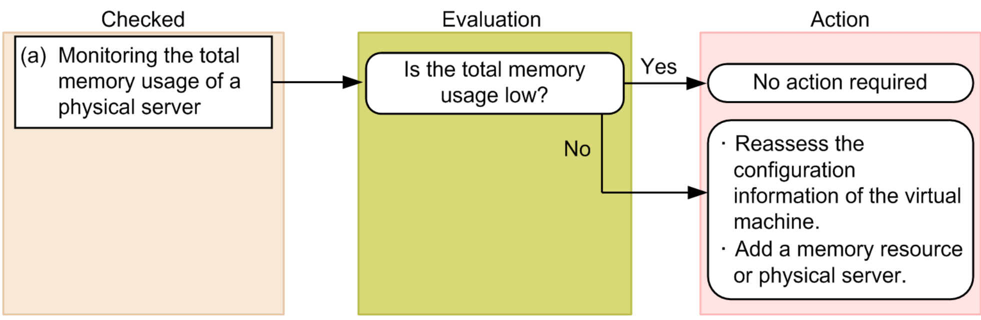
Figure 1–58: Correspondence between records and data collection ranges



(2) Monitoring examples

This subsection uses an example of monitoring a physical server on which a virtual environment is running to explain the factors that cause insufficient memory resources and how to solve such problems. The following figure shows the items monitored here and the flow of actions to take.

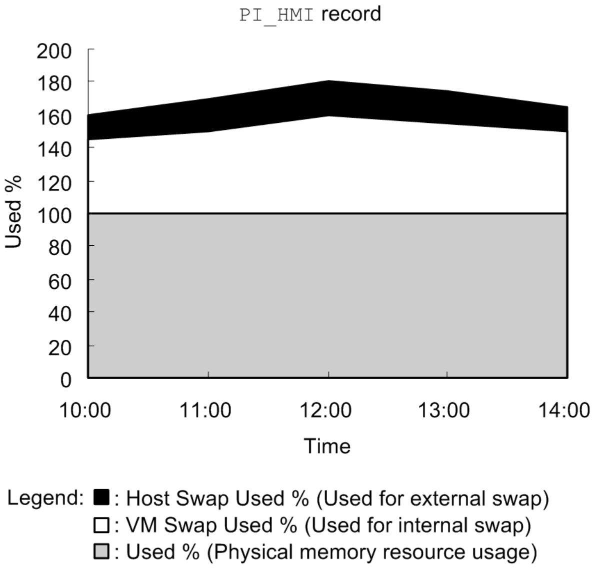
Figure 1–59: Monitored items and flow of actions



(a) Example of monitoring the total memory usage of a physical server

You can monitor the total memory usage of a physical server in the Total Used % field of the PI_HMI record. If this value is large, the memory resources of the physical server are considered insufficient. Note that you can monitor this item by using an alarm that is provided in the monitoring template. The figure below shows a monitoring example.

Figure 1–60: Example of monitoring the total memory usage



Monitoring template report to be checked

Host Memory Used

Monitoring template alarm to be checked

Host Memory Usage

In this example, the `Total Used %` value, which indicates the total of the `Used %`, `VM Swap Used %`, and `Host Swap Used %` values, exceeds 100%. Therefore, it is likely the memory resources for physical servers are insufficient.

In such a case, revise the configuration information for the virtual machines. If the total memory usage value does not improve even after the configuration information has been revised, consider either adding memory resources to the physical server or adding a physical server.

1.6.5 Monitoring disk resources

This subsection explains how to monitor the disk resources of a KVM system.

(1) Overview

In a KVM system, one of the following methods is used to provide disk resources to a virtual machine:

- Allocation to a virtual machine of a virtual hard disk file created on the physical disk
- Allocation of a physical disk to a virtual machine

The OS running on a virtual machine recognizes the allocated disk resources as a normal physical disk.

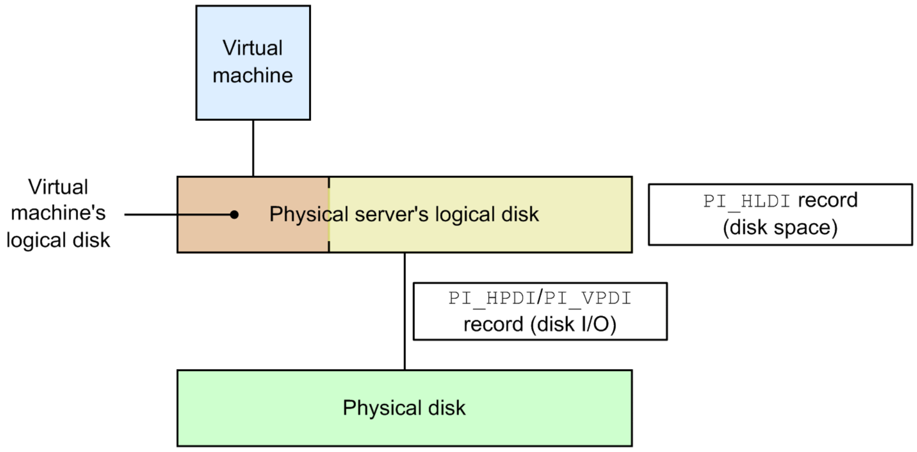
Information on disk resources can be classified as the disk I/O resource (which includes information such as the disk transfer rate) and the disk space resource (which includes information such as the disk capacity). PFM - RM for Virtual

Machine provides records for these two types of disk resources. When you monitor a KVM system, you can use the records listed below to monitor the disk resources. Note that the `PI_VLDI` record that indicates a virtual machine's disk space resource is not supported.

1. `PI_HPDI` record
   This record is used to monitor the performance data of the physical disk. It shows the disk I/O resource as viewed from the physical server.
2. `PI_VPDI` record
   This record is used to monitor the performance data of the physical disk being used by a virtual machine. It shows the disk I/O resource as viewed from the virtual machine.
3. `PI_HLDI` record
   This record is used to monitor the performance data of the physical server's logical disk. It shows the disk space resource as viewed from the physical server.

The following figure shows the range of performance data collected in each record.

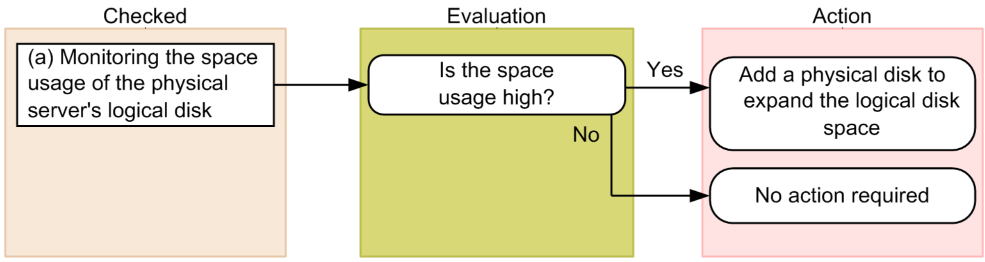
Figure 1–61: Correspondence between records and data collection ranges



(2) Monitoring examples

This subsection uses an example of monitoring `disk 1` and `disk 2` installed on the physical server to explain the problems that might occur in disk resources and how to solve them. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–62: Monitored items and flow of actions

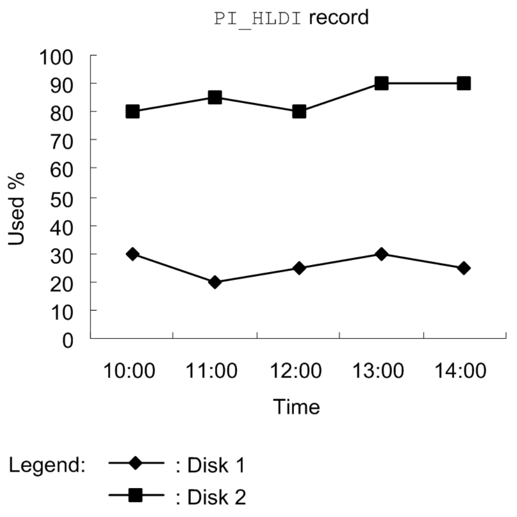


(a) Example of monitoring the space usage of a physical server's logical disk

You can check whether a physical server's logical disk has sufficient free space based on the space usage.

You can check the space usage in the Used % field of the PI_HLDI record. Note that you can monitor this field using an alarm that is provided in the monitoring template. The figure below shows a monitoring example.

Figure 1–63: Example of monitoring the space usage of a physical server's logical disk



Monitoring template report to be checked

Host Disk Used

Monitoring template alarm to be checked

Host Disk Usage

In this example, the space usage of Disk 2 is high. If the space usage is high, you can solve this problem by expanding the logical disk space through the addition of a physical disk.

1.6.6 Monitoring the network resources

This subsection explains how to monitor the network resources of a KVM system.

(1) Overview

In a KVM system, multiple virtual machines share the NIC of the physical server. The NIC allocated to each virtual machine is called the *virtual NIC*.

The OS running on a virtual machine recognizes a virtual NIC as a normal NIC.

In a virtual environment, virtual machines use the physical NIC at the same time, and as a result, the network bandwidth that can be used by each virtual machine becomes narrow. Consequently, each virtual machine's network data send/receive speed may decrease.

By monitoring the network's performance data, you can detect such performance deterioration of the virtual machine, and thus you can take the necessary corrective action.

The two records described below are used for monitoring the network resources. For details about the records, see *5. Records*.

1. PI_HNI record

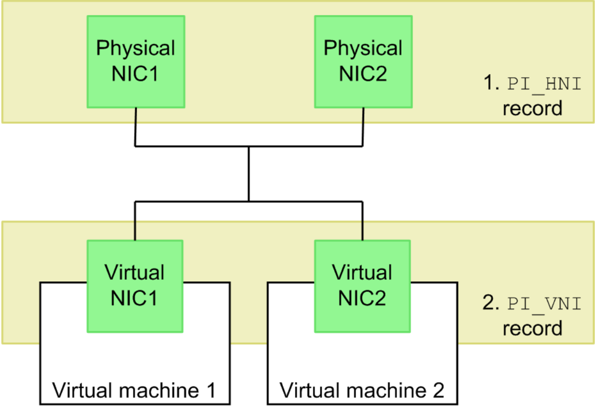
   This record is used to monitor the performance data of the physical NIC.

2. PI_VNI record

This record is used to monitor the performance data of the virtual NIC.

The figure below shows the range of performance data collected in each record.

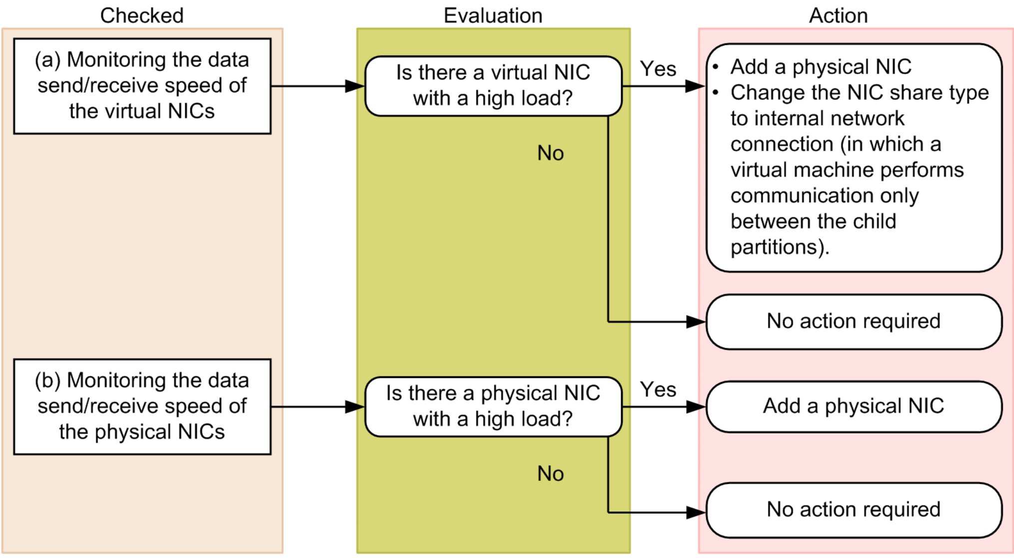
Figure 1–64: Correspondence between records and data collection ranges



(2) Monitoring examples

Using resource monitoring of virtual NIC1 and NIC2 and physical NIC1 and NIC2 as an example, this subsection explains problems that may occur with network resources, and how to solve them. The figure below shows the items monitored here and the flow of actions to take.

Figure 1–65: Monitored items and flow of actions

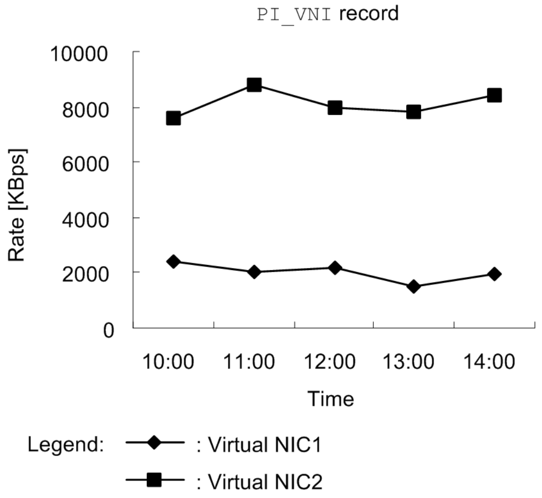


(a) Example of monitoring the data send/receive speed of a virtual NIC

The load placed on a virtual NIC can be evaluated from the data send/receive speed of the virtual NIC. You can check the data send/receive speed of the virtual NIC in the Rate field of the PI_VNI record.

The figure below shows an example of monitoring the data send/receive speed of a virtual NIC.

Figure 1–66: Example of monitoring the data send/receive speed of a virtual NIC



Monitoring template report to be checked

VM Network Data

In this example, the value that indicates the data send/receive speed for virtual NIC 2 is large. This means that virtual NIC 2 is heavily loaded. If there are virtual NICs that are heavily loaded, depending on the communication status of the virtual machine, take either of the following actions:

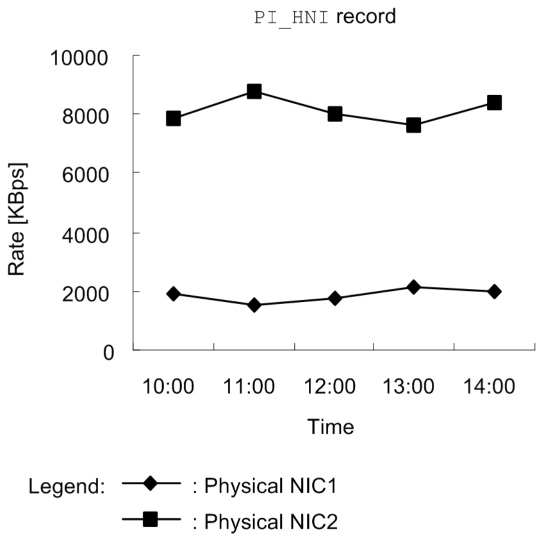
- If the virtual machine communicates with an external network
  Add physical NICs to distribute the load on virtual NICs.
- If the virtual machine communicates with only other virtual machines in the same virtual environment
  Creates a virtual switch for the internal network, and change the NIC that is connected to the other virtual machines.

(b) Example of monitoring the data send/receive speed of a physical NIC

The load on a physical NIC can be evaluated from the data send/receive speed of the physical NIC. You can check the data send/receive speed of the physical NIC in the Rate field of the PI_HNI record.

The figure below shows an example of monitoring the data send/receive speed of a physical NIC.

Figure 1–67: Example of monitoring the data send/receive speed of a physical NIC



Monitoring template report to be checked

Host Network Data

In this example, the data send/receive speed of physical NIC2 is high, indicating a high load level. If there is a physical NIC with a high load, consider adding a physical NIC.

1.7 Example of performance monitoring operation using PFM - RM for Virtual Machine (for Docker environment)

To ensure the stability of operations in a system, it is important to monitor the system's performance and assess its status. When you monitor a Docker environment, items to be monitored vary depending on PFM products.

This section explains how to use PFM - RM for Virtual Machine to monitor the performance of a Docker container.

1.7.1 System resources that are important for performance monitoring in a Docker container

The following are the system resources that are important when you use PFM - RM for Virtual Machine to monitor the performance of a Docker container.

- CPU resource (see *1.7.3*)
  This is the CPU resource used by the Docker container. It is important for performance monitoring of CPU usage and other items.
- Memory resource (see *1.7.4*)
  This is the memory resource used by the Docker container. It is important for performance monitoring of memory usage and other items.
- Disk resource (see *1.7.5*)
  This is the disk resource used by the Docker container. It is important for performance monitoring of disk usage, disk I/O status, and other items.
- Network resource (see *1.7.6*)
  This is the NIC resource used by the Docker container. It is important for performance monitoring of data send/receive speed and other items.

To set up definitions to monitor these important items, PFM - RM for Virtual Machine provides monitoring templates. Therefore, this section mainly explains monitoring methods that use monitoring templates.

1.7.2 Selecting a baseline

See *1.4.2 Selecting a baseline*.

1.7.3 Monitoring the CPU resource

This subsection explains how to monitor the CPU resource of a Docker container.

(1) Overview

Monitoring CPU performance information enables you to understand performance trends of Docker containers.

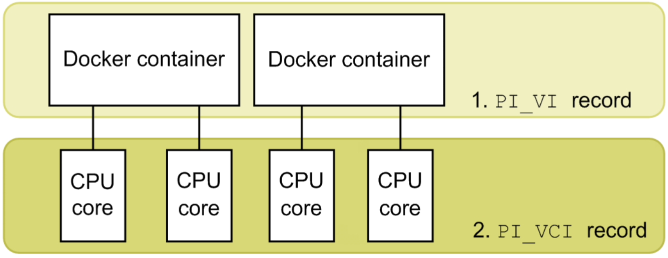
By monitoring CPU performance data, you can detect Docker containers with high CPU usage rate, and thus you can take appropriate corrective action.

The following two records are used to monitor the CPU resource. For details about records, see *5. Records*.The PI and PI_HCI records that indicate the CPU resource of the physical server are not supported.

1. PI_VI record
   This record is used to monitor the performance data of the CPUs that are being used by each Docker container.
2. PI_VCI record
   This record is used to monitor the performance data of the CPU cores that are being used by each Docker container.

The following figure shows the range of performance data collected in each record.

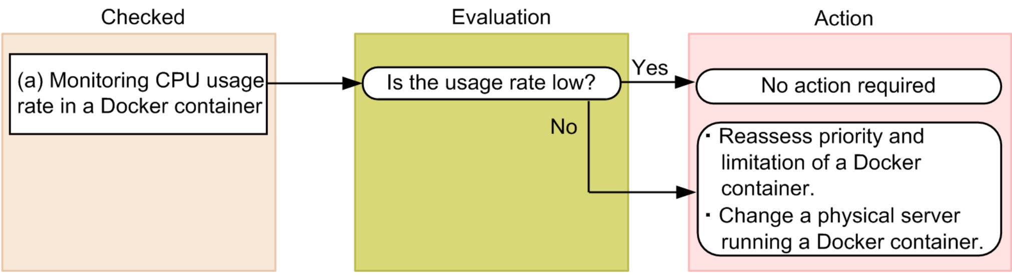
Figure 1–68: Range of performance data collected in each record



(2) Monitoring examples

Using CPU resource monitoring on virtual machines container1 through container2 as an example, this subsection explains the factors that cause insufficient CPU resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–69: Monitored items and flow of actions

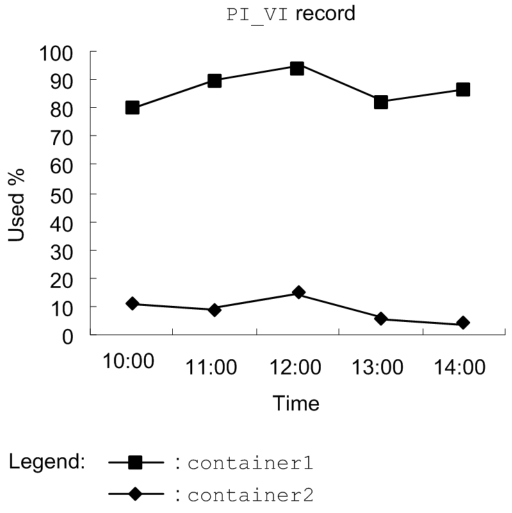


(a) Example of monitoring the CPU usage rate for Docker containers

You can check the CPU usage rate for Docker containers in the Used % field of the PI_VI record.

An example of monitoring is shown below.

Figure 1–70: Example of monitoring the CPU usage rate for Docker containers



Monitoring template report to be checked

VM CPU Trend

In this example, the CPU usage rate of `container1` is high. If the performance of a Docker container has degraded due to high CPU usage rate, review the number of allocated CPU resources and other configuration information. If the problem still cannot be solved, consider whether you can add a physical CPU that runs a Docker container or change the physical server.

1.7.4 Monitoring the memory resource

This subsection explains how to monitor the memory resource of a Docker container.

(1) Overview

Multiple Docker containers share the physical server's memory resource. In a Docker container, processes run within the range of the memory resource that has been allocated to the Docker container.

You can understand used amounts of the memory resource that is being used by each Docker container.

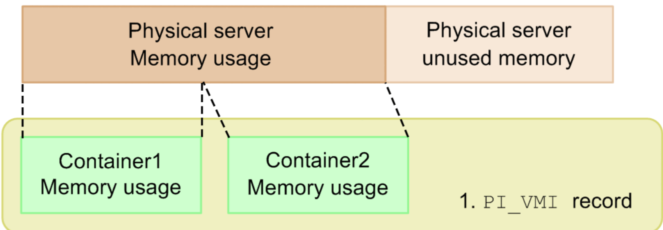
The following `PI_VMI` record a used to monitor the memory resource. For details about records, see *5. Records*. The `PI_HMI` record this indicate the memory resource of the physical server are not supported.

1. `PI_VMI` record

   This record is used to monitor the amount of memory resources allocated to each Docker container.

The following figure shows the range of performance data collected in the `PI_VMI` record.

Figure 1–71: Correspondence between records and data collection ranges

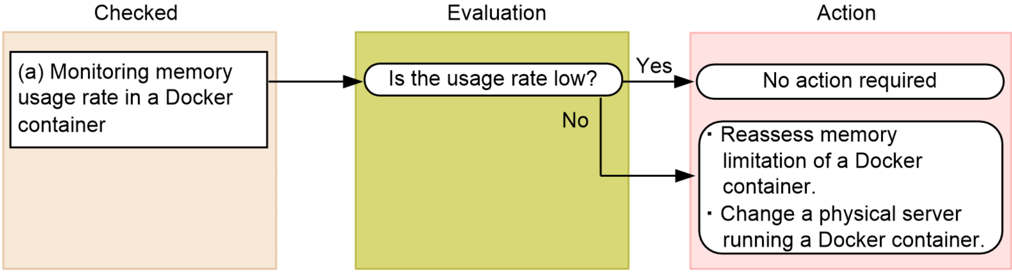


(2) Monitoring examples

This subsection explains the factors that cause insufficient memory resource of Docker containers and how to solve such problems.

The following figure shows the items monitored here and the flow of actions to take.

Figure 1–72: Monitored items and flow of actions

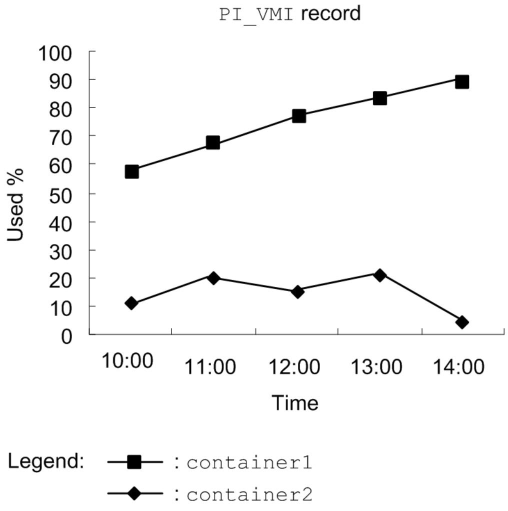


(a) Example of monitoring the total memory usage rate of a Docker container

You can monitor the memory usage rates of Docker containers in the Used % field of the `PI_VMI` record. If this value is large, the memory resources of the Docker container are considered insufficient.

The figure below shows a monitoring example.

Figure 1–73: Example of monitoring the memory usage rate



Monitoring template report to be checked

VM Memory Trend

In this example, as the value of Used % for container1 has increased, insufficient memory resource of the Docker container may occur.

In this case, check the processes that are running in the Docker container. If the problem still cannot be solved, consider changing the Docker container's memory resource, changing the physical server that runs the Docker container, or taking other appropriate action.

1.7.5 Monitoring disk resources

This subsection explains how to monitor the disk resources of a Docker container.

(1) Overview

For Docker containers, the disk I/O resource, which includes information such as the disk transfer rate, can be monitored. The `PI_HPDI` and `PI_HLDI` records that indicate the physical server's disk resource and the `PI_VLDI` record that indicates the virtual machine's disk space resource are not supported.

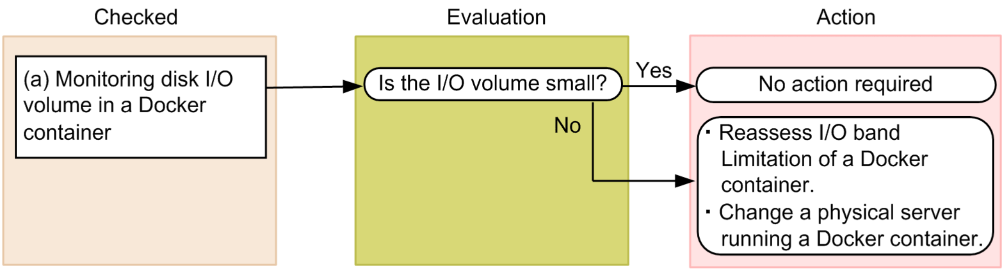
1. `PI_VPDI` record

   This record is used to monitor the performance data of the disk device that is used by a Docker container. It shows the disk I/O resource as viewed from the Docker container.

(2) Monitoring examples

This subsection uses an example of monitoring the Docker container's disk I/O to explain the problems that might occur in disk resources and how to solve them. The following figure shows the items monitored here and the flow of actions to take.

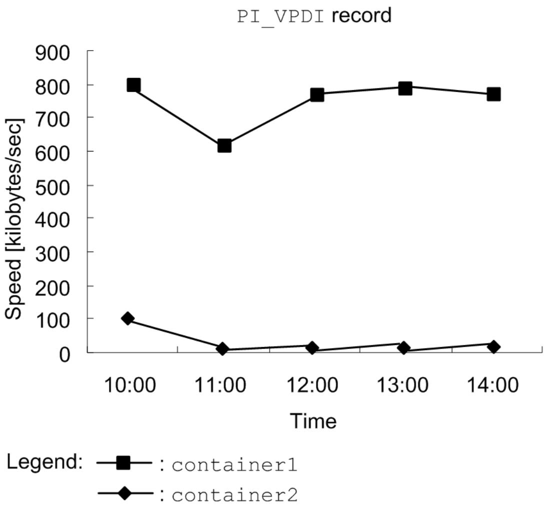
Figure 1–74: Monitored items and flow of actions



(a) Example of monitoring the disk I/O volume of a container

You can check the value of disk I/O volume in the Speed field of the `PI_VPDI` record. The figure below shows a monitoring example.

Figure 1–75: Example of monitoring the disk I/O usage volume



Monitoring template report to be checked

VM Disk I/O

In this example, the value of disk I/O volume for container1 has increased. You can solve this problem by identifying a Docker container with high disk I/O volume.

1.7.6 Monitoring the network resources

This subsection explains how to monitor the network resources of a Docker container.

(1) Overview

A virtual NIC is allocated to each Docker container. The processes that are running in the Docker container recognize a virtual NIC as a normal NIC.

By monitoring the network's performance data of each Docker container, you can detect Docker containers with high traffic volume, and thus you can take appropriate corrective action.

The following `PI_VNI` record a used to monitor the network resources. For details about the records, see *5. Records*. The `PI_HNI` record this indicate the network resources of the physical server are not supported.

1. `PI_VNI` record
   This record is used to monitor the performance data of virtual NICs for Docker containers.

(2) Monitoring examples

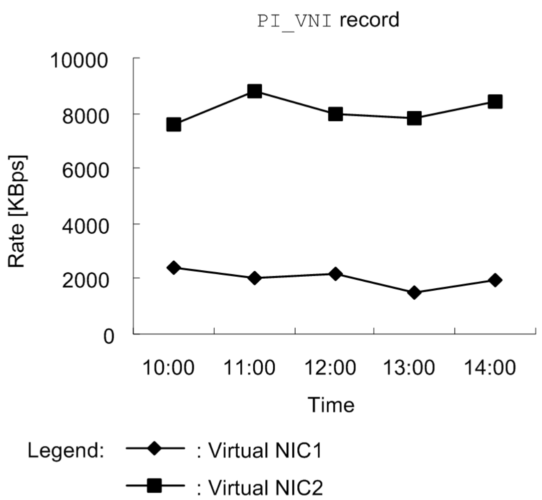
This subsection explains resource monitoring of the virtual NIC1 and NIC2.

(a) Example of monitoring the data send/receive speed of a virtual NIC

The load placed on a virtual NIC can be evaluated from the data send/receive speed of the virtual NIC. You can check the data send/receive speed of the virtual NIC in the Rate field of the `PI_VNI` record.

The figure below shows an example of monitoring the data send/receive speed of a virtual NIC.

Figure 1–76: Example of monitoring the data send/receive speed of a virtual NIC



Monitoring template report to be checked

VM Network Data

In this example, the value of data send/receive speed for the virtual NIC2 has increased. You can solve this problem by identifying a Docker container with high data send/receive speed.

1.8 Example of performance monitoring operation using PFM - RM for Virtual Machine (for Podman environment)

To ensure the stability of operations in a system, it is important to monitor the system's performance and assess its status. When you monitor a Podman environment, items to be monitored vary depending on PFM products.

This section explains how to use PFM - RM for Virtual Machine to monitor the performance of a Podman environment system.

1.8.1 System resources that are important for performance monitoring in a Podman container

The following are the system resources that are important when you use PFM - RM for Virtual Machine to monitor the performance of a Podman container.

- CPU resource (see *1.8.3*)
  This is the CPU resources installed in the physical server. It is important for performance monitoring of CPU usage and other items.
- Memory resource (see *1.8.4*)
  This is the memory resource installed in the physical server. It is important for performance monitoring of memory usage and other items.
- Disk resource (see *1.8.5*)
  This is the disk resource installed in the physical server. It is important for performance monitoring of disk usage, disk I/O status, and other items.
- Network resource (see *1.8.6*)
  This is the NIC resource installed in the physical server. It is important for performance monitoring of data send/receive speed and other items.

To set up definitions to monitor these important items, PFM - RM for Virtual Machine provides monitoring templates. Therefore, this section mainly explains monitoring methods that use monitoring templates.

1.8.2 Selecting a baseline

See *1.4.2 Selecting a baseline*.

1.8.3 Monitoring the CPU resource

This subsection explains how to monitor the CPU resource of a Podman environment.

(1) Overview

In a Podman environment, multiple containers share the physical server's CPUs.

The CPU resources of the physical server are distributed according to each container's CPU resources requirement. However, if the combined total of the CPU resource requirements for the individual containers exceeds the CPU resources on the physical server, the required amount of CPU resources cannot be distributed, resulting in a virtual CPU resource shortage. This adversely affects the performance of the container.

By monitoring CPU performance data, you can detect such performance deterioration in the containers, and thus you can take appropriate corrective action.

The following four records are used to monitor the CPU resource. For details about records, see *5. Records*.

1. `PI` record

   This record is used to monitor the performance data of the physical server's CPUs.

2. `PI_HCI` record

   This record is used to monitor the performance data of each core of the physical CPUs.

3. `PI_VI` record

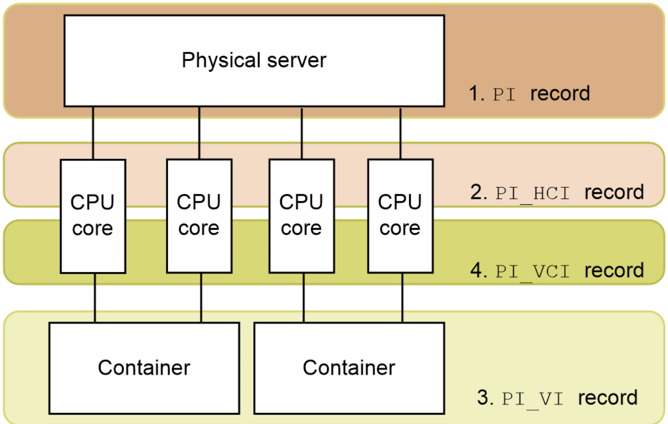
   This record is used to monitor the performance data of the physical server's CPUs that are being used by each container.

4. `PI_VCI` record

   This record is used to monitor the performance data of each core of the physical CPUs.

The following figure shows the range of performance data collected in each record.

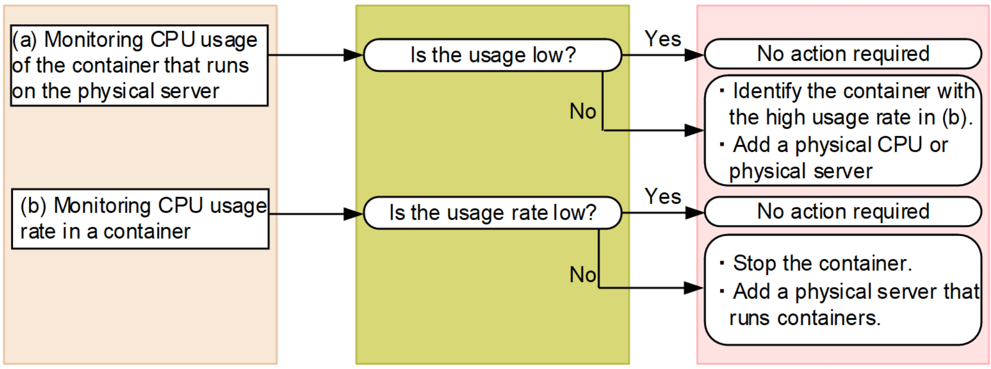
Figure 1–77: Range of performance data collected in each record



(2) Monitoring examples

Using CPU resource monitoring on Physical server as an example, this subsection explains the factors that cause insufficient CPU resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–78: Monitored items and flow of actions

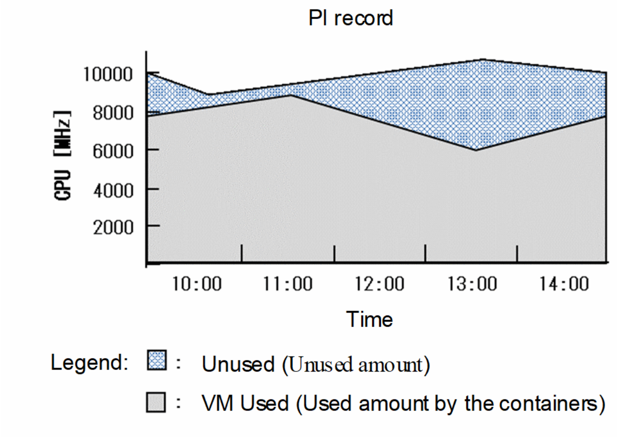


(a) Example of monitoring CPU usage of the container working on the physical server

You can check the CPU usage rate for the container working on the physical server in the VM Used field of the `PI` record.

An example of monitoring is shown below. This example is based on using the monitoring template to display a `Host CPU Used Status` report. The chart in this report displays the VM Used field that indicates the CPU usage for the container. To monitor the VM Used field, check the table that is displayed under the chart.

Figure 1–79: Example of monitoring the CPU usage rate for Physical server



Monitoring template report to be checked

Host CPU Used Status

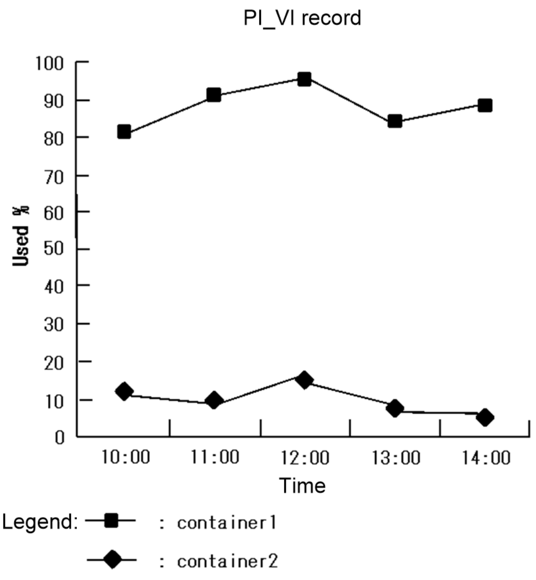
In this example, the CPU usage of the container working on the physical server has increased. If the CPU usage is high and the container's performance is poor, you should stop containers that are not necessary. If the problem persists, consider either adding a physical CPU or adding a physical server.

(b) Example of monitoring the CPU usage rate for containers

You can check the CPU usage rate for each container in the Used % field of the `PI_VI` record.

An example of monitoring is shown below. This example is based on using the monitoring template to display a `VM CPU Trend` report. The chart in this report displays the Used % field that indicates the CPU usage rate for the container. To monitor the Used % field, check the table that is displayed under the chart.

Figure 1–80: Example of monitoring the CPU usage rate for containers



Monitoring template report to be checked

VM CPU Trend

In this example, the CPU usage rate of `container1` is high. If there is a problem with `container1`, stop the container. If the problem persists, consider moving the container to a different physical server.

1.8.4 Monitoring the memory resource

This subsection explains how to monitor the memory resource of a Podman environment.

(1) Overview

In a Podman environment, multiple containers share the physical server's memory resources.

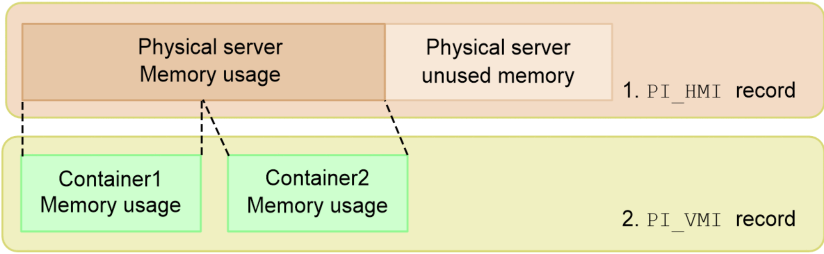
If the physical server does not have enough memory resources, swapping occurs, adversely affecting the performance of the physical server. By monitoring memory performance data, you can detect such performance deterioration in the physical server.

The following two records are used to monitor the memory resource. For details about records, see *5. Records*.

1. `PI_HMI` record
   This record is used to monitor the physical server's physical memory usage and other information.
2. `PI_VMI` record
   This record is used to monitor the memory resources for container.

The following figure shows the range of performance data collected in the `PI_HMI` record and the `PI_VMI` record.

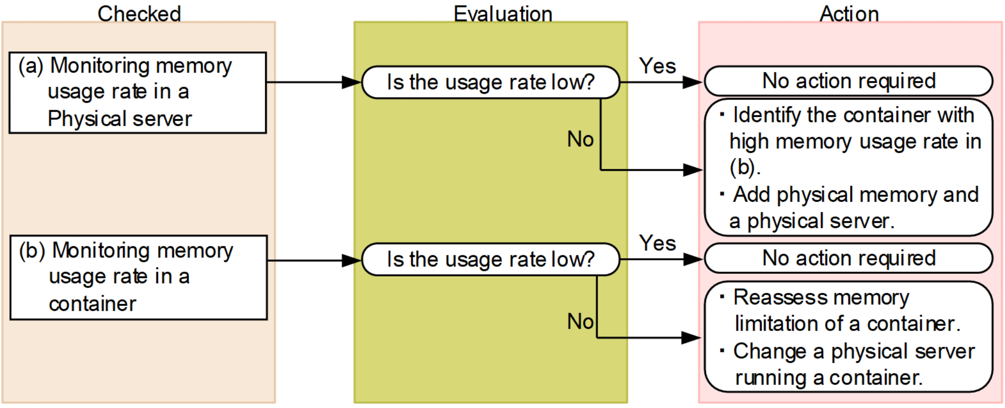
Figure 1–81: Correspondence between records and data collection ranges



(2) Monitoring examples

Using monitoring of a physical server on which the virtual environment is running as an example, this subsection explains the factors that cause insufficient memory resources and how to solve these problems. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–82: Monitored items and flow of actions

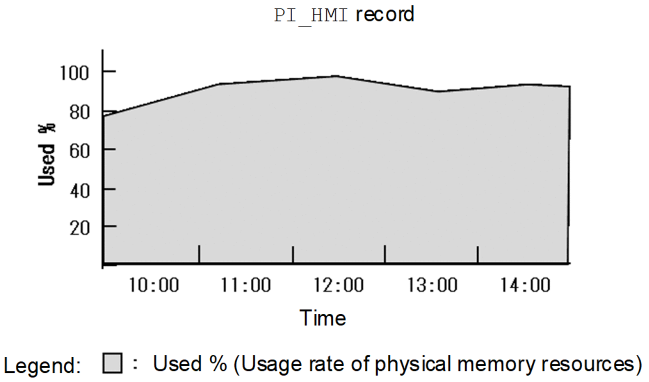


(a) Example of monitoring the memory usage rate of a physical server

You can monitor the memory usage rates of physical server in the Used % field of the PI_HMI record. If this value is large, the memory resources of the physical server are considered insufficient.

The figure below shows a monitoring example.

Figure 1–83: Example of monitoring the memory resources usage rate



Monitoring template report to be checked

Host Memory Used

In this example, as the Used % field shows values between 80% and 100%, there might be a possibility that the physical server does not have enough memory resources.

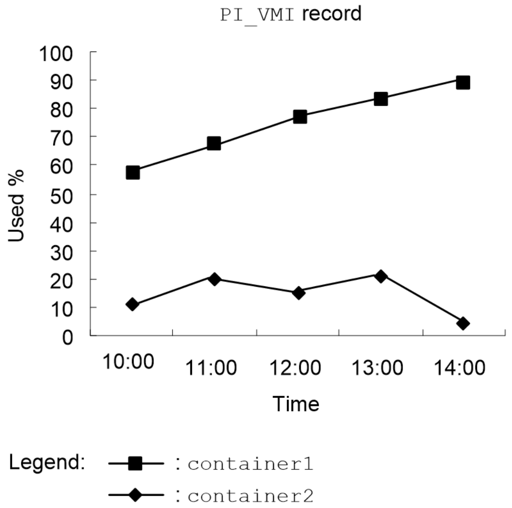
In this case, reassess the memory assignment to the container. If the memory usage value does not improve even after the memory assignment has been reassessed, consider either adding memory resources to the physical server or adding another physical server.

(b) Example of monitoring the memory usage rate of a container

You can monitor the memory usage rates of containers in the Used % field of the `PI_VMI` record.

The figure below shows a monitoring example.

Figure 1–84: Example of monitoring the memory usage rate



Monitoring template report to be checked

VM Memory Trend

In this example, the memory resource usage rate of `container1` increases over time.

If there is a problem with `container1`, stop the container. If the problem persists, consider moving the container to a different physical server.

1.8.5 Monitoring disk resources

This subsection explains how to monitor the disk resources of a Podman environment.

(1) Overview

In a Podman environment, resources can be divided into disk I/O resources indicating the transfer rate of a disk and other data, and disk space resources indicating the disk amount and other data. PFM - RM for Virtual Machine provides records for each of these resources. When monitoring the Podman environment, you can monitor disk resources with the records below. Note that the `PI_VLDI` record, which indicates the disk space resources of the container, is not supported.

1. `PI_HPDI` record

This record is used to monitor the performance data of the physical disk device. It shows the disk I/O resource as viewed from the physical server.

2. PI_VPDI record

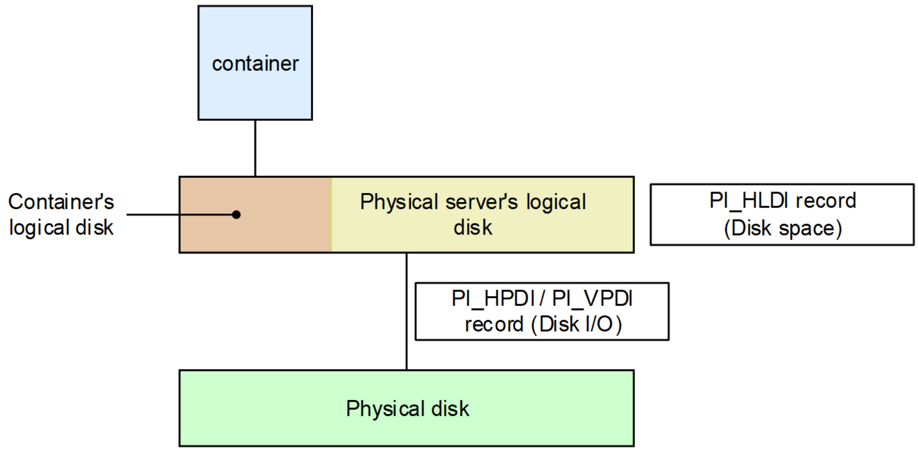
This record is used to monitor the performance data of the physical disk device that is used by a container. It shows the disk I/O resource as viewed from the container.

3. PI_HLDI record

This record is used to monitor the performance data of the physical server's logical disk. It shows the disk space resources as viewed from the physical server.

The following figure shows the range of performance data collected in each record.

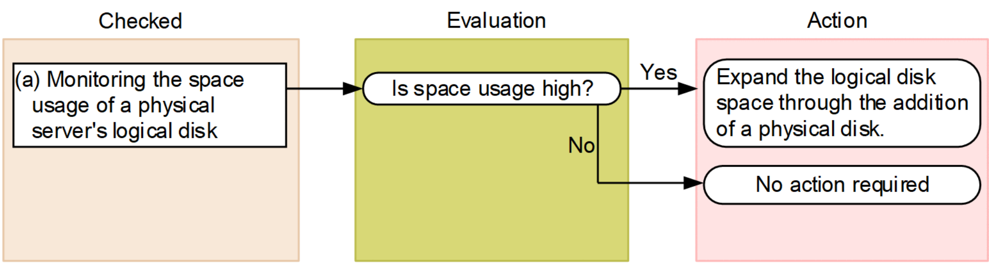
Figure 1–85: Correspondence between records and data collection ranges



(2) Monitoring examples

Using monitoring of disks 1 and 2 installed in the physical server as an example, this subsection explains the problems that might occur in the disk resources and how to solve them. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–86: Monitored items and flow of actions

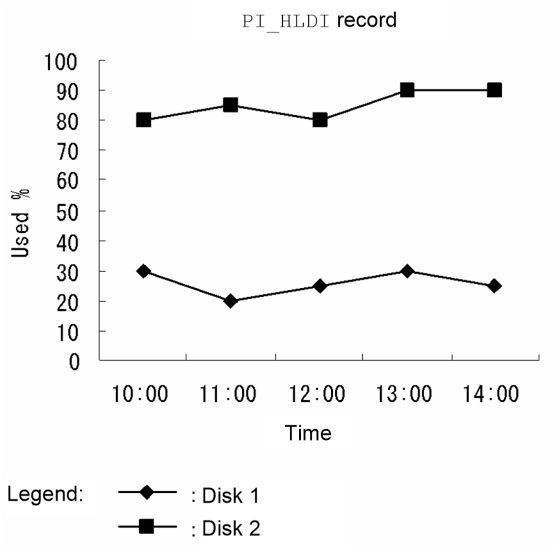


(a) Example of monitoring the space usage of a physical server's logical disk

You can check whether a physical server's logical disk has sufficient free space based on the space usage.

You can check the space usage in the Used % field of the PI_HLDI record. Note that you can monitor this field using an alarm that is provided in the monitoring template. The figure below shows a monitoring example.

Figure 1–87: Example of monitoring the space usage of a physical server's logical disk



Monitoring template report to be checked

Host Disk Used

Monitoring template alarm to be checked

Host Disk Usage

In this example, the space usage of Disk 2 is high. If the space usage is high, you can solve this problem by expanding the logical disk space through the addition of a physical disk.

1.8.6 Monitoring the network resources

This subsection explains how to monitor the network resources of a Podman environment.

(1) Overview

In a Podman environment, multiple containers share the NIC of the physical server. The NIC allocated to each container is called a *virtual NIC*. The container recognizes a virtual NIC as a normal NIC.

In a Podman environment, the containers use the physical NIC at the same time, and as a result, the network bandwidth that can be used by each container becomes narrow. Consequently, each container's network data send/receive speed may decrease.

By monitoring the network's performance data, you can detect containers with high traffic volume, and thus you can take appropriate corrective action.

The following two records are used to monitor the network resource. For details about the records, see *5. Records*.

1. PI_HNI record

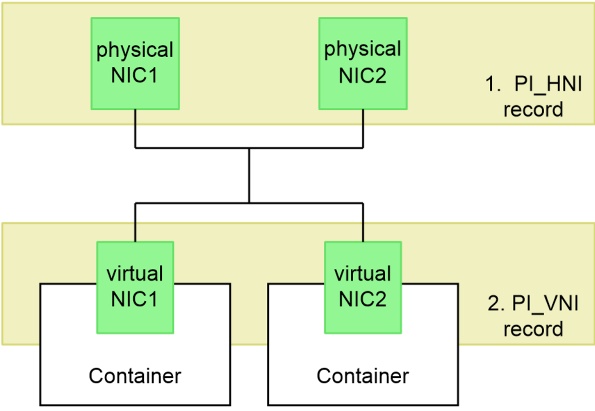
   This record is used to monitor the performance data of physical NICs.

2. PI_VNI record

   This record is used to monitor the performance data of virtual NICs.

The following figure shows the range of performance data collected in each record.

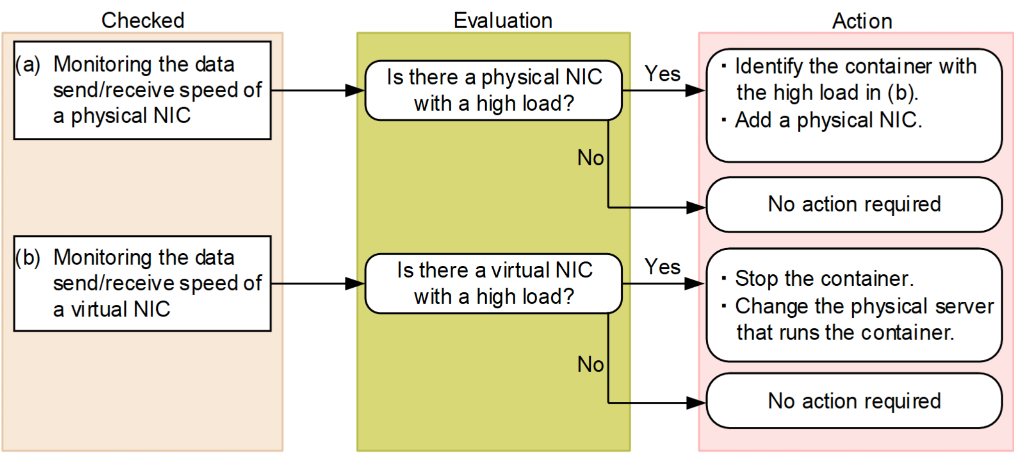
Figure 1–88: Correspondence between records and data collection ranges



(2) Monitoring examples

Using resource monitoring of virtual NICs 1 and 2 and physical NICs 1 and 2 as an example, this subsection explains problems that may occur with network resources, and how to solve them. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–89: Monitored items and flow of actions

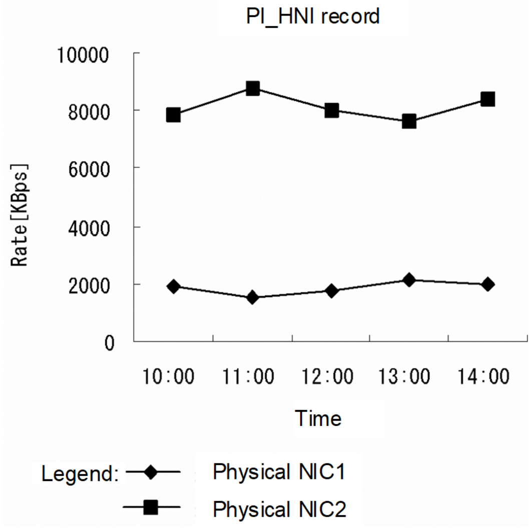


(a) Example of monitoring the data send/receive speed of a physical NIC

The load placed on a physical NIC can be evaluated from the data send/receive speed of the physical NIC. You can check the data send/receive speed of the physical NIC in the Rate field of the `PI_HNI` record.

The figure below shows an example of monitoring the data send/receive speed of a physical NIC.

Figure 1–90: Example of monitoring the data send/receive speed of a physical NIC



Monitoring template report to be checked

Host Network Data

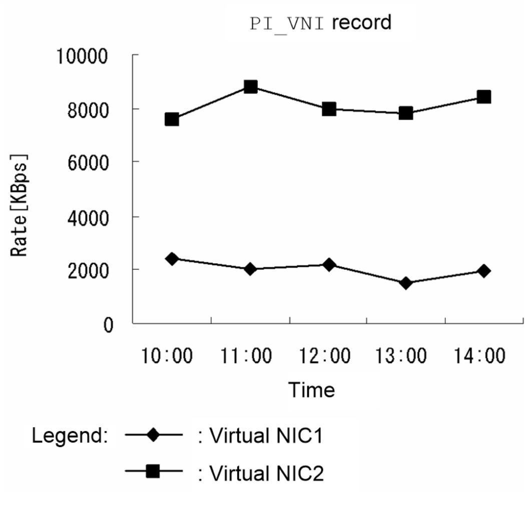
In this example, the data send/receive speed of physical NIC2 is high, indicating a high load level. If there is a physical NIC with a high load, consider adding another physical NIC.

(b) Example of monitoring the data send/receive speed of a virtual NIC

The load placed on a virtual NIC can be evaluated from the data send/receive speed of the virtual NIC. You can check the data send/receive speed of the virtual NIC in the Rate field of the `PI_VNI` record.

The figure below shows an example of monitoring the data send/receive speed of a virtual NIC.

Figure 1–91: Example of monitoring the data send/receive speed of a virtual NIC



Monitoring template report to be checked

VM Network Data

In this example, the data send/receive speed of virtual NIC2 is high, indicating a high load level. If there is a problem with a container, stop the container. If there is no problem with the container and there is a virtual NIC with a high load, take any of the following actions depending on the communication status of the container:

- If the container communicates with an external network

  Add physical NICs to distribute the load on virtual NICs. Alternatively, move the container to a different physical server.
- If the container communicates with only other containers in the same virtual environment

  Configure the container that communicates to work only in the same pod and change the configuration so that it uses the local network in the pod.

1.9 Example of performance monitoring operation using PFM - RM for Virtual Machine (for logical partitioning feature)

To ensure the stability of operations in a system, it is important to monitor the system's performance and assess its status. This section explains how to use PFM - RM for Virtual Machine to monitor the performance of a logical partitioning feature system.

1.9.1 System resources that are important for performance monitoring in a logical partitioning feature system

The following are the system resources that are important when you use PFM - RM for Virtual Machine to monitor the performance of a logical partitioning feature system.

- CPU resource (see *1.9.3*)
  This is the CPU resource installed on the host machine. It is important for performance monitoring of CPU usage and other items.
- Memory resource (see *1.9.4*)
  This is the memory resource installed on the host machine. It is important for performance monitoring of the amount of unused memory.
- Disk resource (see *1.9.5*)
  This is the disk resource installed on the HBA (Host Bus Adapter). It is important for performance monitoring of the number of I/O interrupts.
- Network resource (see *1.9.6*)
  This is the NIC resource installed on the host machine. It is important for performance monitoring of data send/receive speed and other items.

A system with logical partitioning feature consumes CPU resources when processing access to disks and networks. A failure to reserve CPU resources necessary for processing access reduces performance of the host machine and LPARs. Therefore, the CPU resources are most important when the performance of a system with logical partitioning feature is monitored. When configuring monitoring settings, set the monitoring of CPU resources first.

To set up definitions to monitor these important items, PFM - RM for Virtual Machine provides monitoring templates. Therefore, this section mainly explains monitoring methods that use monitoring templates. It also gives examples of compound report definition, which simplifies monitoring.

1.9.2 Selecting a baseline

See *1.4.2 Selecting a baseline*.

1.9.3 Monitoring the CPU resource

This subsection explains how to monitor the CPU resource of a logical partitioning feature system.

(1) Overview

In a system with logical partitioning feature, CPUs on the host machine are allocated to each LPAR and used accordingly. The CPU resource allocated to each LPAR is called a *virtual CPU*. The OS running on a LPAR recognizes a virtual CPU as a normal physical CPU.

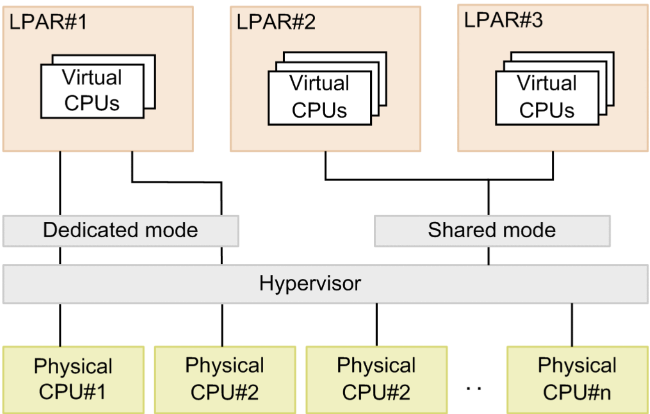
In logical partitioning feature, there are the following two ways to allocate a physical CPU to a LPAR:

- **Dedicated mode**
  This is a mode in which a single LPAR exclusively uses the specified number of physical CPUs. The rate of CPU resource usage per LPAR can be adjusted by changing the number of allocated CPUs.
- **Shared mode**
  This is a mode in which multiple LPARs share the specified number of physical CPUs. The rate of CPU resource usage per LPAR can be adjusted by setting the CPU service rate.

In addition, a hypervisor that manages the system with logical partitioning feature uses all physical CPU resources. The hypervisor can be divided into the kernel component (which is called *SYS1*), and the communication and service component (which is called *SYS2*).

The following figure shows the relationship among LPARs, the hypervisor, and virtual CPUs.

Figure 1–92: Relationship among LPARs, the hypervisor, and virtual CPUs



If more than one LPAR running in shared mode exists, the following problem may occur:

- Although there are free CPU resources allocated in shared mode available for the entire host machine, a specific LPAR runs short of CPU resource.

If this problem occurs, you must revise the settings related to the processor capping function and idle detection function of logical partitioning feature to make effective use of CPU resources allocated in shared mode.

In addition, if a mix of dedicated and shared modes is used for operation, the following problem may occur:

- A shortage of CPU resources for LPARs running in shared mode reduces performance of the LPARs because the LPARs cannot use physical CPU resources in dedicated mode.

If this problem occurs, you must switch the mode of LPARs from dedicated mode to shared mode to distribute the workload of the LPARs.

By monitoring CPU performance data, you can detect such performance deterioration in the LPARs, and thus you can take an appropriate corrective action. The following four records are used to monitor the CPU resource. For details about the records, see *5. Records*.

1. PI record

   This record is used to monitor the performance data of the host machine's CPU.

2. PI_HCI record

   This record is used to monitor the performance data of each core of the physical CPU.

3. PI_VI record

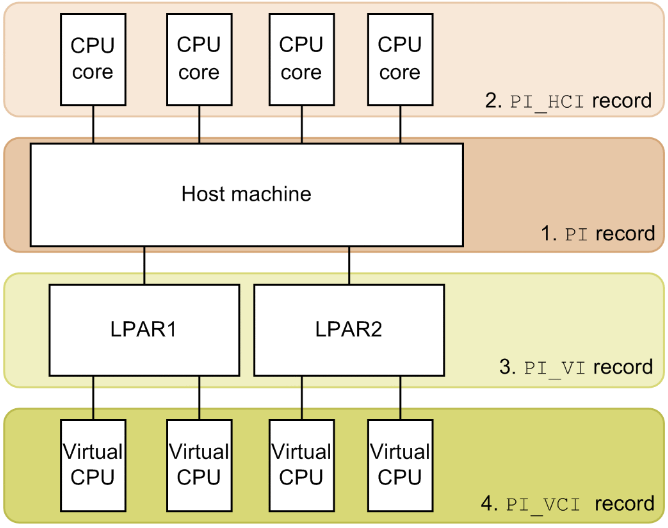
   This record is used to monitor the performance data of the CPU that is being used by each LPAR.

4. PI_VCI record

   This record is used to monitor the performance data of each virtual CPU.

The following figure shows the range of performance data collected in each record.

Figure 1–93: Correspondence between records and data collection ranges



Note that a system with logical partitioning feature uses CPU resources allocated to SYS2 when providing the virtual NIC service. Therefore, performance of CPU resources is affected by virtual NIC usage. By monitoring CPU resources and virtual NIC related resources simultaneously, you can capture the performance of the system with logical partitioning feature more effectively.

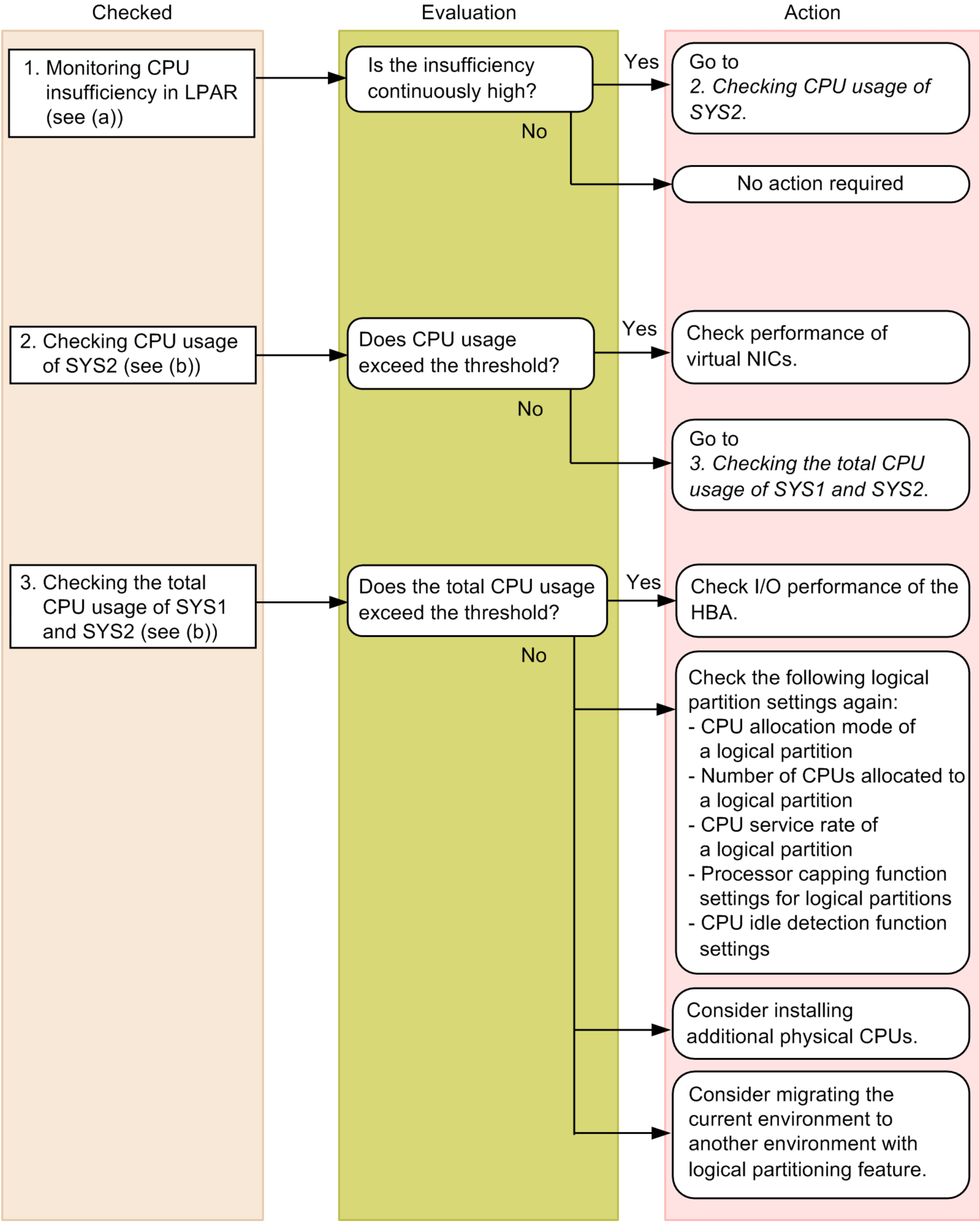
Tip

A system with logical partitioning feature uses the HBA for accessing disks in LPARs. HBA processing consumes CPU resources for SYS1. However, the effects are still less than when the virtual NIC service consumes CPU resources.

(2) Monitoring examples

Using CPU resource monitoring on LPARs vhost1 and vhost2 as an example, this subsection explains the factors that cause insufficient CPU resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–94: Monitored items and flow of actions

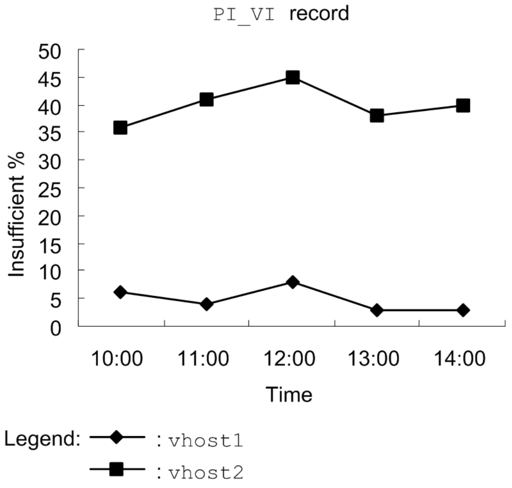


(a) Example of monitoring CPU insufficiency in a LPAR

You can monitor the LPAR's CPU insufficiency in the Insufficient % field of the `PI_VI` record. If a sufficient amount of CPU resources have been allocated to the LPAR, the CPU insufficiency approaches 0%. Note that you can monitor this item with an alarm provided in a monitoring template.

The figure below shows an example of such monitoring.

Figure 1–95: CPU insufficiency monitoring example



Monitoring template report to be checked

VM CPU Insufficient

Monitoring template alarm to be checked

VM CPU Insufficient

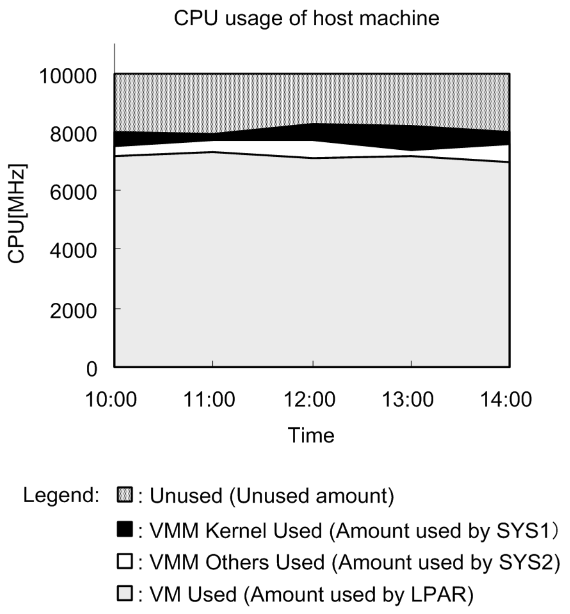
In this example, there appears to be severe insufficiency in the CPU resource of `vhost2`. In this case, check the CPU usage of SYS2.

(b) Example of monitoring CPU usage of the host machine

You can monitor the CPU usage of the host machine in the VM Used, VMM Kernel Used, and VMM Others Used fields of the `PI` record. The VM Used field indicates the CPU usage of each LPAR. The VMM Kernel Used field indicates the CPU usage of SYS1. The VMM Others Used field indicates the CPU usage of SYS2.

The figure below shows an example of such monitoring.

Figure 1–96: Example of monitoring the CPU usage



Monitoring template report to be checked

Host CPU Used Status

If the CPU usage of SYS2 exceeds the threshold, the virtual NIC may be heavily loaded. For details about how to check and handle such a situation, see *1.9.6 Monitoring the network resources*.

> **Tip**
>
> As a guideline, the threshold for the CPU usage of SYS2 should be as much as the usage of two CPU cores. For example, if there are CPUs with eight cores on a system, the threshold would be an amount of use equivalent to 25% of the total usage.

Also, if the total CPU usage of SYS1 and SYS2 exceeds the threshold, the HBA may be heavily loaded when accessing disks. For details about how to check and handle such a situation, see *1.9.5 Monitoring disk resources*.

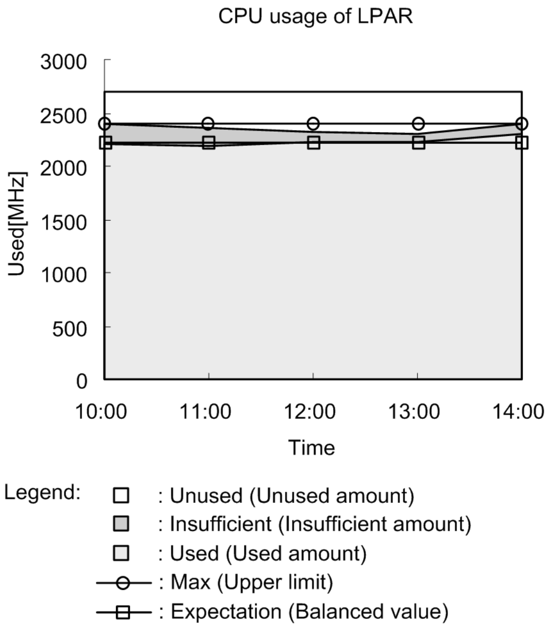
> **Tip**
>
> As a guideline for the total CPU usage of SYS1 and SYS2, the threshold is 90% of the total usage.

(c) Example of checking the maximum CPU allocation size and CPU allocation balancing value of a LPAR

You can check the maximum CPU allocation size of a LPAR in the Max field of the `PI_VI` record. You can also check the CPU allocation balancing value of a LPAR in the Expectation field of the `PI_VI` record. Comparing the maximum CPU allocation sizes and CPU allocation balancing values enables you to investigate a cause of lack of CPU resources in LPARs.

The figure below shows an example of such monitoring.

Figure 1–97: Example of monitoring the maximum CPU allocation size and CPU allocation balancing value



Compound report to be checked (see *1.10*)

Monitoring of virtual machine CPU allocation upper limit

If CPU resources in a LPAR are insufficient, compare the values in the Max and Expectation fields. By using the comparison results, you may be able to address a lack of CPU resources, as described below:

- If the value in the Max field is more than the one in the Expectation field

  The CPU service rate of the LPAR is set to a low value. Check and, if necessary, revise the setting value of the service rate.
- If the value in the Max field is equal to the one in the Expectation field

  The processor capping function restricts the amount of CPU resources allocated to LPARs. Check and, if necessary, revise the setting value of the processor capping function.

(3) Settings of the CPU service rate, processor capping function, and CPU idle detection function

A system with logical partitioning feature provides the functions associated with CPU allocations to LPARs that are described below. Note that in some cases, depending on the settings of these functions, CPU resources cannot be appropriately allocated to LPARs.

CPU service rate and processor capping function

Based on the CPU service rate setting, a CPU allocation to LPARs can be specified by percentage. Also, if the processor capping function is enabled, the CPU service rate value will be the upper limit of the amount to be allocated even if the CPU allocation to LPARs is insufficient.

If the CPU service rate is set to a low value and the processor capping function is enabled for a LPAR that consumes more CPU resources, sufficient CPU resources may not be allocated to the LPAR.

CPU idle detection function

This function detects whether CPUs allocated to a LPAR is in idle state. In a system with logical partitioning feature, if a CPU for a LPAR is in idle state, its resources are allocated to LPARs with insufficient CPU resources allocated. When the CPU idle detection function is disabled, CPUs are not allocated to other LPARs even if they are in idle state, possibly leading to ineffective use of CPU resources.

1.9.4 Monitoring the memory resource

This subsection explains how to monitor the memory resource of a logical partitioning feature system.

(1) Overview

In a virtual environment of a system with logical partitioning feature, the hypervisor and each LPAR use memory exclusively allocated to them from the physical memory on the host machine. The amount of memory is specified by the user at the time of building the environment with logical partitioning feature, and exclusively allocated to LPARs.

The hypervisor reserves a fixed amount of memory of 1,280 MB necessary for self-operation, and exclusively allocates the remainder of physical memory to LPARs when they are started (activated). At this time, if sufficient memory is not allocated to LPARs, a low-memory condition occurs, possibly causing performance of the LPARs to deteriorate. In addition, when a LPAR is started (activated), the LPAR fails to start if the amount of memory specified by the user cannot be allocated from the physical memory.

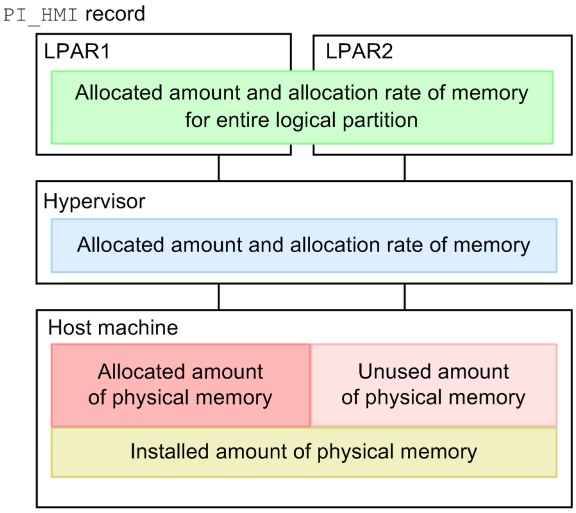
Note that PFM - RM for Virtual Machine cannot capture memory usage, utilization rate, and insufficiency rate data of LPARs. Therefore, you must monitor whether there is a lack of memory in LPARs by using PFM - Agent for Platform or PFM - RM for Platform. When PFM - RM for Virtual Machine is used, if a LPAR runs out of memory, you can take an appropriate corrective action by checking the installed and unused amounts of physical memory on the host machine, and the amount of memory allocated to LPARs.

You can monitor memory resources by using the record shown below. For details about each record, see *5. Records*.

- `PI_HMI` record
  This record is used to monitor the installed, allocated, and unused amount of physical memory on the host machine, as well as the allocated amount and allocation rate of memory for the hypervisor and for the entire LPAR.

The following figure shows the range of performance data collected in the `PI_HMI` record.

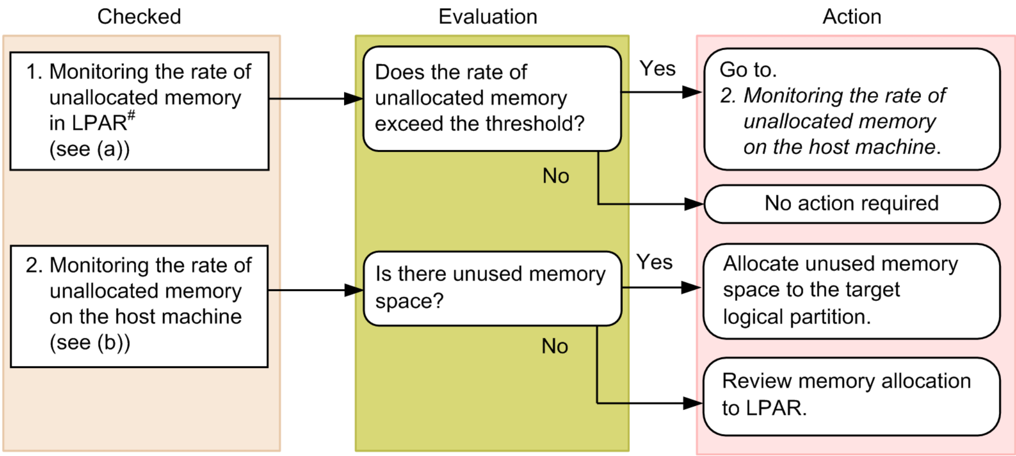
Figure 1–98: Correspondence between records and data collection ranges



(2) Monitoring examples

Using monitoring on the host machine running a virtual environment as an example, this subsection explains the factors that cause insufficient memory resources, and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–99: Monitored items and flow of actions



#: PFM - Agent for Platform or PFM - RM for Platform must be used.

(a) Example of monitoring the rate of unallocated memory in LPARs

You monitor the rate of unallocated memory in LPARs through a combined use of monitoring templates from PFM - Agent for Platform or PFM - RM for Platform, and monitoring templates from PFM - RM for Virtual Machine.

By monitoring the physical memory usage collected by PFM - Agent for Platform or PFM - RM for Platform (from logical partitioning feature's point of view, it seems to monitor memory usage of LPARs), you can determine which LPAR is running out of memory.

For PFM - Agent for Platform

Monitoring template report to be checked

System Overview

Monitoring template alarm to be checked

Available Memory

Monitoring of the combined report of *System Overview* above and *Host Memory Used* from PFM - RM for Virtual Machine enables you to capture the relationship between the LPAR running out of memory and logical partitioning feature.

For PFM - RM for Platform

Monitoring template report to be checked

Memory Used Status

Monitoring template alarm to be checked

Available Memory

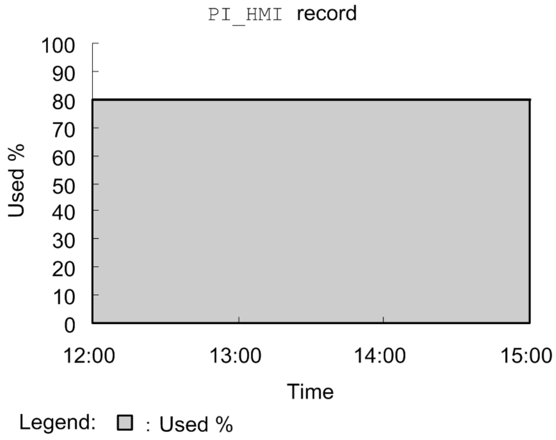
Monitoring of the combined report of *Memory Usage Status* above and *Host Memory Used* from PFM - RM for Virtual Machine enables you to capture the relationship between the LPAR running out of memory and logical partitioning feature.

(b) Example of monitoring the rate of unallocated memory on the host machine

You can monitor the rate of unallocated memory on the host machine in the Used % field of the `PI_HMI` record.

The following figure shows an example of monitoring the rate of unallocated memory on the host machine.

Figure 1–100: Example of monitoring the rate of unallocated memory



Monitoring template report to be checked

Host Memory Used

In this example, you can see that the rate of allocated memory in logical partitioning feature is 80% and 20% of the memory is not used. If there is any unused space, the host machine has an extra amount of memory, which can be reallocated to LPARs running out of memory. If the memory is reallocated, increment the allocation amount of the LPAR running out of memory by 256 MB. Also, reduce the amount of memory allocated to other LPARs without lack of memory.

1.9.5 Monitoring disk resources

This subsection explains how to monitor the disk resources of a logical partitioning feature system.

(1) Overview

Each LPAR on a system with logical partitioning feature uses the SAN boot to start. The system with logical partitioning feature assigns the HBA installed on the physical host of the host machine to each LPAR as a virtual HBA. The LPAR works recognizing the virtual HBA as a physical HBA.

PFM - RM for Virtual Machine obtains the number of I/O interrupts on the physical and virtual HBAs as performance data. If this value remains high, it is considered that the HBA is overloaded by the LPAR, and thus you can take an appropriate corrective action.

Note that PFM - RM for Virtual Machine cannot monitor whether there is a lack of physical disk resources on the SAN storage that exists for each LPAR. Therefore, you must monitor it by using PFM - Agent for Platform or PFM - RM for Platform.

Also, if you want to detect lack of disk resources, failure in HBAs, and other errors, you can use alert monitoring through the LPAR Manager screen.

The following two records are used to monitor the disk resources. For details about each record, see *5. Records*.

1. `PI_HPDI` record

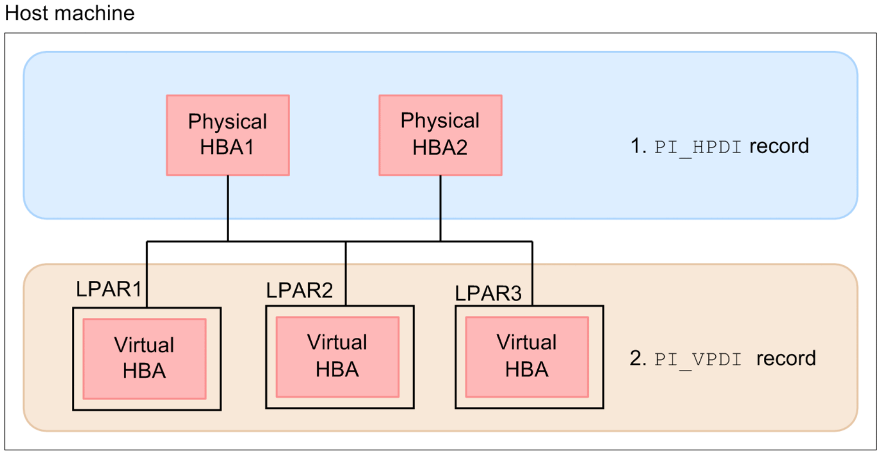
   This record is used to obtain the number of I/O interrupts on the HBA installed on the host machine as performance data. This enables you to monitor the load on the physical HBA.

2. `PI_VPDI` record

   This record is used to obtain the number of I/O interrupts on the entire virtual HBA allocated to LPARs as performance data. By monitoring the number of I/O interrupts per LPAR, you can monitor the disk access load of LPARs.

The following figure shows the range of performance data collected in each record.

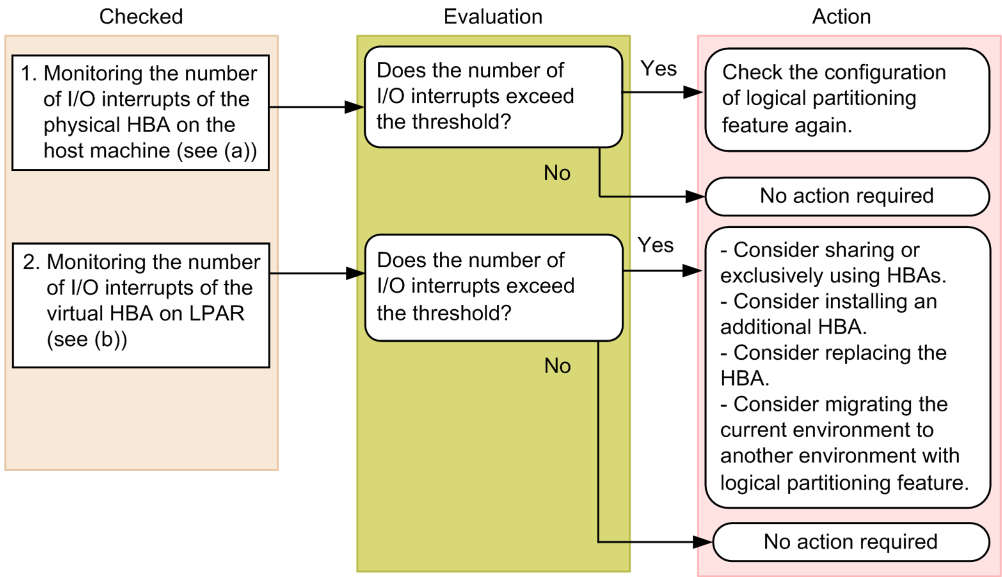
Figure 1–101: Correspondence between records and data collection ranges



(2) Monitoring examples

Using monitoring of the number of I/O interrupts of the physical HBA on the host machine running the virtual environment and that of the virtual HBA for LPARs as an example, this subsection explains the factors that cause insufficient disk resources and how to solve this problem. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–102: Monitored items and flow of actions

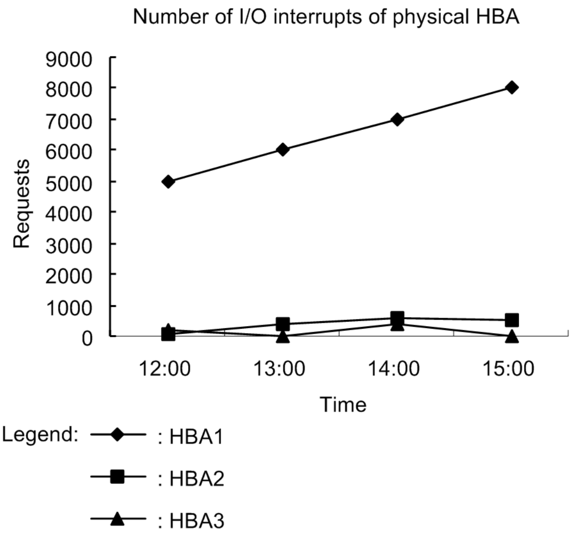


(a) Example of monitoring the number of I/O interrupts of the physical HBA on the host machine

You can monitor the number of I/O interrupts of the physical HBA on the host machine in the Requests field of the `PI_HPDI` record.

The following figure shows an example of monitoring the number of I/O interrupts of the physical HBA on the host machine.

Figure 1–103: Example of monitoring the number of I/O interrupts of the physical HBA on the host machine



Monitoring template report to be checked

Host Disk I/O#

#

The *Host Disk I/O* report does not display the Requests field in a graph. To monitor the number of I/O interrupts of the physical HBA on the host machine, customize the report definition before operation. For details about how to customize the definition, see *(3) Customizing monitoring template reports*.

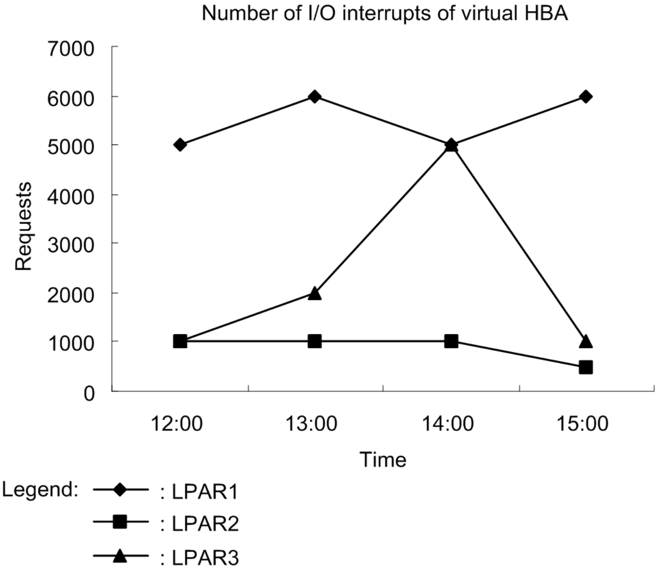
In this example, HBA1 has more I/O interrupts, which means interrupts occur often and continually, and some measures must be taken.

(b) Example of monitoring the number of I/O interrupts of the virtual HBA for LPARs

You can monitor the number of I/O interrupts of the virtual HBA for LPARs in the Requests field of the `PI_VPDI` record.

The following figure shows an example of monitoring the number of I/O interrupts of the virtual HBA for LPARs.

Figure 1–104: Example of monitoring the number of I/O interrupts of the virtual HBA for LPARs



Monitoring template report to be checked

VM Disk I/O#

#

The *VM Disk I/O* report does not display the Requests field in a graph. To monitor the number of I/O interrupts of the virtual HBA for LPARs, customize the report definition before operation. For details about how to customize the definition, see *(3) Customizing monitoring template reports*.

In this example, LPAR1 has more continual interrupts and some measures need to be taken.

(c) Example of monitoring insufficient disk space for LPARs

To monitor whether disk space for LPARs is insufficient, use monitoring templates from PFM - Agent for Platform or PFM - RM for Platform. By using PFM - Agent for Platform or PFM - RM for Platform to monitor the items shown below, you can monitor what percentage of disks for LPARs are not used and identify which LPAR is running out of disk space.

For PFM - Agent for Platform

Monitoring template report to be checked

Free Megabytes - Logical Drive Status

Monitoring template alarm to be checked

Disk Space

For PFM - RM for Platform

Monitoring template report to be checked

Free Megabytes - Logical Disk

Monitoring template alarm to be checked

Disk Space

(3) Customizing monitoring template reports

This subsection explains how to show the Requests field in a graph by customizing the *Host Disk I/O* and *VM Disk I/O* reports in the monitoring template.

(a) Copying the underlying report

Before customizing the reports, you must copy the underlying report to the `User Reports` folder.

To copy the report:

1. In the navigation frame of PFM - Web Console, select the **Reports** tab.
2. In the Reports window, click `System Reports` and then `RM VirtualMachine` to expand.
3. Select the `Host Disk I/O` report or `VM Disk I/O` report that is stored in a desired folder.
4. Copy the selected report to the `User Reports` folder.

(b) Customizing the report

To customize the copied report:

1. In the navigation frame of PFM - Web Console, select the **Reports** tab.
2. In the Reports window, select `User Reports` and then select the `Host Disk I/O` report or `VM Disk I/O` report.
3. In the method frame, select the **Edit** method.
   Click the **Next** button until the Edit > Components window appears.
4. Clear the check box under **Graph** for the **Speed** field, and then select the check box under **Graph** for the **Requests** field.
   Click the **Next** button to show the Edit > Graph window.
5. Change the value of **Y-axis** in **Axis labels** from **Speed[Kbytes/Sec]** to **Requests**.
6. Click the **Finish** button.

1.9.6 Monitoring the network resources

This subsection explains how to monitor the network resources of a logical partitioning feature system.

(1) Overview

In a virtual environment of a system with logical partitioning feature, multiple LPARs share the NIC of the host machine (including the machine built into the chassis). The NIC allocated to each LPAR is called the *virtual NIC*. The LPAR recognizes the virtual NIC as a typical NIC.

A server module is equipped with a *built-in NIC*, and connects to an external LAN via a switch module and management module built into the chassis. The built-in NIC equipped with the server module can be allocated by being shared or exclusively used by LPARs.

When PFM - RM for Virtual Machine is used, physical NICs equipped with the server module (built-in and external NICs) and virtual NICs associated with them are to be monitored as network resources.

By monitoring network performance data, you can keep track of the loads on these NICs, and thus you can take an appropriate corrective action.

The following two records are used to monitor the network resources. For details about each record, see *5. Records*.

1. `PI_HNI` record

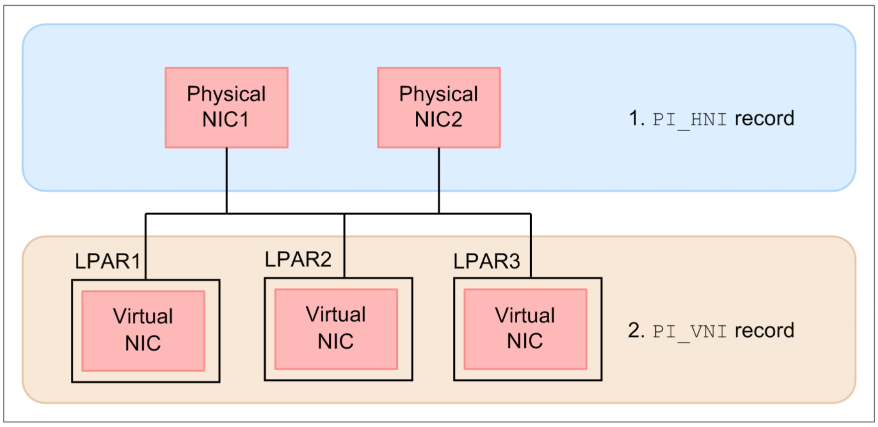
   This record is used to monitor performance data of physical NICs on the host machine. By monitoring the amount of data traffic passing through the physical NICs, you can monitor the loads on the NICs. An overloaded NIC can affect the CPU utilization rate.

2. `PI_VNI` record

   This record is used to monitor performance data of virtual NICs for LPARs. By monitoring the amount of data traffic passing through the allocated virtual NICs, you can monitor whether I/Os are concentrated on a specific virtual NIC.

The following figure shows the range of performance data collected in each record.

Figure 1–105: Correspondence between records and data collection ranges

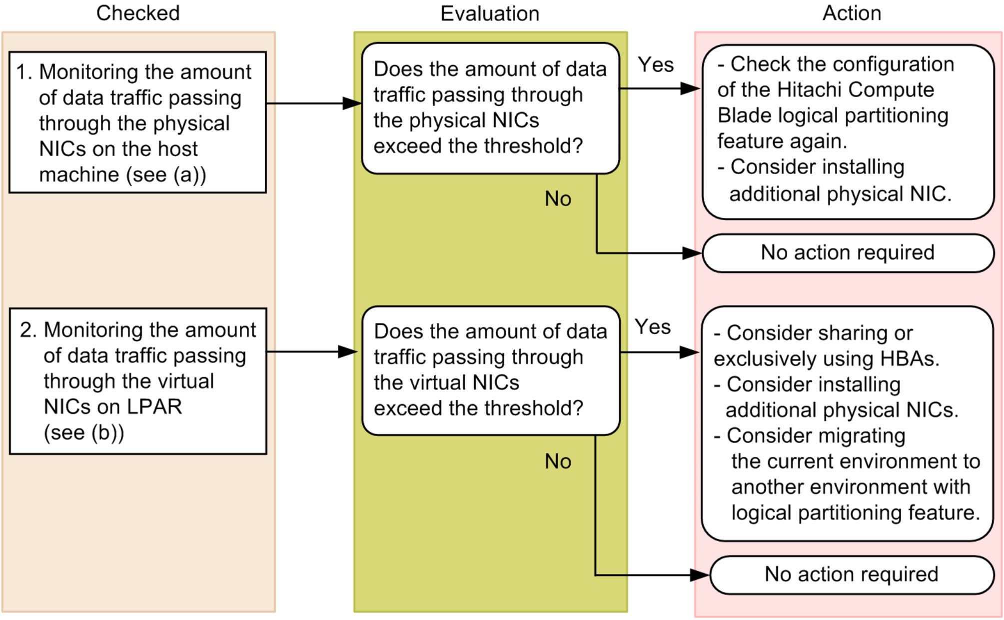


> **Tip**
>
> If an NIC is allocated in shared mode to a LPAR, CPUs on SYS2 process data, and therefore the amount of data traffic passing through the NIC affects the CPU load on SYS2. Thus, if the amount of data traffic passing through the NIC increases, CPU resource usage of SYS2 may increase excessively, causing performance deterioration in the LPARs. Note that if the NIC is allocated in dedicated mode to a LPAR, it does not affect the CPU load on SYS2.

(2) Monitoring examples

Using monitoring of the amounts of data traffic passing through the physical NICs on the host machine and passing through the virtual NICs for LPARs as examples, this subsection explains possible problems related to network resources, and how to solve those problems. The following figure shows the items monitored here and the flow of actions to take.

Figure 1–106: Monitored items and flow of actions

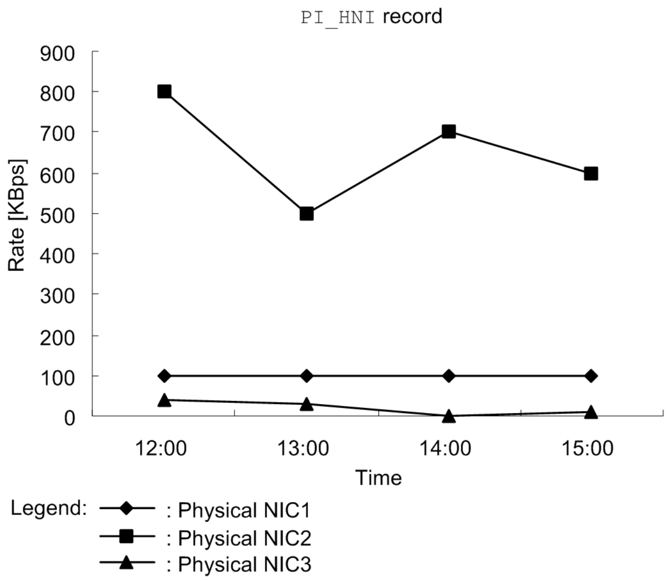


(a) Example of monitoring the amount of data traffic passing through the physical NICs

You can estimate the load on a physical NIC from the data send/receive speed on that physical NIC. You can monitor the data send/receive speed on the physical NIC in the Rate field of the `PI_HNI` record.

The following figure shows an example of monitoring the data send/receive speed on the physical NICs.

Figure 1–107: Example of monitoring the data send/receive speed on the physical NICs



Monitoring template report to be checked

Host Network Data

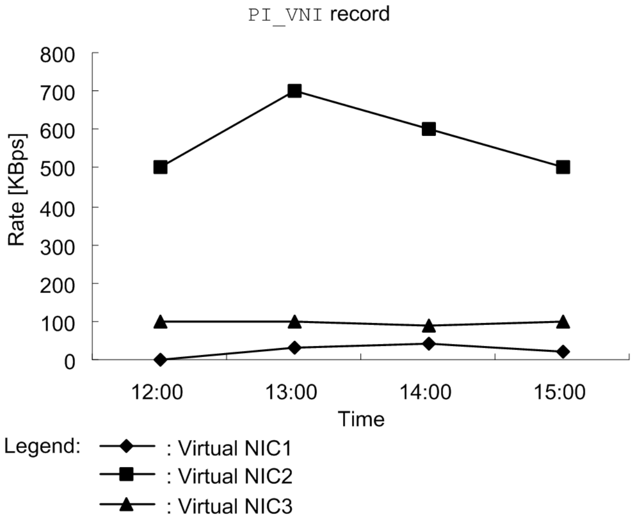
In this example, the traffic volume is low in the physical NIC1 and physical NIC3. On the other hand, the physical NIC2 remains overloaded. Take some actions, such as reviewing the configuration of logical partitioning feature or installing additional physical NICs.

(b) Example of monitoring the amount of data traffic passing through the virtual NICs

You can estimate the load on a virtual NIC from the data send/receive speed on that virtual NIC. You can monitor the data send/receive speed on the virtual NIC in the Rate field of the `PI_VNI` record.

The following figure shows an example of monitoring the data send/receive speed on the virtual NICs.

Figure 1–108: Example of monitoring the data send/receive speed on the virtual NICs



Monitoring template report to be checked

VM Network Data

In this example, the traffic volume is low in the virtual NIC1 and virtual NIC3. On the other hand, the virtual NIC2 remains overloaded. In this case, the problem can be solved by reviewing how NICs are shared or exclusively used. If the problem persists, consider installing additional physical NICs or migrating the current environment to any other environment with logical partitioning feature.

1.10 Description of compound report settings

The table below describes the items in the compound report mentioned in *1.4.4 Monitoring the memory resource* and *1.9.3 Monitoring the CPU resource*. For details about how to create a compound report, see the chapter that explains the creation of reports for operation analysis in the *JP1/Performance Management User's Guide*.

Points

- A compound report is created for each monitored physical server.
- If you create a compound report without specifying any field display conditions, all virtual machines running on the physical server are displayed. To monitor the information of each virtual machine separately, specify a virtual machine name as a field display condition.

Table 1–1: Compound report setting details

Compound report name	Monitoring template report name	Field display condition	Compound bookmark and group setting				
			Group	Graph type	Group name	Max	Min
Monitoring of virtual machine CPU allocation upper limit	`VM CPU Allocation Value`	Change the asterisks (`*`) to the target virtual machine name.	Group 1	Line	VM CPU Allocation Value [MHz]	CPU resource (`PI.Clocks` value)	`0`
	`VM CPU Used`	Change the asterisks (`*`) to the target virtual machine name.	Group 2	Stacked area	CPU [MHz]	CPU resource (`PI.Clocks` value)	`0`
Monitoring of virtual machine memory allocation upper limit	`VM Memory Allocation Value`	Change the asterisks (`*`) to the target virtual machine name.	Group 1	Line	VM Memory Allocation Value [MB]	Memory size (`PI_VMI.Size` value)	`0`
	`VM Swap Used`	Change the asterisks (`*`) to the target virtual machine name.	Group 2	Stacked area	Used [MB]	Memory size (`PI_VMI.Size` value)	`0`
Monitoring of working set size of all virtual machines	`Host Memory Size`	--	Group 1	Line	Memory Size [MB]	Memory size (`PI_VMI.Size` value)	`0`
	`VM Working Size - Total`	--	Group 2	Stacked area	Working Size [MB]	Memory size (`PI_VMI.Size` value)	`0`

Legend:

--: Do not specify.

Max: Maximum value

Min: Minimum value

2 Installation and Setup

This chapter explains how to install and set up PFM - RM for Virtual Machine. For instructions on how to install and set up the entire Performance Management system, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

2.1 Installation and setup

This section explains how to install and set up PFM - RM for Virtual Machine.

2.1.1 Preparation for installation and setup

This subsection explains the items you must check before you install and set up PFM - RM for Virtual Machine.

(1) Prerequisite OS

PFM - RM for Virtual Machine runs under the following OS:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

(2) Setting up the network environment

The following explains the network environment necessary to run Performance Management.

(a) Setting IP addresses

For the PFM - RM for Virtual Machine host, set up an environment in which an IP address can be resolved using a host name. PFM - RM for Virtual Machine cannot start in an environment in which an IP address cannot be resolved.

For the monitoring host name (the name used as the host name of a Performance Management system), you can use either a real host name or an alias name.

- When a real host name is used for the monitoring host name
  Set up the environment such that the host name that can be confirmed using the execution result of the `hostname` command can be used to resolve the IP address.
- When an alias name is used for the monitoring host name
  Set up the environment such that the alias name that has been set can be used to resolve the IP address.

For details about how to set up a monitoring host name, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

Use one of the following to set a host name and an IP address:

- The host information setting file (`jpchosts` file) of Performance Management
- The `hosts` file
- DNS

Note that IP addresses set in the `jpchosts` file are not used to resolve the IP address of a monitoring-target host.

> **Important**
>
> - Performance Management can operate in a DNS environment, but it does not support host names in FQDN format. Therefore, specify a monitoring host name without a domain name.
> - Performance Management cannot operate on a host to which a dynamic IP address is assigned by DHCP. Therefore, set fixed IP addresses for all hosts on which Performance Management will be installed.
> - If Performance Management is being used in multiple LAN environments, set IP addresses in the `jpchosts` file. For details, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.
>
>   When the virtual environment to be monitored is provided by VMware, Hyper-V, KVM, Docker environment, or Podman environment, set up the virtual environment so that an IP address can be resolved using the host name.
> - When the virtual environment to be monitored is one with logical partitioning feature, its host name is not used for management, and therefore, it does not have to be resolved to an IP address. Specify the IP address that is set when the environment with logical partitioning feature is configured at the time of instance generation during setup. However, if you want to use the host name for management, you can use HVM ID, which was defined during the configuration of the environment with logical partitioning feature, as a host name. In this case, configure your system so that HVM ID can be resolved to the IP address. HVM ID must also be unique in the system. Set HVM ID and the IP address by using either of the following methods:
>
>   - `hosts` file
>
>   - DNS

To learn how to change the network configuration or rename host names in a Performance Management system, see the chapter on changing the Performance Management system configuration in the *JP1/Performance Management Planning and Configuration Guide*.

(b) Settings for using IPv6

Because Performance Management supports network configurations for IPv4 environments and IPv6 environments, you can use Performance Management in a network configuration that contains both IPv4 and IPv6 environments.

PFM - RM for Virtual Machine can communicate with PFM - Manager by using IPv6.

Note, however, that this applies only when Windows or Linux runs on a host with PFM - Manager installed.

PFM - RM for Virtual Machine can communicate with monitored VMware, Hyper-V, KVM, Docker environment and Podman environment with IPv6. In addition, communication with Logical partitioning feature to be monitored uses IPv4, it can not be monitored in IPv6 only environment.

For details about the scope of communication in IPv4 and IPv6 environments, see *K. About Communication in IPv4 Environments and IPv6 Environments*.

To use IPv6 for communication, IPv6 must be enabled on both the PFM - Manager host and the PFM - RM host by using the `jpcconf ipv6 enable` command.

In addition, before installing PFM - RM for Virtual Machine, you need to enable the use of IPv6 on the PFM - RM for Virtual Machine host. You have to execute the `jpcconf ipv6 enable` command to enable this setting. If this setting is already enabled, however, you do not need to execute the command. If you want to check whether the use of IPv6 is enabled, execute the `jpcconf ipv6 display` command.

For details about the `jpcconf ipv6 enable` command and the `jpcconf ipv6 display` command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*. For details about the conditions or occasions for executing the `jpcconf ipv6 enable` command, see the chapter that describes network configuration examples in an environment that includes IPv6 in the *JP1/Performance Management Planning and Configuration Guide*.

When you use IPv6 for communication between a monitored host and PFM - RM for Virtual Machine, specify the name of a monitored host where name resolution can be performed.

Communication between PFM - RM for Virtual Machine and a monitoring target is performed with an IP address that can be resolved. Also, if an IPv4 environment and an IPv6 environment are both used, and communication between PFM - RM for Virtual Machine and the monitoring target fails with an IP address that can be resolved, the communication is not retried by using another IP address.

For example, if communication fails when IPv4 is used, IPv6 is not used to retry communication. Similarly, if communication fails when IPv6 is used, IPv4 is not used to retry communication. Make sure beforehand that a connection can be established.

(c) Setting port numbers

The services of Performance Management programs are assigned to the default port numbers shown in the table below. Port numbers that are not being used in the system are automatically assigned to all other services or programs every time a service is started. If you are using Performance Management in a firewall environment, assign fixed port numbers. For details about how to assign fixed port numbers, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

Table 2–1: Default port numbers and Performance Management program services

Service explanation	Service name	Parameter	Port number	Remarks
Service configuration information management function	Name Server	`jp1pcnsvr`	`22285`	Port number used by the Name Server service of PFM - Manager. This port number is set for all hosts of Performance Management.
Service status management function	Status Server	`jp1pcstatsvr`	`22350`	Port number used by the Status Server service of PFM - Manager and PFM - Base. This port number is set for hosts on which PFM - Manager and PFM - Base are installed.
Monitoring console communication facility	View Server	`jp1pcvsvr`	`22286`	The port number used by the View Server service of PFM - Manager. This port is set up on the hosts on which PFM - Manager is installed.
Web service facility	Web Service	--	`20358`	The port number used by the Web Service service of PFM - Web Console.
Web container facility	Web Console	--	`20359` `20360`	The port number used by the Web Console service of PFM - Web Console.
JP1/SLM linkage facility	JP1/ITSLM	--	`20905`	The port number set by JP1/SLM.

Legend:

--: None

Configure the network in such a way that these port numbers used by PFM - RM for Virtual Machine can be used for communications.

(d) SSL/TLS utilization setup

PFM - RM for Virtual Machine acquires the performance data of a virtual environment through a network. If the virtual environment to be monitored is VMware and Docker environment, SSL/TLS must be enabled on the RM host and the monitoring target because communication between PFM - RM for Virtual Machine and the virtual environment is encrypted by SSL/TLS.

When the virtual environment to be monitored is one with VMware, see *2.5.1 For VMware*.

When the virtual environment to be monitored is one with Docker environment, see *2.5.4 For Docker environment*.

(3) OS user permissions necessary for installation

When you install PFM - RM for Virtual Machine, you must have the following permissions:

When the UAC function is enabled

You must log in to the host to be installed with Administrators permissions, or be upgraded to Administrators permissions when you start the installer.

When the UAC function is disabled

You must log in to the host to be installed with Administrators permissions.

(4) Prerequisite programs

This subsection explains the programs required for installing PFM - RM for Virtual Machine. The following figure shows the program configuration diagram.

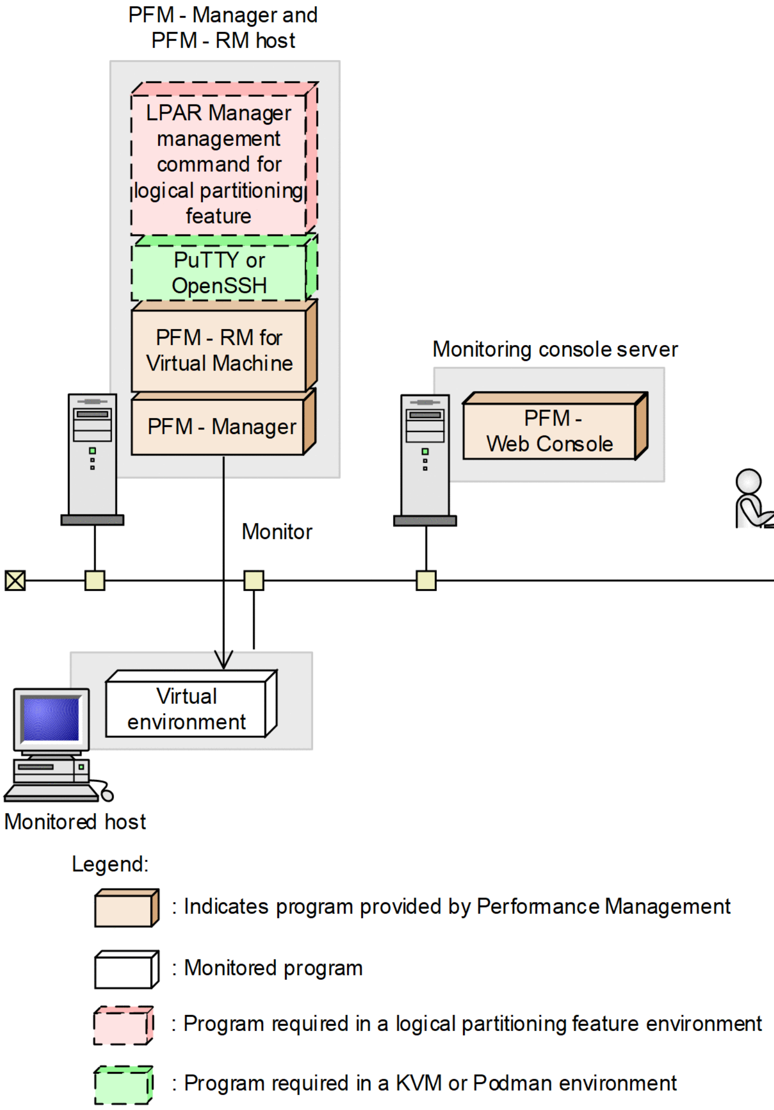
Note that the term *PFM - RM host* means a host on which PFM - RM for Virtual Machine is installed.

Also note that this subsection explains two typical program configurations separately. Decide which configuration you want to use depending on the system environment.

Installing PFM - RM for Virtual Machine on a PFM - Manager host:

In this program configuration, PFM - RM for Virtual Machine is installed on a host on which PFM - Manager is installed. The following figure shows an example of this program configuration.

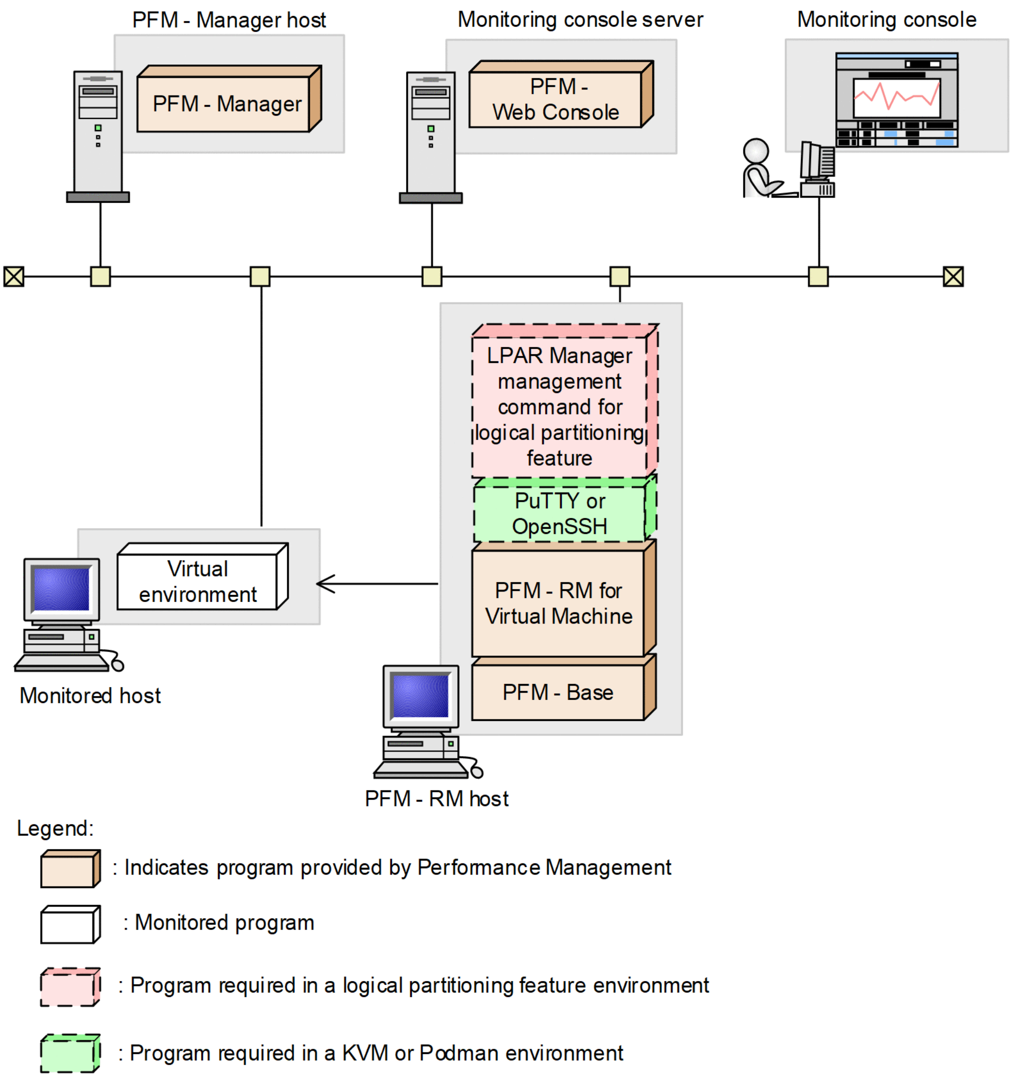
Figure 2–1: Configuration in which PFM - RM for Virtual Machine is installed on a host on which PFM - Manager is installed



Installing PFM - RM for Virtual Machine on a host on which PFM - Manager is not installed:

In this program configuration, PFM - RM for Virtual Machine is installed on a host on which PFM - Manager is not installed. To use this configuration, PFM - Base must also be installed on the host on which PFM - RM for Virtual Machine is installed. The following figure shows an example of this program configuration.

Figure 2–2: Configuration in which PFM - RM for Virtual Machine is installed on a host on which PFM - Base is installed



If the virtual environment of the monitoring target is a VMware or KVM environment, PFM - RM for Virtual Machine uses the Microsoft XML Parser Version 3.0 component that is included in Internet Explorer. Therefore, Internet Explorer must be installed.

(a) Performance Management programs

Before you can install PFM - RM for Virtual Machine on a host, PFM - Manager or PFM - Base must be installed on that host as a prerequisite program for Performance Management.

If you want to install PFM - RM for Virtual Machine on a host on which PFM - Manager is installed, you do not need to install PFM - Base. If you want to install multiple PFM - RM instances on a host on which PFM - Base is installed, only one instance of PFM - Base is required.

In addition, PFM - Web Console is required to monitor the operating status of monitoring targets by using PFM - RM for Virtual Machine.

(b) Virtual environments supported as the monitoring target

The following virtual environments are supported as the monitoring target.

For details about virtual environments that can be monitored, see the *Release Notes*.

For VMware:

- VMware vSphere ESXi 5
- VMware vSphere ESXi 6

For Hyper-V:

- 2012 Hyper-V
- 2012 R2 Hyper-V
- 2016 Hyper-V
- 2019 Hyper-V

For KVM:

- KVM (Red Hat Enterprise Linux 6 (64-bit x86_64))
- KVM (Red Hat Enterprise Linux(R) Server 7)
- KVM (Red Hat Enterprise Linux(R) Server 8)

For Docker environment

- Red Hat Enterprise Linux(R) Server 7
- Windows Server 2016
- Windows Server 2019

For Podman environment

- Red Hat Enterprise Linux(R) Server 8

For logical partitioning feature:

- Hitachi Compute Blade 320
- Hitachi Compute Blade 500
- Hitachi Compute Blade 2000
- Hitachi Compute Blade 2500

> **Important**
>
> Logical partitioning feature supports only IPv4 networks.

(c) Hardware running logical partitioning feature that is supported as the monitoring target

PFM - RM for Virtual Machine can monitor the following sets of hardware running logical partitioning feature:

For details about hardware running logical partitioning feature that can be monitored, see the *Release Notes*.

- Hitachi Compute Blade 320
- Hitachi Compute Blade 500
- Hitachi Compute Blade 2000
- Hitachi Compute Blade 2500

(d) LPAR Manager management command for logical partitioning feature

For the HvmSh command that is necessary for PFM - RM for Virtual Machine to collect information about logical partitioning feature, use the LPAR Manager management command from the management tool for logical partitioning feature provided with the feature.

If you have more than one version of the HvmSh command, use the latest version.

For details about the HvmSh command for hardware running logical partitioning feature, see the *Release Notes*.

> **Important**
>
> The user authentication function of the HvmSh command is non-support.

(5) Tools that must be started on the virtual environment side when VMware is monitored

Before you monitor the information of a virtual machine running on VMware, start vmware-tools on the virtual machine you intend to monitor.

The host name of the virtual machine (VM Host Name field) can be acquired only when vmware-tools is running.

VM Logical Disk Status (PI_VLDI) records of the virtual machine and host names of each record (VM Host Name field) can be acquired only when vmware-tools is running.

- When the OS of the monitored virtual machine is Windows
  During installation of VMware Tools, a dialog box for selecting program functions opens. In this dialog box, leave all default settings intact when you install VMware Tools. The following items are selected as the default settings:
  - SCSI driver
  - SVGA driver
  - Mouse driver
  - VMXNET NIC driver
  - Memory control driver
  - FileSystem Sync Driver
  - VMWCI-Driver
- When the OS of the monitored virtual machine is Linux
  During installation of VMware Tools, a dialog box for selecting program functions is not displayed. Follow the normal installation procedure.
- When the OS of the monitored virtual machine is neither Windows nor Linux
  For instructions on how to install VMware Tools, contact VMware Corporation.

(6) Installation and setup in a cluster system

When you install and set up PFM - RM for Virtual Machine in a cluster system, the prerequisite network environment and program configurations are different from those used to set up a normal configuration. Tasks at the executing node and standby node are also required. For details, see *3. Operations in a Cluster System*.

(7) Notes

This subsection provides notes related to the installation and setup of Performance Management.

(a) Note on the registry

Operation of PFM - RM for Virtual Machine is supported only in an OS environment with standard settings. Performance data might not be collected correctly if special OS settings are specified in some way, such as using the registry editor to edit the registry directly. This is also true if the special settings have been published in the Microsoft Knowledge Base.

(b) Note on the environment variable

Performance Management uses `JPC_HOSTNAME` as the environment variable; therefore, the user must not specify it as an environment variable. If it is specified as an environment variable, Performance Management will not operate correctly.

(c) Notes on installing and setting up multiple Performance Management programs on the same host

You can install multiple Performance Management programs; that is, PFM - Manager, PFM - Web Console, and PFM - RM for Virtual Machine, on the same host. When you do so, note the following.

- To maximize system performance and reliability, we recommend that you run PFM - Manager, PFM - Web Console, and PFM - RM for Virtual Machine on separate hosts.
- If you install PFM - Manager and PFM - RM for Virtual Machine on the same host, you do not need PFM - Base. In this case, since the prerequisite program for PFM - RM for Virtual Machine is PFM - Manager, first install PFM - Manager and then install PFM - RM for Virtual Machine.
- You cannot install PFM - Base and PFM - Manager on the same host. To install PFM - Manager on a host that has PFM - Base and PFM - RM for Virtual Machine installed, you must first uninstall all Performance Management programs except PFM - Web Console, and then install PFM - Manager, followed by PFM - RM for Virtual Machine. Likewise, to install PFM - Base on a host that has PFM - Manager and PFM - RM for Virtual Machine installed, first uninstall all Performance Management programs except PFM - Web Console, and then install PFM - Base, followed by PFM - RM for Virtual Machine.
- When you install PFM - RM for Virtual Machine on a host that has PFM - Manager installed, the PFM - Manager of the local host becomes the PFM - Manager at the connection destination. In this case, you cannot change the PFM - Manager at the connection destination to the PFM - Manager of a remote host. To connect to the PFM - Manager of a remote host, make sure that the host on which you intend to install PFM - RM for Virtual Machine does not have PFM - Manager installed.
- When you install PFM - Manager on a host that has PFM - RM for Virtual Machine installed, the PFM - Manager at the connection destination of PFM - RM for Virtual Machine is reset to the local host name. Check the setting result that is output to the common message log.
- To install PFM - RM for Virtual Machine on a host that has PFM - Web Console installed, close all browser windows before you perform installation.
- When you perform a new installation of a Performance Management program, the status management function is enabled by default. However, when you perform an upgrade installation from a version earlier than 08-00 to version 08-00 or a later version, the status management function settings remain as they were in the earlier version. For details about how to change the settings for the status management function, see the chapter that explains Performance Management error detection in the *JP1/Performance Management User's Guide*.

(d) Notes on version upgrading

For details about notes on upgrading the versions of Performance Management programs, see the section describing the notes on version upgrading in the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

Observe the following precautions when upgrading the version of PFM - RM for Virtual Machine:

- You cannot install PFM - Base and PFM - Manager on the same host. To install PFM - Manager on a host that has PFM - Base and PFM - RM for Virtual Machine installed, you must first uninstall all Performance Management programs except PFM - Web Console, and then install PFM - Manager, followed by PFM - RM for Virtual Machine. Likewise, to install PFM - Base on a host that has PFM - Manager and PFM - RM for Virtual Machine installed, first uninstall all Performance Management programs except PFM - Web Console, and then install PFM - Base, followed by PFM - RM for Virtual Machine.

(e) Other notes

- Before you perform new installation of Performance Management programs in an environment that does not yet have any of these programs installed, make sure that the target folder does not contain any files or folders.
- If you install a Performance Management program while another program (e.g., Windows Event Viewer) that references a Performance Management program, service, or file is running, a message asking the user to restart the system may be displayed. In this case, restart the system as instructed by the message, and then complete the installation process.
- If you install a Performance Management program while another program (e.g., Windows Event Viewer) that references a Performance Management program, service, or file is running, file expansion may fail if there is a disk space shortage, or if you do not have the proper folder permissions. If other programs are running that reference a Performance Management program, service, or file, stop them all, and then reinstall the Performance Management program. If there is a disk space shortage, or if you do not have the proper folder permissions, solve these problems first, and then reinstall the Performance Management program.
- Before you install a Performance Management program, check whether any of the security-related programs are installed. If one is installed, take the necessary steps explained below.
    - For a security-monitoring program

      Either stop the security-monitoring program or change its settings so that it will not prevent installation of Performance Management programs.
    - For a virus-detection program

      We recommend that you first stop the virus-detection program and then install Performance Management programs.

      If a virus-detection program is running while you are trying to install a Performance Management program, it may slow down or prevent installation, or it may result in an incorrect installation.
    - For a process-monitoring program

      Either stop the process-monitoring program or change its settings so that it will not monitor any Performance Management service or process, or any service or process of a shared component.

      If a process-monitoring program starts or stops these programs or services during the installation of a Performance Management program, the installation process may fail.
- You can install PFM - RM for Virtual Machine on one of the virtual machines in a virtual environment, and monitor the virtual environment with the virtual machine installed PFM - RM for Virtual Machine. However, in this operating mode, PFM - RM for Virtual Machine will stop if the virtual environment stops. Therefore, we recommend that you install PFM - RM for Virtual Machine in an environment that is different from the monitored virtual environment.
- Precautions when installing this product as an overwrite installation.

    - An overwrite installation of PFM - RM for Virtual Machine may require a restart.
    - If you open the files of this product or display the Event Viewer, close the files or the dialog to release before installation.
- Before you install this product, make sure you stop any Performance Management programs or services that are running, by using the `jpcspm stop` command. You must also stop any Performance Management services that are running on any logical host that has been set up.
- Attempting to extract the file necessary for installation sometimes fails, depending on the system conditions. If installation fails, please try again. If the problem persists, acquire the following files in the %WINDOWS%\TEMP\HCDINST folder, and contact the system administrator.
    - HCDMAIN.LOG or HCDMAIN#.LOG (# is a numeric character.)
    - HCDINST.LOG or HCDINST#.LOG (# is a numeric character.)
    - P-CC2A2C-5VCL.LOG
- When you upgrade the OS to Windows Server 2012 or later version from Windows Server 2003 or earlier version, uninstall all Performance Management products before upgrading.
- You must restart your computer when prompted to do so at completion of an installation.
- This software is a Hitachi program product that conforms to the disk copy installation of JP1/ServerConductor/Deployment Manager and Hitachi Compute Systems Manager Deployment Manager Plug-in, or the copy functionality using conversion to image files that is provided by a virtual platform. About the disk copy installation, see notes on installing replicated disks in the manual *JP1/Performance Management Planning and Configuration Guide*.

(8) Preparation for collecting data in the event of an error

If a problem occurs, memory dumps, crash dumps, user mode process dumps, and other data might be required. To obtain these dumps when a problem has occurred, use one of the following methods to set up output of these dumps in advance.

(a) Setting method in Windows

- Output settings for user-mode process dump

  By entering the following registry, you can instantly collect user-mode process dump for investigation purposes if an application program terminates abnormally:

  ```
  \\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\Lo
  calDumps
  ```

  Specify the following registry values in the registry key:

    - `DumpFolder : REG_EXPAND_SZ` <*dump-output-destination-folder-name*>

      (You must have permissions to write data to the output destination folder.)
    - `DumpCount : REG_DWORD` <*number-of-dumps-to-be-saved*>
    - `DumpType : REG_DWORD 2`

    > **Important**
    >
    > - Setting up this registry entry enables you to output user-mode process dumps in JP1 as well as other application programs. Be aware of this point when deciding to output user-mode process dump.

- When a user-mode process dump is output, it uses disk space. Therefore, when you decide to output user-mode process dump, make sure that sufficient disk space is allocated to the specified dump output destination folder.

2.1.2 Installation and setup workflow

This subsection explains the flow for installing and setting up PFM - RM for Virtual Machine.

Figure 2–3: Installation and setup flow

For details about the installation and setup procedures for PFM - Manager and PFM - Web Console, see the chapter on installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

Note that setup commands that require information to be entered by the user can be selected to run interactively or non-interactively.

When a command is run interactively, a prompt is issued to the user requesting entry of a required value.

When a command is run non-interactively, the user is not prompted. Instead, the input information required for execution of the command is provided by means of option specifications and definition files. By automating setup tasks through batch processing and remote execution, you can reduce the administrative burden and operational costs.

For details about commands, see the chapter on commands in the manual *JP1/Performance Management Reference*.

2.1.3 Installation procedure

This subsection explains how to install PFM - RM for Virtual Machine programs, including how to install programs from the provided CD-ROM.

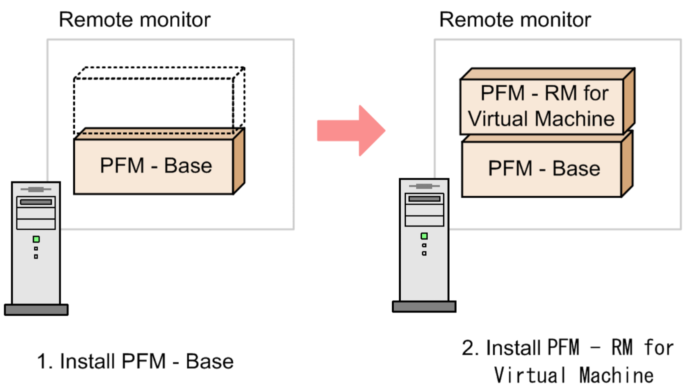
(1) Program installation sequence

First install PFM - Base, and then install PFM - RM for Virtual Machine. You cannot install PFM - RM for Virtual Machine on a host that does not have PFM - Base installed.

To install PFM - RM for Virtual Machine on the same host as PFM - Manager, first install PFM - Manager, followed by PFM - RM for Virtual Machine.

If more than one PFM - RM is installed on the same host, the order of PFM - RM installation does not matter.

Figure 2–4: Program installation sequence



(2) Program installation procedure

You can install Performance Management programs on a Windows host either by using the supplied medium or by using JP1/SD to perform a remote installation. For details about how to use JP1/Software Distribution, see the *JP1/Software Distribution Administrator's Guide Volume 1*, for Windows systems.

Notes:

- If Performance Management programs or services have been activated on the target hosts, stop all of these programs and services. Stop all services on both physical hosts and logical hosts. For details about how to stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.
- If user account control functionality (UAC) is enabled on a Windows host, the User Account Control dialog box might be displayed during installation. If this dialog box is displayed, click the **Continue** button to continue installation, or click the **Cancel** button to cancel installation.

To install Performance Management programs using the supplied medium:

1. Log on to the host on which you intend to install the programs as an administrator.

2. If any Performance Management services are running on the local host, stop all of them.
   The services you are going to stop are the Performance Management services running on both the physical and logical hosts. For details about how to stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

3. Insert the supplied medium into the machine and execute the installer.
   Proceed with installation by following the instructions of the installer that starts.
   The following items, which have been set upon the installation of PFM - Manager or PFM - Base, are displayed for your information:
   - User information
     Enter information such as a user name.
   - Installation folder
     Specify the folder in which to install the Performance Management programs.
     The installation folder is created when you specify a folder in the Select Directory dialog box and click **OK**. If you create a folder by mistake, delete it after installation.
   - Program folder
     Specify the program menu name to be registered in **Start** - **All Programs** in Windows.
     By default, **Performance Management** is registered.

   > **Note**
   >
   > The installation folder for Performance Management programs other than PFM - Web Console can be specified when such a program is first installed on the target host.
   >
   > In the second and subsequent installations of Performance Management programs, the installation folder specified during the first installation of the Performance Management programs will be used as the installation folder.

4. Click the **Install** button to start the installation process.

2.1.4 Setting up PFM - RM for Virtual Machine

This subsection explains the setup for operating PFM - RM for Virtual Machine. The setup procedure depends on the virtual environment to be monitored. The icon VMware, Hyper-V, KVM , VMware , Hyper-V , logical partitioning feature ,, KVM , Docker environment or Podman environment indicates a setup item required for the indicated virtual environment.

The icon Optional indicates either a setup item that may be required depending on the environment that is used, or an optional setup item that is available for changing default settings.

(1) Registering PFM - RM for Virtual Machine VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

To use PFM - Manager and PFM - Web Console to centrally manage PFM - RM for Virtual Machine, you need to register PFM - RM for Virtual Machine in PFM - Manager and PFM - Web Console.

If PFM - RM for Virtual Machine is already registered in PFM - Manager and PFM - Web Console, you do not have to follow the procedure described below. If PFM - RM for Virtual Machine is not registered yet, manually register PFM - RM for Virtual Machine according to the procedure.

You can determine whether manual registration of PFM - RM for Virtual Machine is necessary by referring to the conditions described below.

Manually registering PFM - RM for Virtual Machine in PFM - Manager

When all of the following conditions apply, manually register PFM - RM for Virtual Machine in PFM - Manager:

- The PFM - RM for Virtual Machine to be installed is of a product version that is not specified in the *Release Notes* for PFM - Manager.
- PFM - RM for Virtual Machine is installed on a host other than PFM - Manager.

Manually registering PFM - RM for Virtual Machine in PFM - Web Console

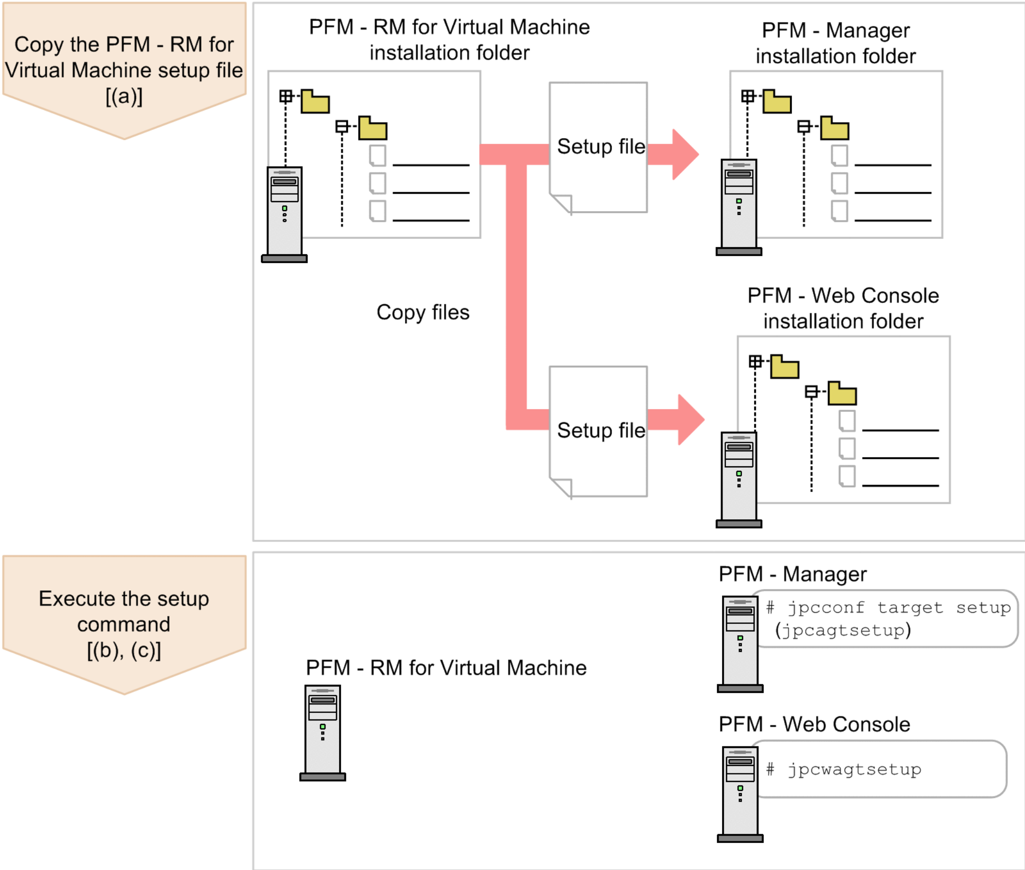
When the following condition applies, manually register PFM - RM for Virtual Machine in PFM - Web Console:

- The PFM - RM for Virtual Machine to be installed is of a product version that is not specified in the *Release Notes* for PFM - Web Console.

If, however, the *Release Notes* for PFM - RM for Virtual Machine state that it is necessary to execute the setup command, execute the setup command.

The figure below shows the flow for registering PFM - RM for Virtual Machine.

Figure 2–5: PFM - RM for Virtual Machine registration flow



Legend:

[ ] : Referenced section(s)

Important

- Register PFM - RM for Virtual Machine before setting up the instance environment.
- If you add the same version of PFM - RM for Virtual Machine to a Performance Management system in which PFM - RM for Virtual Machine information is already registered, there is no need to register PFM - RM for Virtual Machine.
- To install different versions of PFM - RM for Virtual Machine on different hosts, set up the earlier version first, and then set up the later version.
- If you install PFM - RM for Virtual Machine on the same host as PFM - Manager, the `jpcconf agent setup` command is automatically executed. When the message `KAVE05908-I New agent setup ended successfully` is output to the common message log, check the result. If the command was not executed correctly, execute it again. For details about how to execute the command, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.
- During the task of registering the PFM - RM for Virtual Machine information, a folder named `RM VirtualMachine` is created under the **Reports** and **Alarms** tabs in PFM - Web Console. If you had already created a folder or file named `RM VirtualMachine` under the **Reports** tab on your own, change its name before you proceed with registration.

(a) Copying the PFM - RM for Virtual Machine setup files

Copy the setup files located on the host on which you installed PFM - RM for Virtual Machine to the host on which you installed PFM - Manager and PFM - Web Console.

To copy the setup files:

1. If PFM - Web Console is running, stop it.
2. Copy the setup files of PFM - RM for Virtual Machine in the binary mode.
   The table below shows the location where the files are stored and the location to which the files are to be copied.

Table 2–2: Setup files to copy

Copying source (PFM - RM for Virtual Machine setup file)	Copying destination		
	PFM program name	OS	Destination folder
installation-folder`\setup\jpcagt8w.EXE`	PFM - Manager	Windows	*PFM-Manager-installation-folder*`\setup\`
installation-folder`\setup\jpcagt8u.Z`		UNIX	`/opt/jp1pc/setup/`
installation-folder`\setup\jpcagt8w.EXE`	PFM - Web Console	Windows	*PFM-Web-Console-installation-folder*`\setup\`
installation-folder`\setup\jpcagt8u.Z`		UNIX	`/opt/jp1pcwebcon/setup/`

(b) Using the PFM - Manager host to execute the setup command

Use PFM - Manager host to execute the following command for setting up PFM - RM for Virtual Machine:

```
jpcconf agent setup -key RMVM (jpcagtsetup agt8)
```

Although an example of interactive command execution is shown here, the `jpcconf agent setup` command can be also executed non-interactively. For details about the `jpcconf agent setup` command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*.

> **Important**
>
> If you execute the `jpcconf agent setup` command when the Performance Management programs and services of the local host on which you are executing the setup command have not stopped completely, an error may occur. In this case, first make sure that the Performance Management programs and services have stopped completely, and then re-execute the `jpcconf agent setup` command.

After this task has been completed, you can delete the PFM - RM for Virtual Machine setup file located on the PFM - Manager host.

(c) Using the PFM - Web Console host to execute the setup command

Use PFM - Web Console host to execute the following command for setting up PFM - RM for Virtual Machine:

```
jpcwagtsetup
```

After this task has been completed, you can delete the PFM - RM for Virtual Machine setup file located on the PFM - Web Console host.

(2) Setting PFM - RM for Virtual Machine

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Optional

If you want to change the following operations of PFM - RM for Virtual Machine, you need to set it up:

- The collection process starts collecting data based on a reference point that is the time 0 seconds after the service starts. (`UseOffset`)
- PFM - RM for Virtual Machine monitors VMware with Hyper-Threading enabled based on the physical processor cores rather than the logical processor cores. (`UseHTPhysicalClocks`)

The following explains how to perform the setting up PFM - RM for Virtual Machine.

(a) Editing the PFM - RM for Virtual Machine setup file

Edit the following file to change setting up PFM - RM for Virtual Machine:

```
Installation-folder\agt8\agent\jpcagt8cfg.ini
```

(b) Contents of the PFM - RM for Virtual Machine setup file

The following explains the format of the PFM - RM for Virtual Machine setup file.

```
[common]
UseOffset={N | Y}
[vmware]
UseHTPhysicalClocks={N | Y}
```

Description:

`[common]` is the section name. The settings in the `[common]` section are applied to all the monitoring targets.

`UseOffset` is set to disable (N) or enable (Y).

Disable (N) (default value):

The collection process starts collecting data based on a reference point that is the time when the Remote Monitor Collector service starts. The process ignores the `Offset` value specified during setup of the instance environment. When set to disable, the message KAVL20516-W can be output repeatedly, depending on the time when the Remote Monitor Collector service starts. To prevent this, we recommend that you set the `Interval` value (the collection process interval) to half of the `Collection Interval` value (the collection interval of the Remote Monitor Collector service).

Enable (Y):

The collection process starts collecting data based on a zero reference point. Specifically, the process starts in the period specified in the `Offset` value during startup of the instance environment. It starts at a fixed time, without relying on the time the Remote Monitor Collector service starts. Note that if multiple instances are monitored with the same `Offset` value, the load on the PFM - RM for Virtual Machine host can increase significantly. Therefore, if you set this setting to enable, you need to set a different `Offset` value for each instance.

For details about the `UseOffset` setting and the `Offset` setting of instance environments, see *Appendix N. What to Do When the Message KAVL20516-W Is Output Repeatedly*.

`[vmware]` is the section name. The settings in the `[vmware]` section are applied only if the monitoring target is a VMware environment.

`UseHTPhysicalClocks key` is set to disable (N) or enable (Y).

Disable (N) (default value):

Specify N when Hyper-Threading is disabled in all monitored VMware environments (including environments without Hyper-Threading function).

Enable (Y):

Specify Y when a monitored environment has VMware with Hyper-Threading enabled.

If you change this setting, the value of the processor core changes in the Host Status (PI) record or VM Status (PI_VI) record of VMware with Hyper-Threading enabled. For details, see *Appendix M. Fields Affected by Setting the PFM - RM for Virtual Machine*.

(c) Setting UseHTPhysicalClocks

The following table shows the correspondence between VMware with Hyper-Threading and the UseHTPhysicalClocks key setting.

Table 2–3: Correspondence between VMware with Hyper-Threading and the UseHTPhysicalClocks key setting

No.	Hyper-Threading in the VMware environment	Setting UseHTPhysicalClocks key	
		Disable (N) (default value)	Enable (Y)
1	Disabled (including environments without Hyper-Threading function)	Displays the data based on the physical processor cores.	Displays the data based on the physical processor cores. Does not depend on whether Hyper-Threading is disabled or enabled. #2
2	Enabled	Displays the data based on the logical processor cores. #1	

#1

The value of the Used % field in the Host Status (PI) record or the value of the Host Used % field in the VM Status (PI_VI) record depends on the CPU resources used, is about 50 - 70 percent of the maximum.

#2

The value of the Clocks field in Host Status (PI) record is based on the physical processor cores. Therefore, the value of the Used % field in the Host Status (PI) record or the value of the Host Used % field in the VM Status (PI_VI) record depends on the CPU resources used, is 100 percent of the maximum.

Notes:

If the PFM - RM for Virtual Machine setup file does not exist or the file contents are invalid, the system performs same as for the earlier version (for example, `UseHTPhysicalClocks=N`).

When N is specified for the UseHTPhysicalClocks key and monitoring is of VMware with Hyper-Threading enabled, if you do not set thresholds and conditions that match the monitoring target, the alarm will not be detected correctly. In such a case, set thresholds and conditions that match the monitoring target, or specify `UseHTPhysicalClocks=Y`.

Example:

If the CPU resources used are high, no alarm is detected.

If you set a warning condition for when the value of the Used field exceeds 80 %, and specify UseHTPhysicalClocks=N and monitor VMware with Hyper-Threading enabled, the value of the Used field will not exceed 80 % and so no alarm is detected.

(3) Setting up an instance environment VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

To set up an instance environment, you specify instance information. To set up multiple instance environments, you repeat the setup procedure.

Specify the instance information of PFM - RM for Virtual Machine. You specify this information from the PFM - RM host.

The table below shows the types of instance information to specify. Before you begin the setup procedure, check the following information.

Table 2–4: Instance information of PFM - RM for Virtual Machine

Item#1	Explanation	Value that can be set	Default value
`VM_Type`#2	Specifies the type of virtual environment to be monitored.	Character string not exceeding 8 bytes. Specify the following value, depending on the monitoring target: • For VMware: `vmware` • For Hyper-V: `hyperv` • For logical partitioning feature: `virtage` • For KVM: `kvm` • For Docker environment: `docker` • For Podman environment: `podman`	`vmware`
`Interval`	Specifies the collection process execution interval.#3	60 through 3,600 (units: seconds)	300
`Offset`	Specifies the offset value for starting data collection.#3#4	1 through 3,600 (units: seconds)	10
`Std_Category`#5	Specifies whether the collection process outputs the basic information (PI, VI, and VM records) to a temporary performance information file.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Y
`Cpu_Category`#5	Specifies whether the collection process outputs the CPU information (HCI, VCI, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, CPU-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Y
`Memory_Category`#5	Specifies whether the collection process outputs the memory information (HMI, VMI, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, memory-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Y
`Disk_Category`#5	Specifies whether the collection process outputs the disk information (HPDI, VPDI, HLDI, VLDI, VVDI, VDKD, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, disk-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Y

Item#1	Explanation	Value that can be set	Default value
`Network_Category`#5	Specifies whether the collection process outputs the network information (HNI, VNI, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, network-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Y
`HostUserID`#6#7	Specifies the user ID of the PFM - RM host.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(None)
`HostPassword`#6	Specifies the password for the user ID that was specified in `HostUserID`. The characters entered for this item are not displayed on the screen. When you enter a value for this item, the system will ask you to re-enter the value.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(No password)
`HostDomain`#6	Specifies the name of the domain that the PFM - RM host belongs to. This specification is unnecessary if the host belongs to a work group.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(None)
`SSH_Type`	Specifies the type of the SSH client.	Either of the following strings (lowercase): • putty `plink.exe` of PuTTY is used as the SSH client. • windows `ssh.exe` of OpenSSH (which comes with Windows Server 2019) is used as the SSH client.	putty
`SSH_Client`#8	Specifies the absolute path of the execution module (`plink.exe` or `ssh.exe`) of the SSH client (PuTTY or OpenSSH). You do not need to enclose the path name in double quotation marks (`"`) even if it includes spaces.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(None)
`Log_Size`	Specifies the maximum size of a single file for the collected logs.#9	`1-32` (megabytes) A value of 16 or greater is recommended.	`16`
`UseVcpuMax`	Specifies which is to be used as the CPU resource clock frequency: the frequency assigned to the virtual machine or the clock frequency of the physical CPU. To use the frequency assigned to the virtual machine, specify `Y`. To use the frequency of the physical CPU, specify `N`.#10	`{Y|N}` Specification of this item is valid only for monitoring targets that use VMware. For other monitoring targets, specification of this item is ignored.	N

#1

To execute the jpcconf target setup command in non-interactive mode, use this item name as the product-specific label in the definition file. For details about executing the command in non-interactive mode, see the chapter on commands in the manual *JP1/Management Reference*.

#2

Cannot be reset using the jpcconf inst setup command.

#3

Depending on the intervals at which Remote Monitor Collector collects information and at which a temporary performance information file is created, the contents of the file might not be updated, causing the same data to be collected. If this problem occurs, review and correct the settings of these intervals. For details about setting these intervals, see *1.3.1 General procedure for performance data collection tasks*.

#4

If you set the UseOffset value to enable (Y) in *2.1.4 (2) Setting PFM - RM for Virtual Machine*, set the Offset value to a smaller value than the Interval value.

If you set the UseOffset value to disable (N), the Offset value is ignored.

#5

In order to acquire performance data from a temporary performance information file during record collection, collection of the records collected when the Std_Category, Cpu_Category, Memory_Category, Disk_Category, and Network_Category items are enabled takes precedence over other records.

If records collected when any of these items are enabled are unnecessary, disable the relevant items by setting N. Doing so reduces the number of times the monitoring target is accessed. Remember that if you set N for any of the above items, you can no longer collect performance data for the relevant records.

For example, if you set Std_Category to N (disable collection), behavior for PI records is as follows:

- PI records are not saved in the Store database.
- A KAVJS5002-I error message is displayed when an attempt is made to display a real-time report on PI records from PFM - Web Console.
- If an alarm that uses a PI record is bound, the alarm does not function.

For the general procedure for performance data collection tasks, see *1.3.1 General procedure for performance data collection tasks*.

#6

If PFM - RM for Virtual Machine is used in a cluster system, set the same user name and password on both the executing and standby nodes so that the PFM - RM host account can access both nodes.

#7

PFM - RM for Virtual Machine uses the user account specified by HostUserID to start the collection process. If the profile for the user account does not exist, acquisition of performance data might fail.

The profile information for a user account is created at the first logon with the user account.

When you add or change a PFM - RM for Virtual Machine instance environment, you might want to create a new user account and specify it for HostUserID. In such a case, the profile for the user account might have not been created yet. To avoid this, immediately after creating a new user account, log on to Windows with that user account.

#8

If PFM - RM for Virtual Machine is operated in a cluster system, specify it with the file path accessible from both the executing and standby nodes.

#9

For collected log data, a maximum of eight files are created per monitoring target. If the hard disk does not have sufficient free space, a collected log output error will occur. For details about the collected log, see *7.3 Log information*.

#10

The VM Status (PI_VI) record is the only record that is collected if this item is enabled. For details about the fields of this record, see *VM Status (PI_VI)* in *5. Records*.

To check the CPU resource usage of the physical host itself that runs the virtual machine by using the clock frequency of the physical CPU as the CPU resource clock frequency, specify N. You can specify N if, for example, you want to move or stop a virtual machine without exhausting CPU resources.

To check the CPU resource usage of a virtual machine by using the clock frequency assigned to the virtual machine as the CPU resource clock frequency, specify Y. You can specify Y when there are sufficient CPU resources on the physical host that runs the virtual machine.

Important

Even if the setting for the connection environment specified in the instance environment is invalid, the instance environment generation command terminates normally. However, when you begin collecting records in that instance environment, no performance data is collected. For the corrective action to take in such a case, see *7.2.1(1) No performance data is collected even when PFM - RM for Virtual Machine is started*.

The items that need to be specified during instance environment setup depend on the virtual environment to be monitored. The table below shows the items that need to be specified for each virtual environment.

Table 2–5: Instance information items to be specified for each virtual environment

Item	Virtual environment					
	VMware	Hyper-V	KVM	Docker environment	Podman environment	logical partitioning feature
VM_Type	N	Y	Y	Y	Y	Y
Interval	D	D	D	D	D	D
Offset	D	D	D	D	D	D
Std_Category	D	D	D	D	D	D
Cpu_Category	D	D	D	D	D	D
Memory_Category	D	D	D	D	D	D
Disk_Category	D	D	D	D	D	D
Network_Category	D	D	D	D	D	D
HostUserID	Y	Y	Y	Y	Y	N
HostPassword	Y	Y	Y	Y	Y	N
HostDomain	D	D	D	D	D	N
SSH_Type	N	N	Y	N	Y	N
SSH_Client	N	N	Y	N	Y	N
Log_Size	D	D	D	D	D	D

Item	Virtual environment					
	VMware	Hyper-V	KVM	Docker environment	Podman environment	logical partitioning feature
`UseVcpuMax`	D	N	N	N	N	N

Legend:

Y: Specification is required.

D: Specify only to change the default value.

N: Specification is not necessary.

To create an instance environment, use the `jpcconf inst setup` command. The procedure for configuring an instance environment is described below.

This example shows execution in the interactive mode, but you can also execute the `jpcconf inst setup` command in the non-interactive mode. For details about this command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*.

For an instance environment setting example, see *2.6 Examples of setting up an instance environment and monitoring targets*.

To configure an instance environment:

1. Execute the `jpcconf inst setup` command.

   You can specify any instance name. Make sure that you specify an easy-to-identify PFM - RM for the Virtual Machine instance name for convenience of management. The following shows an example of the command that builds an instance named `inst1`:

   ```
   jpcconf inst setup -key RMVM -inst inst1
    (jpcinssetup agt8 -inst inst1)
   ```

   Note that the instance name you specify must consist of only single-byte alphanumeric characters. For details, see the manual *JP1/Performance Management Reference*.

   For details about the `jpcconf inst setup` command, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

2. Set up the instance information of PFM - RM for Virtual Machine.

   Enter the items listed in Table 2-5 as instructed by the command. All items are required. If you choose to use the default value that is displayed for an item to input, simply press the **Enter** key.

Once all of the information has been input, an instance environment is created. To change the instance information that was input during configuration, execute the `jpcconf inst setup` command again to update the instance environment. For details about how to update an instance environment, see *2.4.2 Updating an instance environment*. You can change some of the information that was specified by editing the properties from PFM - Web Console. For details about the information that can be specified, see *E.2 Remote Monitor Collector service property list*.

The instance environment that is created is described below.

- Folder structure of the instance environment

  The instance environment is created under the following folder:

  For a physical host: *installation-folder*`\agt8`

  For a logical host: *environment-folder*#`\jp1pc\agt8`

#

The environment folder is the folder on a shared disk that was specified during creation of the logical host.

The table below shows the folder configuration of the instance environment that is created.

Table 2–6: Folder configuration of the instance environment

Folder name, file name			Explanation
Remote Monitor	*instance-name*	groups	Storage folder for group agents
		log	Storage folder for log files
		targets	Storage folder for remote agents
		GARULES.DAT	Grouping rule file
		grouplist.ini	Group list file
		inssetup.bat	Expansion of the `jpcconf inst setup` command for PFM - RM for Virtual Machine
		jpcagt.ini	Remote Monitor Collector service startup initialization file
		jpcagt.ini.lck	Lock file for a remote monitor collector service startup initialization file (this file is created for each instance)
		jpcagt.ini.model#	Model file for the Remote Monitor Collector service startup initialization file
		plugin.ini	Virtual environment connection definition file
		status.dat	Intermediate file for internal processing
		targetlist.ini	Monitoring-target list file
		tstatuses.dat	Virtual-agent status information file
store	*instance-name*	backup	Backup folder
		dump	Export folder
		import	Import folder
		log	Storage folder for log files
		partial	Partial backup folder
		*.DAT	Data model definition file
		*.DB	Performance data file
		*.IDX	Index file for a performance data file
		*.LCK	Lock file for a performance data file
		jpcsto.ini	Remote Monitor Store service startup initialization file
		jpcsto.ini.model#	Model file for the Remote Monitor Store service startup initialization file
		status.dat	Intermediate file for internal processing

#

Use this file when you need to restore the instance environment to the settings that were in effect when it was created.

- Service ID of the instance environment
  The service ID of the instance environment is as follows.

When the product name display function is enabled

instance-name`[`*host-name*`]<RM VirtualMachine>`

When the product name display function is disabled

`8` *function-ID instance-number instance-name* `[`*host-name*`]`

For PFM - RM for Virtual Machine, the instance name specified by the `jpcconf inst setup` command is displayed for the instance name. For details about service IDs, see the naming rules described in the appendix section of the *JP1/Performance Management Planning and Configuration Guide*. For details about the product name display function, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.

- Windows service name of the instance environment

  The Windows service name of the instance environment is as follows.

  - Remote Monitor Collector service: PFM - RM *for Virtual Machine-instance-name* `[`*logical-host-name*`]`
  - Remote Monitor Store service: PFM - RM *Store for Virtual Machine-instance-name* `[`*logical-host-name*`]`

  For details about the Windows service name, see the naming rules described in the appendix section of the *JP1/ Performance Management Planning and Configuration Guide*.

  For details about the Windows service name when Performance Management is operated on a logical host, see the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

(4) Setting up monitoring targets VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

Set information about the monitoring-target virtual environments for the instance set up in *(3) Setting up an instance environment* above. You can set a maximum of 50 monitoring-target virtual environments. To set multiple monitoring-target virtual environments for one instance, repeat the procedure described in *(3)*. Note, however, that if there are many monitoring-target virtual environments, performance might be reduced, depending on the performance and environment of the machine. If this is the case, reduce the number of monitoring-target virtual environments. We recommend that you perform sufficient testing before you start working in the production environment.

Note on setting up a monitoring-target host:

- The command that generates a monitoring target terminates normally even if incorrect values are specified during setup of the monitoring-target host. However, record collection will not collect performance data if the settings are incorrect. For corrective measures you can take in such a case, see *7.2.1(1) No performance data is collected even when PFM - RM for Virtual Machine is started*.

Setup of a monitoring target is performed on the PFM - RM for Virtual Machine host.

The table below lists and describes the items that can be specified during setup of a monitoring target. Check the settings of these items before starting operation.

Table 2–7: Items that can be specified during setup of a PFM - RM for Virtual Machine monitoring target

Item[#1]	Explanation	Specifiable value	Default	Whether re-updating by the jpcconf target setup command is possible
`Target Host`	Specifies the host name of a monitoring-target physical server. Make sure that the host name can resolve to an IP address.[#2] The specified value is used for collection of performance information and for health checks. For linkage with JP1/IM, the specified value is also used as the event host name.	A character string that is 32 bytes long and consists of only single-byte alphanumeric characters and hyphens (`-`) can be specified. However, a value that begins with a hyphen (`-`) cannot be specified. Make sure that you specify a value that is unique within the instance.[#3]	(None)	Y
`VM_Host`	Specifies the host name of a monitoring-target physical server.[#2] Specify this item if the destination host has already been created with a name that cannot be specified for `Target Host`. In the case of logical partitioning feature, an IP address can be specified. The specified value is used for collection of performance information. If no value is specified for this item, the value specified by `Target Host` is used to perform collection of performance information.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character In the case of VMware, specify a character string consisting of single-byte alphanumeric characters, hyphens (`-`), and periods (`.`). You cannot use a host name that contains underscores (_). The IP address of a host with logical partitioning feature is specified in the following IPv4 format: *xxx*.*xxx*.*xxx*.*xxx* Example: `192.168.41.1`	(None)	Y
`Security`	Specifies whether to use SSL/TLS for communication with the monitoring-target physical server.	`{0|1|2|3}` • For VMware and Docker environment: `0`: SSL/TLS is not used. `0` cannot be used depending on the version of ESXi. Check the release notes for details. `0` cannot be specified in the Docker environment. `1`: SSL/TLS is used[#8]. If there is a problem with the server certificate, performance data continues to be collected, and a warning message is output to the common message log. When the warning message is output, you must change the invalid certificate to a valid one. `2`: SSL/TLS is used[#8]. If there is a problem with the server certificate, performance data continues to	`1`	Y

Item[#1]	Explanation	Specifiable value	Default	Whether re-updating by the jpcconf target setup command is possible
`Security`	Specifies whether to use SSL/TLS for communication with the monitoring-target physical server.	be collected, but a warning message is not output to the common message log. If the default certificate of VMware is used for operation, output of warning messages can be suppressed by specifying the value `2`. If you want operation without using the server certificate of Docker environment, specify the value `2` so that a warning message can be prevented from outputting. `3`: SSL/TLS is used[#8]. If there is a problem with the server certificate, performance data is not collected, and a warning message is output to the common message log. If you want to use a valid certificate for operation and collect performance data only from trusted monitoring targets, specify the value `3`. • For Hyper-V, logical partitioning feature, KVM, and Podman environment: SSL/TLS cannot be used, so ignores the value of this item is always assumed to be `0`.	`1`	Y
`Port`	Specifies the port number that is used for communication with the monitoring-target physical server.	`0-65535` • For VMware: If `0` is specified, `443` (HTTPS default port number) • For logical partitioning feature: The fixed port number `623` is used. This port number cannot be changed. Even if a value other than `623` is specified for this item, the port number `623` will be assumed. • For KVM: If `0` is specified, `22` (SSH default port number) will be assumed. • For Docker environment: Specify the port number used for connection. If `0` is specified, performance data is not collected and a warning message is output to the common message log. • For Podman environment: If `0` is specified, `22` (SSH default port number) will be assumed.	`0`	Y

Item#1	Explanation	Specifiable value	Default	Whether re-updating by the jpcconf target setup command is possible
`UserID`	Specifies the user ID that is used to connect to the monitoring-target physical server.#4, #5	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(None)	Y
`Password`	Specifies the password that is used to connect to the monitoring-target physical server. The character string entered for this item is not displayed. This item requires you to enter a value twice. If the monitoring target uses VMware and you specify a password that includes special characters, make sure that you specify a sanitized string.	Character string not exceeding 256 bytes.#6 Note that the following character cannot be used: • Tab character	(None)	Y
`Domain`	Specifies the name of the domain that the monitoring-target physical server belongs to. If the server belongs to a work group, you do not need to specify this item. If the monitoring target uses VMware, KVM, logical partitioning feature ,Docker environment or Podman environment, you do not need to specify this item.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(None)	Y
`Private_Key_File`	Specifies the absolute path name of the private key file that is used for SSH public key encryption. Specify the path name of the private key file created by the procedure described in *2.5.7(4) Settings related to SSH connection*. You do not need to enclose the path name in double quotation marks (`"`) even if it includes spaces.#7	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	(None)	Y

Legend:

Y: Re-updating is possible.

#1

If the `jpcconf target setup` command is executed in non-interactive mode, use this item name as the product-specific label in the definition file. For details about executing the command in non-interactive mode, see the chapter on commands in the manual *JP1/Performance Management Reference*.

#2

The first thing you must do before you can perform collection of performance information and do health checks is to ensure that the PFM - RM host name can resolve to an IP address. To use the JP1/IM linkage facility, the JP1/IM host name must be resolvable to an IP address.

#3

The string All, which is a reserved word for group agents, cannot be used.

#4

If the monitoring-target virtual environment uses VMware, the account you specify must have a *Read-only* or a higher VMware role. For details about how to specify the role settings, see the VMware documentation.

If the monitoring-target virtual environment uses Hyper-V, for the permission of the account you specify, see *2.5.2(1)(a) Environment settings required for connection*.

If the monitoring-target virtual environment uses KVM, always log in as superuser.

If the monitoring-target virtual environment uses Podman environment, always log in as superuser. The container (rootless container) of a general user cannot be monitored.

#5

If the monitoring-target virtual environment uses KVM, use bash, bsh, or ksh as the login shell for the user you specify.

If the monitoring-target virtual environment uses Podman environment, use bash as the login shell for the user you specify.

#6

If the monitoring-target virtual environment uses VMware and any of the symbols listed in the following table are used in the password on the VMware side, replace them with symbol entities when specifying a value for Password.

Symbol	Characters entered
<	<
>	>
&	&
'	'
"	"

For example, if the string you want to specify for Password is abc<def>xyz, enter abc<def>xyz.

#7

The private key file is accessed every time record collection is performed. Make sure that the private key file at the path you specify can be accessed during record collection.

#8

If there is a problem with the client certificate, the connection is rejected on the Docker environment side. Thus, performance data cannot be collected.

The items that must be specified during setup of a monitoring target differ depending on the monitoring-target virtual environment. The following table shows the required setting items for each virtual environment.

Table 2–8: Required setting items for each virtual environment

Item	Virtual environment					
	VMware	Hyper-V	KVM	Docker environment	Podman environment	logical partitioning feature
`Target Host`	Y	Y	Y	Y	Y	Y
`VM_Host`	D	D	D	D	D	D
`Security`	D	N	N	D	N	N
`Port`	D	N	D	Y	D	N
`UserID`	Y	Y	Y	N	Y	N
`Password`	Y	Y	N	N	N	N
`Domain`	N	Y	N	N	N	N
`Private_Key_File`	N	N	Y	N	Y	N

Legend:

Y: Specification is required.

D: Specify only to change the default value.

N: Specification is not necessary.

A monitoring target is set up by using the `jpcconf target setup` command. To set up a monitoring target:

1. Execute the `jpcconf target setup` command.

   For PFM - RM for Virtual Machine, we recommend that you specify the host name of the physical server as the monitoring target name.

   The following shows an example of executing a command when setting the host named `targethost1` as the monitoring target of the instance named `inst1`.

   ```
   jpcconf target setup -key RMVM -inst inst1 -target targethost1
   ```

   For details about the `jpcconf target setup` command, see the chapter on commands in the manual *JP1/ Performance Management Reference*.

2. Set up the PFM - RM for Virtual Machine monitoring target.

   Specify settings of the monitoring target as the command instructs. For details about these settings, see Table 2-8. After specifying a value for a setting item, press the return key to set the value. If you accept the initial value as is, only press the return key.

The following shows an example of specifying settings when the monitoring target is a VMware environment.

Conditions of the monitoring target that you want to set up:

- Host name of the physical server: `targethost1`
- User: `user1`
- Password: `pass1`
- Domain: `domain1`

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key RMVM -inst
inst1 -target targethost1
Target Host [] : targethost1 <Enter>
```

```
VM_Host [] : <Enter>
Security [1] : <Enter>
Port [0] :<Enter>
UserID:user1 <Enter>
Password :pass1#1<Enter>
Re-enter :pass1#1<Enter>
Domain [] :<Enter>
Private_Key_File [] :<Enter>
KAVE05361-I The monitoring target is now being added. (servicekey#2=RMVM,i
nst=inst1, target=targethost1)
KAVE05362-I The monitoring target has been added. (servicekey#2=RMVM,inst=
inst1, target=targethost1)
```

#1

You are prompted to enter a password twice. The string entered as the password is not displayed.

#2

If the product name display function of PFM - Manager is disabled, `agt8` is displayed for `servicekey`.

After you have specified all settings, the monitoring-target environment is set up. The following table shows the folders created in the environment.

Table 2–9: Folders created in the monitoring-target environment

No.	Folder	File name	Description
1	*PFM-RM-for-Virtual-Machine-installation-folder*# `\agt8\agent\`*instance-name*`\targets`	*monitoring-target-name*`.ini`	Monitoring-target settings file
2		*monitoring-target-name*`.ini.model`	Model file for the monitoring-target settings file
3	*PFM-RM-for-Virtual-Machine-installation-folder*# `\agt8\agent\`*instance-name*`\targets` `\`*monitoring-target-name*	--	Work folder for the monitoring target

Legend:

--: Not applicable

#

If you use PFM - RM for Virtual Machine on a logical host, replace *PFM-RM-for-Virtual-Machine-installation-folder* with the *environment-folder*`\jp1pc`.

Note that a service ID is added during setup of the monitoring target.

Added service ID:

- Remote Agent service

  `8A`*ab*`[`*monitoring-target-name*`@`*host-name*`]` (*a*: *instance-number*, *b*: *instance-name*)

  For *b* (*instance-name*) and *monitoring-target-name*, specify the corresponding values that are specified in the `jpcconf target setup` command.

  For example, if the PFM - RM host name is `host1`, the instance name is `inst1`, and the monitoring target name is `targethost1`, the service ID is as follows:

  `8A1inst1[targethost1@host1]`

For details about service IDs, see the naming conventions in the appendix of the *JP1/Performance Management Planning and Configuration Guide*.

To change the information about a monitoring target, re-execute the `jpcconf target setup` command. For details about updating a monitoring target, see *2.4.3 Updating monitoring targets*.

Some of the items that are set can be changed by editing the properties of PFM - Web Console. For details about the items that can be changed by editing these properties, see *E.3 Remote agent and group agent property list*.

(5) Settings for each monitoring target Optional

(a) For VMware VMware

You use SSL/TLS to communicate with VMware, you must setting of encrypted communication and embed a CA certificate. For details, see *2.5.1 For VMware*.

Use user defined records to monitor performance information that is not retrieved by PFM - RM for Virtual Machine. For details about the user defined records, *2.5.1(6) User defined records*.

(b) For Hyper-V Hyper-V

You must set up WMI on the PFM - RM host and the monitored host. For details, see *2.5.2 For Hyper-V*.

(c) For KVM KVM

The host must be connected to an SSH server in order to collect performance data from the host. For details about SSH settings, see *2.5.3 For KVM*.

(d) For Docker environment Docker environment

To use SSL/TLS to communicate with Docker environment, the CA certificate and client certificate must be embedded. For details, see *2.5.4 For Docker environment*.

(e) For Podman environment Podman environment

The host must be connected to an SSH server in order to collect performance data from the host. For details about SSH settings, see *2.5.5 For Podman environment*.

(f) For logical partitioning feature logical partitioning feature

Specify the IP address of the machine on which the monitoring agent is installed for that environment. For details, see *2.5.6 For logical partitioning feature*.

(6) Network settings VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Optional

This is necessary only when you want to change the settings according to the network configuration in which Performance Management is used.

In the network settings, you can specify the following two items:

- IP address
  Set an IP address when you use Performance Management in a network that is connected to multiple LANs. To set multiple IP addresses, define host names and IP addresses in the `jpchosts` file. Use the same `jpchosts` file throughout the entire Performance Management system.

For details about how to set IP addresses, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

- Port number

  You can set the port number to be used by Performance Management. To avoid confusion during operations, use the same port number and service name throughout the entire Performance Management system.

  For details about how to set port numbers, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

(7) Changing the log file size VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Optional

The operation status of Performance Management is output to a log file specific to Performance Management. This log file is called the *common message log*. This setting is necessary only when you want to change the size of this file.

For details, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

(8) Changing the performance data storage destination VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Optional

This setting is necessary only when you want to change the folder at the save destination, the backup destination, or the export destination for the database that stores the performance data managed by PFM - RM for Virtual Machine.

By default, performance data is saved in the following locations:

- Save destination: *installation-folder*`\agt8\store\`*instance-name*`\`
- Backup destination: *installation-folder*`\agt8\store\`*instance-name*`\backup\`
- Partial backup destination: *installation-folder*`\agt8\store\`*instance-name*`\partial\`
- Export destination: *installation-folder*`\agt8\store\`*instance-name*`\dump\`
- Import destination: *installation-folder*`\agt8\store\`*instance-name*`\import\`

Important

For the default save destinations when Performance Management is operated on a logical host, replace *installation-folder* with *environment-folder*`\jp1pc`.

For details, see *2.4.1 Changing the performance data storage destination*.

(9) Setting up PFM - Manager at the connection destination of PFM - RM for Virtual Machine VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

On the host on which PFM - RM for Virtual Machine is installed, set up the PFM - Manager that manages the PFM - RM for Virtual Machine. To set up PFM - Manager at the connection destination, use the `jpcconf mgrhost define` command.

> **Important**
>
> - Even when multiple PFM - RMs are installed on the same host, you can specify only one PFM - Manager at the connection destination. You cannot set up a different PFM - Manager for each PFM - RM.
> - When PFM - RM for Virtual Machine and PFM - Manager are installed on the same host, the PFM - Manager of the local host becomes the PFM - Manager at the connection destination. In this case, you cannot change the PFM - Manager at the connection destination to another PFM - Manager.

To set up PFM - Manager at the connection destination:

1. Stop Performance Management programs and services.

   Before you perform a setup operation, if Performance Management programs and services have been activated on the local host, stop all of them. For details about how to stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

   If the Performance Management programs and services are active during the execution of the `jpcconf mgrhost define` command, a message asking the user whether to stop these programs and services is displayed.

2. Specify the host name of the PFM - Manager host at the connection destination, and then execute the `jpcconf mgrhost define` command.

   For example, if the PFM - Manager at the connection destination is on the host `host01`, specify the following:

   ```
   jpcconf mgrhost define -host host01
    (jpcnshostname -s host01)
   ```

(10) Setting the action log output

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Optional

This setting is necessary if you want to output an action log when you start or stop a PFM service, or when you change the connection status with PFM - Manager. An *action log* contains history information that is output in conjunction with the alarm function that monitors items such as the system load threshold.

For details about the setting procedure, see *I. Outputting Action Log Data*.

2.2 Uninstallation and unsetup

This section explains how to uninstall and unsetup PFM - RM for Virtual Machine.

2.2.1 Before uninstallation and unsetup

Note the following points before you perform uninstallation and unsetup of PFM - RM for Virtual Machine.

(1) Note about the OS user permissions required for uninstallation

- To uninstall PFM - RM for Virtual Machine, you must perform the procedure as a user with Administrator permissions.

(2) Note about the network

- Even when you uninstall Performance Management programs, the port numbers defined in the `services` file are not deleted.

(3) Notes about programs

- If you uninstall a Performance Management program while another program (e.g., Windows Event Viewer) that references a Performance Management program, service, or file is running, files or folders might remain. In this case, manually delete all the files and folders under the installation folder.
- If you uninstall a Performance Management program while another program (e.g., Windows Event Viewer) that references a Performance Management program, service, or file is running, a message asking the user to restart the system may be displayed. In this case, restart the system as instructed by the message, and then complete the uninstallation process.
- If both PFM - Base and PFM - RM for Virtual Machine have been installed on a host, you must first uninstall PFM - RM for Virtual Machine before you can uninstall PFM - Base. In this case, uninstall PFM - RM for Virtual Machine first, followed by PFM - Base. Likewise, if both PFM - Manager and PFM - RM for Virtual Machine have been installed on a host, you must first uninstall PFM - RM for Virtual Machine before you can uninstall PFM - Manager. In this case, uninstall PFM - RM for Virtual Machine first, followed by PFM - Manager.

(4) Note about services

Uninstalling PFM - RM for Virtual Machine alone does not delete the service information that can be displayed using the `jpctool service list` command. In this case, use the `jpctool service delete` command to delete the service information.

To get the PFM - Web Console host to recognize that an instance environment has been deleted, execute the `jpctool service sync` command. This synchronizes the agent information on the PFM - Manager host and the PFM - Web Console host.

For details, see the section about deleting services in the chapter that explains installation and setup in the *JP1/ Performance Management Planning and Configuration Guide*.

(5) Other notes

- To uninstall Performance Management programs from a host on which PFM - Web Console has been installed, close all browser windows before you perform uninstallation.

- If message "KAVE05085-E The instance environment could not be deleted." is displayed, Delete instance of logical partitioning feature, KVM or Podman environment after executing following steps.
    1. Log in to the PFM - RM for Virtual Machine host with Administrators permissions.
    2. Input the following command to command prompt.
       - *In logical partitioning feature*
       `tasklist|find "HvmSh.exe"`
       - *In KVM or Podman environment*
       When putty is specified for SSH_Type
       `tasklist|find "plink.exe"`
       When windows is specified for SSH_Type
       `tasklist|find "ssh.exe"`
       (Note to case sensitive.)
    3. If the following message is displayed, input the command same as the step 2 more than 30 seconds later.
       - *In logical partitioning feature*
       `HvmSh.exe 5120 Services 0 112K`
       - *In KVM or Podman environment*
       When putty is specified for SSH_Type
       `plink.exe 5120 Services 0 112K`
       When windows is specified for SSH_Type
       `ssh.exe 5120 Services 0 112K`
    4. If the outputs step 2 and 3 displayed process "`HvmSh.exe`", "`plink.exe`", or "`ssh.exe`" which have same process ID, take note the process ID of incomplete `HvmSh.exe` process, `plink.exe` process, or `ssh.exe` process (Process ID is in 2nd column. In above example, "5120"is process ID.)
    5. Terminate the HvmSh.exe process by input the following message to command prompt with specifying the process ID noted in step 4.
       `taskkill /F /PID <process ID>`
    6. Execute the command of step 2 and make sure that message of step 3 is not displayed.

2.2.2 Unsetup procedure

This subsection explains how to unset up PFM - RM for Virtual Machine. The unsetup procedure depends on the virtual environment to be monitored. The icon VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment indicates an unsetup item required for the indicated virtual environment.

(1) Deleting a monitoring target VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

To delete a monitoring target, you need to first confirm the name of the monitoring target and then delete the monitoring target on the PFM - RM host. To confirm the name of the monitoring target that you want to delete, use the `jpcconf target list` command. To delete the monitoring target, use the `jpcconf target unsetup` command.

You do not need to stop the PFM - RM for Virtual Machine service when deleting a monitoring target.

To delete a monitoring target:

1. Check the name of the monitoring target that you want to delete.

   Execute the `jpcconf target list` command by specifying the service key and the instance name that identify the target PFM - RM for Virtual Machine instance.

   ```
   jpcconf target list -key RMVM -inst inst1
   ```

   The names of hosts monitored by the instance are displayed.

   ```
   Targets:
   targethost1
   targethost2
   Groups:
   All
   ```

2. Delete the objective monitoring target.

   Execute the `jpcconf target unsetup` command by specifying the service key, instance name, and monitoring target name that identify the target PFM - RM for Virtual Machine instance.

   ```
   jpcconf target unsetup -key RMVM -inst inst1 -target targethost1
   ```

   When the `jpcconf target unset` command terminates normally, `targethost1` is no longer monitored.

Notes:

- If you use the `jpcconf target unsetup` command to delete a monitoring target, service information is automatically deleted from PFM - Manager. Therefore, you do not need to execute the `jpctool service delete` command. The following describes when service information is deleted.

  • If the `jpcconf target unsetup` command is executed while the services of PFM - Manager and the target PFM - RM for Virtual Machine instance are running, the instance requests PFM - Manager to delete service information. Service information is deleted at this time.

  • If the `jpcconf target unsetup` command is executed while the service of PFM - Manager or the target PFM - RM for Virtual Machine instance is not running, service information is deleted at the following time: when the service of PFM - RM for Virtual Machine is restarted and connected to PFM - Manager.

- For the PFM - Web Console host to recognize that an instance environment has been deleted, execute the `jpctool service sync` command to synchronize the agent information on the PFM - Manager host and the PFM - Web Console host.

  Note that the folder and files shown below are not deleted when a monitoring target is deleted. Delete these folder and files manually.

  installation-folder#`\agt8\agent\`*instance-name*`\targets\`*monitoring-target-name*

  installation-folder#`\agt8\agent\`*instance-name*`\log\`*VM_Type_monitoring-target-nameN*

#

If you use PFM - RM for Virtual Machine on a logical host, replace *installation-folder* with the *environment-folder*`\jp1pc`.

(2) Unsetup of the instance environment VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

To unset up the instance environment, first confirm the instance name and then delete the instance environment. Delete the instance environment from the PFM - RM host. To confirm the instance name, use the `jpcconf inst list` command. In addition, to delete an instance environment that has been created, use the `jpcconf inst unsetup` command.

To delete an instance environment:

1. Confirm the instance name.

   Specify the service key indicating PFM - RM for Virtual Machine and then execute the `jpcconf inst list` command.

   ```
   jpcconf inst list -key RMVM
    (jpcinslist agt8)
   ```

   If the instance name that was set is `inst1`, `inst1` is displayed.

2. If any PFM - RM for Virtual Machine services in the instance environment are active, stop them.

   For details about how to stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

3. Delete the instance environment.

   Specify the service key indicating the PFM - RM for Virtual Machine and the instance name, and then execute the `jpcconf inst unsetup` command. If the instance name that was set is `inst1`, specify the following:

   ```
   jpcconf inst unsetup -key RMVM -inst inst1
    (jpcinsunsetup agt8 -inst inst1)
   ```

   When the `jpcconf inst unsetup` command terminates normally, the folders, service IDs, and Windows services that were created as the instance environment are deleted.

   Important

   Even when you unset up the instance environment, the service information that can be displayed using the `jpctool service list` command is not deleted. You use the `jpctool service delete` command to delete the service information.

   To get the PFM - Web Console host to recognize that an instance environment has been deleted, execute the `jpctool service sync` command. This synchronizes the agent information on the PFM - Manager host and the PFM - Web Console host.

   A specification example follows.

   - Instance name: `inst1`
   - Host name: `host03`
   - Remote Monitor Collector service service ID: `8A1inst1[host03]`
   - Remote Monitor Store service service ID: `8S1inst1[host03]`
   - Group Agent service service ID: `8Ainst1[All@host03]`

     ```
     jpctool service delete -id 8?1inst1[host03] -host host03
     jpctool service delete -id 8?1inst1[*@host03] -host host03
     ```

   For details about commands, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

(3) Deleting settings from PFM - Manager VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment

From PFM - Web Console, log into PFM - Manager and delete the definitions related to the PFM - RM for Virtual Machine instance that you want to unset up.

To delete settings from PFM - Manager:

1. Delete the agent from PFM - Web Console.
2. Delete the agent information of PFM - Manager.
   For example, to delete the service information of the PFM - RM for Virtual Machine instance of the host `targethost1`, specify the following and execute the command:
   ```
   jpctool service delete -id service-ID -host targethost1
   ```
   For *service-ID*, specify the service ID of the agent you want to delete.
3. Restart the PFM - Manager service.
   For details about how to start services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.
4. Apply the service information of the PFM - Manager host.
   To get the PFM - Web Console host to recognize that the service information has been deleted, synchronize the agent information on the PFM - Manager host with the information on the PFM - Web Console host. To synchronize agent information, use the `jpctool service sync` command.

(4) Deleting the LPAR Manager management command for logical partitioning feature logical partitioning feature

Delete the `HvmSh` command (`HvmSh.exe`) from the management tool for logical partitioning feature, which has been copied to the following folder during setup:

```
installation-folder\agt8\plugin\jpcagt5virtage.d\
```

2.2.3 Uninstallation procedure

This subsection explains how to uninstall PFM - RM for Virtual Machine.

To uninstall PFM - RM for Virtual Machine:

1. Log on to the host from which you want to uninstall PFM - RM for Virtual Machine as a user with Administrator permissions.
2. On the local host, stop Performance Management programs and services.
   Display the service information to make sure services are not running.
   If any Performance Management programs and services are active on the local host, stop all of them. You must stop all services on both the physical host and on the logical hosts. For details about how to display services or stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

3. Select the Performance Management programs you want to uninstall.

   From Windows **Control Panel**, choose **Add or Remove Programs**#, and select the Performance Management programs to uninstall.

   #: The names might vary depending on the Windows version.

4. Choose **Remove** and click **OK**.

   The selected programs are uninstalled.

Note:

The WMI connection settings might be changed (*2.5.2(1) Setting up WMI connection*). If these settings have been changed, we recommend that you restore the initial settings.

The private key and public key that are used for SSH public key authentication might be set (*2.5.7 SSH connection settings*). If these keys have been set but are not necessary, delete them. If PuTTY or OpenSSH is unnecessary, uninstall it.

Precautions regarding uninstallation in a Windows environment

- If user account control functionality (UAC) is enabled on the operating system, the User Account Control dialog box might be displayed during uninstallation. If this dialog box is displayed, click the **Continue** button to continue uninstallation, or click the **Cancel** button to cancel uninstallation.
- If you uninstall this product in either of the following cases, files or folders might remain after the uninstallation:

  - When a Performance Management program or service is running.

  - When a file or folder under the *installation-folder*\agt8 is being accessed.

  If files or folders remain, delete the *installation-folder*\agt8 and all files and folders in it. When a logical host environment has been set up, also delete, from the shared disk, the *environment-folder (directory)* \jp1pc\agt8 and all the files and folders in it.

2.3 Changing the PFM - RM for Virtual Machine system configuration

If the network configuration of the monitored system or its host name changes, you may need to modify the system configuration of PFM - RM for Virtual Machine.

When you modify the system configuration of PFM - RM for Virtual Machine, you must also modify the settings of PFM - Manager and PFM - Web Console. For details about how to modify the system configuration of Performance Management, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*. Note that when you change the physical host name or alias name, some PFM - RMs may require specific additional tasks. However, in the case of PFM - RM for Virtual Machine, no specific additional tasks are required.

To change the setting for a running PFM - RM for Virtual Machine system, perform the procedure described in *2.1.4(2) Setting PFM - RM for Virtual Machine*, and then restart all services in the changed instances environment. After restarting the services, the settings are updated.

Also, the performance data changes after the settings are changed, so you need to check the thresholds of the fields as described in *M. Fields Affected by Setting the PFM - RM for Virtual Machine*.

2.4 Changing the PFM - RM for Virtual Machine operation method

If there are changes in the way the collected operation monitoring data is used, you may need to modify the PFM - RM for Virtual Machine operating procedures. For details about how to modify operating procedures for the entire Performance Management system, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

2.4.1 Changing the performance data storage destination

Performance data collected by PFM - RM for Virtual Machine is managed in the Store database of the Remote Monitor Store service of PFM - RM for Virtual Machine. This subsection explains how to change the performance data storage destination.

To change the data storage folders listed below that collect performance data managed in the Store database, use the `jpcconf db define` command to specify a new folder. If you need to save the performance data that was collected before the Store database storage folders were changed, use the `-move` option of the `jpcconf db define` command. For details about the `jpcconf db define` command, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

- Save folder
- Backup folder
- Partial backup folder
- Export folder
- Import folder

The table below describes the option names and the ranges of values that can be set in the `jpcconf db define` command.

Table 2–10: Items that can be set in the command for changing the performance data storage destination

Explanation	Option name	Value that can be set#1	Default value#2
Performance data creation folder	`sd`	Path name of 1-214 bytes	*installation-folder*`\agt8\store\`*instance-name*
Performance data backup folder	`bd`	Path name of 1-211 bytes	*installation-folder*`\agt8\store\`*instance-name*`\backup`
Performance data partial backup folder	`pbd`	Path name of 1-214 bytes	*installation-folder*`\agt8\store\`*instance-name*`\partial`
Largest generation number for saving performance data	`bs`	1-9	`5`
Performance data export folder	`dd`	Path name of 1-127 bytes	*installation-folder*`\agt8\store\`*instance-name*`\dump`
Performance data import folder	`id`	Path name of 1-222 bytes	*installation-folder*`\agt8\store\`*instance-name*`\import`

#1

Specify the folder name using a relative or absolute path from the default Store database storage folder (*installation-folder*`\agt8\store`).

#2

For the default values when Performance Management is operated on a logical host, replace *installation-folder* with *environment-folder*`\jp1pc`.

2.4.2 Updating an instance environment

To update an instance environment, confirm the instance name and then update the instance information. You specify instance information from the PFM - RM host.

Use the following table to confirm the information you need to update.

Table 2–11: PFM - RM for Virtual Machine instance information

Item	Explanation	Value that can be set	Default value
`VM_Type`	Specifies the type of virtual environment to be monitored.	Cannot be changed	Previous setting
`Interval`	Specifies the collection process execution interval.	60 through 3,600 (units: seconds)	Previous setting
`Offset`	Specifies the offset value for starting data collection.	1 through 3,600 (units: seconds)	Previous setting
`Std_Category`	Specifies whether the collection process outputs the basic information (PI, VI, and VM records) to a temporary performance information file.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Previous setting
`Cpu_Category`	Specifies whether the collection process outputs the CPU information (HCI, VCI, PODI, POCI, and PODD records) to a temporary performance information file. # For the PODI and POCI records, CPU-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Previous setting
`Memory_Category`	Specifies whether the collection process outputs the memory information (HMI, VMI, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, memory-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Previous setting
`Disk_Category`	Specifies whether the collection process outputs the disk information (HPDI, VPDI, HLDI, VLDI, VDKD, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, disk-related fields are output.	Either of the following values: • `Y`: Outputs the information. • `N`: Does not output the information.	Previous setting

Item	Explanation	Value that can be set	Default value
Network_Category	Specifies whether the collection process outputs the network information (HNI, VNI, PODI, POCI, and PODD records) to a temporary performance information file. #: For the PODI and POCI records, network-related fields are output.	Either of the following values: • Y: Outputs the information. • N: Does not output the information.	Previous setting
HostUserID	Specifies the user ID of the PFM - RM host.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
HostPassword	Specifies the password for the user ID that was specified in HostUserID. The characters entered for this item are not displayed on the screen. When you enter a value for this item, the system will ask you to re-enter the value.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
HostDomain	Specifies the name of the domain that the PFM - RM host belongs to. This specification is unnecessary if the host belongs to a work group.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
SSH_Type	Specifies the type of the SSH client.	Either of the following strings (lowercase): • putty plink.exe of PuTTY is used as the SSH client. • windows ssh.exe of OpenSSH (which comes with Windows Server 2019) is used as the SSH client.	Previous setting
SSH_Client	Specifies the absolute path of the execution module (plink.exe or ssh.exe) of the SSH client (PuTTY or OpenSSH). . You do not need to enclose the path name in double quotation marks (") even if it includes spaces.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
Log_Size	Specifies the maximum size of a file for the collected log.	1-32 (megabytes) A value of 16 or greater is recommended.	Previous setting
UseVcpuMax	Specifies which is to be used as the CPU resource clock frequency: the frequency assigned to the virtual machine or the clock frequency of the physical CPU. To use the frequency assigned to the virtual machine, specify Y. To use the frequency of the physical CPU, specify N.	{Y|N}	Previous setting

The items that need to be specified during instance environment setup depend on the virtual environment to be monitored. This also applies to the items that can be updated. The table below shows whether the items can be updated for each virtual environment.

Table 2–12: Instance information items that can be updated for each virtual environment

Item	Virtual environment					
	VMware	Hyper-V	KVM	Docker environment	Podman environment	logical partitioning feature
`VM_Type`	N	N	N	N	N	N
`Interval`	Y	Y	Y	Y	Y	Y
`Offset`	Y	Y	Y	Y	Y	Y
`Std_Category`	Y	Y	Y	Y	Y	Y
`Cpu_Category`	Y	Y	Y	Y	Y	Y
`Memory_Category`	Y	Y	Y	Y	Y	Y
`Disk_Category`	Y	Y	Y	Y	Y	Y
`Network_Category`	Y	Y	Y	Y	Y	Y
`HostUserID`	Y	Y	Y	Y	Y	--
`HostPassword`	Y	Y	Y	Y	Y	--
`HostDomain`	Y	Y	Y	Y	Y	--
`SSH_Type`	--	--	Y	--	Y	--
`SSH_Client`	--	--	Y	--	Y	--
`Log_Size`	Y	Y	Y	Y	Y	Y
`UseVcpuMax`	Y	--	--	--	--	--

Legend:

Y: Can be updated

N: Cannot be updated

--: Updating is not necessary because the setting has no effect on the PFM - RM for Virtual Machine operation

To confirm an instance name, use the `jpcconf inst list` command. To update the instance environment, use the `jpcconf inst setup` command.

The procedure for updating the instance environment is as follows. To update multiple instance environments, repeat this procedure.

To update the instance environment:

1. Confirm the instance name.
   Specify the service key indicating PFM - RM for Virtual Machine, and then execute the `jpcconf inst list` command.
   ```
   jpcconf inst list -key RMVM
   ```
   If the instance name that was set is `inst1`, `inst1` is displayed.

2. If any PFM - RM for Virtual Machine services are active for the instance environment you are updating, stop them.
   For details about how to stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

If the instance environment you are updating is active when you try to execute the `jpcconf inst setup` command, a confirmation message is displayed and you can stop the service. If you stop the service, the updating process continues. If you do not stop the service, the updating process stops.

3. Specify both the service key indicating PFM - RM for Virtual Machine and the instance name, and then execute the `jpcconf inst setup` command.

   To update an instance environment in which the instance name is `inst1`, specify the following and then execute the command:

   ```
   jpcconf inst setup -key RMVM -inst inst1
   ```

4. Update the instance information of PFM - RM for Virtual Machine.

   Enter the items listed in Table 2-12 as instructed by the command. The current settings are displayed. If you choose not to change the displayed value, simply press the **Enter** key. Once all information has been input, the instance environment is updated.

5. Restart the service(s) of the updated instance environment.

   For details about how to start services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

For details about commands, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

2.4.3 Updating monitoring targets

To update a monitoring target, check the name of the monitoring target, and then update the information about the monitoring target. Set the new information about the monitoring target on the PFM - RM host.

The table below lists and describes the information items that can be updated. Check the items that you want to change beforehand.

Table 2–13: Setting items for a monitoring target of PFM - RM for Virtual Machine

Item	Explanation	Value that can be set	Default value
`Target Host`	Specifies the host name of a monitoring-target physical server. Make sure that the host name can resolve to an IP address.#1 The specified value is used for collection of performance information and for health checks. For linkage with JP1/IM, the specified value is also used as the event host name.	A character string that is 32 bytes long and consists of only single-byte alphanumeric characters and hyphens (`-`) can be specified. However, a value that begins with a hyphen (`-`) cannot be specified. Make sure that you specify a value that is unique within the instance.#2	Previous setting
`VM_Host`	Specifies the host name of a monitoring-target physical server. Specify this item if the destination host has already been created with a name that cannot be specified for `Target Host`. In the case of logical partitioning feature, an IP address can be specified. The specified value is used for collection of performance information. If no value is specified for this item, the value	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character In the case of VMware, specify a character string consisting of single-byte alphanumeric characters, hyphens (`-`), and periods (`.`). You cannot use a host name that contains underscores (`_`). The IP address of a host with logical partitioning feature is specified in the following IPv4 format:	Previous setting

Item	Explanation	Value that can be set	Default value
VM_Host	specified by Target Host is used to perform collection of performance information.	*xxx*.*xxx*.*xxx*.*xxx* Example: 192.168.41.1	Previous setting
Security	Specifies whether to use SSL for communication with the monitoring-target physical server.	{0|1|2|3} • For VMware and Docker environment: 0: SSL/TLS is not used. 0 cannot be specified, must use SSL/TLS connection. 1: SSL/TLS is used. If there is a problem with the server certificate, performance data continues to be collected, and a warning message is output to the common message log. When the warning message is output, you must change the invalid certificate to a valid one. 2: SSL/TLS is used. If there is a problem with the server certificate, performance data continues to be collected, but a warning message is not output to the common message log. If the default certificate of VMware is used for operation, output of warning messages can be suppressed by specifying the value 2. If you want operation without using the server certificate of Docker environment, specify the value 2 so that a warning message can be prevented from outputting. 3: SSL/TLS is used. If there is a problem with the server certificate, performance data is not collected, and a warning message is output to the common message log. If you want to use a valid certificate for operation and collect performance data only from trusted monitoring targets, specify the value 3. • For Hyper-V, logical partitioning feature, KVM, and Podman environment: SSL/TLS cannot be used, so ignores the value of this item is always assumed to be 0.	Previous setting
Port	Specifies the port number that is used for communication with the monitoring-target physical server.	0-65535 • For VMware: If 0 is specified, 443 (HTTPS default port number) • For logical partitioning feature: The fixed port number 623 is used. This port number cannot be changed. Even if a value other than 623 is specified for this item, the port number 623 will be assumed. • For KVM: If 0 is specified, 22 (SSH default port number) will be assumed. • For Docker environment:	Previous setting

Item	Explanation	Value that can be set	Default value
`Port`	Specifies the port number that is used for communication with the monitoring-target physical server.	Specify the port number used for connection. If `0` is specified, performance data is not collected and a warning message is output to the common message log. • For Podman environment: If `0` is specified, `22` (SSH default port number) will be assumed.	Previous setting
`UserID`	Specifies the user ID that is used to connect to the monitoring-target physical server.#3	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
`Password`	Specifies the password that is used to connect to the monitoring-target physical server. The character string entered for this item is not displayed. This item requires you to enter a value twice. If the monitoring target uses VMware and you specify a password that includes special characters, make sure that you specify a sanitized string.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
`Domain`	Specifies the name of the domain that the monitoring-target physical server belongs to. If the server belongs to a work group, you do not need to specify this item. If the monitoring target uses VMware, KVM, logical partitioning feature, Docker environment, or Podman environment, you do not need to specify this item.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting
`Private_Key_File`	Specifies the absolute path name of the private key file that is used for SSH public key encryption. You do not need to enclose the path name in double quotation marks (`"`) even if it includes spaces.	Character string not exceeding 256 bytes. Note that the following character cannot be used: • Tab character	Previous setting

#1

The first thing you must do before you can perform collection of performance information and do health checks is to ensure that the PFM - RM host name can resolve to an IP address. To use the JP1/IM linkage facility, the JP1/IM host name must be resolvable to an IP address.

#2

The string `All`, which is a reserved word for group agents, cannot be used.

#3

If the monitoring-target virtual environment uses VMware, the account you specify must have a *Read-only* or a higher VMware role. For details about how to specify the role settings, see the VMware documentation.

For the permission required for the account to be specified if the monitoring-target virtual environment uses Hyper-V, see the notes in *2.5.2 For Hyper-V*.

If the monitoring-target virtual environment uses KVM, always log in as superuser. For details about how to specify settings after login and notes, see *2.5.3 For KVM*.

If the monitoring-target virtual environment uses Podman environment, always log in as superuser. The container (rootless container) of a general user cannot be monitored.

For details about how to specify settings after login and notes, see *2.5.5 For Podman environment*.

Note that the items that must be specified during setup of a monitoring target differ depending on the virtual environment. Therefore, the items that can be updated also differ depending on the virtual environment. The following table shows the updatable items for each virtual environment.

Table 2–14: Updatable items of a monitoring target for each virtual environment

Item	Virtual environment					
	VMware	Hyper-V	KVM	Docker environment	Podman environment	logical partitioning feature
`Target Host`	Y	Y	Y	Y	Y	Y
`VM_Host`	Y	Y	Y	Y	Y	Y
`Security`	Y	--	--	Y	--	--
`Port`	Y	--	Y	Y	Y	--
`UserID`	Y	Y	Y	--	Y	--
`Password`	Y	Y	--	--	--	--
`Domain`	--	Y	--	--	--	--
`Private_Key_File`	--	--	Y	--	Y	--

Legend:

Y: Can be updated

--: Updating is not necessary because the setting has no effect on the PFM - RM for Virtual Machine operation

To check the name of the monitoring target that you want to update, use the `jpcconf target list` command. To update the monitoring target, use the `jpcconf target setup` command.

The following procedure shows how to update a monitoring target. To update multiple monitoring targets, repeat this procedure:

1. Check the name of the monitoring target that you want to update.
   To display the names of monitoring targets for instance `inst1`, execute the `jpcconf target list` command by specifying the service key that indicates PFM - RM for Virtual Machine.
   ```
   jpcconf target list -key RMVM -inst inst1
   ```
   If the only monitoring target that is set is `targethost1`, the command displays `targethost1`.

2. If you change the name of the monitored host (`Target Host`), use the `jpctool alarm unbind` command to unbind the bound alarm table.
   To unbind an alarm table with the alarm table name of `alarmtable1` from the monitored host with the service ID of `8Atargethost1`, execute the following command:
   ```
   jpctool alarm unbind -key RMVM -table alarmtable1 -id 8Atargethost1
   ```

3. Execute the `jpcconf target setup` command by specifying the service key that indicates PFM - RM for Virtual Machine, the instance name, and the name of the monitoring target.

4. Update the information about the monitoring target of PFM - RM for Virtual Machine.

Specify values for the items indicated in Table 2-14. The current value is displayed for each item in the initial status. To accept the displayed value, just press the return key. When all entries are completed, the monitoring target is updated.

5. If you unbound the alarm table in step 2, execute the `jpctool alarm bind` command to bind the alarm table.
   To bind the alarm table with the alarm table name of `alarmtable1` with the monitored host with the service ID of `8Atargethost1`, execute the following command:
   ```
   jpctool alarm bind -key RMVM -table alarmtable1 -id 8Atargethost1
   ```

For details about commands, see the chapter on commands in the manual *JP1/Performance Management Reference*.

2.5 Settings for each virtual environment

This section describes the settings required for each virtual environment.

2.5.1 For VMware

If the virtual environment to be monitored uses VMware environment, communication between PFM - RM for Virtual Machine and the virtual environment is encrypted using SSL/TLS#.

Therefore, the following settings are required:

- Setting of encrypted communication (For the PFM - RM for Virtual Machine host of Windows Server 2016 or later)
- Confirmation the certificate used in the monitored VMware ESX
- Installing a CA certificate in the PFM - RM for Virtual Machine host

#

The SSL/TLS communication protocol is using the Internet Options of the user account set for HostUserID in the instance environment settings.

If you change the settings, log in to the PFM - RM for Virtual Machine host by using the user account set for HostUserID in the instance environment settings, in the Internet Options dialog box, click the Advanced tab, and then change the settings in the Security category. If you do not use SSL 3.0, uncheck Use SSL 3.0 and check Use TLS 1.0, Use TLS 1.1, and Use TLS 1.2 in the Security category.

(1) Setting of encrypted communication (For the PFM - RM for Virtual Machine host of Windows Server 2016 or later)

One of the following methods is required for the PFM - RM for Virtual Machine host of Windows Server 2016 or later:

Important

If the settings are not made, the status will be as follows and the report cannot be displayed.

- Status field of `Host Status Detail (PD)` record is `SUCCESS`
- VM Count field of `Host Status Detail (PD)` record is `0`
- VM Active field of `Host Status Detail (PD)` record is `0`
- No data other than `Host Status Detail (PD)` records is collected.

(a) Add the Web site to the Restricted Sites or Trusted Sites zones

Using by the user specified as a HostUserID in the "Internet Options" dialog box for instance setting, select the "Security" tab, specify a host for monitoring target to "Add this website to the zone:" in "Local Intranet" or "Trusted sites" as followings:

If you have more than one monitoring target, add each of them to the zone for the Web site.

https://(monitoring-target-hosts-name #)

#

When VM_Host is not specified by setting for the monitoring target, specify the name of Target Host. When VM_Host is specified, specify the name of VM_Host.

(b) Turn off Internet Explorer Enhanced Security Configuration

For the group (Administrators or Users groups) includes the user specified as a HostUserID in the Internet Options dialog box for instance setting, turn off on "Internet Explorer Enhanced Security Configuration" dialog box.

In this case, be careful that the settings are applied for the other users in the same group.

If a user other than the user specified as a `HostUserID` turns off this configuration, the user with `HostUserID` needs to log on to the PFM - RM host to apply the configuration change.

(2) Confirmation the certificate used for the monitored VMware ESX

Check the certificate used by VMware ESX to be monitored. To check it with Internet Explorer, use the following procedure.

(a) Open the login page of VMware ESX to be monitored

In the address bar of Internet Explorer, enter `https://`*name-of-the-monitored-host* to open the login page of VMware ESX.

If you see the `This site is not secure` message, select **Go on to the webpage (not recommended)** under **More information**.

(b) Show the certificate

In the address bar of Internet Explorer, click **Certificate Error** or the key icon, and in the pop-up window that appears, select **View Certificate**.

(c) Check the certificate

In the **General** tab of the Certificate dialog box, check the **Issued by** field. The following types of certificates are available:

- Default certificate
  If the issuer is VMware Installer, the certificate will be a default certificate.
- vCenter Server certificate
  If the issuer is CA and the **Issuer** field in the **Details** tab of the Certificate dialog box is VMware Engineering, the certificate will be a vCenter Server certificate.
- Your own certificate
  The certificates issued by any issuer other than the above will be your own certificate.

If VMware ESX to be monitored uses the default certificate, you do not have to install the CA certificate in the PFM - RM for Virtual Machine host. However, when monitoring using the default certificate, see *(3) When using the default certificate*.

If a vCenter Server certificate or an own certificate is used for the monitored VMware ESX, a CA certificate must be installed in the PFM-RM for Virtual Machine host.

If you are using a vCenter Server certificate on the monitored VMware ESX or using your own certificate, see *(4) When obtaining and operating a CA certificate*.

Furthermore, check the certificate of VMware ESX to see if:

- The validity period for the certificate of VMware ESX is valid.
- The name specified in the **Issued to** field of the VMware ESX certificate can be resolved by the PFM - RM for Virtual Machine host.
- The name specified in the **Issued to** field of the certificate is the same as that of the host to be specified as the target monitored by PFM - RM for Virtual Machine.

  If these host names are inconsistent, the certificate is handled as an invalid one.

  You need to configure the monitored host differently according to the **Issued to** field of the VMware ESX certificate. Based on your environment, configure it as follows:

  - If the **Issued to** field of the VMware ESX certificate contains a domain name

    Specify the name entered in the **Issued to** field of the VMware ESX certificate for the `VM_Host` parameter of the settings for the monitored host.
  - If the **Issued to** field of the VMware ESX certificate contains a host name, but not a domain name

    Specify the name entered in the **Issued to** field of the VMware ESX certificate for the `Target Host` parameter of the settings for the monitored host.

    Specify no value or specify the same value as that of the `Target Host` parameter for the `VM_Host` parameter of the settings for the monitored host.

If you monitor the target without installing the CA certificate in the PFM - RM for Virtual Machine host when VMware ESX to be monitored uses the vCenter Server certificate or your own certificate, see *(3) When using the default certificate*, just like the case where the default certificate is used.

(3) When using the default certificate

If you choose to use a default certificate of VMware Installer, the following precautions need to be taken.

- For an environment that cannot communicate with the Windows Update site

  The Update Root Certificates function works for communications that use a certificate. When the Update Root Certificates function verifies the certificate, the function does so by downloading the latest information from the Windows Update site. When the Update Root Certificates function is enabled, if the environment does not allow the host that runs PFM - RM for Virtual Machine to communicate with the Windows Update site, verifying the certificate might take a long time.

  If certificate verification takes too much time, the KAVL20014-W warning message is output to the common message log and the monitoring cannot be performed.

  > **Important**
  >
  > If certificate verification takes a long time, the following problems may occur.
  >
  > - When collecting performance data, it takes a long time for the response from the connection destination VMware, a KAVL20516-W warning message is output to the common message log, and the collection of performance data does not complete within the collection interval.
  > - When collecting performance data, it takes a long time for the response from the connection destination VMware to occur, a KAVL20014-W warning message is output to the common message log, and collection fails.
  >
  > Take one of the following actions:
  >
  > - Review the network environment so that the Windows root certificate update function operates.

- Change the Windows settings (the security policy settings of the OS) so that the Update Root Certificates function does not communicate with the Windows Update site.
- Modify the network environment so that the Update Root Certificates function can run normally.
- Change the Windows settings (the security policy settings of the OS) so that the Update Root Certificates function does not communicate with the Windows Update site.

- Ignore the KAVL20205-W warning message that is output to the common message log

  If `1` is specified for `Security` in the monitoring target settings, the KAVL20205-W warning message is output to the common message log because the default certificate of VMware is not a valid certificate. In this case, make sure that the message can be safely ignored for normal operation.

  If the warning message is not necessary, specify `2` for `Security` in the monitoring target settings.

  Note that when the default certificate of VMware is used, monitoring does not function if `3` is specified for `Security` in monitoring target settings.

- Operation using a certificate that cannot be trusted

  The default certificate of VMware is determined to be a certificate that cannot be trusted by certificate verification. Make sure that a certificate that cannot be trusted does not cause problems that affect operation.

(4) When obtaining and operating a CA certificate

To operate with a CA certificate, obtain the issuing CA certificate and import it to the PFM-VM for Virtual Machine host.

(a) Obtain the CA certificate

- When the vCenter Server certificate is used

  If you used the vCenter Server for VMware ESX management, obtain CA certificate from vCenter Server. The vCenter Server's CA certificate is available from "Download trusted root CA certificates" on the page where "Getting Started" of vCenter Server is displayed.

  For details about, the following knowledge base. For questions about knowledge base, contact VMware.

  KB 2108294 : How to download and install vCenter Server root certificates to avoid Web Browser certificate warnings

- When your own certificate is used

  Obtain the CA certificate from the administrator of the certificate.

Check whether the obtained CA certificate is the validity period.

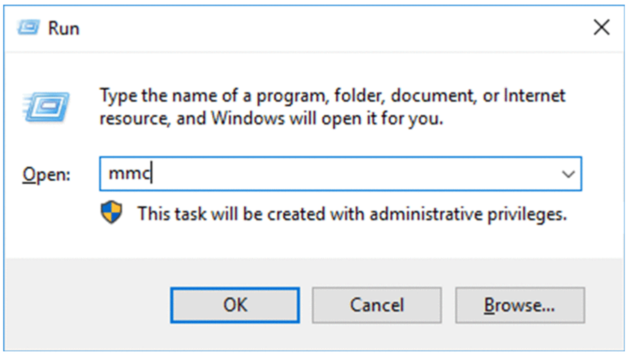
In addition, if the validity period of the certificate is expired, you need to obtain a new CA certificate and import it.

(b) Importing the default certificate for VMware

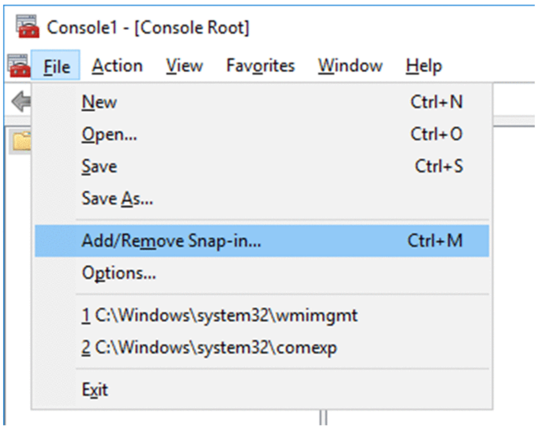
After you have prepared a CA certificate for VMware as described in *(1)* above, import the certificate onto the PFM-VM host. To import the certificate:

1. In Windows, choose **Start** and then **Run**.

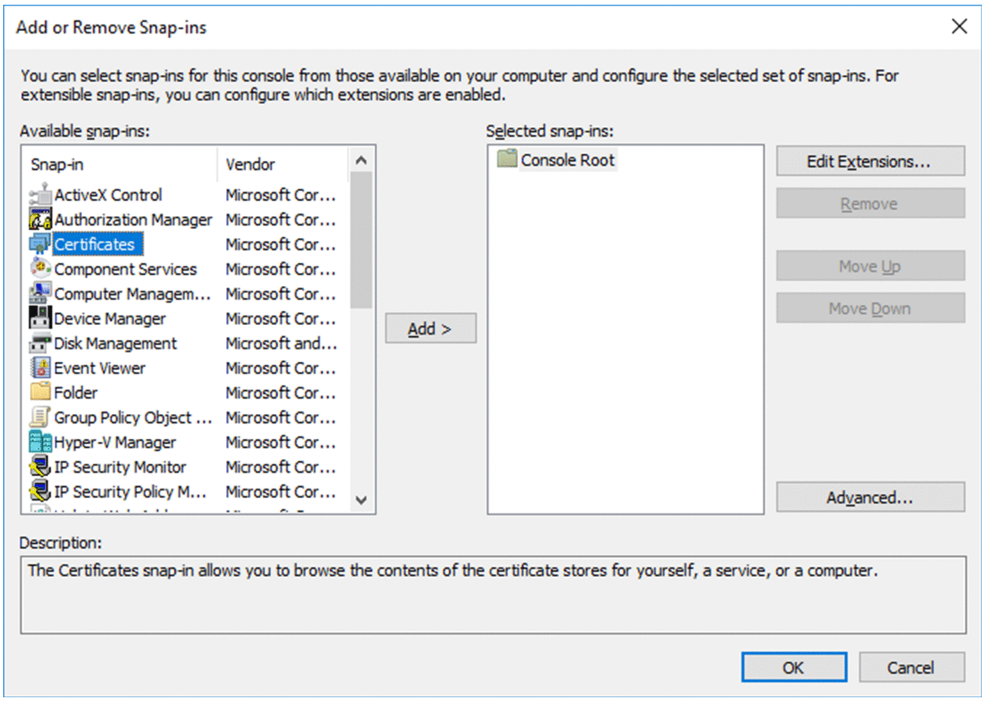
   The Run dialog box opens.



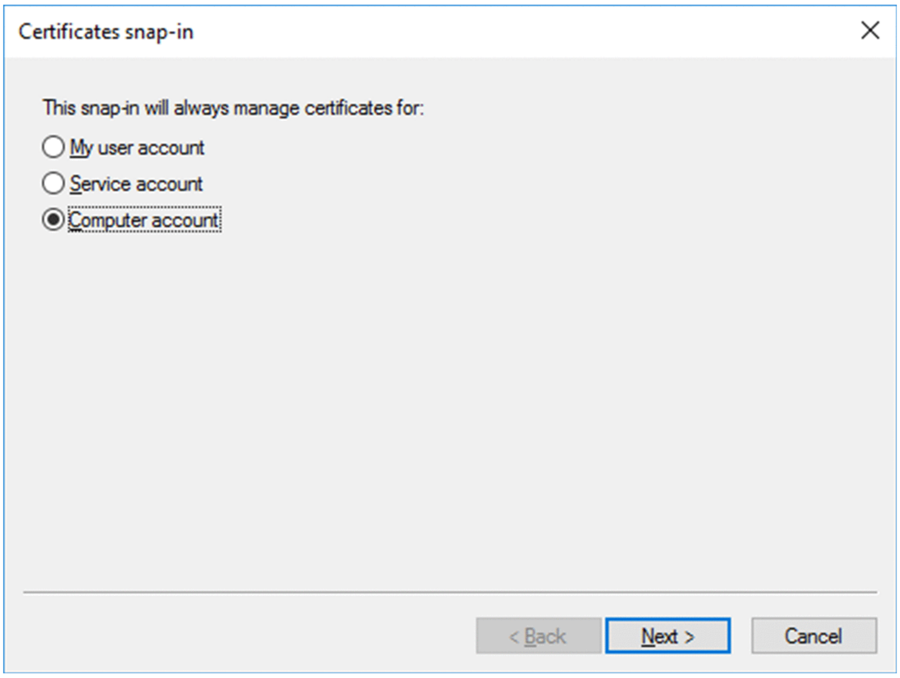
2. In the Run dialog box, enter `mmc` and click **OK**.
   Management Console starts.



3. In **Console1**, choose **File** and then **Add/Remove Snap-in**.
   The Add/Remove Snap-in dialog box opens.

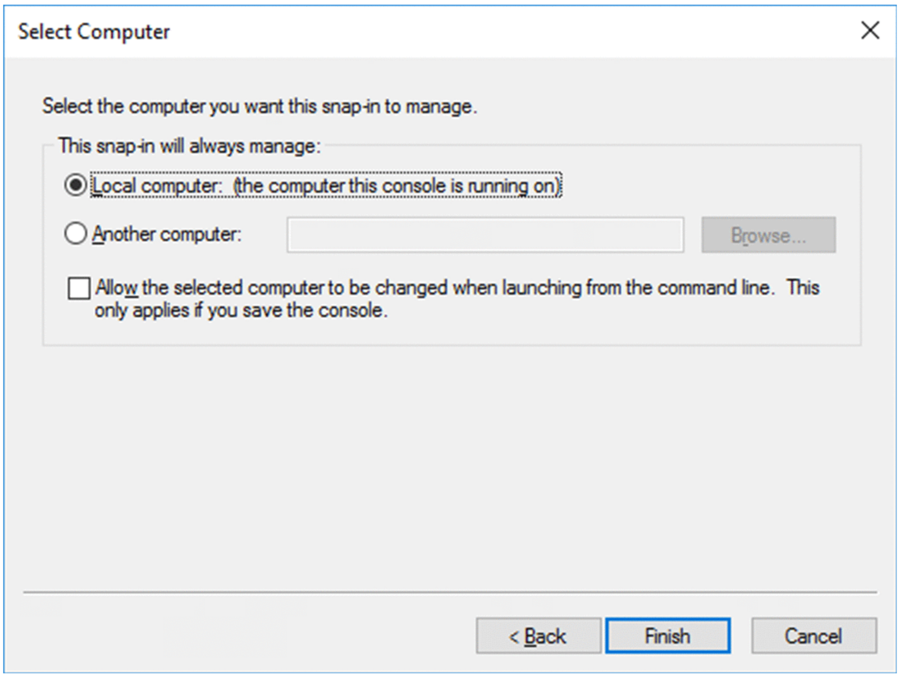


4. Choose **Certificates** and then click **Add**.
   The Certificates snap-in dialog box opens.

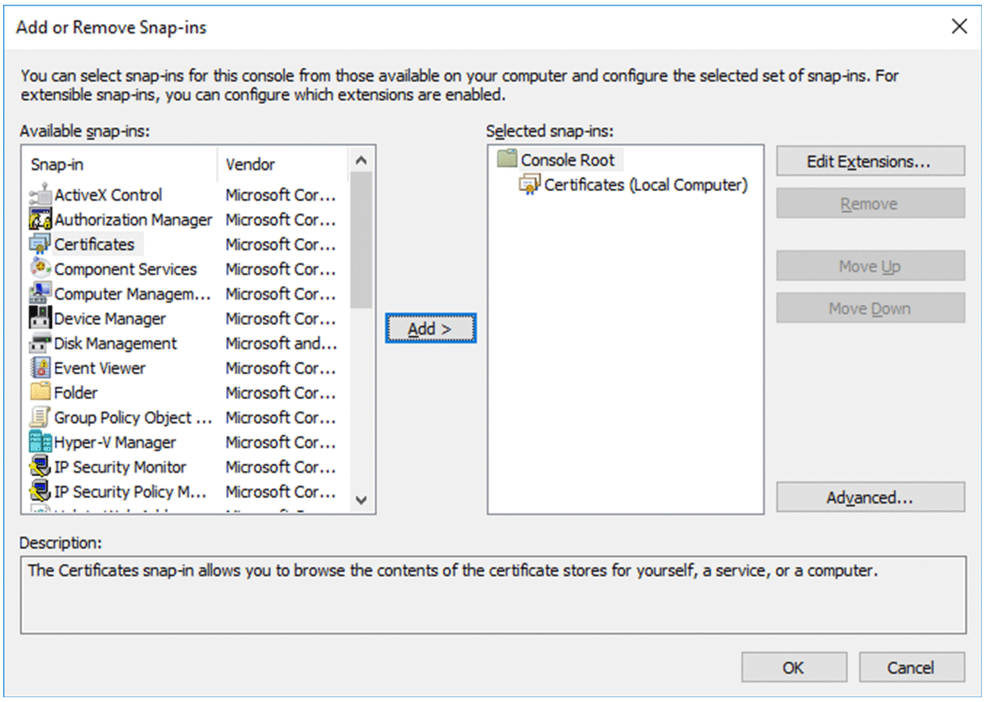


5. Choose **Computer account** and then click **Next**.

   The Select Computer dialog box opens.

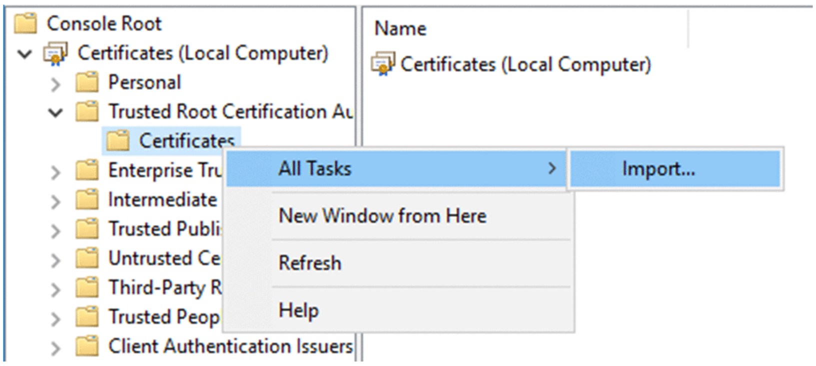


6. Choose **Local computer** and click **Finish**.

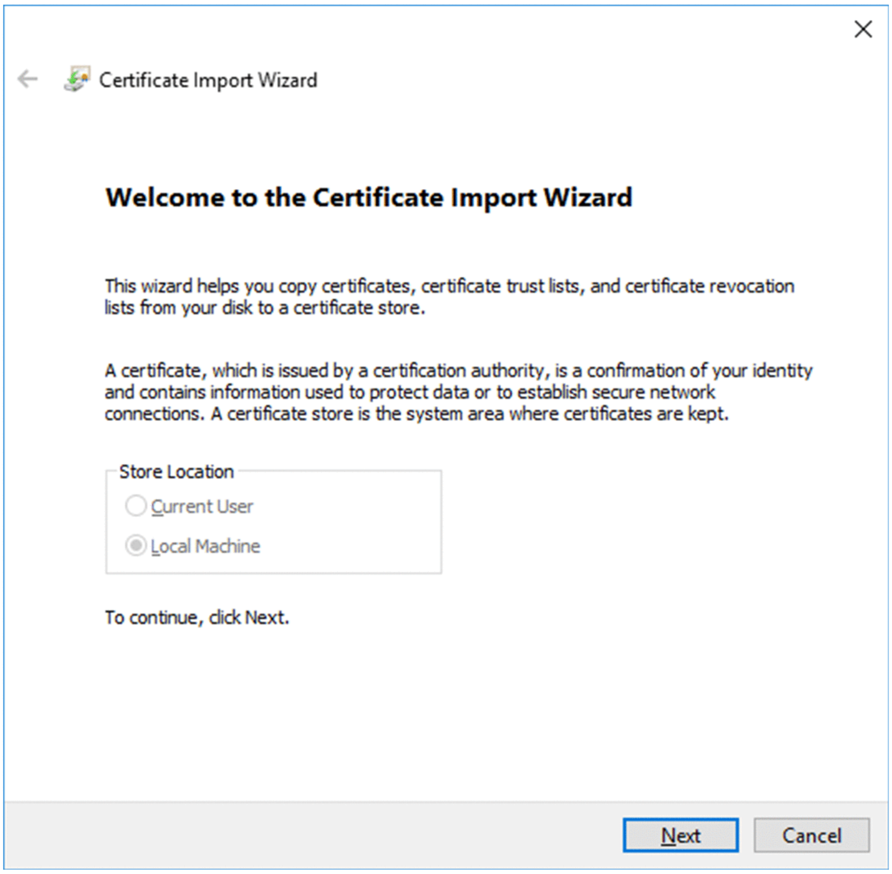


7. Check that **Certificates (Local Computer)** is added to **Selected snap-ins** and click **OK**.

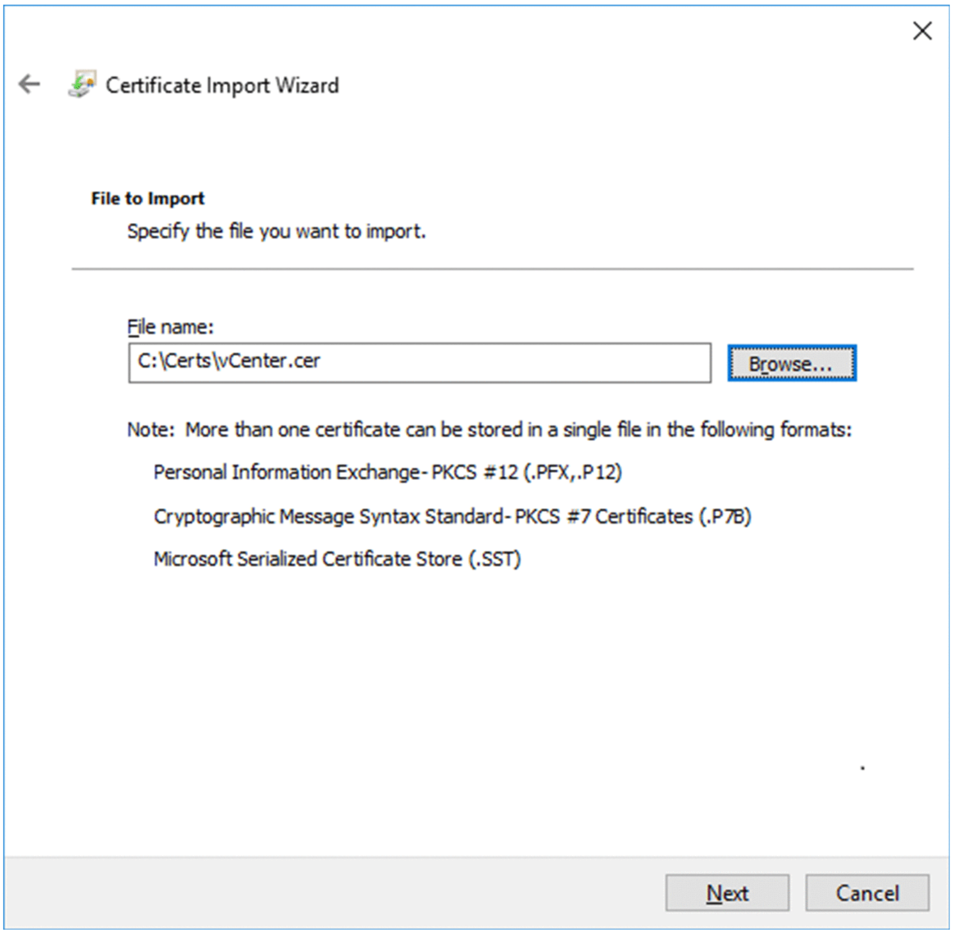
8. Expand **Certificates (Local Computer)** and right-click **Certificates** under **Trusted Root Certification Authorities**. Then click **All Tasks** and **Import** from the displayed menu items.



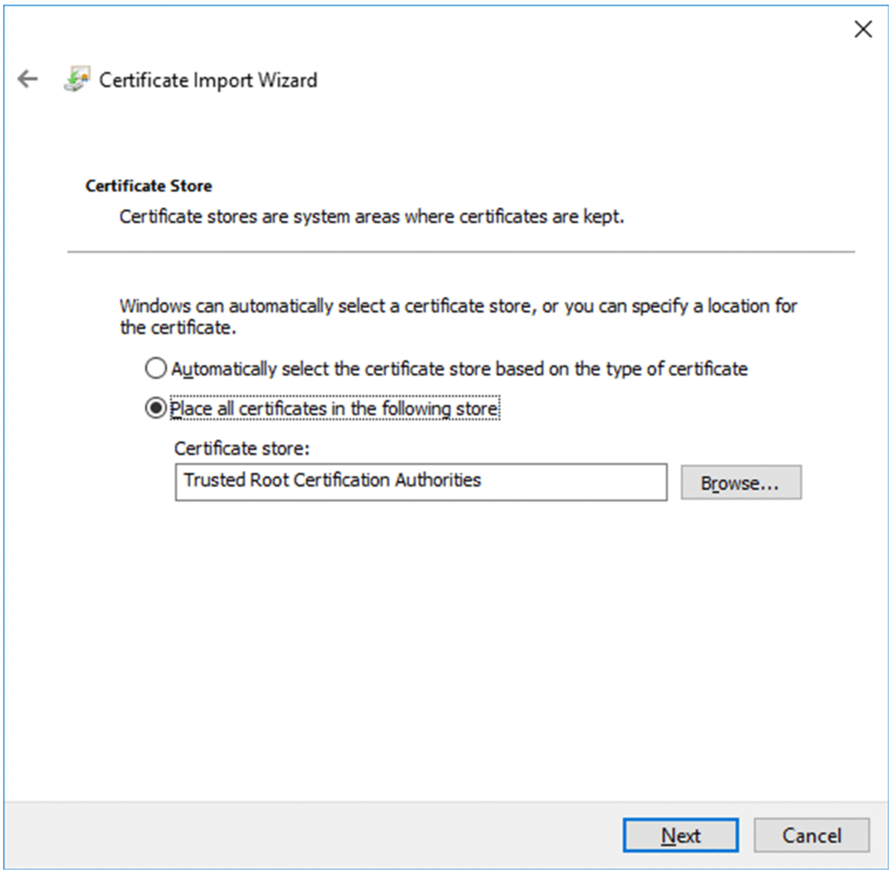
9. In the right pane of **Console1**, right-click **Trusted Root Certification Authorities**, then **All Tasks** and **Import**.
   The Certificate Import Wizard dialog box opens.



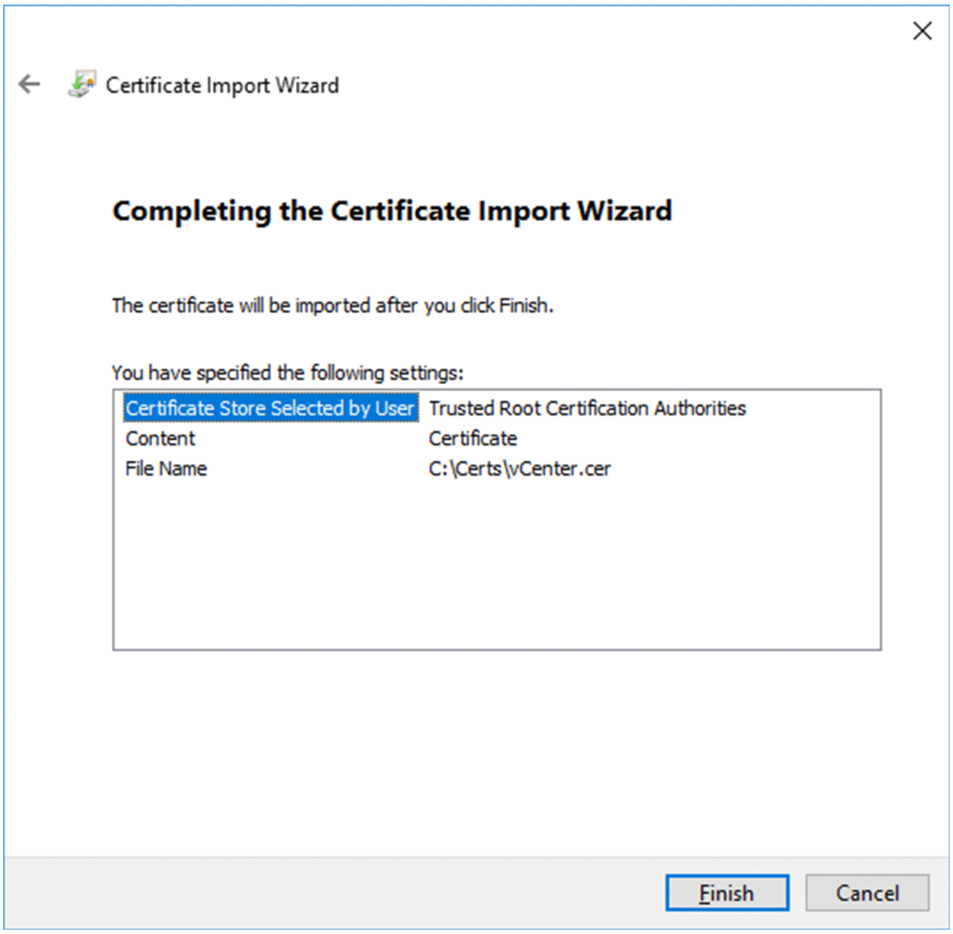
10. Click **Next**.



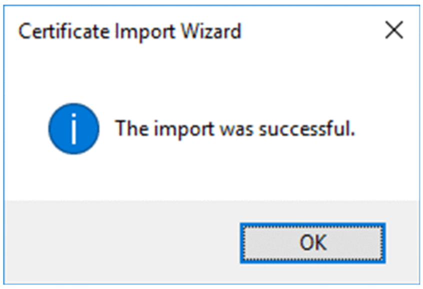
11. In the **File name** text box, enter the file name under which to save the certificate, and then click **Next**.



12. Choose **Place all certificates in the following store,** and then click **Next**.



13. Click **Finish**.



14. Click **OK**.

(5) Confirmation of Host Logical Disk Status (PI_HLDI) record

The PI_HLDI record source data is equivalent to the amount of free space displayed for the storage on the Configuration page when the connection-destination VMware ESX is displayed in vSphere Client.#

#

This value might be different from the amount of free space displayed in vSphere Client that is connected to vCenter Server. If vSphere Client is connected to vCenter Server, recheck the value by connecting to VMware ESX.

As a result, if you monitor free space on the VMware datastore by using the `Used`, `Free`, and `Used %` fields of the `PI_HLDI` record, which are retrieved by PFM - RM for Virtual Machine, the value might not be changed when collecting performance data or evaluating the alert.

By using the Last Update field of the PI_HLDI record, make sure that the *last update time* of the VMware datastore is periodically updated.
If it is not periodically updated, consider the following knowledge base.
For further information about countermeasures or other topics, contact VMware.
KB2008367: Amount of free space reported on the host is incorrect in vCenter Server

(6) User defined records

If the monitoring target is a VMware environment, you can use user defined records to monitor performance information that is not retrieved by PFM - RM for Virtual Machine.

(a) User defined record collection

The following records are used in user defined record collection:

- Host Generic Data Detail (PD_HGDD)
- Host Generic Data Interval (PI_HGDI)
- VM Generic Data Detail (PD_VGDD)
- VM Generic Data Interval (PI_VGDI)

Each record has different record types and monitoring targets. Use the record that meets your needs.

The following table shows the details of the records.

Record name	Record type	Monitoring target	Use it to:
Host Generic Data Detail (PD_HGDD)	PD record	Physical server	Get the physical server's system status at a given point in time.
Host Generic Data Interval (PI_HGDI)	PI record	Physical server	Analyze changes or trends in the physical server's system status over time.
VM Generic Data Detail (PD_VGDD)	PD record	Virtual machine	Get the virtual machine's system status at a given point in time.
VM Generic Data Interval (PI_VGDI)	PI record	Virtual machine	Analyze changes or trends in the virtual machine's system status over time.

(b) Items collected in user defined records

You can see which items are collected in user defined records by checking the performance chart of VMware vSphere Client.

For example, if you want to monitor utilization of the physical server's CPU core, you check the following items:

Items collected for CPU core utilization
Chart Options: CPU
Rollup: average
Internal Name: coreUtilization

(c) Deploying the user-defined record definition file

Add the user-defined record definition file with the name recorddef.ini to the folder of the instance whose VM_Type is vmware.

For physical hosts:

installation-folder\agt8\agent*instance-name*\recorddef.ini

For logical hosts:

environment-folder\jp1pc\agt8\agent*instance-name*\recorddef.ini

The same definition file is used for all the monitoring targets in the instance.

Examples where the instance name is `inst01`:

- For physical hosts:
  installation-folder\agt8\agent\inst01\recorddef.ini
- For logical hosts:
  environment-folder\jp1pc\agt8\agent\inst01\recorddef.ini

The user-defined record define information file is validated when the Remote Monitor Collector service starts. If there is no error in the file, the message KAVL20528-I is output. If there is an error in the file, the message KAVL20527-W is output, and the Remote Monitor Collector service starts without collecting user defined records.

Note

If you change the user-defined record definition file, restart the Remote Monitor Collector service. The definition file that was loaded at startup is not updated until the service is restarted.

(d) Creating the user-defined record definition file

You can specify the user defined records to be monitored and the items to be collected to create the user-defined record definition file.

Format of the definition file

Configure the user-defined record definition file (in `ini` format) as follows:

```
[record-ID]
[[section-name]]
TYPE=chart-option
NAME=internal-name
ROLLUP=rollup
```

Items of the definition file

The following table shows the items you need to specify.

Item	Parameter	Explanation
1	[*record-ID*]	Specify the record ID in which data is stored. Values can be: • `PD_HGDD`: Host Generic Data Detail record • `PI_HGDI`: Host Generic Data Interval record • `PD_VGDD`: VM Generic Data Detail record • `PI_VGDI`: VM Generic Data Interval record
2	[[*section-name*]]	Specify the section name with up to 32 bytes of single-byte alphanumeric characters, hyphens (-), and underscores (_). The specified section name is stored in the `SECTION_NAME` field.
3	TYPE=*chart-option*	Specify the chart option# with up to 32 bytes of single-byte alphanumeric characters. If the specified chart option does not exist, no performance data will be collected.

Item	Parameter	Explanation
4	`NAME=`*internal-name*	Specify the name of the performance counter (internal name) with up to 64 bytes of single-byte alphanumeric characters and periods (`.`). If the specified internal name does not exist, no performance data will be collected.
5	`ROLLUP=`*rollup*	Specify the rollup. Values can be: • Average value: `average` • Latest value: `latest` • Summary value: `summation` If the specified rollup does not exist, no performance data will be collected.

#

The following table shows the values to be specified for each chart option.

Item	Chart option	Value to be specified	Corresponding record ID
1	CPU	`cpu`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
2	vSphere Replication	`hbr`	`PD_HGDD` and `PI_HGDI`
3	System	`sys`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
4	Storage adapter	`storageAdapter`	`PD_HGDD` and `PI_HGDI`
5	Storage path	`storagePath`	`PD_HGDD` and `PI_HGDI`
6	Disk	`disk`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
7	Datastore	`datastore`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
8	Network	`net`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
9	Memory	`mem`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
10	Virtual flash	`vflashModule`	`PD_HGDD` and `PI_HGDI`
11	Virtual disk	`virtualDisk`	`PD_VGDD` and `PI_VGDI`
12	Power source	`power`	`PD_HGDD`, `PI_HGDI`, `PD_VGDD`, and `PI_VGDI`
13	Information on the management agent	`managementAgent`	`PD_HGDD` and `PI_HGDI`

Definition example for utilization of the physical server's CPU core

```
[PD_HGDD]
[[CPU_CORE_UTIL]]
TYPE=cpu
NAME=coreUtilization
ROLLUP=average
```

Notes

- If there is no performance data that matches the combination of the specified items, the type name (`TYPE`), counter name (`NAME`), and rollup type (`ROLLUP`), no user defined record will be collected. Create a report for the user defined record to make sure the items are output.

- The number of the section names in the user-defined record definition file must not exceed 100. Collecting performance data can take a long time when many section names are listed in the file.
- If a large number of virtual machines monitored are running, the number of the PI_VGDI records increases. Keep this in mind, and monitor the PFM-RM host to prevent a poor disk performance or a storage shortage.
- The user-defined record definition file is loaded during startup of the PFM - RM for Virtual Machine service. If you change the definition file, restart the service.
- If there is an error in the user-defined record definition file, the message KAVL20527-W is output to the common message log. Check the line number and the details in the message to correct the file.
- If there is an error in the user-defined record definition file, the PFM - RM for Virtual Machine service starts while ignoring the record type that includes the error. You can check the valid defined record types in the message KAVL20528-I, which is output to the common message log.
- If the message KAVL20528-I is not output to the common message log, no user defined records will be collected. Make sure that the location where the user-defined record definition file is stored, the file name, and the contents of the file are correct.

(e) Examples of the definition and output

Here are examples of the definition and output to collect CPU wait time and memory wait time for physical servers and virtual machines.

Definition example:

```
[PD_HGDD]
[[CPU_WAIT]]
TYPE=cpu
NAME=wait
ROLLUP=summation
[[MEM_LATENCY]]
TYPE=mem
NAME=latency
ROLLUP=average

[PI_VGDI]
[[CPU_WAIT]]
TYPE=cpu
NAME=wait
ROLLUP=summation
[[MEM_LATENCY]]
TYPE=mem
NAME=latency
ROLLUP=average
```

Output example:

Host Generic Data Detail (PD_HGDD) record

Section Name	Data Name	Object Name	String Data	Double Data
CPU_WAIT	wait	0	0	0.0
CPU_WAIT	wait	1	24	24.0
MEM_LATENCY	latency	n/a	0	0.0

VM Generic Data Interval (PI_VGDI) record

Section Name	Data Name	Object Name	String Data	Double Data

CPU_WAIT	wait	0	1.2	1.2
CPU_WAIT	wait	1	1	1.0
MEM_LATENCY	latency	n/a	0.2	0.2

2.5.2 For Hyper-V

When the virtual environment to be monitored is Hyper-V, PFM - RM for Virtual Machine uses WMI to collect performance data from the monitored host. If a WMI connection has not been set up, performance data cannot be collected. Therefore, you must set up WMI on the PFM - RM host and the monitored host.

(1) Setting up WMI connection

This subsection explains how to set up WMI connection.

The following are the settings for connecting WMI:

- DCOM
  You must set DCOM on both the PFM - RM host and the monitored host.
  If you run the PFM - RM host in a cluster system, set DCOM at both the executing node and the standby node.
- Firewall
  Set a firewall on the monitored host, if necessary.
- WMI namespace
  Set a WMI namespace on the monitored host, if necessary.

When the above settings have been made, check that connection can be established from the PFM - RM host to the monitored host. For details about how to check the connection, see *(2) Checking the WMI connection status*.

Note about setting up WMI connection:

Data cannot be collected if the startup type of the Windows Management Instrumentation service (service name: `WinMgmt`) that provides system management information for the OS of the monitored host is disabled.

(a) Environment settings required for connection

The following settings are required in order to set WMI:

- User account settings
  To use WMI, you must have accounts for the PFM - RM host and the monitored host.
  - Account for the PFM - RM host
    To set up an account, specify the appropriate values for `HostUserID`, `HostPassword`, and `HostDomain`, as shown in Table 2-5. You specify this account when you set up an instance.
    If you run PFM - RM for Virtual Machine in a cluster system, set up an account for the PFM - RM host in such a manner that the same user name and password can be used to log on to both the running system and the standby system.
    If Hyper-V on a PFM - RM host is included in the target hosts to monitor their instances, use either of the following users with the instance settings:
    1. Built-in Administrator user
    2. Member of Performance Log Users or Performance Monitor Users group and Hyper-V Administrators group

Also, in a domain environment, specify the computer name of the PFM - RM host for `HostDomain` (it might be different from the host name in the `hosts` file or DNS system).

- Account for the monitored host

  To set up an account, specify the appropriate values for `UserID`, `Password`, and `Domain`, as shown in Table 2-8. You specify this account when you set up the monitoring target.

  The account to be used either following User.

  For Domain account user :

  1. Domain Administrator's user.

  For Local account user :

  1. Built-in Administrator's user.

  2. Member of Administrators group#.

  3. Member of Performance Log Users or Performance Monitor Users group and Hyper-V Administrators group.

  If you monitor Hyper-V on a PFM - RM host, do not specify the account for the monitoring-target host (`UserID`, `Password`, and `Domain`).

#

If used member of the administrators group(other than Built-in Administrator's account user), UAC restricts the permissions, causing the user to log on as a general user. As a result, performance information might not be acquired due to denial of access.

To prevent this, disable UAC or execute the following command on the monitoring target:

```
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v
LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
```

To override the settings specified by the above command and restore the previous status, execute the following command:

```
reg delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies
\System /v LocalAccountTokenFilterPolicy /f
```

- WMI service settings

  Set the monitored host's WMI service startup to a setting other than **Disabled**. If it is set to **Disabled**, performance data cannot be collected.

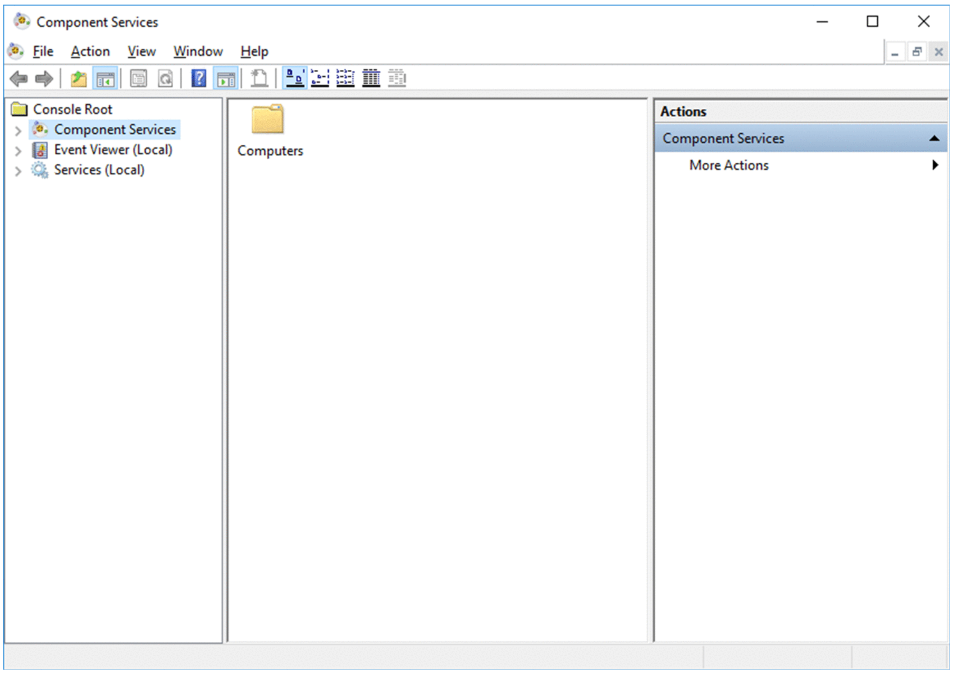
(b) Setting DCOM

This subsection describes how to set DCOM on the PFM - RM host and the monitored host.

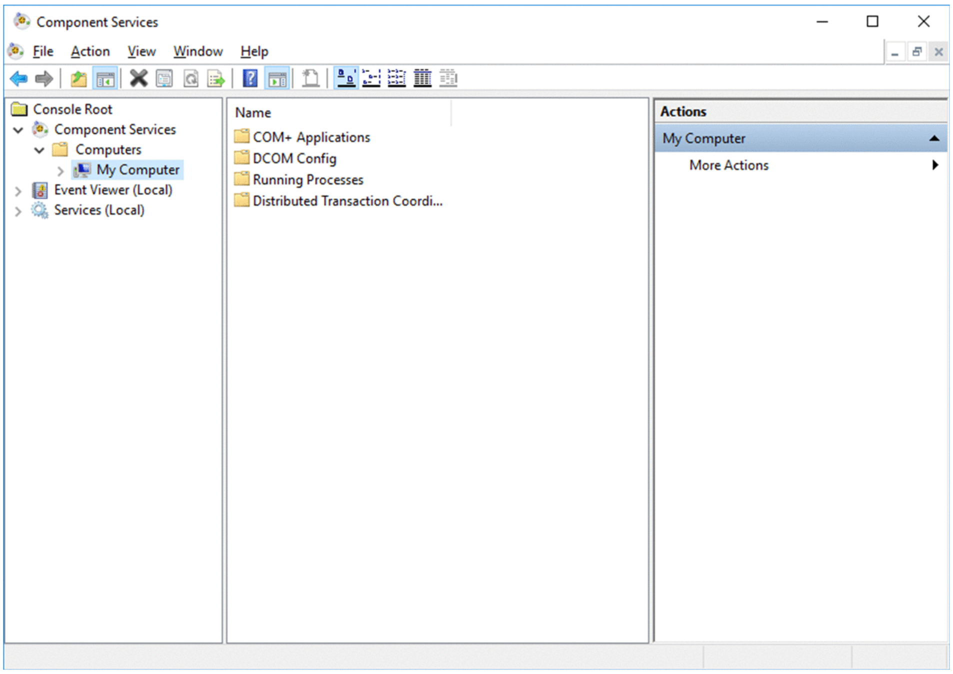
- Setting on the PFM - RM host

To set DCOM on the PFM - RM host:

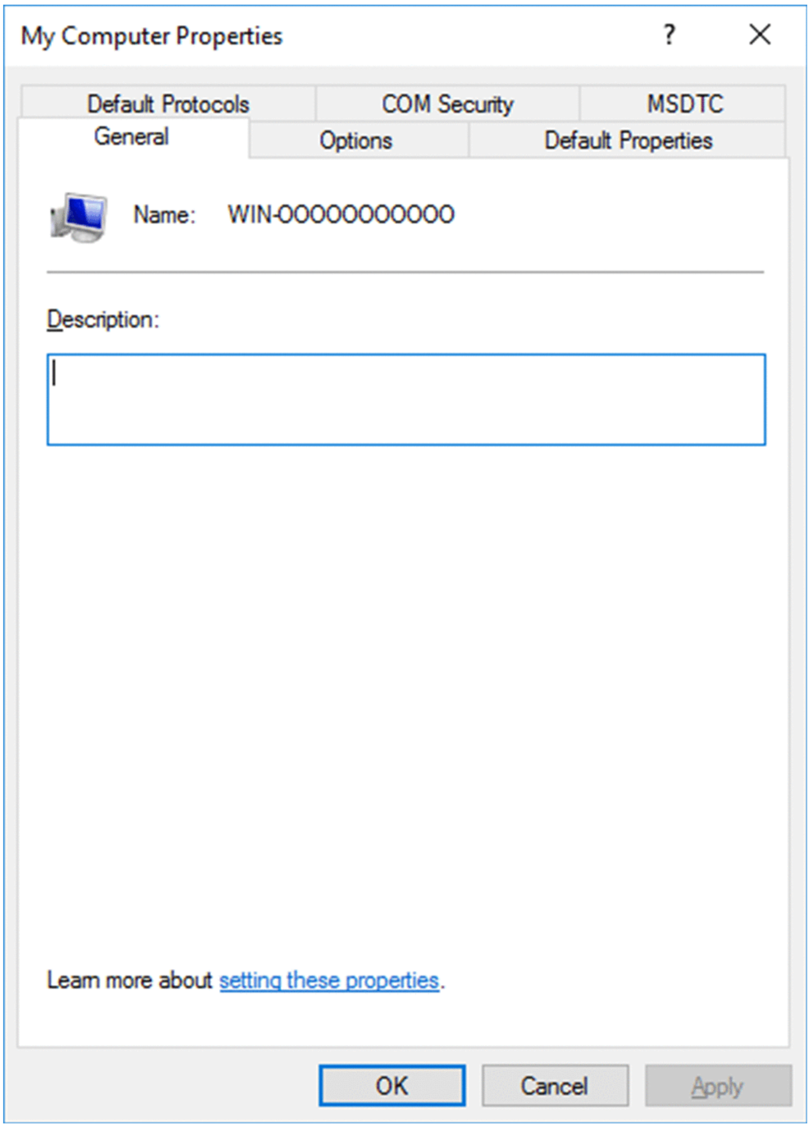
1. In Windows, choose **Start**, and then **Run**.
2. Enter `dcomcnfg.exe`, and then click the **OK** button.
   The Component Services window appears.



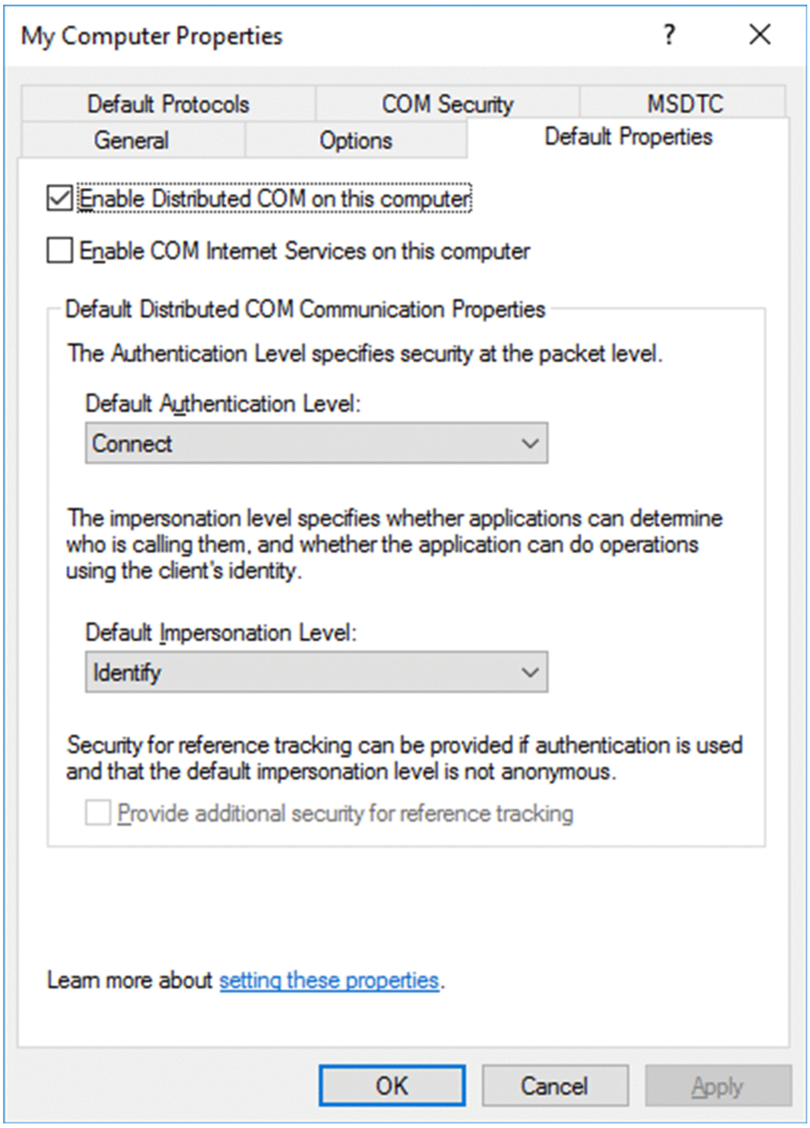
3. Click **Component Services**, then **Computers** to expand the tree.



4. Choose **My Computer** and then, from the right-click menu, choose **Properties**.
   The My Computer Properties dialog box appears.



5. Choose the **Default Properties** tab, and then select **Enable Distributed COM on this computer**.



6. Click the **OK** button.
   The My Computer Properties dialog box closes.

7. Restart the host.
   This is necessary only when the **Enable Distributed COM on this computer** setting has been changed.

- Setting on the monitored host
  This subsection describes how to set DCOM on the monitored host.

To set DCOM on the monitored host:

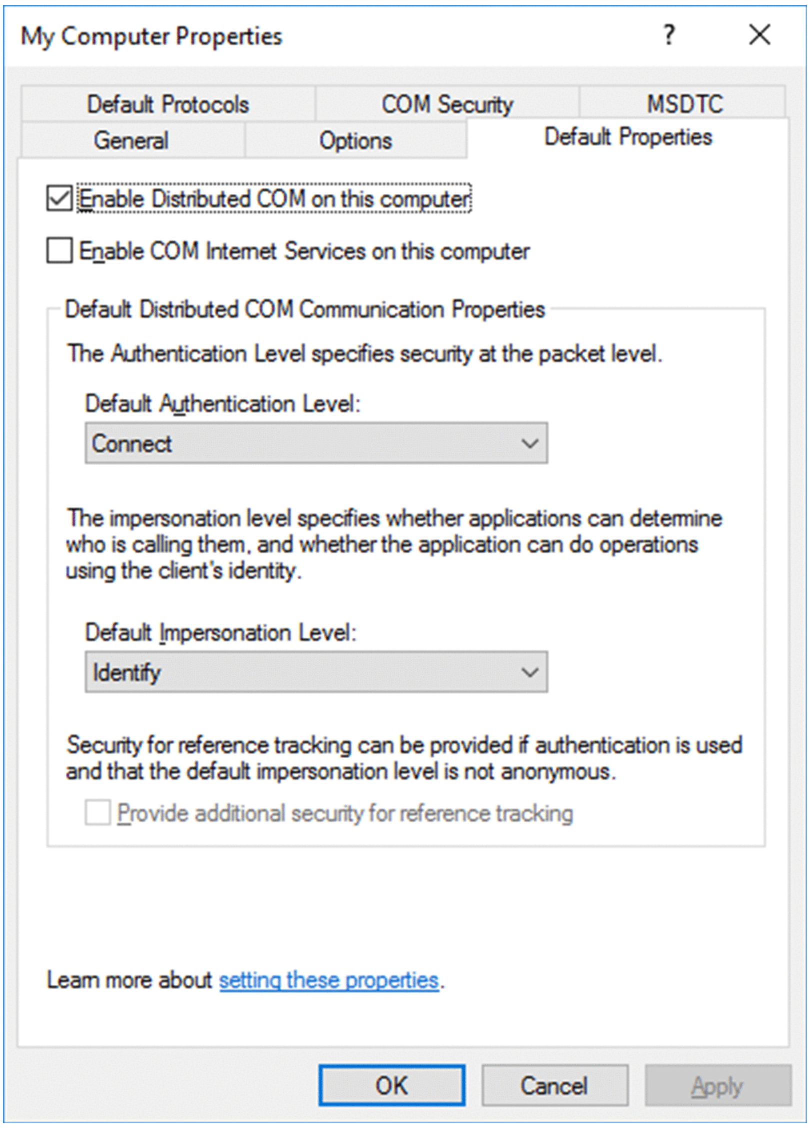
1. In Windows, choose **Start** and then **Run**.

2. Enter `dcomcnfg.exe`, and then click the **OK** button.
   The Component Services window appears.

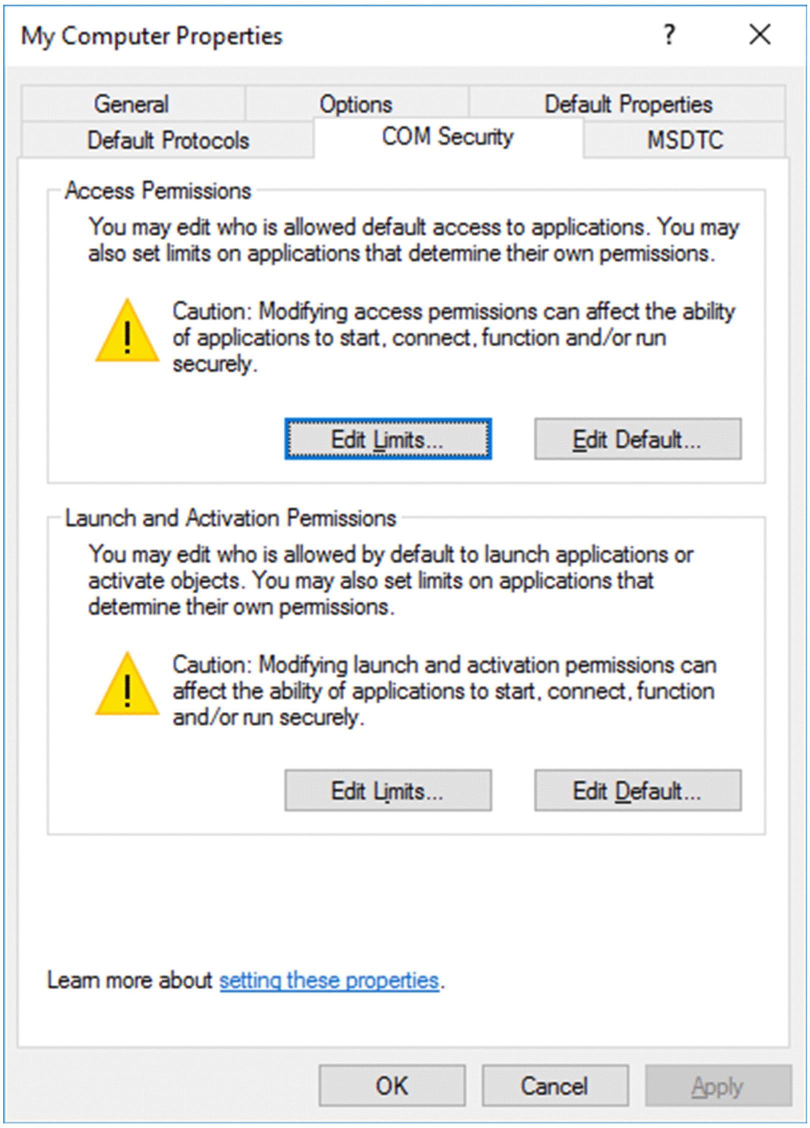
3. Click **Component Services**, then **Computers**, to expand the tree.

4. Choose **My Computer** and then, from the right-click menu, choose **Properties**.
   The My Computer Properties dialog box appears.

5. Choose the **Default Properties** tab, and then select **Enable Distributed COM on this computer**.

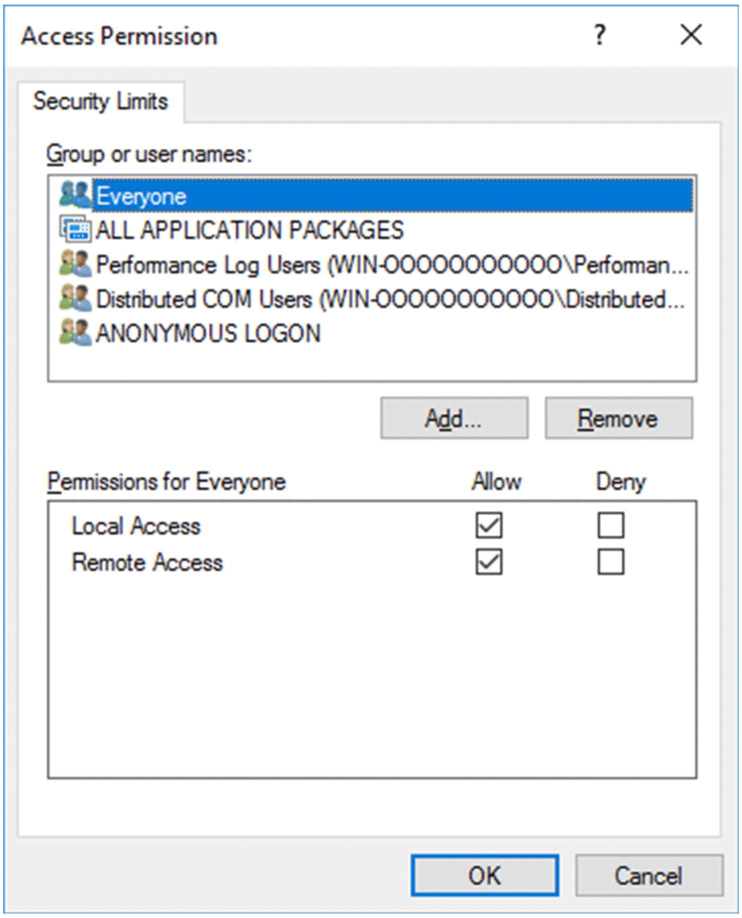


6. Select the **COM Security** tab and then, in **Access Permissions**, click the **Edit Limits** button.



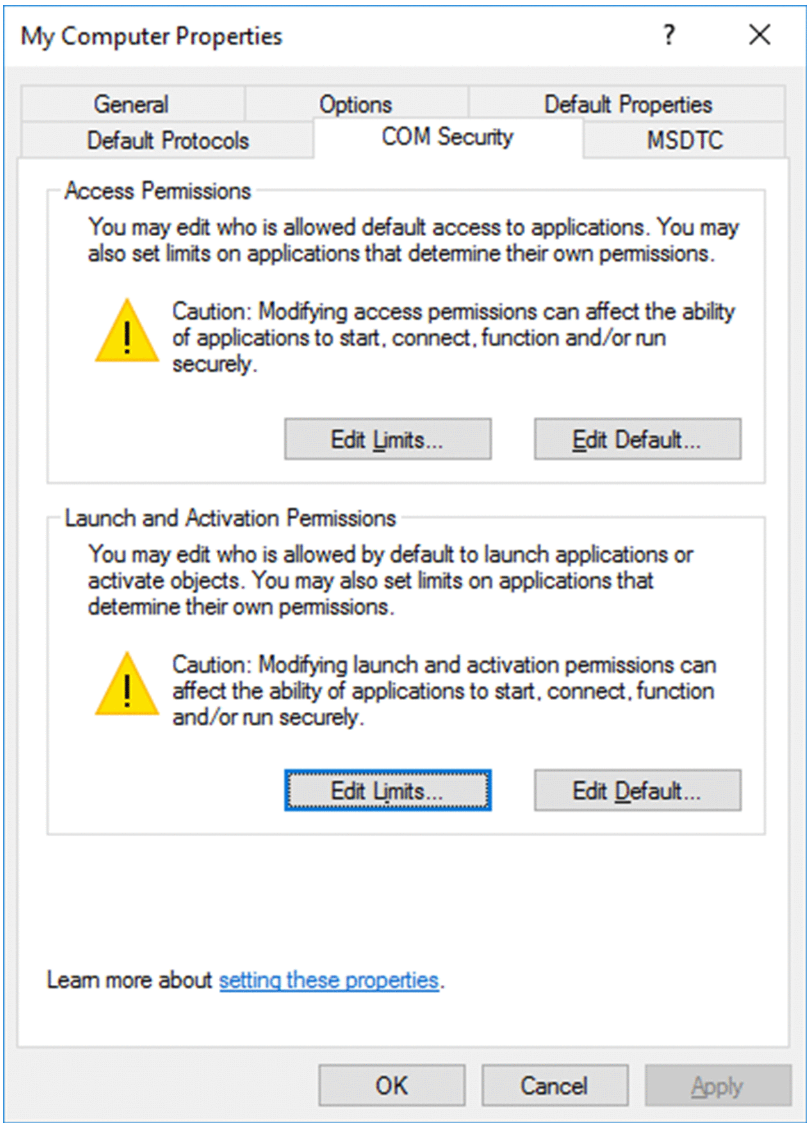
The Access Permission dialog box appears.
Check if the user who connects to the monitored host or the group to which this user belongs is displayed in **Group or user names**.
If the user or its group is not displayed, click the **Add** button and add the user or the group.

7. In **Group or user names**, select the user that connects to the monitored host or the group to which this user belongs. Check that the check box for **Allow** is selected for **Remote Access**. If this check box is cleared, select it.



8. Click the **OK** button.
   The Access Permission dialog box closes.

9. Select the **COM Security** tab and then, in **Launch and Activation Permissions**, click the **Edit Limits** button.

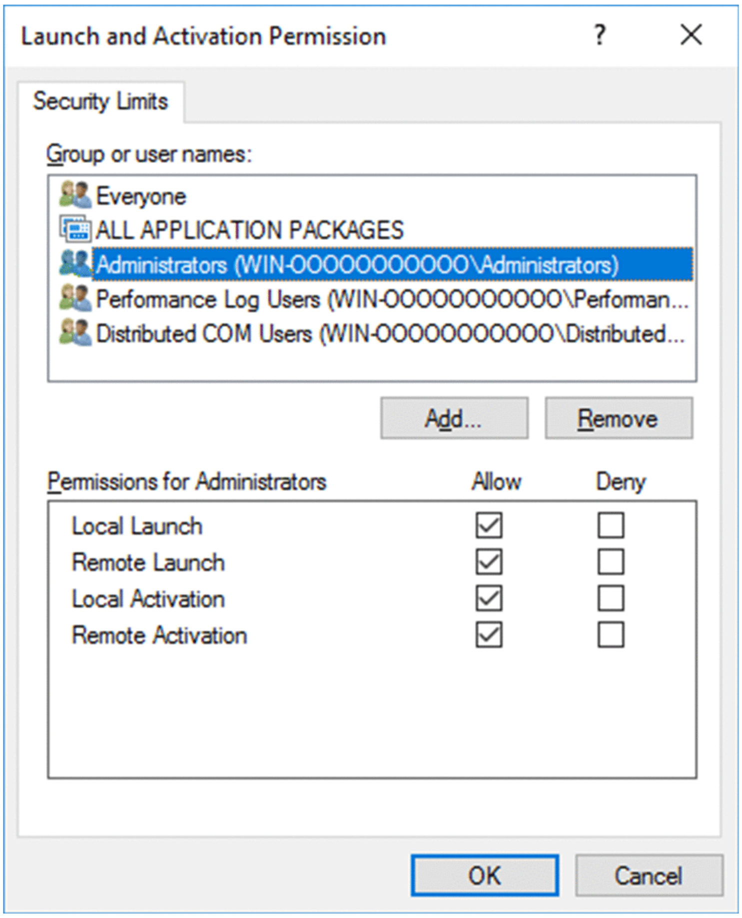


The Launch and Activation Permission dialog box is displayed.

Check if the user who connects to the monitored host or the group to which this user belongs is displayed in **Group or user names**.

If the user or its group is not displayed, click the **Add** button and add the user or the group.

10. In **Group or user names**, select the user that connects to the monitored host or the group to which this user belongs. Check that the check boxes for **Allow** are selected for **Remote Launch** and **Remote Activation**. If these check boxes are cleared, select them.



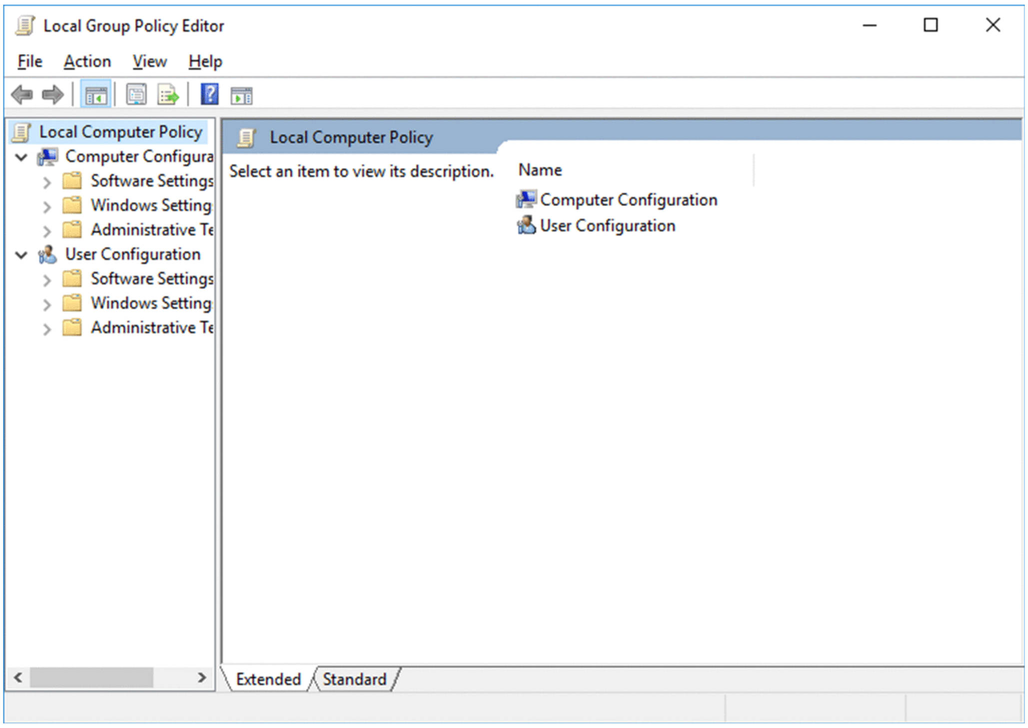
11. Click the **OK** button.
    The Launch and Activation Permission dialog box closes and the My Computer Properties dialog box is displayed.
12. Click the **OK** button.
    The My Computer Properties dialog box closes.
13. Restart the host.
    This is necessary only when the **Enable Distributed COM on this computer** setting has been changed.

(c) Setting a firewall

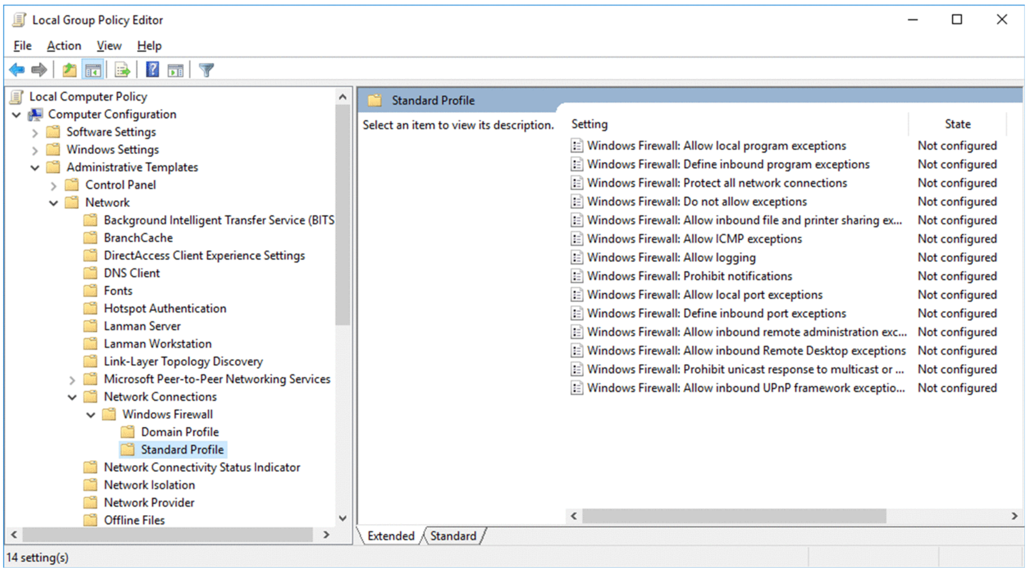
If the Windows firewall is enabled, you must perform this setting.

To set a firewall:

1. In Windows, choose **Start** and then **Run**.
2. Enter `gpedit.msc`, and then click the **OK** button.
   The Local Group Policy Editor window is displayed.

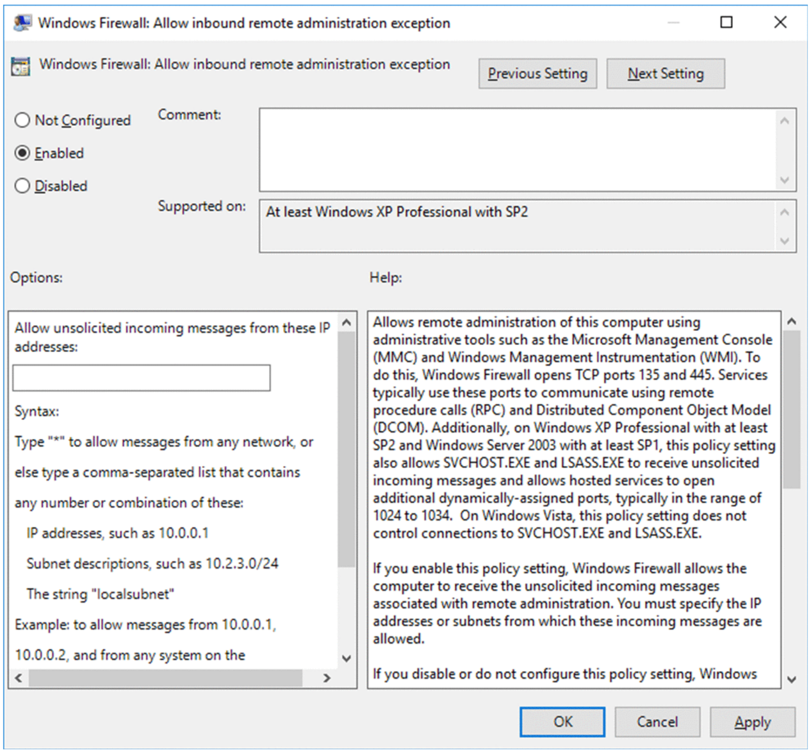


3. Click **Computer Configuration**, **Administrative Templates**, **Network**, **Network Connections**, and **Windows Firewall** in this order to expand the tree.



4. Click **Standard Profile**# and then, in the right-hand pane, from the right-click menu of **Windows Firewall: Allow inbound remote administration exception**, choose **Properties**.

   The Windows Firewall: Allow inbound remote administration exception Properties dialog box is displayed.



#

If the host is a domain environment, click **Domain Profile** instead.

5. Choose the **Setting** tab, and then select **Enabled**.

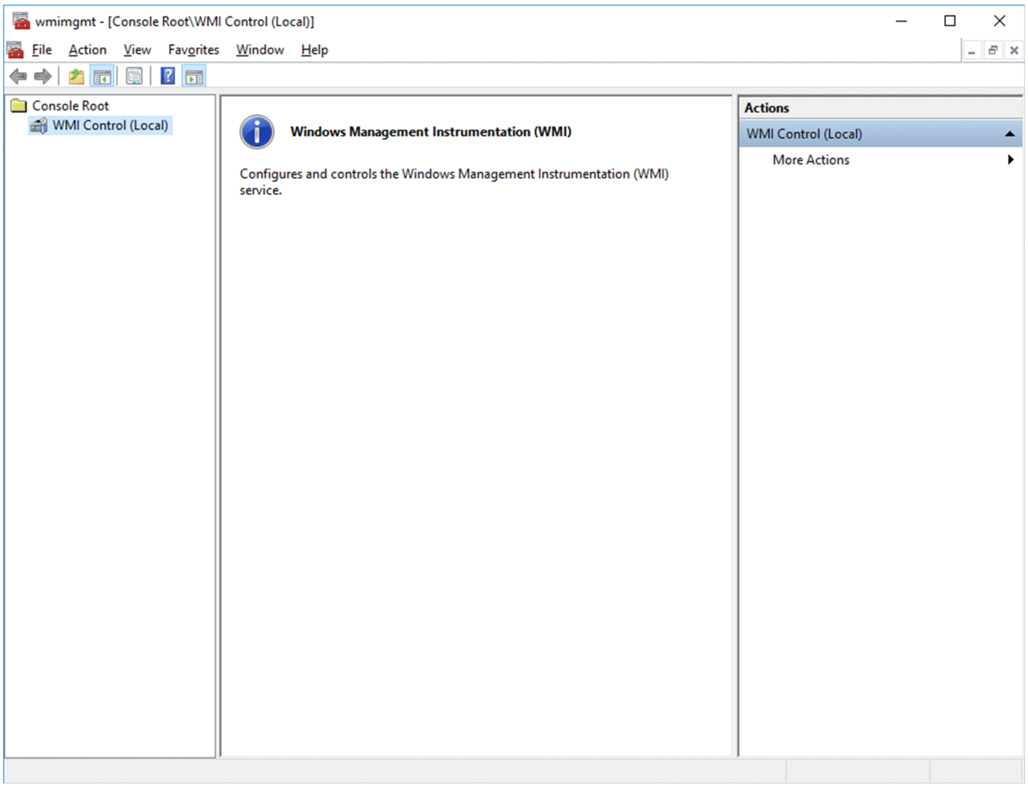
6. Click the **OK** button.
   The Windows Firewall: Allow inbound remote administration exception Properties dialog box closes.

(d) Setting a WMI namespace

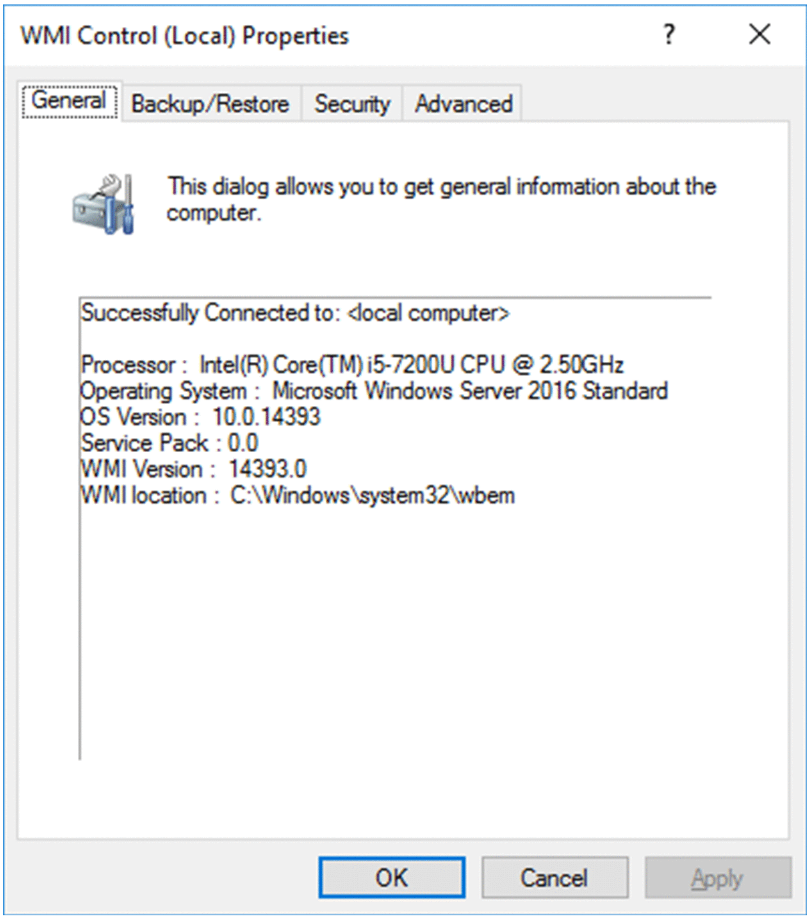
To set a WMI namespace:

1. In Windows, choose **Start** and then **Run**.

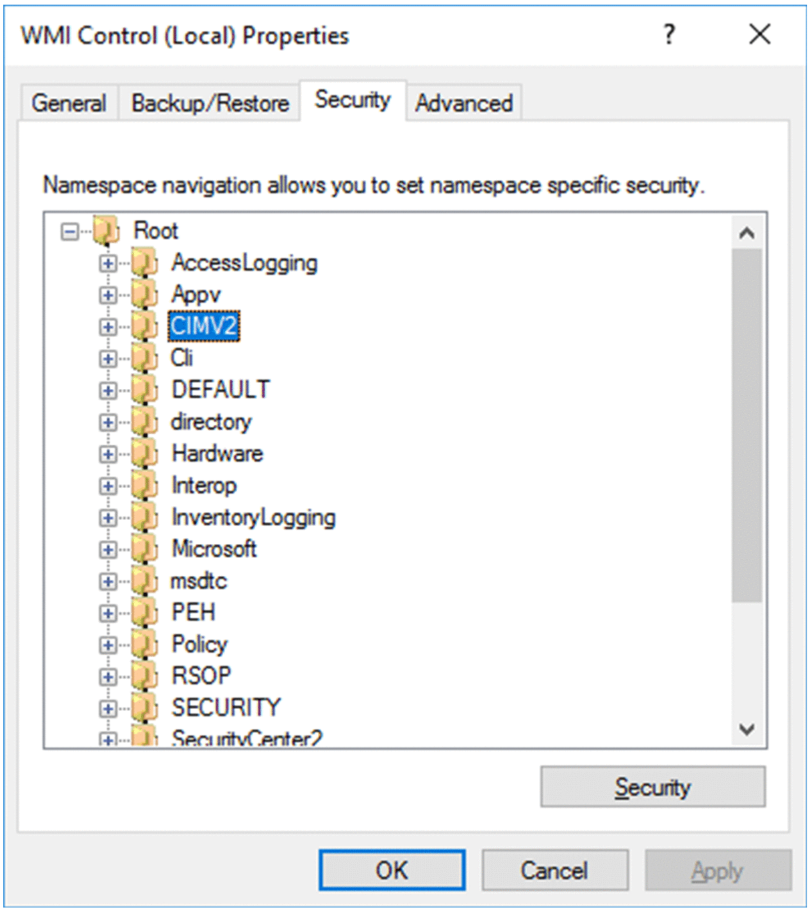
2. Enter `wmimgmt.msc`, and then click the **OK** button.
   The wmimgmt - [Console Root\WMI Control (Local)] window is displayed.



3. Choose **WMI Control (Local)** and then, from the right-click menu, choose **Properties**.
   The WMI Control (Local) Properties dialog box is displayed.



4. On the **Security** page, click **Root** and then **CIMV2**, and then select **CIMV2**.



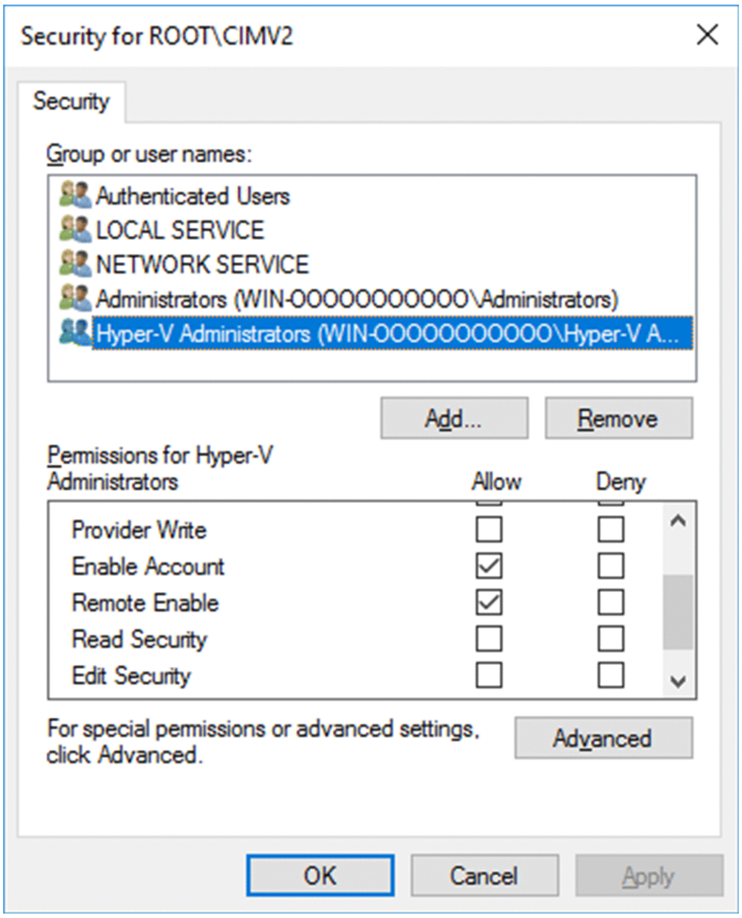
5. Click the **Security** button.

   The Security for ROOT\CIMV2 dialog box is displayed.

   Check if the user who connects to the monitored host or the group to which this user belongs is displayed in **Group or user names**. If the user or its group is not displayed, click the **Add** button and add the user or the group.

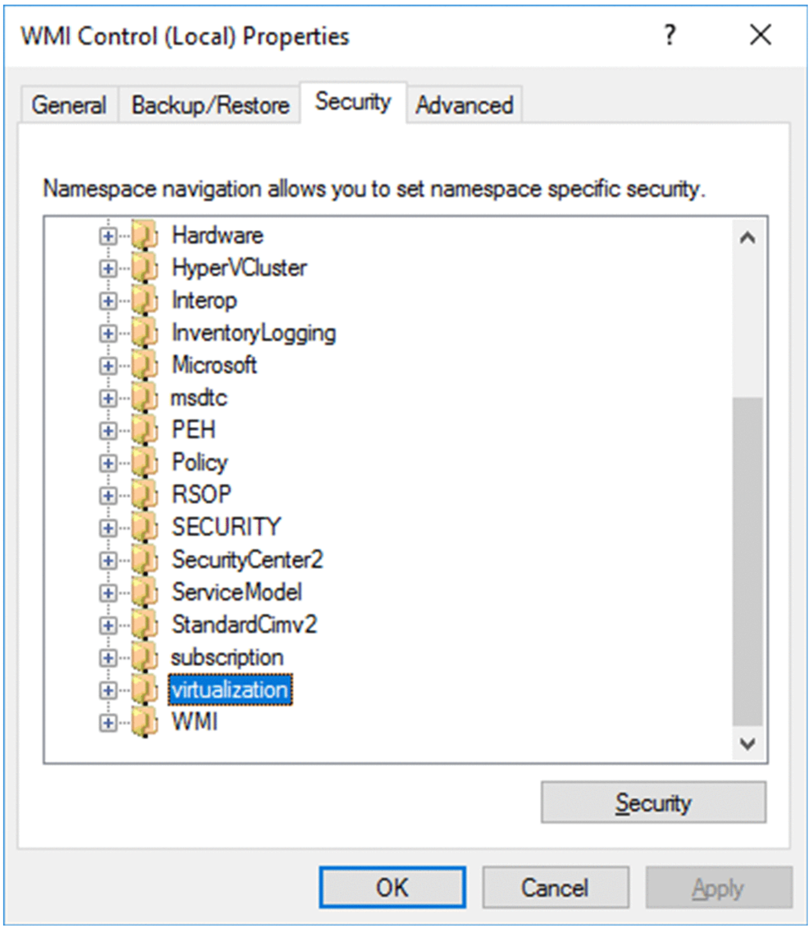
6. In **Group or user names**, select the user that connects to the monitored host or the group to which this user belongs.

   Check if the check boxes for **Allow** are selected for **Enable Account** and **Remote Enable**. If these check boxes are cleared, select them.



7. Click the **OK** button.
   The Security for ROOT\CIMV2 dialog box closes and the WMI Control (Local) Properties dialog box is displayed.

8. On the **Security** page, click **Root** and then **Virtualization**, and then select **Virtualization**.



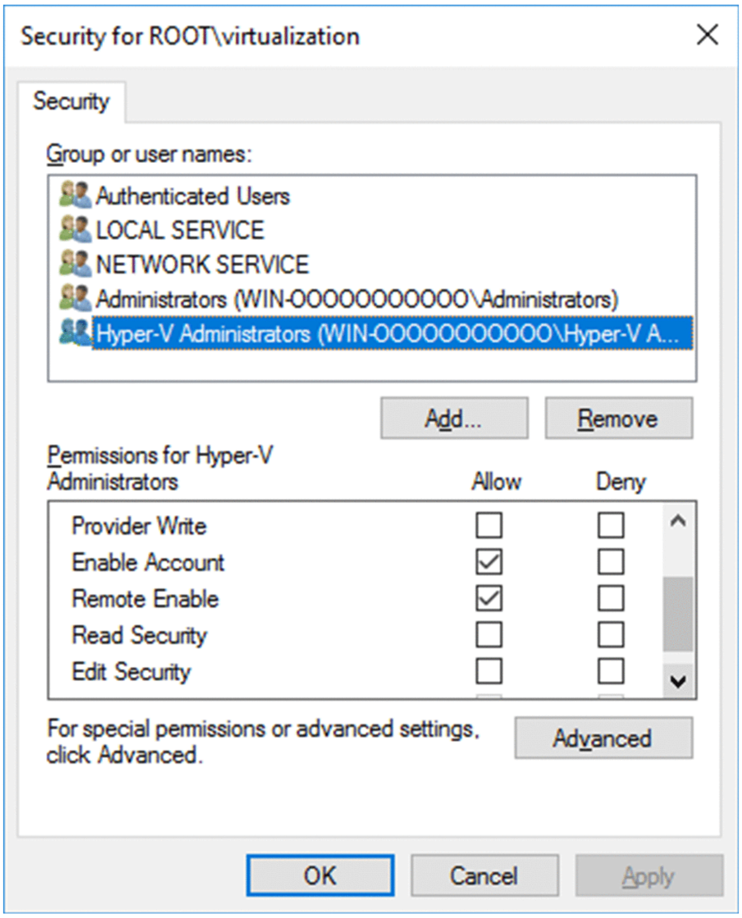
9. Click the **Security** button.

   The Security for ROOT\virtualization dialog box is displayed.

   Check if the user who connects to the monitored host or the group to which this user belongs is displayed in **Group or user names**. If the user or its group is not displayed, click the **Add** button and add the user or the group.

10. In **Group or user names**, select the user that connects to the monitored host or the group to which this user belongs.

    Check if the check boxes for **Allow** are selected for **Enable Account** and **Remote Enable**. If these check box are cleared, select them.



11. Click the **OK** button.

    The Security for ROOT\virtualization dialog box closes and the WMI Control (Local) Properties dialog box is displayed.

12. On the **Security** page, click **Root**, **virtualization**, and **V2**, and then select **V2**. (In Windows Server 2012 R2 or later)

13. Click the **Security** button. (In Windows Server 2012 R2 or later)

    The Security for ROOT\virtualization\v2 dialog box is displayed.

    Check if the user who connects to the monitored host or the group to which this user belongs is displayed in **Group or user names**. If the user or its group is not displayed, click the **Add** button and add the user or the group.

14. In **Group or user names**, select the user that connects to the monitored host or the group to which this user belongs. (In Windows Server 2012 R2 or later)

    Check if the check boxes for **Allow** are selected for **Enable Account** and **Remote Enable**. If these check boxes are cleared, select them.

15. Click the **OK** button. (In Windows Server 2012 R2 or later)

    The Security for ROOT\virtualization\v2 dialog box closes and the WMI Control (Local) Properties dialog box is displayed.

16. Click the **OK** button.

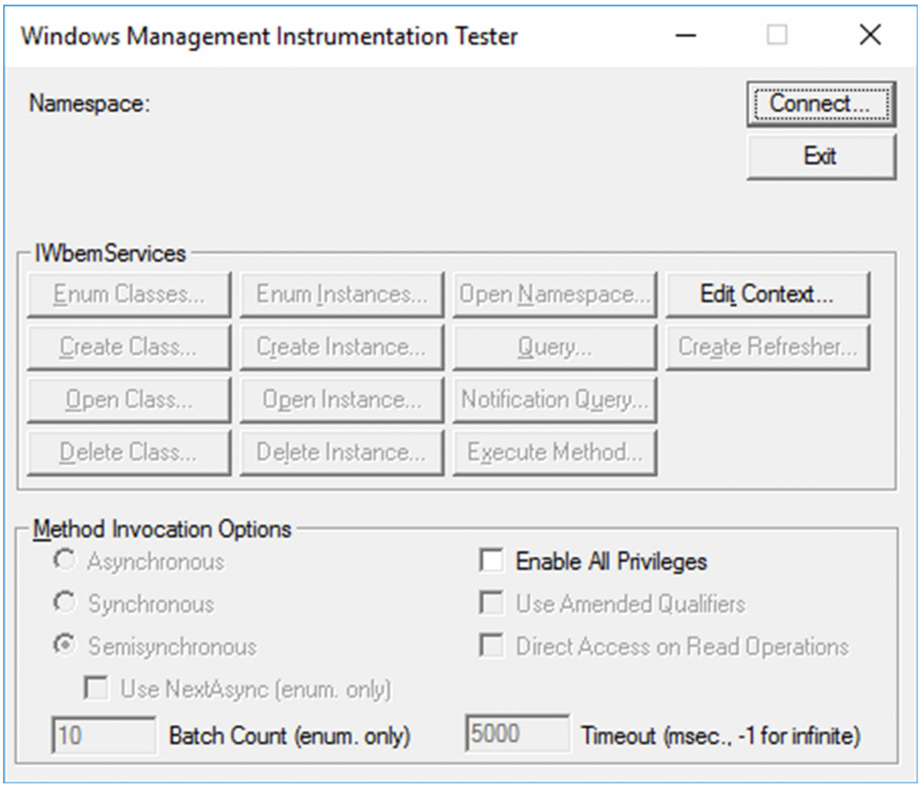
    The WMI Control (Local) Properties dialog box closes.

17. In the wmimgmt - [Console Root\WMI Control (Local)] window, from the **File** menu, choose **Exit** to close the window.

(2) Checking the WMI connection status

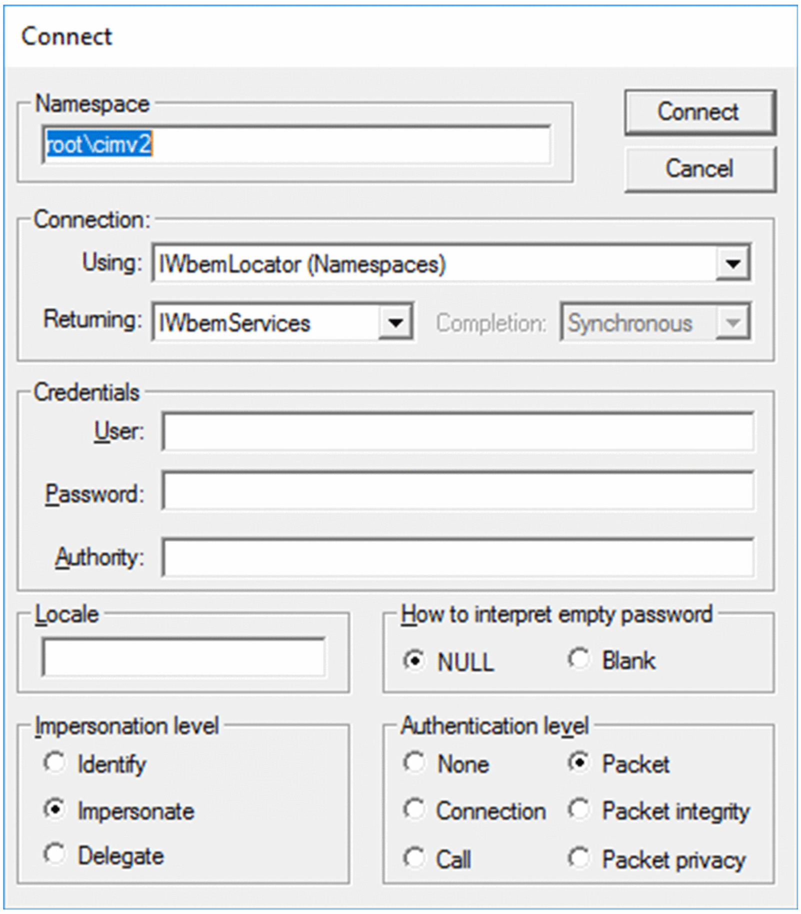
Use the `wbemtest.exe` Windows tool to check if the PFM - RM host is connected to the monitored host.

To check the WMI connection status (perform this procedure on the PFM - RM host):

1. At the command prompt, execute the following command:
   `runas /user:`*`user-name`*` wbemtest`
   If you specify a user who belongs to a domain, specify the user name in *user-name*`@`*domain-name* format.
   The Windows Management Instrumentation Tester window appears.
   

   Make sure that, for the user name, you specify the same value that is specified for `HostUserID` and `HostDomain` during setup of the instance environment. When the command prompts you to enter a password, specify the same value that is specified for `HostPassword`.
   For details about **HostUserID**, **HostDomain**, and **HostPassword**, see Table 2-5.

2. Click the **Connect** button.
   The Connect dialog box is displayed.



3. Enter appropriate information in **Namespace**, **User**, **Password**, and **Authority**.

   The following explains the information to be entered:

   - **Namespace**

     Enter `\\`*monitored-host-name*`\root\cimv2` or `\\`*monitored-host-name*`\root\virtualization`. For *monitored-host-name*, specify the value that is set for **Target_Host** in the monitoring target settings.

   - **User**

     Enter the user name used to log on to the monitored host. For this user name, specify the value set for **UserID** in the monitoring target settings.

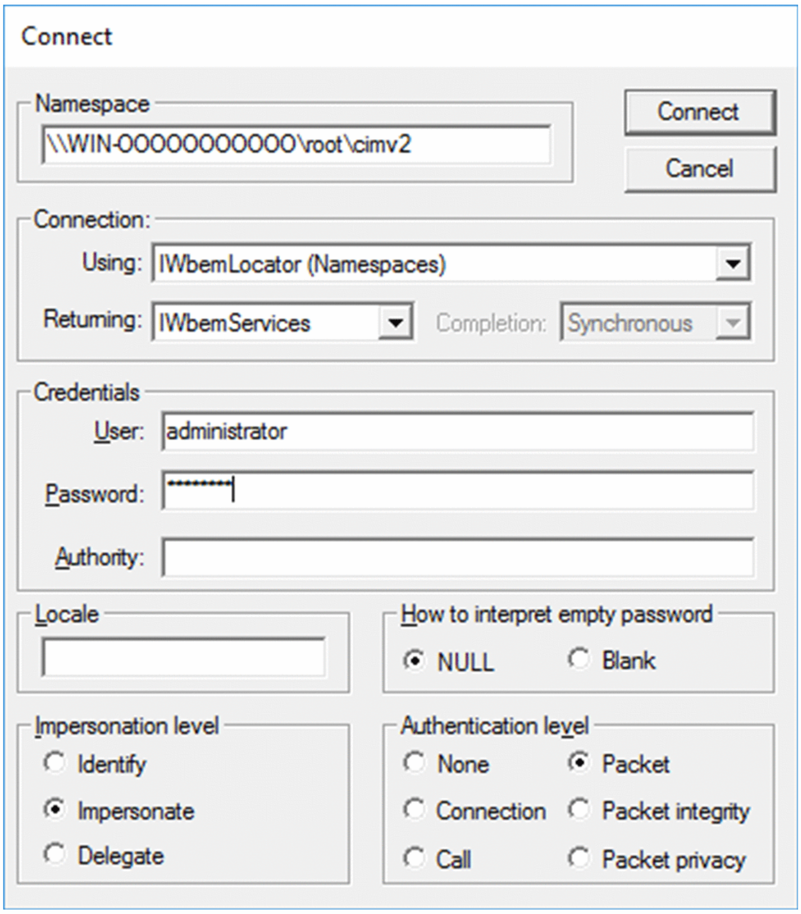
   - **Password**

     Enter the user's password. For the user password, specify the value set for **Password** in the monitoring target settings.

   - **Authority**

     Enter `ntlmdomain`**:***monitored-host's-domain-name*. If the monitored host is a work group, leave this item blank. For the domain name of the monitored host, specify the value set for **Domain** in the monitoring target settings.

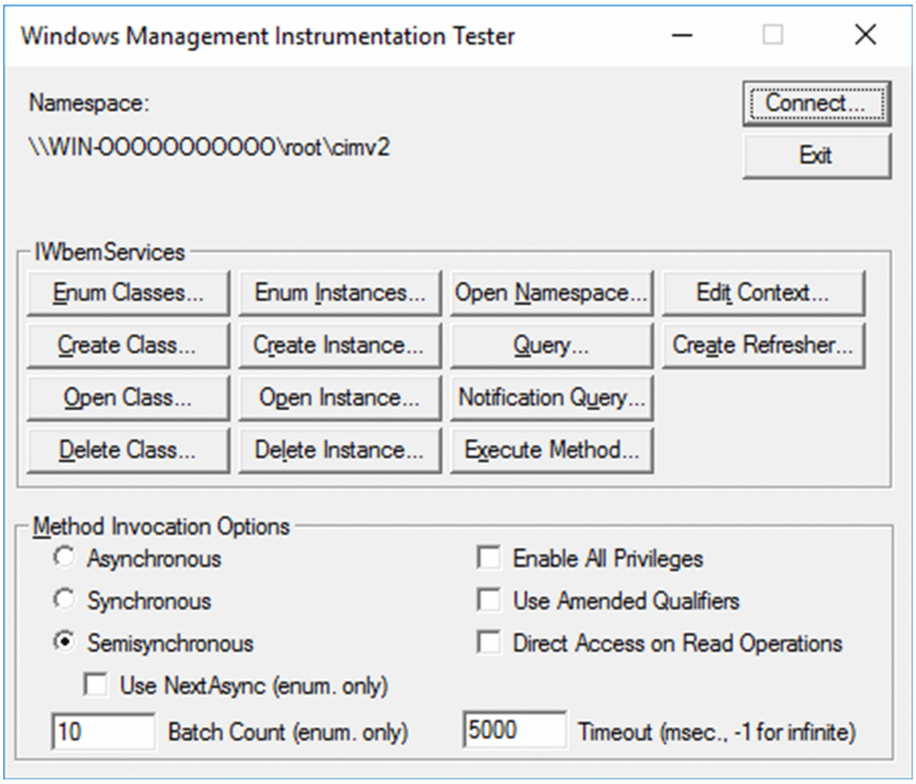
   The following shows an example:



For details about **Target_Host**, **UserID**, **Password**, and **Domain**, see Table 2-8.

4. Click the **Connect** button.

   If connection is established successfully, the Connect dialog box closes and all the buttons in the Windows Management Instrumentation Tester dialog box are enabled.



If an error dialog box is displayed, check the settings on the basis of the displayed error number. The error numbers and their causes are listed below.

Note that if you change settings while the `wbemtest.exe` tool is running, an error might result when you retry connection establishment. If this happens, restart the tool and then retry establishing connection.

- `0x8001011c`

  DCOM has not been set up on the PFM - RM host.

- `0x80070005`

  Possible causes are as follows:

  • DCOM has not been set up on the PFM - RM host.

  • DCOM has not been set up on the monitored host.

  • The user name, password, or domain name used to connect to the monitored host is invalid.

- `0x80041003`

  On the monitored host, **Namespace** has not been set for WMI.

- `0x80041008`

  The value specified in **Authority** does not begin with `ntlmdomain:`.

- `0x800706`*XX*

  Possible causes are as follows:

  • The monitored host name is invalid.

  • The monitored host is not running.

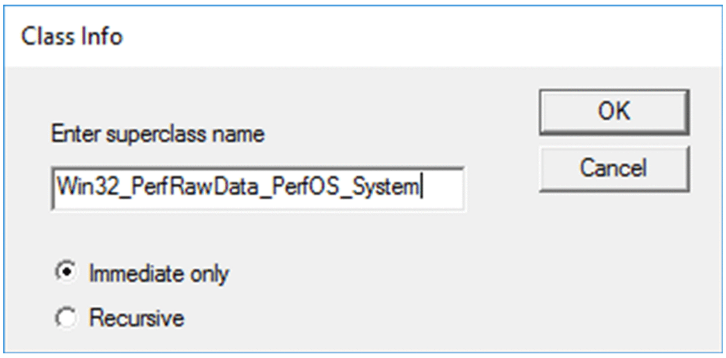
  • A firewall has not been set up on the monitored host.

  • The password for the user logging on to the monitored host has expired.

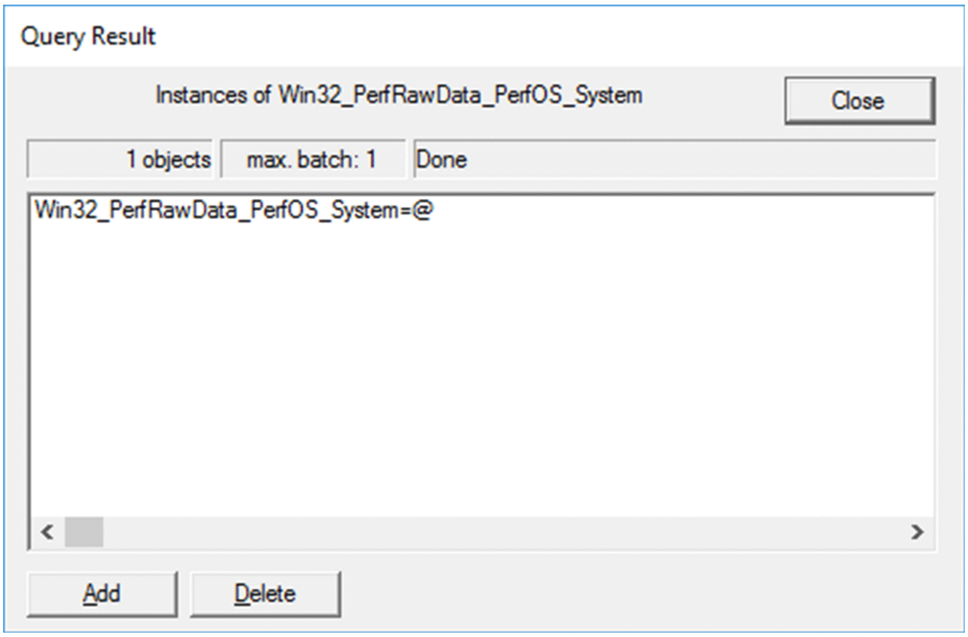
5. Click the **Enum Instances** button.

   The Class Info dialog box is displayed.

6. In **Enter superclass name**, enter `Win32_PerfRawData_PerfOS_System`, and then click the **OK** button.



The Query Result dialog box is displayed.
Check if the list contains the entry `Win32_PerfRawData_PerfOS_System=@`. If an error dialog box is displayed or this value is not displayed in the list, the user who connects to the monitored host might not be a member of the Administrators, Performance Log Users, or Performance Monitor Users group.
Note that if you change settings while the `wbemtest.exe` tool is running, an error might result when you retry instance enumeration. If this happens, restart the tool and then check again.



2.5.3 For KVM

If the monitoring-target host uses KVM, the host must be connected to an SSH server in order to collect performance data from the host.

For details about the SSH connection settings, see *2.5.7 SSH connection settings*.

2.5.4 For Docker environment

If the virtual environment to be monitored uses Docker environment, communication between PFM - RM for Virtual Machine and the virtual environment is encrypted using SSL/TLS. Therefore, the following certificates and passwords are required:

For the physical server running Docker environment to be monitored:

- Root certificate of the certificate authority

  A root certificate of the certificate authority is needed. In this case, the certificate authority must be the same certificate authority that issues a certificate of the physical server.

- Certificate of the physical server running Docker environment

  A certificate issued to the physical server is needed. In this case, the host name of the physical server is designated as an issuing destination of the certificate. This certificate is required for each physical server running Docker environment.

- Private key for the certificate of the physical server

  A private key, which is used when the certificate of the physical server is issued, is needed.

For the Windows server running PFM - RM for Virtual Machine:

- Root certificate of the certificate authority

  A root certificate of the certificate authority is needed. In this case, the certificate authority must be the same certificate authority that issues a certificate of the physical server.

- Client certificate used for connecting with Docker environment (Personal Information Exchange format)

  A client certificate must be issued from the certificate authority that issues the certificate of the physical server. If the issuers of the client certificate and the certificate of the physical server differ, monitoring cannot be performed. When all certificates of the physical server are issued by the same certificate authority, monitoring can be performed with a single client certificate. When the certificates of the physical server are issued from various certificate authorities, each client certificate must be issued from each certificate authority.

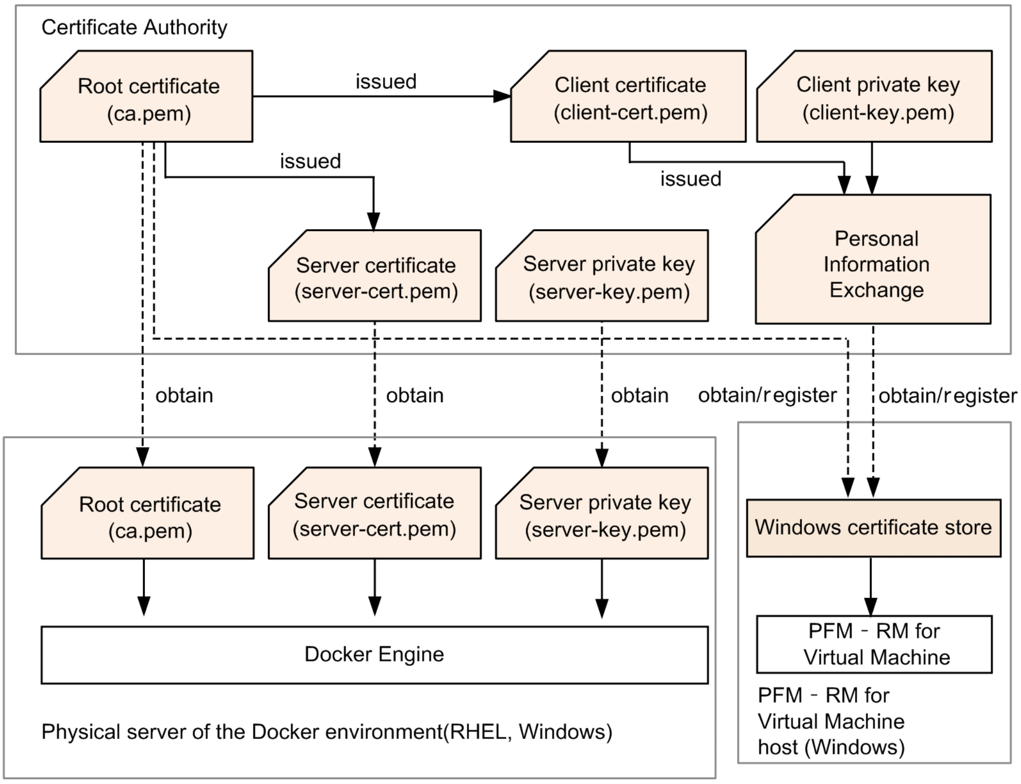
- Password for Personal Information Exchange

  For client certificates with Personal Information Exchange format, a password is set to protect the private key. This password is required when you register a certificate with the Windows certificate store.

  > **Important**
  >
  > If no CA certificate has been embedded when you use SSL/TLS to communicate with Docker environment, the following problems may occur:
  >
  > - During performance data collection, it may take quite a while to receive a response from the Docker environment at the connection destination.
  > - Because of the delay in receiving a response from the Docker environment at the connection destination, performance data collection might not be completed within the prescribed collection interval, resulting in a collection failure.
  >
  > If the client certificate is not embedded, collection fails due to connection rejection by the Docker environment.

The following figure shows how to place each certificate.

Figure 2–6: Placement of certificates



The Docker environment specifies root certificate files, server certificate files and server private key files using arguments during execution of the Docker Engine.

PFM - RM for Virtual Machine uses the certificates and private keys that have been registered with the Windows certificate store.

(1) Settings of target for monitoring

The Docker environment daemon enables certificates and TCP to change firewall settings of the physical server.

(a) Placing certificates

Place the root certificate, server certificates, and private keys in a location where the certificates and keys can be accessed by the Docker Engine.

For Docker environment (Linux)

An example of placing on `/etc/docker/certs.d`

Root certificates : `/etc/docker/certs.d/ca.pem`

Server certificates : `/etc/docker/certs.d/server-cert.pem`

Server private keys : `/etc/docker/certs.d/server-key.pem`

For Docker environment (Windows)

An example of placing on `C:\ProgramData\docker\certs.d`

Root certificates : `C:\ProgramData\docker\certs.d\ca.pem`

Server certificates : `C:\ProgramData\docker\certs.d\server-cert.pem`

Server private keys : `C:\ProgramData\docker\certs.d\server-key.pem`

(b) Enabling TCP connection and SSL/TLS

Set a TCP port number and certificates necessary for encrypted communication so that you can connect with the Docker environment remotely.

For Docker environment (Linux)

Add `-H`, `--tlsverify`, `--tlscacert`, `--tlscert`, and `--tlskey` options as arguments of `OPTIONS` for the `/etc/sysconfig/docker` file.

An example of placing the certificate files on `/etc/docker/certs.d` and receiving them at the port number `XXXX` is shown below.

```
OPTIONS='--selinux-enabled --log-driver=journald
  --tlsverify --tlscacert=/etc/docker/certs.d/ca.pem
  --tlscert=/etc/docker/certs.d/server-cert.pem
  --tlskey=/etc/docker/certs.d/server-key.pem
  -H unix:///var/run/docker.sock -H tcp://0.0.0.0:XXXX'
```

To enable the changes, restart the Docker Engine.

An example of a command to restart the Docker daemon is shown below.

```
systemctl restart docker
```

For Docker environment (Windows)

Add the `hosts`, `tlsverify`, `tlscacert`, `tlscert`, and `tlskey` options to the file `C:\ProgramData\Docker\config\daemon.json`.

An example of placing the certificate files on `C:\ProgramData\docker\certs.d` and receiving them at the port number `XXXX` is shown below.

```
{
"hosts": ["tcp://0.0.0.0:XXXX", "npipe://"],
"tlsverify": true,
"tlscacert": "C:\\ProgramData\\docker\\certs.d\\ca.pem",
"tlscert": "C:\\ProgramData\\docker\\certs.d\\server-cert.pem",
"tlskey": "C:\\ProgramData\\docker\\certs.d\\server-key.pem"
}
```

To enable the changes, restart the Docker Engine service.

The Docker Engine service you need to restart:

Service name : Docker

Indication name : Docker Engine

(c) Changing the firewall settings

If firewall is enabled, change the settings to allow remote connection.

For Docker environment (Linux)

An example of a port number `XXXX` is shown below.

```
firewall-cmd --permanent --zone=public --add-port=XXXX/tcp
firewall-cmd --reload
```

For Docker environment (Windows)

An example of a port number XXXX is shown below.

```
netsh advfirewall firewall add rule name="Docker Engine"
protocol=TCP dir=in localport=XXXX action=allow
```

(2) Setting PFM - RM for Virtual Machine

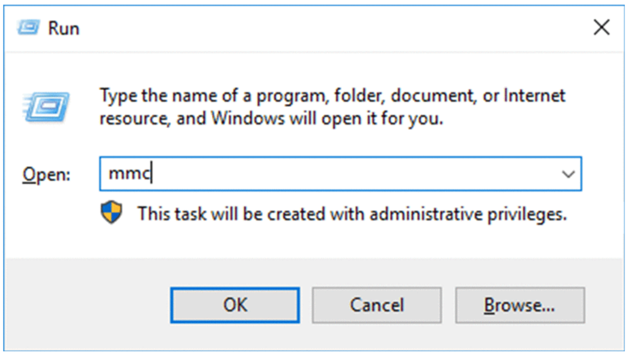
When you set target for monitoring for PFM - RM for Virtual Machine, set the following parameters:

- Target Host
  Set the host name of the physical server running the Docker environment to be monitored. The host name of the physical server must be resolved by the PFM - RM for Virtual Machine host.
- Security
  Specify a value in the range from 1 to 3 for encrypted communication.
- Port
  Set a TCP port number specified for the Docker environment.

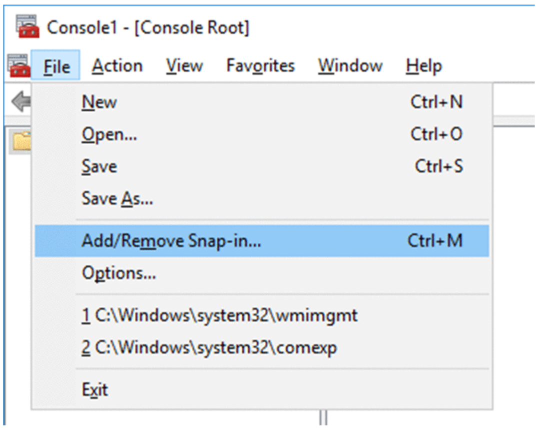
To enable encrypted communication with the Docker environment, register certificates.

(a) Registering a root certificate

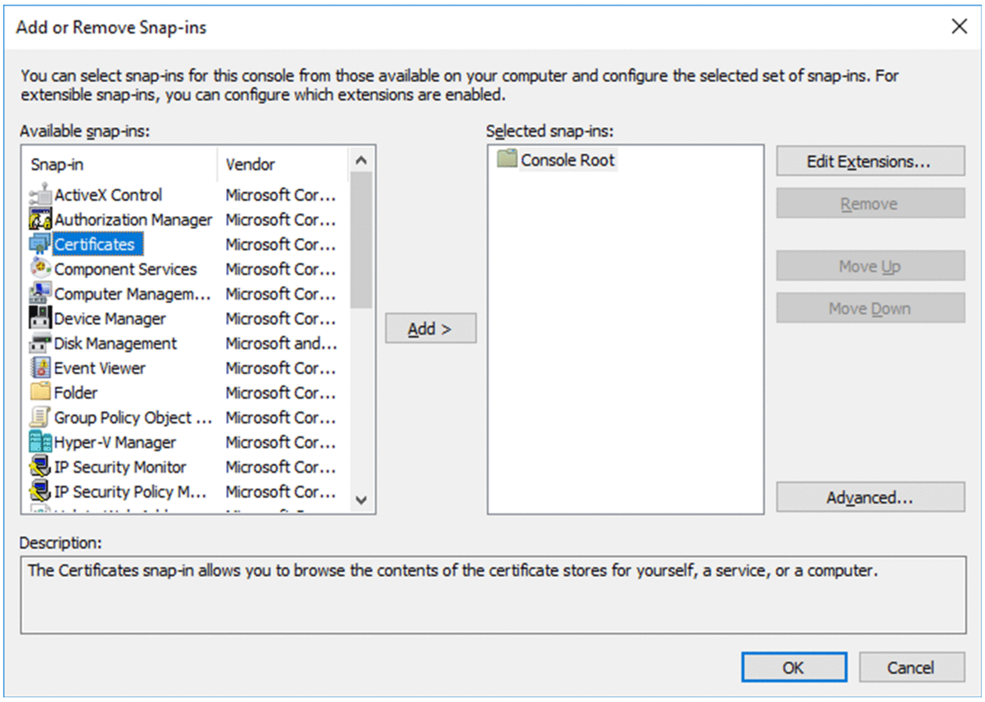
1. In Windows, choose **Start** and then **Run**.
   The Run dialog box opens.



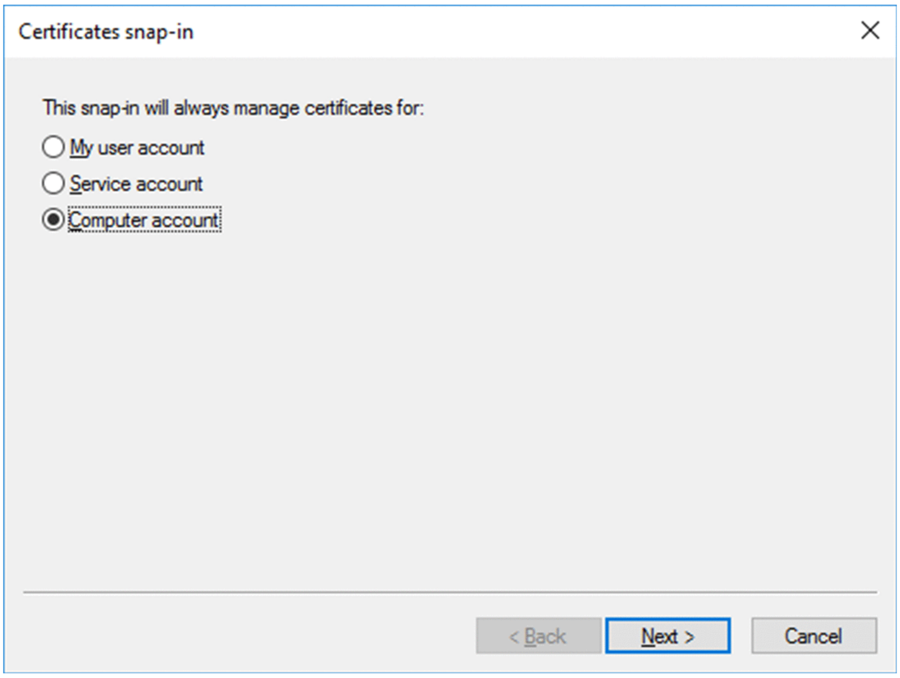
2. In the Run dialog box, enter mmc and click **OK**.
   Management Console starts.



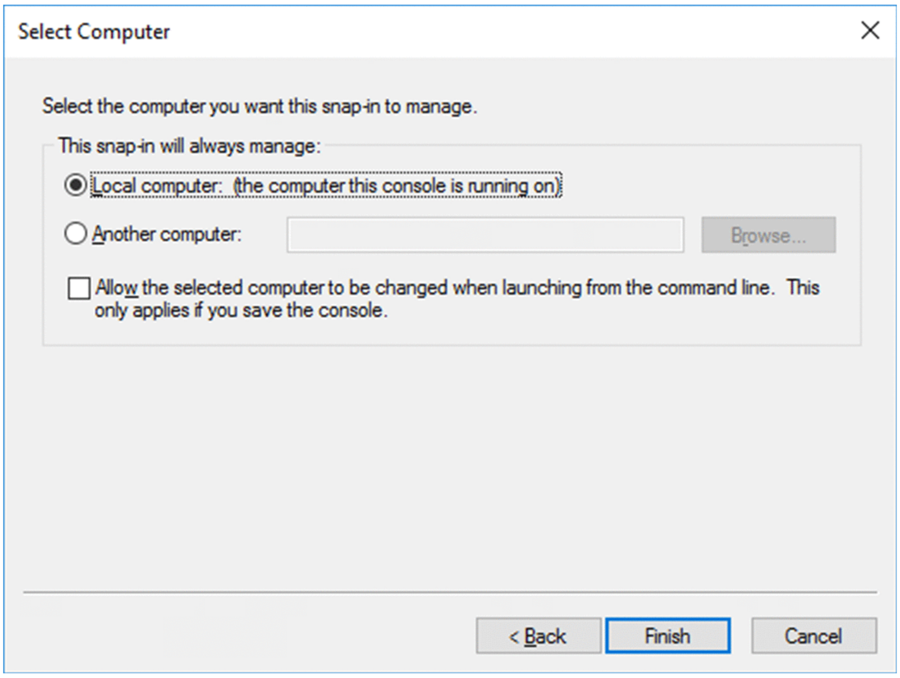
3. In **Console1**, choose **File** and then **Add/Remove Snap-in**.
   The Add/Remove Snap-in dialog box opens.



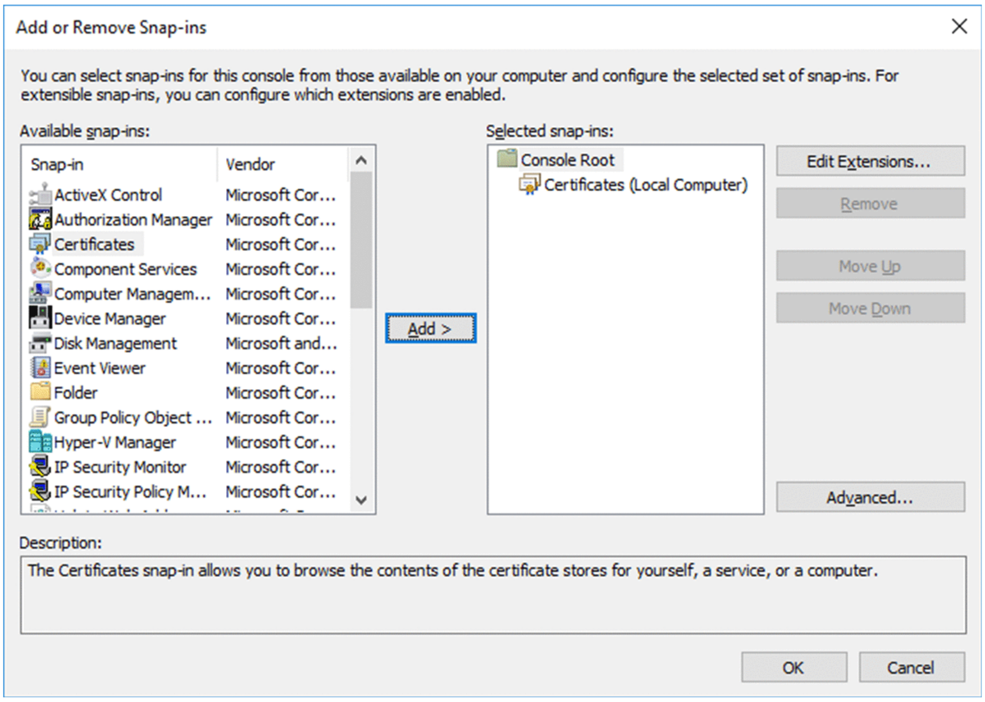
4. Choose Certificates and then click **Add**.
   The Certificates snap-in dialog box opens.



5. Choose **Computer account** and then click **Next**.
   The Select Computer dialog box opens.

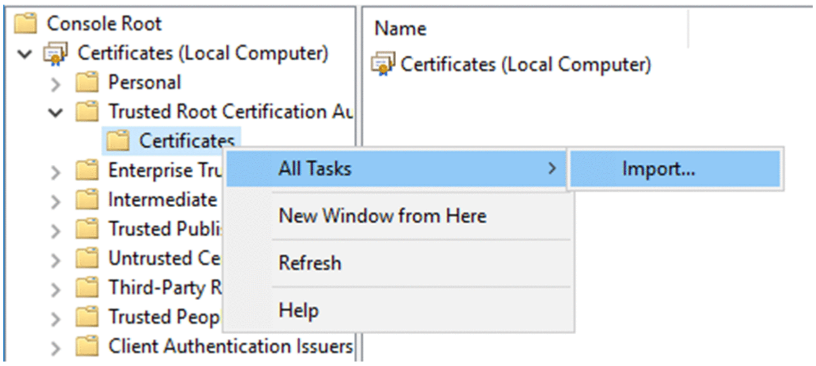


6. Choose **Local computer** and click **Finish**.

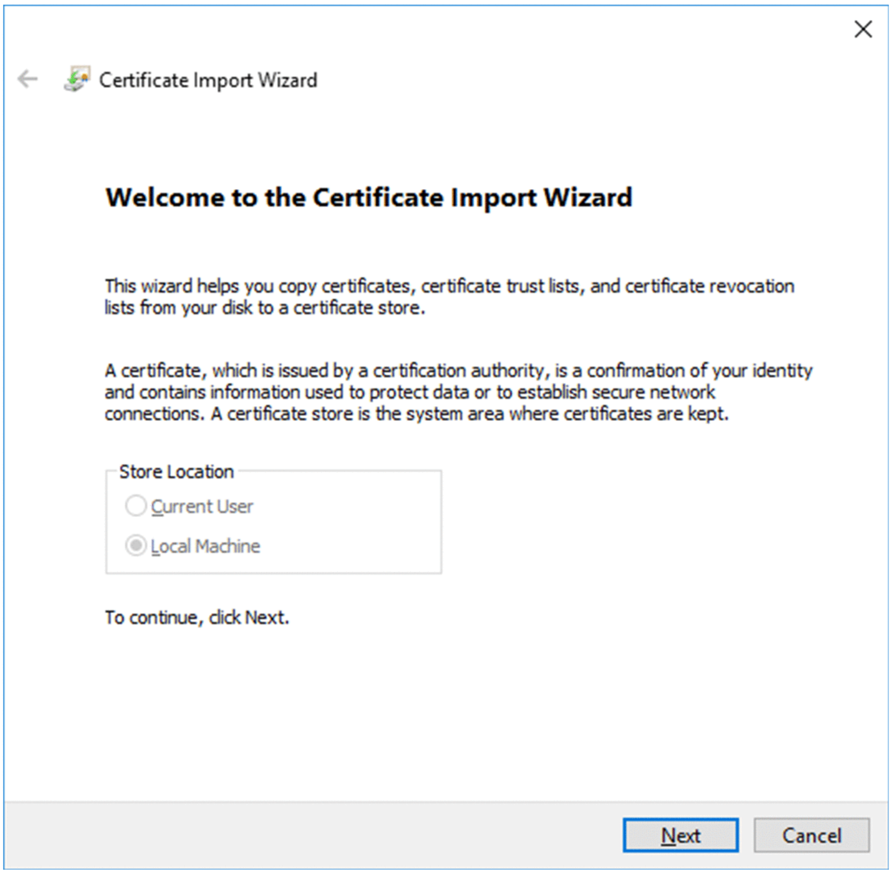


7. Check that **Certificates (Local Computer)** is added to **Selected snap-ins** and click **OK**.

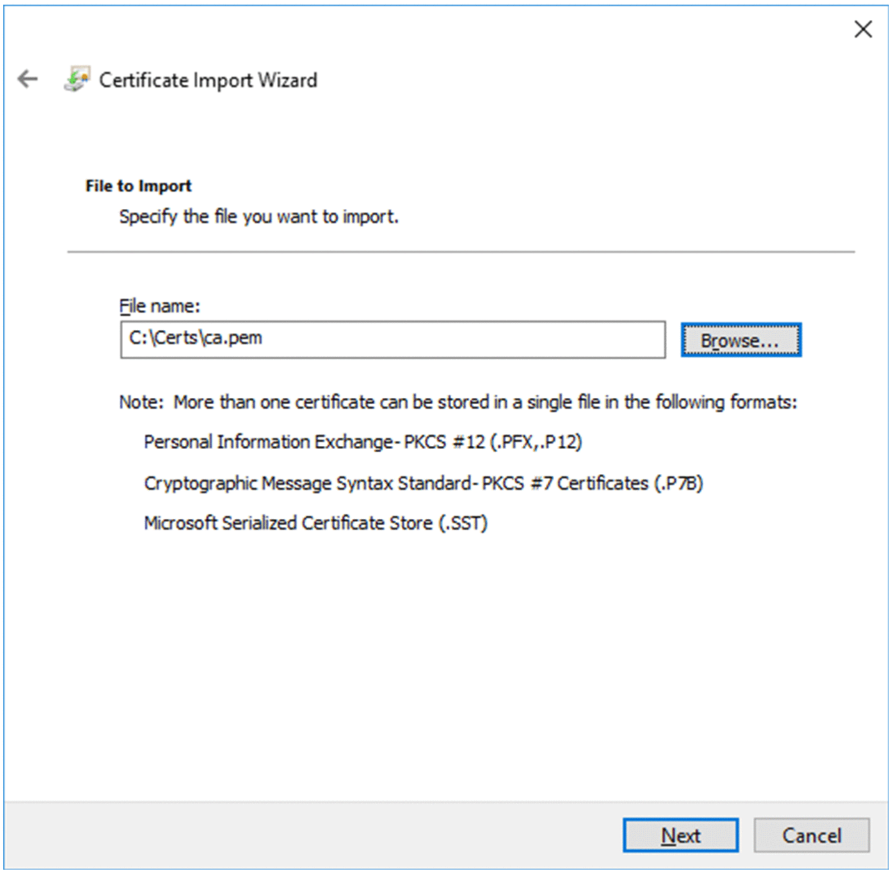
8. Expand **Certificates (Local Computer)** and right-click **Certificates** under **Trusted Root Certification Authorities**. Then click **All Tasks** and **Import** from the displayed menu items.



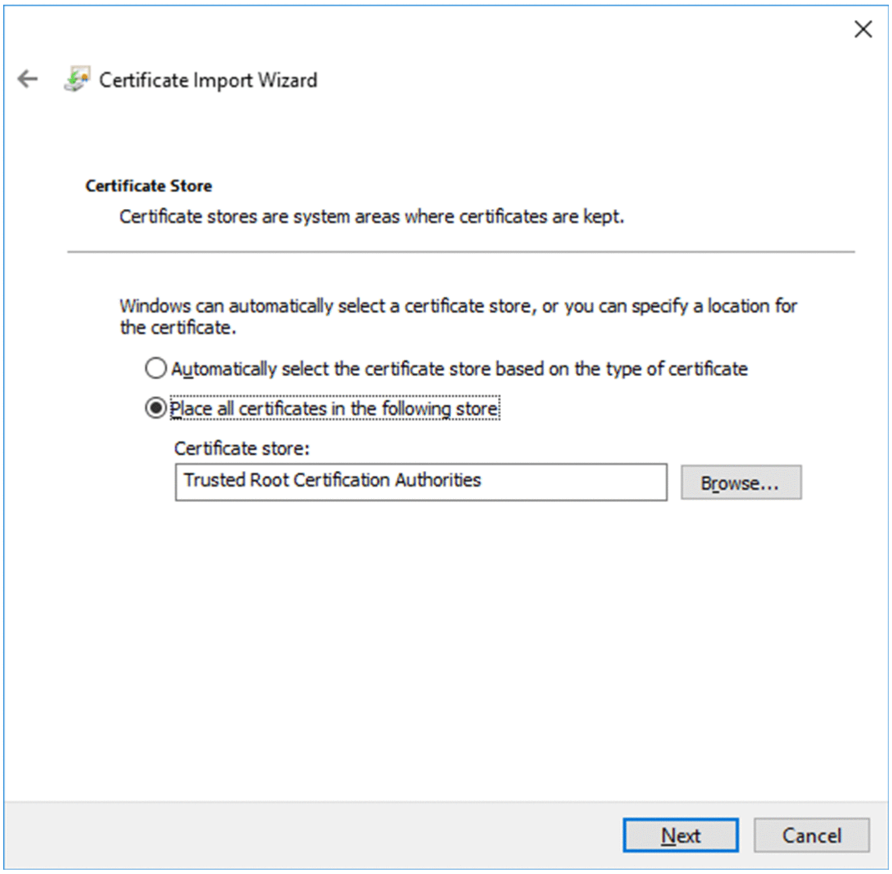
The Certificate Import Wizard dialog box opens.



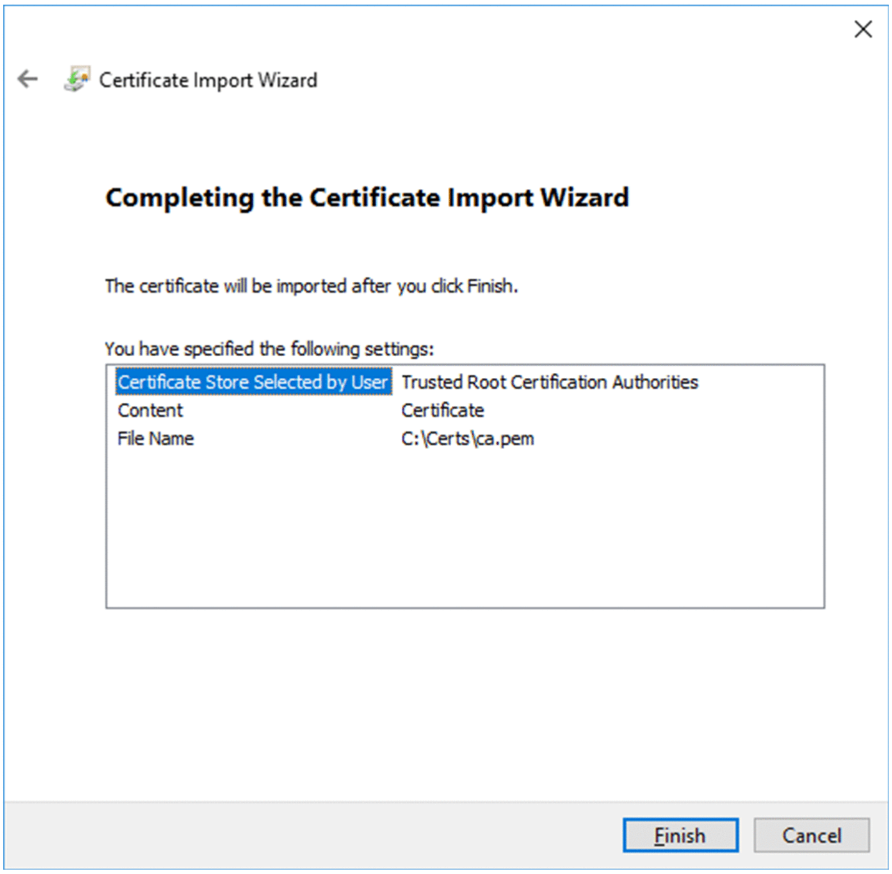
9. Click **Next**.



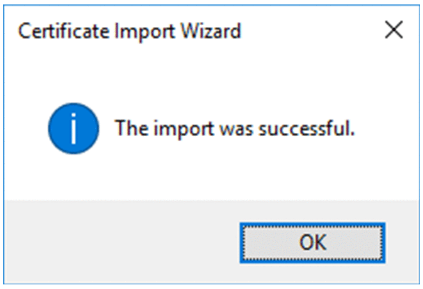
10. In the **File name** text box, enter the file name under which to save the certificate, and then click **Next**.
    Here, `C:\Certs\ca.pem` is entered as an example.
    Check that the certificate store is set as **Trusted Root Certification Authorities**.



11. Choose **Place all certificates in the following store,** and then click **Next**.



12. Click **Finish**.

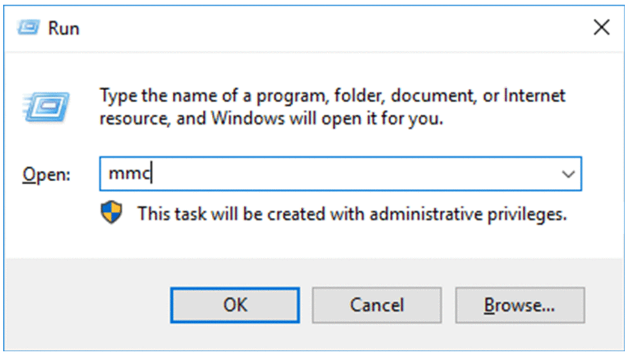


13. Click **OK** and check that the certificate has been successfully imported.

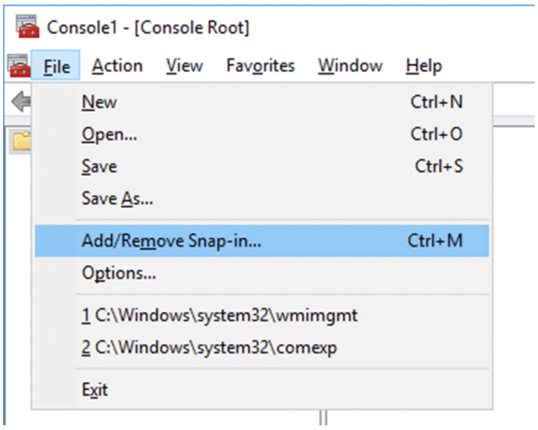
(b) Registering a client certificate

Log on as a user of the PFM - RM host (that is, user specified as `HostUserID` in the instance environment setting).

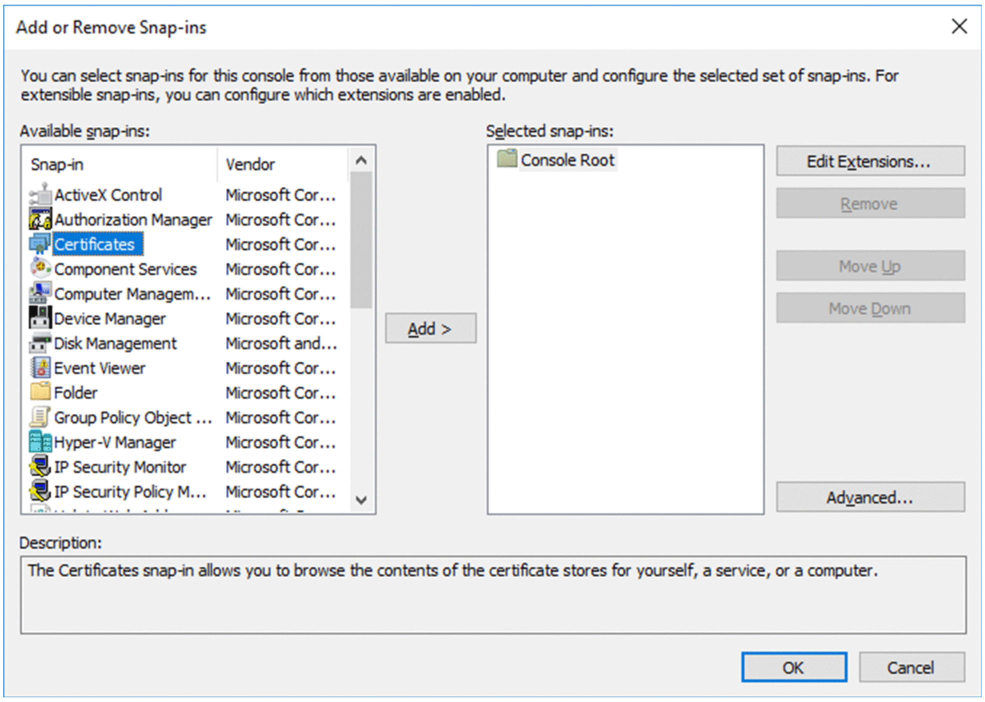
1. In Windows, choose **Start** and then **Run**.
   The Run dialog box opens.



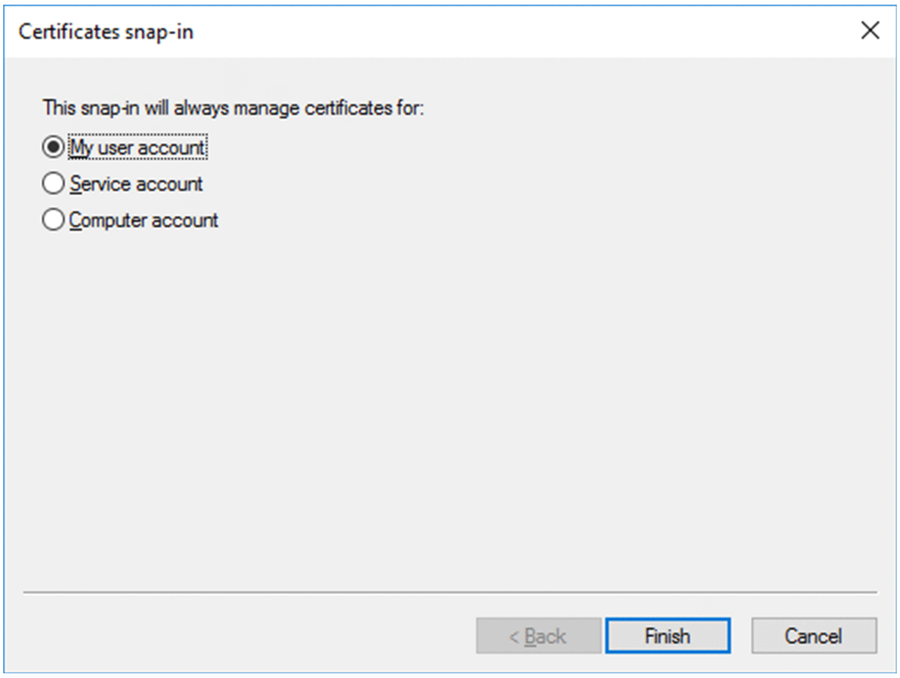
2. In the Run dialog box, enter `mmc` and click **OK**.
   Management Console starts.



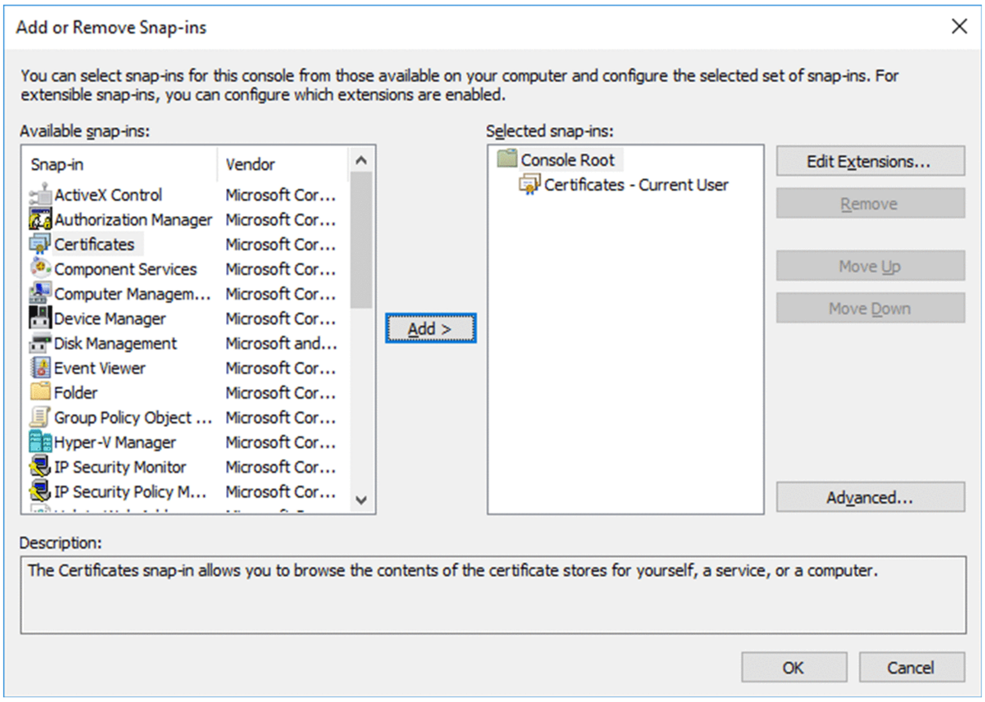
3. In **Console1**, choose **File** and then **Add/Remove Snap-in**.
   The Add/Remove Snap-in dialog box opens.



4. Choose Certificates and then click **Add**.
   The Certificates snap-in dialog box opens.

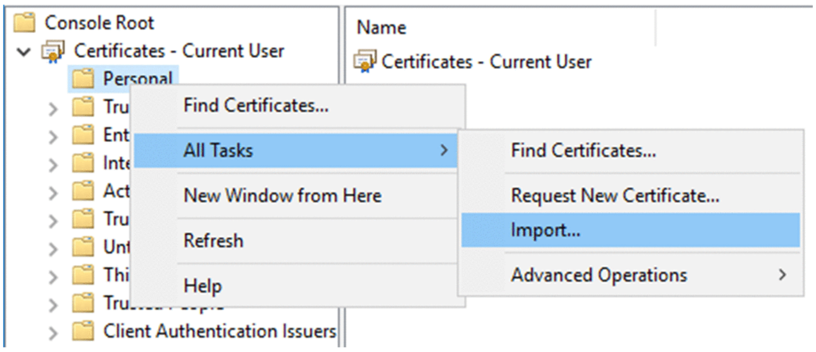


5. Select **My user account** and click **Finish**.

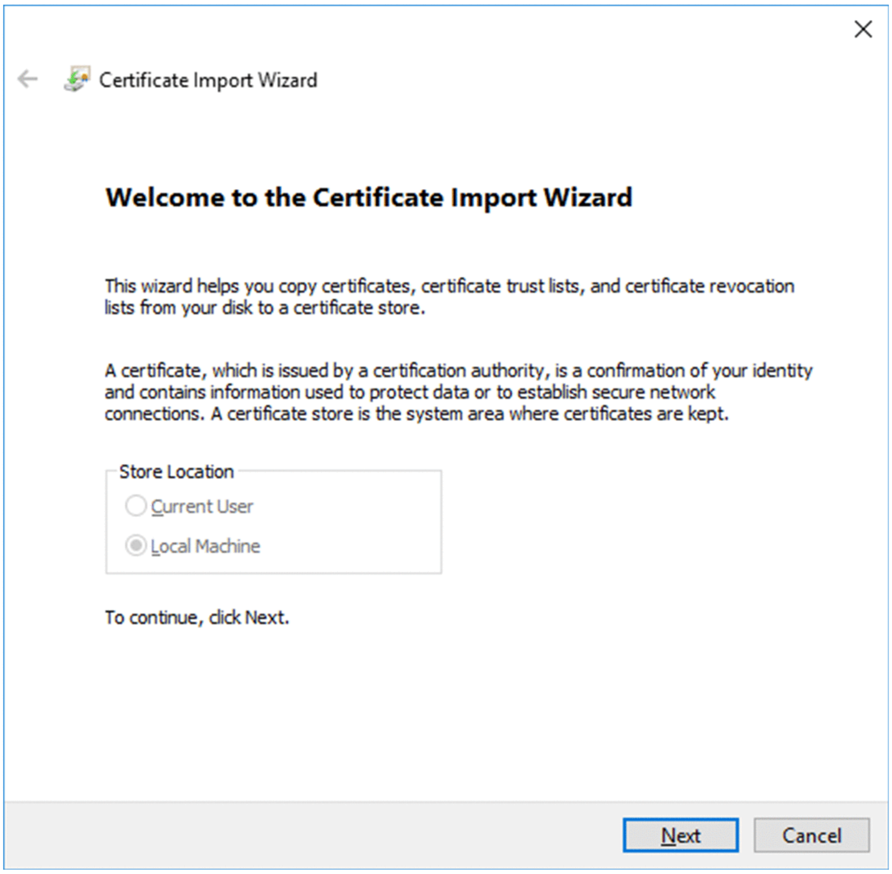


6. Check that **Certificates - Current User** is added to **Selected snap-ins** and click **OK**.

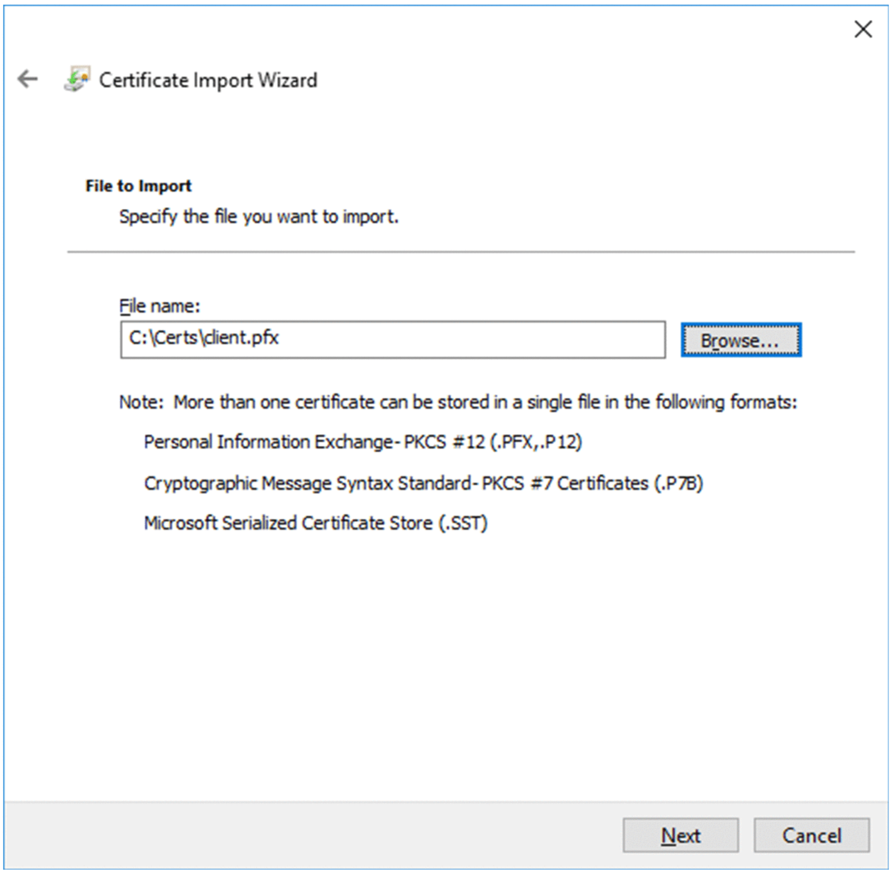
7. Expand **Certificates - Current User** and right-click **Personal**. Then click **All Tasks** and **Import** from the displayed menu items.



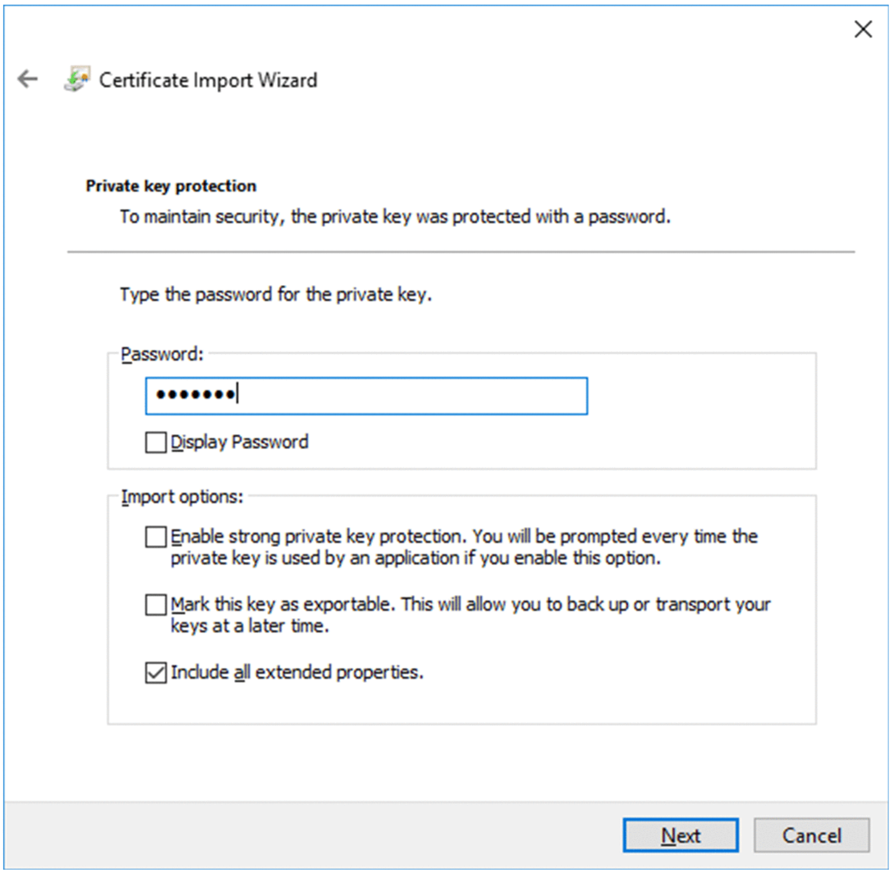
The Certificate Import Wizard dialog box opens.


Certificate Import Wizard
Welcome to the Certificate Import Wizard
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.
A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.
Store Location
Current User
Local Machine
To continue, click Next.
Next
Cancel

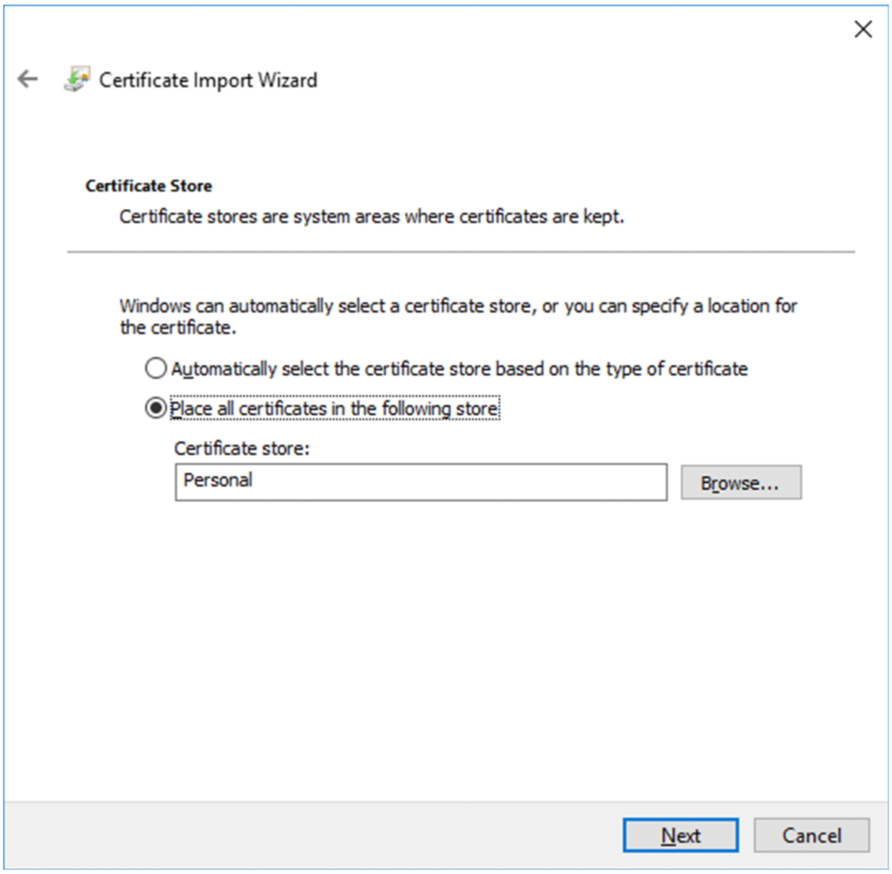
8. Click **Next**.


Certificate Import Wizard
File to Import
Specify the file you want to import.
File name:
C:\Certs\client.pfx
Browse...
Note: More than one certificate can be stored in a single file in the following formats:
Personal Information Exchange- PKCS #12 (.PFX,.P12)
Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)
Microsoft Serialized Certificate Store (.SST)
Next
Cancel

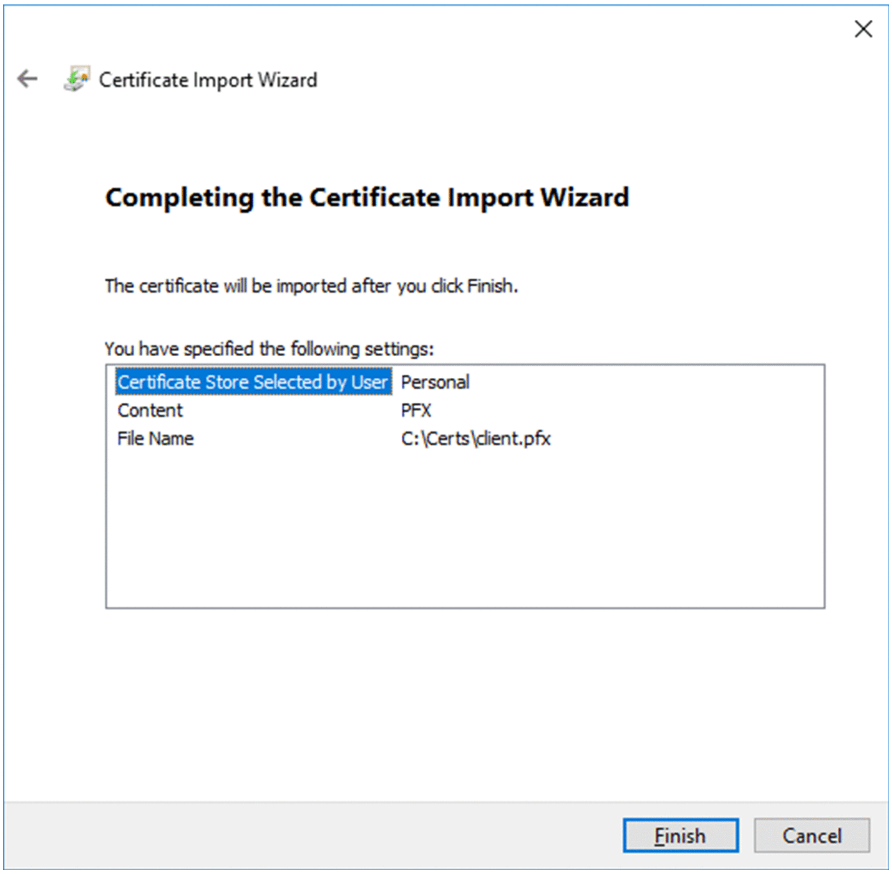
9. In the **File name** text box, enter the file name under which to save the certificate, and then click **Next**.
Here, `C:\Certs\client.pfx` is entered as an example.



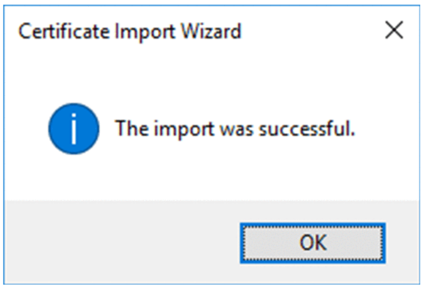
10. Specified the Personal Information Exchange's password and then click **Next**.
Check that the certificate store is set as **Personal**.



11. Choose **Place all certificates in the following store**, and then click **Next**.



12. Click **Finish**.



13. Click **OK** and check that the certificate has been successfully imported.

2.5.5 For Podman environment

If the monitoring-target host uses Podman environment, the host must be connected to an SSH server in order to collect performance data from the host.

For details about the SSH connection settings, see *2.5.7 SSH connection settings*.

2.5.6 For logical partitioning feature

When the virtual environment to be monitored is one with logical partitioning feature, you must specify the IP address of the machine on which the monitoring agent is installed for that environment, in addition to performing setup through PFM - RM for Virtual Machine. However, if the monitoring agent has multiple IP addresses, specify the preferred IP address to be selected and used by the OS of the monitoring agent, that is, the IP address used by the monitoring agent, for the environment with logical partitioning feature. Also, when the system operates in a cluster system, specify the IP addresses of the executing node and standby node for the environment with logical partitioning feature, not the logical IP addresses.

For details about how to set up the IP address of the monitoring agent host in an environment with logical partitioning feature, see the manual *Hitachi Compute Blade User's Guide*.

When the virtual environment to be monitored is one with logical partitioning feature, you must configure both the host on which the monitoring agent is installed and the environment with that feature.

To configure the environment:

1. Copy the `HvmSh` command (`HvmSh.exe`) from the management tool for Hitachi Compute Blade logical partitioning feature, which is included with that feature, to the following folder on the PFM - RM host:

```
installation-folder\agt8\plugin\jpcagt5virtage.d\
```

2. In the environment with logical partitioning feature to be monitored, set the IP address of the machine on which the monitoring agent is installed.

Important

The collection interval of records of the monitoring logical partitioning feature When you monitor logical partitioning feature using this product, Set collection interval of all performance data for instances to monitor logical partitioning feature to under ten minutes. When you monitor logical partitioning feature and set collection interval of performance data more than ten minutes, the performance data are not collected by the specifications of logical partitioning feature firmware and the `HvmSh` command.

2.5.7 SSH connection settings

This subsection describes how to set up an SSH connection. To use SSH, the PFM - RM host must have PuTTY or OS-standard OpenSSH installed. If SSH connection settings are not specified, PFM - RM for Virtual Machine will not be able to collect performance data. The settings for public key authentication must also be specified because public key authentication is used to authenticate the SSH server. In addition, because performance data is collected by using OS commands, necessary software and RPM packages might need to be installed on the PFM - RM host and the monitoring-target host.

Notes on installing PuTTY:

- Perform installation as a member of the Administrators group.
- Make sure that the name of the installation folder does not include multi-byte characters.

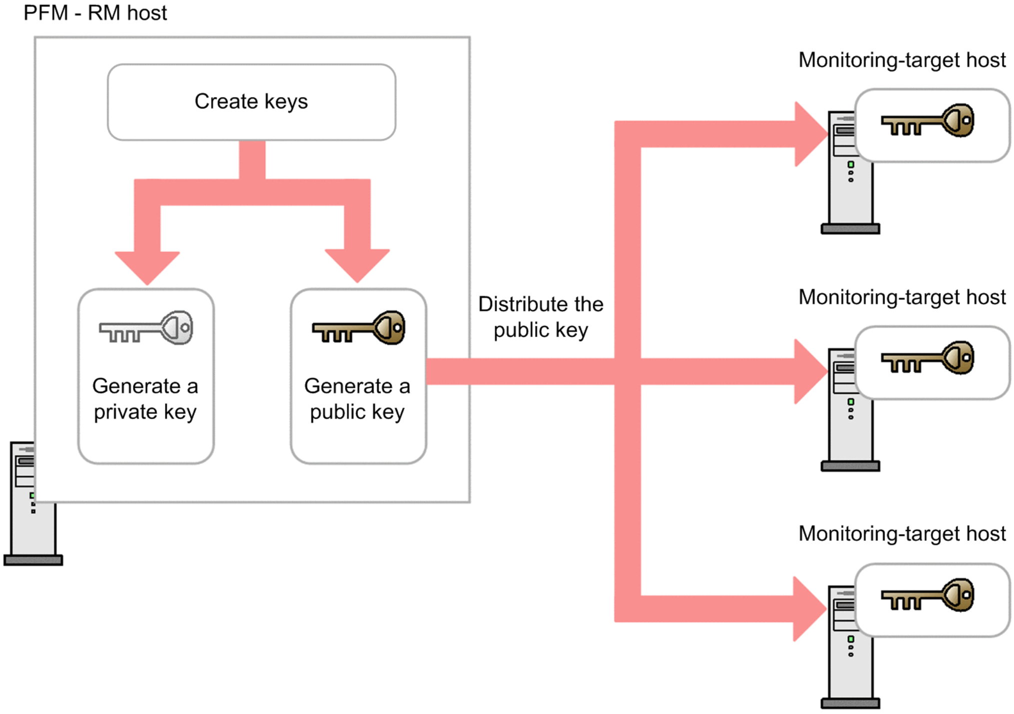
(1) Specifying the SSH connection settings

To enable connection to an SSH server, the following operations must be performed:

- Enable public key authentication of the SSH server.
  Do this on the monitoring-target host.
- Create keys.
  Do this on the PFM - RM host.
- Deploy the private key on the PFM - RM host.
  Do this on the PFM - RM host.
- Deploy the public key on the monitoring-target host.
  Do this on the monitoring-target host.

The following figure shows the concept of public key authentication.

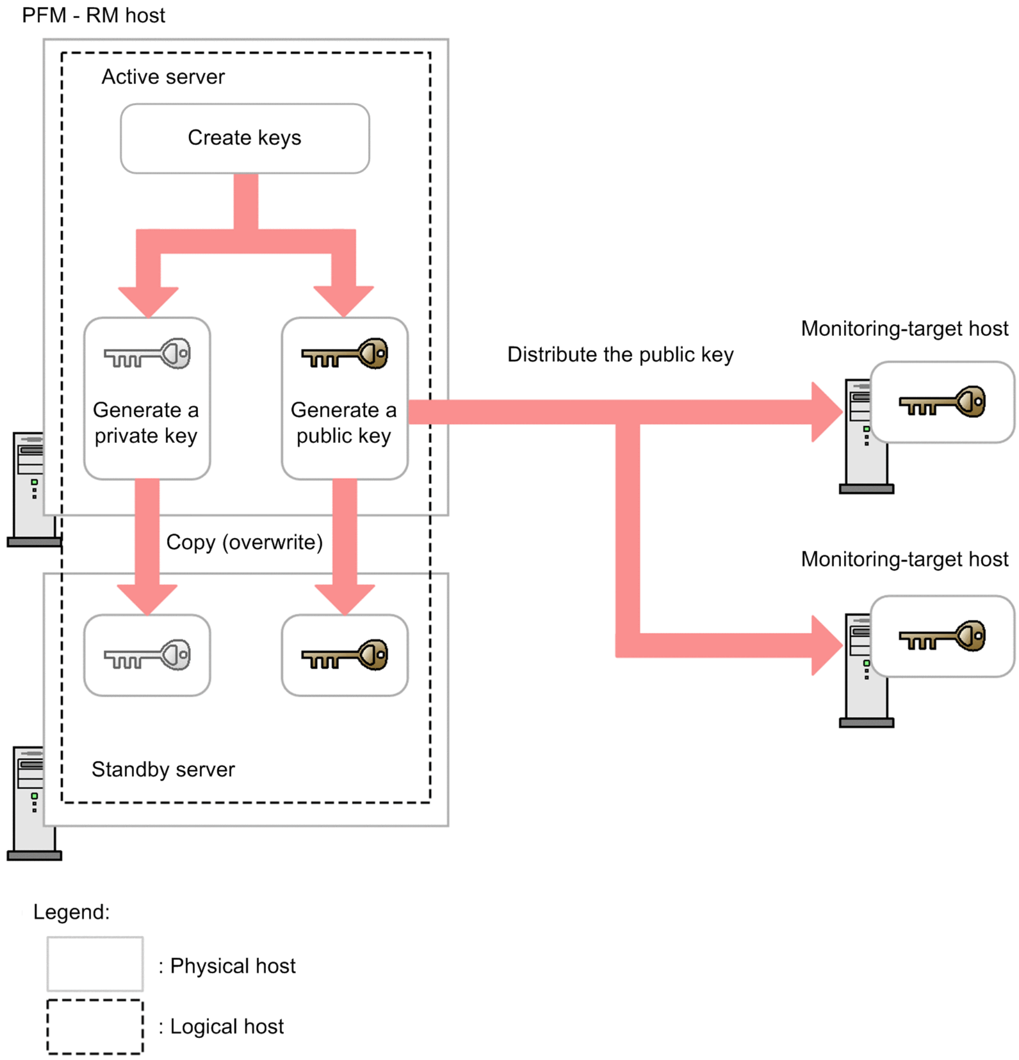
Figure 2–7: Concept of public key authentication



In a cluster system, there are two methods of public key authentication. One method uses the same keys for the executing and standby nodes, and the other method uses different keys for those nodes.

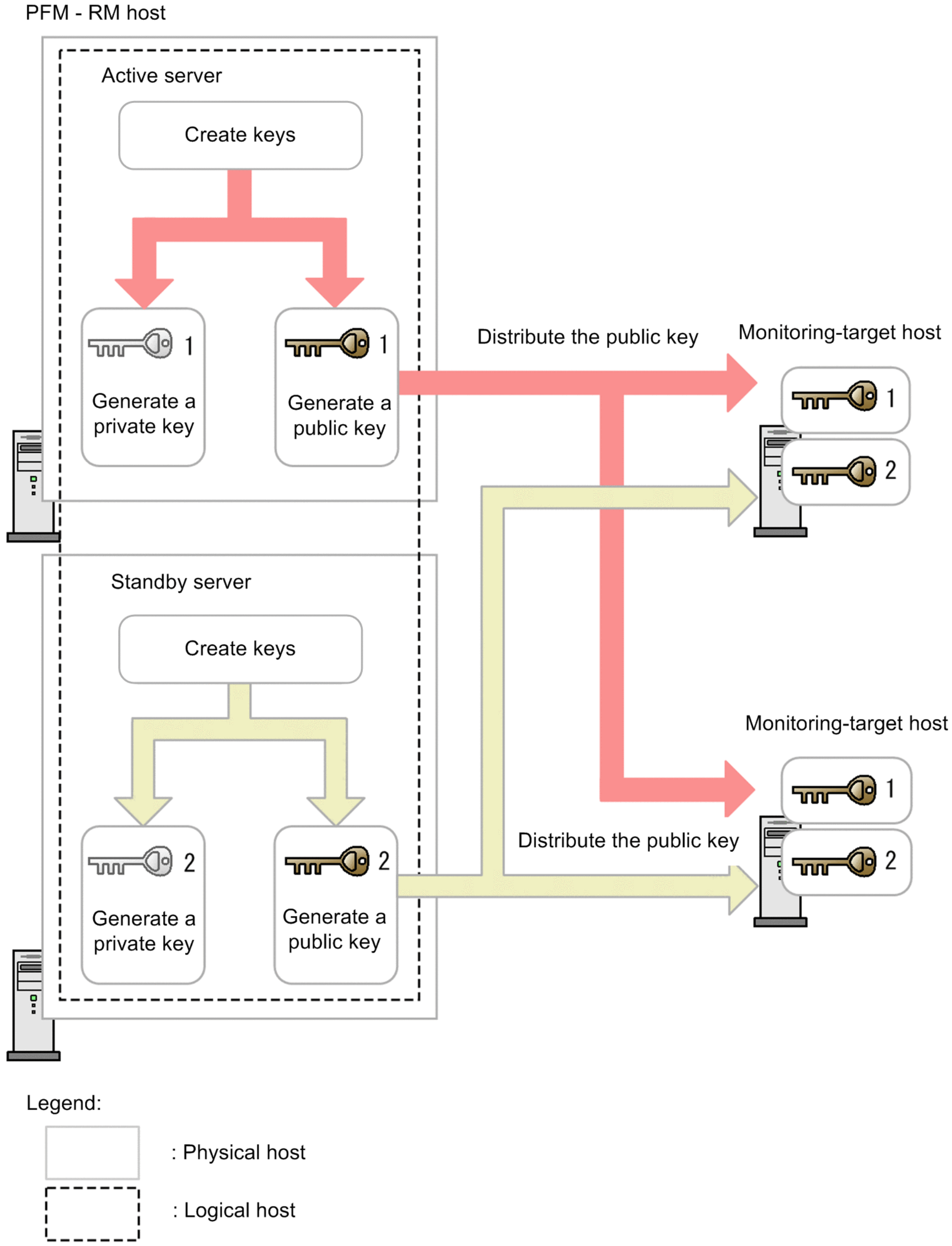
If you choose to use the same keys for both the executing and standby nodes, copy the key files on the executing node to the standby node, overwriting the existing key files on the standby node. The following figure shows public key authentication using the same keys for both nodes.

Figure 2–8: Public key authentication using the same keys for both the executing and standby nodes



If you choose to use different keys for the executing and standby nodes, register the keys on the monitoring-target host. The following figure shows public key authentication using different keys for the executing and standby nodes.

Figure 2–9: Public key authentication using different keys for the executing and standby nodes



(2) User account settings

To use SSH, accounts of both the PFM - RM host and the monitoring-target host are required.

- PFM - RM host account

  Set values that are described as `HostUserID`, `HostPassword`, and `HostDomain` in Table 2-5. The account that is set is specified during instance setup.

  If you use PFM - RM for Virtual Machine in a cluster system, specify the same user name and password on the executing and standby nodes. This will let the account logon to both of these nodes.

- Monitoring-target host account

  Use superuser as the monitoring-target host account.

(3) Installing necessary software and RPM packages

Different monitoring targets need different software and packages. The following describes what are required for each monitoring target.

(a) For KVM

■ Software required on the PFM - RM host

The table below lists the software that is required for PFM - RM for Virtual Machine to acquire monitoring-target information. For details, see the *Release Notes*.

Table 2–15: Software required to acquire monitoring-target information

Software name	OS	Version	Default
PuTTY	Windows Server 2012	• Plink 0.62 or later • PuTTYgen 0.62 or later • Pscp 0.62 or later	N
	Windows Server 2012 R2	• Plink 0.62 or later • PuTTYgen 0.62 or later • Pscp 0.62 or later	N
	Windows Server 2016	• Plink 0.62 or later • PuTTYgen 0.62 or later • Pscp 0.62 or later	N
	Windows Server 2019	• Plink 0.62 or later • PuTTYgen 0.62 or later • Pscp 0.62 or later	N
OpenSSH	Windows Server 2019	OpenSSH_for_Windows_7.7p1, LibreSSL 2.6.5 or later	Y

Legend:

Y: Installed by default.

N: Not installed by default.

■ RPM packages required on the monitoring-target host

In order for PFM - RM for Virtual Machine to acquire monitoring-target information, the RPM packages listed below are required.

Software name	OS	RPM package name	Default
OpenSSH	Red Hat Enterprise Linux 6 (64-bit x86_64)	`openssh-5.3p1-20.el6` or later `openssh-server-5.3p1-20.el6` or later	Y
	Red Hat Enterprise Linux(R) Server 7	`openssh-6.6.1p1-11.el7` or later `openssh-server-6.6.1p1-11.el7` or later	Y
	Red Hat Enterprise Linux(R) Server 8	`openssh-7.8p1-4.el8` or later `openssh-server-7.8p1-4.el8` or later	Y

Legend:

Y: Installed by default.

■ Packages and commands required on the monitoring-target host

The command required for record collection differs depending on the record to be collected, and the required RPM package also differs depending on the command. To check the required package for a command, execute the following:

```
/bin/rpm -qf full-path-name-of-the-prerequisite-command
```

■ Records and the commands required for collecting the records

The following table lists records and the commands that are required for collecting the records.

Table 2–16: Records and the commands required for collecting the records

No.	Record	Command
1	Host CPU Status (PI_HCI)	/bin/date /bin/cat
2	Host Logical Disk Status (PI_HLDI)	/bin/date /bin/df
3	Host Memory Status (PI_HMI)	/bin/date /bin/ps /usr/bin/free /usr/bin/getconf /usr/bin/vmstat
4	Host Network Status (PI_HNI)	/bin/date /sbin/ifconfig /usr/bin/virsh
5	Host Physical Disk Status (PI_HPDI)	/bin/date /usr/bin/iostat
6	Host Status Detail (PD)	/usr/bin/virsh /bin/hostname
7	Host Status (PI)	/bin/date /bin/cat /bin/ps /usr/bin/top
8	VM CPU Status (PI_VCI)	/bin/date /usr/bin/virsh
9	VM Logical Disk Status (PI_VLDI)	--
10	VM Memory Status (PI_VMI)	/bin/date /usr/bin/pmap /usr/bin/virsh
11	VM Network Status (PI_VNI)	/bin/date /sbin/ifconfig /bin/cat /usr/bin/virsh
12	VM Physical Disk Status (PI_VPDI)	/bin/date /usr/bin/virsh
13	VM Status Detail	/usr/bin/virsh

No.	Record	Command
13	(PD_VM)	/usr/bin/virsh
14	VM Status (PI_VI)	/bin/date /usr/bin/virsh /bin/ps /usr/bin/top

The following lists the RPM packages required by the commands that are required for record collection.

Table 2–17: RPM packages required by the commands that are required for record collection (For Red Hat Enterprise Linux 6 (64-bit x86_64))

No.	Command name	Package name	Default
1	/bin/cat	coreutils-8.4-13.el6 or later	Y
2	/bin/date	coreutils-8.4-13.el6 or later	Y
3	/bin/df	coreutils-8.4-13.el6 or later	Y
4	/bin/ps	procps-3.2.8-17.el6 or later	Y
5	/usr/bin/free	procps-3.2.8-17.el6 or later	Y
6	/usr/bin/getconf	glibc-common-2.12-1.25.el6 or later	Y
7	/usr/bin/iostat	sysstat-9.0.4-18.el6 or later	N
8	/usr/bin/pmap	procps-3.2.8-17.el6 or later	Y
9	/usr/bin/top	procps-3.2.8-17.el6 or later	Y
10	/usr/bin/virsh	libvirt-client-0.8.7-18.el6 or later	N
11	/usr/bin/vmstat	procps-3.2.8-17.el6 or later	Y
12	/sbin/ifconfig	net-tools-1.60-105.el6 or later	Y
13	/bin/hostname	net-tools-1.60-105.el6 or later	Y

Legend:

Y: Installed by default.

N: Not installed by default.

Table 2–18: RPM packages required by the commands that are required for record collection (For Red Hat Enterprise Linux(R) Server 7)

No.	Command name	Package name	Default
1	/bin/cat	coreutils-8.22-11.el7 or later	Y
2	/bin/date	coreutils-8.22-11.el7 or later	Y
3	/bin/df	coreutils-8.22-11.el7 or later	Y
4	/bin/ps	procps-ng-3.3.10-3.el7 or later	Y
5	/usr/bin/free	procps-ng-3.3.10-3.el7 or later	Y
6	/usr/bin/getconf	glibc-common-2.17-78.el7 or later	Y
7	/usr/bin/iostat	sysstat-10.1.5-7.el7 or later	N

No.	Command name	Package name	Default
8	/usr/bin/pmap	procps-ng-3.3.10-3.el7 or later	Y
9	/usr/bin/top	procps-ng-3.3.10-3.el7 or later	Y
10	/usr/bin/virsh	libvirt-client-1.2.8-16.el7 or later	N
11	/usr/bin/vmstat	procps-ng-3.3.10-3.el7 or later	Y
12	/bin/hostname	hostname-3.13-3.el7 or later	Y

Legend:

Y: Installed by default.

N: Not installed by default.

Table 2–19: RPM packages required by the commands that are required for record collection (For Red Hat Enterprise Linux(R) Server 8)

No.	Command name	Package name	Default
1	/bin/cat	coreutils-8.30-6.el8 or later	Y
2	/bin/date	coreutils-8.30-6.el8 or later	Y
3	/bin/df	coreutils-8.30-6.el8 or later	Y
4	/bin/ps	procps-ng-3.3.15-1.el8 or later	Y
5	/usr/bin/free	procps-ng-3.3.15-1.el8 or later	Y
6	/usr/bin/getconf	glibc-common-2.28-42.el8 or later	Y
7	/usr/bin/iostat	sysstat-11.7.3-2.el8 or later	Y
8	/usr/bin/pmap	sysstat-11.7.3-2.el8 or later	Y
9	/usr/bin/top	procps-ng-3.3.15-1.el8 or later	Y
10	/usr/bin/virsh	libvirt-client-4.5.0-23.module+el8+2800+2d311f65 or later	N
11	/usr/bin/vmstat	procps-ng-3.3.15-1.el8 or later	N
12	/bin/hostname	hostname-3.20-6.el8 or later	Y

Legend:

Y: Installed by default.

N: Not installed by default.

(b) For Podman environment

■ Software required on the PFM - RM host

Software required by PFM - RM for Virtual Machine to acquire monitoring-target information is the same as that for KVM. See *Table 2-15 Software required to acquire monitoring-target information*.

■ RPM packages required on the monitoring-target host

In order for PFM - RM for Virtual Machine to acquire monitoring-target information, the RPM packages listed below are required.

Software name	OS	RPM package name	Default
OpenSSH	Red Hat Enterprise Linux(R) Server 8	`openssh-7.8p1-4.el8` or later `openssh-server-7.8p1-4.el8` or later	Y

Legend:

Y: Installed by default.

■ Packages and commands required on the monitoring-target host

The command required for record collection differs depending on the record to be collected, and the required RPM package also differs depending on the command. To check the required package for a command, execute the following:

```
/bin/rpm -qf full-path-name-of-the-prerequisite-command
```

■ Records and the commands required for collecting the records

The following table lists records and the commands that are required for collecting the records.

Table 2–20: Records and the commands required for collecting the records

No.	Record	Command

1	Host CPU Status
	(PI_HCI)
		`/usr/bin/date`
		`/usr/bin/cat`
		`/usr/bin/getconf`

2	Host Logical Disk Status
	(PI_HLDI)
		`/usr/bin/date`
		`/usr/bin/df`
		`/usr/bin/mount`

3	Host Memory Status
	(PI_HMI)
		`/usr/bin/date`
		`/usr/bin/cat`
		`/usr/bin/getconf`
		`/usr/bin/vmstat`
		`/usr/bin/podman`

4	Host Network Status (PI_HNI)	`/usr/bin/date` `/usr/bin/cat`

5	Host Physical Disk Status
	(PI_HPDI)
		`/usr/bin/date`
		`/usr/bin/cat`
		`/usr/bin/ls`

6	Host Status Detail (PD)	`/usr/bin/podman`

7	Host Status
	(PI)
		`/usr/bin/date`
		`/usr/bin/cat`
		`/usr/bin/getconf`
		`/usr/bin/podman`

8	VM CPU Status
	(PI_VCI)
		`/usr/bin/date`
		`/usr/bin/cat`
		`/usr/bin/getconf`
		`/usr/bin/podman`

9	VM Memory Status (PI_VMI)	`/usr/bin/date` `/usr/bin/cat`

No.	Record	Command
9	VM Memory Status (PI_VMI)	/usr/bin/getconf /usr/bin/podman
10	VM Network Status (PI_VNI)	/usr/bin/date /usr/bin/cat /usr/bin/podman
11	VM Physical Disk Status (PI_VPDI)	/usr/bin/date /usr/bin/cat /usr/bin/podman
12	VM Status Detail (PD_VM)	/usr/bin/podman
13	VM Status (PI_VI)	/usr/bin/date /usr/bin/cat /usr/bin/getconf /usr/bin/podman
14	POD Status Detail (PD_PODD)	/usr/bin/podman
15	POD Status Interval (PI_PODI)	/usr/bin/date /usr/bin/cat /usr/bin/podman When Cpu_Category or Memory_Category is set to Y /usr/bin/getconf
16	POD Container Status Interval (PI_POCI)	/usr/bin/date /usr/bin/cat /usr/bin/podman When Cpu_Category or Memory_Category is set to Y /usr/bin/getconf

The following lists the RPM packages required by the commands that are required for record collection.

Table 2–21: RPM packages required by the commands that are required for record collection (For Red Hat Enterprise Linux 6 (64-bit x86_64))

No.	Command name	Package name	Default
1	/usr/bin/cat	coreutils-8.30-6.el8 or later	Y
2	/usr/bin/date	coreutils-8.30-6.el8 or later	Y
3	/usr/bin/df	coreutils-8.30-6.el8 or later	Y
4	/usr/bin/ls	coreutils-8.30-6.el8 or later	Y
5	/usr/bin/mount	util-linux-2.32.1-8.el8 or later	Y
6	/usr/bin/vmstat	procps-ng-3.3.15-1.el8 or later	N
7	/usr/bin/getconf	glibc-common-2.28-42.el8 or later	Y
8	/usr/bin/podman	podman-1.0.0-2.git921f98f.module+el8+2785+ff8a053f or later	N

Legend:

Y: Installed by default.

N: Not installed by default.

(4) Settings related to SSH connection

The settings that enable SSH connection must be specified on both the PFM - RM host and the monitoring-target host. The following describe the procedures for specifying these settings.

(a) Enabling public key authentication of the SSH server

To enable public key authentication:

1. Log in to the monitoring-target host as the superuser.
2. Open the `/etc/ssh/sshd_config` file.
3. Change the value of `PubkeyAuthentication` to `yes`.
4. Change the value of `PermitRootLogin` to `yes`.
5. Save and close the `/etc/ssh/sshd_config` file.
6. Restart the `sshd` service by executing the command shown below.
   Note that the command below assumes that the host named `targethost1` is set as a monitoring target.

```
[root@targethost1.ssh]$ /etc/rc.d/init.d/sshd restart
```

> **Important**
>
> To allow the superuser to collect information, open the `/etc/ssh/sshd_config` file, and change the value of `PermitRootLogin` to `yes`. Then, restart the `sshd` service.

(b) Creating keys

The procedure for creating keys is described below.

Keys can be created by logging on to the PFM - RM host, and then by using the function provided by the SSH client. The cryptography and the length of the key should be determined according to the documentation of the OS to be monitored. If you monitor a RHEL 8 virtual environment, the RSA encryption key requires 2,048 bits or more in length at the encryption policy level for the entire default RHEL 8 system. In this subsection, the procedure for creating RSA keys is described. To create RSA keys:

If you use PuTTY as the SSH client

1. From the Windows **Start** menu, select **Programs**, **PuTTY**, and then **PuTTYgen**.
   PuTTYgen starts, and the PuTTY Key Generator window appears.
2. Under **Parameters**, confirm that **Type of key to generate** is **SSH-2 RSA** and **Number of bits in a generated key:** has a value longer than the length acceptable to the SSH client as its key length, and then click the **Generate** button.
   The key generation progress bar appears in **Key**.

Because PuTTY uses the version 2 of the SSH protocol by default, **SSH-2 RSA** is selected. However, you might want to change the protocol version to 1. For details about how to change the protocol version to 1, see the PuTTY documentation.

3. Randomly move the mouse pointer in the dialog box until the progress reaches 100% to create a random number that is required to create keys.
   When the progress reaches 100%, the created random number is displayed in **Key**, and keys are created.
4. Click the **Save private key** button to save a private key.
   If no values are entered in **Key passphrase** and **Confirm passphrase**, a dialog box appears. In this dialog box, click the **Yes** button without entering anything in **Key passphrase** and **Confirm passphrase**.
5. Click the **Save public key** button to save a public key.

If you use OpenSSH (which comes with Windows Server 2019) as the SSH client

You can select the RSA encryption as the type of key.

1. Log on to the PFM - RM host.
2. Execute the `ssh-keygen -t rsa` command.
3. Determine the output destination and name of the private key.
   They are defaulted to `%userprofile%\.ssh\id_rsa`.
4. Press the **Enter** key twice.
   When you are asked to enter the passphrase for the private key, press the **Enter** key without typing anything. When you are asked to confirm it, press the **Enter** key again without typing anything.

The following shows an example of executing the `ssh-keygen` command:

```
C:\work>ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\username\.ssh\id_rsa): <Ente
r>
Enter passphrase (empty for no passphrase): <Enter>
Enter same passphrase again: <Enter>
Your identification has been saved in C:\Users\username\.ssh\id_rsa.
Your public key has been saved in C:\Users\username\.ssh\id_rsa.pub.
The key fingerprint is:
SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx username @PFM - RM host
 name
```

(c) Deploying the public key (PFM - RM host)

If there are multiple monitoring-target hosts, perform the following procedure on all of the hosts.

■ Transfer the public key

Transfer the public key to the `.ssh` directory under the home directory on a monitoring-target host. To do this:

1. Log in to the monitoring-target host as the superuser (account specified for `UserID` during setup of the monitoring target).
2. Use the `cd` command to move to the `.ssh` directory under the home directory.
   If the `.ssh` directory does not exist under the home directory, create the `.ssh` directory. For the access permission attribute of the `.ssh` directory, set `700` or `755`. For the owner and group, set values that are appropriate for the user specified during setup of the monitoring-target host.

If the home directory, `.ssh` directory attribute, owner, and group settings are not correct, SSH connection might fail.

For details about how to set the directory attribute, see the OS documentation.

3. On the PFM - RM host, open the Command Prompt window, and then execute the following commands.

   If you use PuTTY as the SSH client:

   Navigate to the folder where PuTTY is installed, and then execute the `pscp` command, which is provided by PuTTY.

   The following shows an example of executing the command when the public key is located in the PuTTY installation directory:

```
C:\Program Files\PuTTY>pscp.exe agt8.pub ClientUser@TargetHost:.ssh
ClientUser@TargetHost's password:password (Enter the superuser's password
here.)
agt8.pub                  | 0 kB |   0.3 kB/s | ETA: 00:00:00 | 100%
C:\Program Files\PuTTY>
```

   If a message asking you whether you want to register the fingerprint appears, enter `n`.

   If you use OpenSSH (which comes with Windows Server 2019) as the SSH client:

   The following shows an example of executing the command when the public key is located in the `.ssh` directory:

```
C:\Users\username\.ssh\>scp.exe id_rsa.pub ClientUser@TargetHost:.ssh
The authenticity of host 'PFM - RM hostname' can't be established.
ECDSA key fingerprint is SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added xxx.xxx.xxx.xxx (ECDSA) to the list of known ho
sts.

ClientUser@TargetHost's password:password
agt8.pub                                     100%  404     0.4KB/s   00:00
```

■ Registering the public key (monitoring-target host)

Log in to the monitoring-target host as the superuser that was set during setup of the monitoring target (account specified for `UserID`), and then register the public key. To do this:

1. Log in to the monitoring-target host as the superuser that was set during setup of the monitoring target.

2. Use the `cd` command to move to the `.ssh` directory.

3. Execute the following command.

   If you use PuTTY as the SSH client:

   Execute the `ssh-keygen` command with the `-i` and `-f` options specified. When you execute the command, the public key created by PuTTY is converted into the authentication key file format available to OpenSSH.

   If you use OpenSSH (which comes with Windows Server 2019) as the SSH client:

   Execute the `cat` command with the public key file and the authentication key file as the redirect destination specified. When you execute the command, the contents of the public key file are redirected to the authentication key file. Furthermore, the contents of the received public key are added to the authentication key file.

4. Use the `rm` command to delete the public key file that was received by the procedure in *Transferring the public key*.

5. Execute the `chmod` command to change the attribute of the key authentication file to `600`.

The following shows an example of executing the commands in steps 2 through 5:

If you use PuTTY as the SSH client:

```
[root @targethost1 ~]$ cd .ssh
[root @targethost1 .ssh]$ ssh-keygen -i -f agt8.pub >> authorized_keys
[root @targethost1 .ssh]$ rm agt8.pub
[root @targethost1 .ssh]$ chmod 600 authorized_keys
```

If you use OpenSSH (which comes with Windows Server 2019) as the SSH client:

```
[root@targethost1 ]$ cd .ssh
[root@targethost1 .ssh]$ cat id_rsa.pub >> authorized_keys
[root@targethost1 .ssh]$ rm id_rsa.pub
[root@targethost1 .ssh]$ chmod 600 authorized_keys
```

The name of the key authentication file is set by `AuthorizedKeysFile` in the `/etc/ssh/sshd_config` file.

By default, `~/.ssh/ authorized_keys` is set.

(d) Confirming connectivity and registering the fingerprint (PFM - RM host)

To confirm connectivity and register the fingerprint:

1. Log in to the PFM - RM host.
   Make sure that you use the account that was set for `HostUserID` during setup of the instance environment.
2. Open the Command Prompt window.
3. Execute the following command using the private key that has been created.
   If you use PuTTY as the SSH client:
      `plink.exe` of PuTTY
   If you use OpenSSH (which comes with Windows Server 2019) as the SSH client:
      `ssh.exe` of OpenSSH
   Connection is attempted.
4. Upon achieving the initial connection, register the fingerprint.
   Enter `y` to register the fingerprint of the public key on the monitoring-target host.
   When `y` is entered, the prompt of the monitoring-target host is displayed.
5. Log out.
   When the prompt of the monitoring-target host is displayed, enter `exit` to log out from the host.
6. Execute the PuTTY `plink` command to reconnect to the monitoring-target host.
   If you are not prompted to enter anything and reconnection succeeds, the connection settings are completed. Enter `exit` to log out from the monitoring-target host.
   If an error occurs or you are prompted to enter something, check for problems with operations performed by the procedure.

The following shows an example of performing the procedure for checking connectivity:

If you use PuTTY as the SSH client:

```
C:\WINDOWS\system32>"C:\Program Files\PuTTY\plink.exe" -ssh -noagent -i "C:\
Program Files\PuTTY\agt8.ppk" -l root -P 22 targethost1
The server's host key is not cached in the registry. You have no guarantee t
hat the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
If you trust this host, enter "y" to add the key to PuTTY's cache and carry
on connecting.
If you want to carry on connecting just once, without adding the key to the
cache, enter "n".
If you do not trust this host, press Return to abandon the connection.
Store key in cache? (y/n) y
Using username "root".
Last login: Wed Aug  4 13:29:55 2010 from xxx.xxx.xxx.xxx
[root@targethost1]$ exit
logout
C:\WINDOWS\system32>"C:\Program Files\PuTTY\plink.exe" -ssh -noagent -i "C:\
Program Files\PuTTY\agt8.ppk" -l root -P 22 targethost1
Using username "root".
Last login: Wed Aug  4 13:30:00 2010 from xxx.xxx.xxx.xxx
[root@targethost1]$ exit
logout
C:\WINDOWS\system32>
```

If you use OpenSSH (which comes with Windows Server 2019) as the SSH client:

```
C:\Users\username\.ssh>ssh -i "C:\Users\username\.ssh\id_rsa" -l root -p 22
targethost1
The authenticity of host '[xxx.xxx.xxx.xxx]:22 ([xxx. xxx. xxx. xxx]:22)' ca
n't be established.
RSA key fingerprint is SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[xxx.xxx.xxx.xxx]:22' (RSA) to the list of known
 hosts.
Last login: Wed Sep 25 09:08:14 2019 from xxx.xxx.xxx.xxx
[root@targethost1]$ exit
logout
C:\Users\username\.ssh>
```

Important

- PFM - RM for Virtual Machine assumes that fingerprint authentication has already finished. If the fingerprint has not yet been registered before PFM - RM for Virtual Machine connects to the SSH client for the first time, always register the fingerprint at the initial connection.

  In a cluster environment, make sure that you also check connectivity and register the fingerprint on the standby node.
- If you changed the user account specified for `HostUserID` during the setup of the instance environment, register the fingerprint again.
- In the PFM - RM host, execute the `uname` or other commands on the monitored host to confirm that you will get a response in less than 10 seconds.

If you use OpenSSH, which comes with Windows Server 2019, as the SSH client, the connection may fail when users other than the user specified for `HostUserID` in the instance settings can access the private key file. In this case, go to **Properties**, **Security**, and **Advanced** for the private key file, and remove the permissions of the users other than the user specified for `HostUserID`.

2.6 Examples of setting up an instance environment and monitoring targets

This section provides examples of setting up an instance environment and monitoring targets to monitor a VMware environment.

2.6.1 For VMware

This subsection provides an example of setting up an instance environment to monitor VMware.

(1) Assumed VMware environment

The explanation in this subsection assumes that the user has the following VMware environment:

Monitored host

- Virtual environment software: VMware ESXi
- Physical server host name: `vm-host`
- Login user name#: `user01`
- Login password#: `pass01`
- Domain name: None

#

The user must be able to connect to VMware ESXi from a management tool such as VMware vSphere Client

PFM - RM host

- OS: Windows
- Login user name: `vmuser`
- Login password: `vmpass`
- Domain name: `vmdomain`

(2) Example of setting up an instance environment

A command execution example for setting up an instance environment follows.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf inst setup -key RMVM -inst vmwa
re1   ...1
 VM_Type          [vmware]           :<Enter>          ...2
 Interval         [300]              :<Enter>          ...3
 Offset           [10]               :<Enter>          ...4
 Std_Category     [Y]                :<Enter>          ...5
 Cpu_Category     [Y]                :<Enter>          ...6
 Memory_Category  [Y]                :<Enter>          ...7
 Disk_Category    [Y]                :<Enter>          ...8
 Network_Category [Y]                :<Enter>          ...9
 HostUserID       []                 :vmuser<Enter>   ...10
 HostPassword     []                 :vmpass<Enter>   ...11
                             Re-enter :vmpass<Enter>
 HostDomain       []                 :vmdomain<Enter>  ...12
```

```
 SSH_Type          [putty]             :<Enter>          ...13
 SSH_Client        []                  :<Enter>          ...14
 Log_Size (MB)     [16]                :<Enter>          ...15
 UseVcpuMax        [N]                 :<Enter>          ...16
KAVE05080-I Creating an instance environment. (servicekey=RMVM, inst=vmware1
)
KAVE05081-I An instance environment has been created. (servicekey=RMVM, inst
=vmware1)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *16*: Number of the corresponding step in the setup procedure.

To set up an instance environment:

1. Execute the `jpcconf inst setup` command.
   Specify any name as the instance name.
2. Set up `VM_Type`.
   Since the default value (`vmware`) will be used, press the **Enter** key without entering any value.
3. Set up `Interval`.
   Since the default value (`300`) will be used, press the **Enter** key without entering any value.
4. Set up `Offset`.
   Since the default value (`10`) will be used, press the **Enter** key without entering any value.
5. Set up `Std_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
6. Set up `Cpu_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
7. Set up `Memory_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
8. Set up `Disk_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
9. Set up `Network_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
10. Set up `HostUserID`.
    Enter the PFM - RM host's `vmuser` user name to be used for connection and press the **Enter** key.
11. Set up `HostPassword`.
    Enter `vmpass` as the password for the user ID set by `HostUserID`, and then press the **Enter** key. The system will ask you to re-enter the password for confirmation purposes. Re-enter the `vmpass` password and then press the **Enter** key.
    Note:
    The entered character strings are not displayed on the screen.

12. Set up `HostDomain`.

    Enter `vmdomain` as the name of the domain that the PFM - RM host belongs to, and then press the **Enter** key.

13. Set up `SSH_Type`.

    Press the **Enter** key without entering any value because this item is not used by VMware.

14. Set up `SSH_Client`.

    Press the **Enter** key without entering any value because this item is not used by VMware.

15. Set up `Log_Size`.

    Since the default value (16 megabytes) will be used, press the **Enter** key without entering any value.

16. Set up `UseVcpuMax`.

    Since the default value (N) will be used, press the **Enter** key without entering any value.

(3) Example of setting up monitoring targets

The following shows an example of executing a command to set up a monitoring target.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key RMVM -inst vm
ware1 -target vm-host   ...1
 Target Host         []           :vm-host <Enter>       ...2
 VM_Host             []           :<Enter>               ...3
 Security            [1]          :<Enter>               ...4
 Port                [0]          :<Enter>               ...5
 UserID                           :user01<Enter>         ...6
 Password            []           :pass01<Enter>         ...7
                         Re-enter :pass01<Enter>
 Domain              []           :<Enter>               ...8
 Private_Key_File    []           :<Enter>               ...9
KAVE05361-I The monitoring target is now being added. (servicekey=RMVM, inst
=vmware1, target=vm-host)
KAVE05362-I The monitoring target has been added. (servicekey=RMVM, inst=vmw
are1, target=vm-host)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *9*: Number of the corresponding step in the setup procedure.

To set up a monitoring target (as in the above example):

1. Execute the `jpcconf target setup` command.

   Specify `vm-host`, which is the host name of a physical server, as the name of a monitoring target.

2. Set `Target Host`.

   Enter `vm-host`, which is the host name of a physical server, and then press the **Enter** key.

3. Set `VM_Host`.

   Since the `Target Host` value will be used, press the **Enter** key without entering any value.

4. Set `Security`.

   Since the default value (`1`) will be used, press the **Enter** key without entering any value.

5. Set `Port`.

   Since the default value (`0`) will be used, press the **Enter** key without entering any value.

6. Set `UserID`.

   Enter `user01` as the user name that is used for connection, and then press the **Enter** key.

7. Set `Password`.

   Enter `pass01` as the password that is used for connection, and then press the **Enter** key. When you are prompted to enter the same password again, re-enter `pass01`, and then press the **Enter** key.

   Note:

   The password string that you enter is not displayed.

8. Set `Domain`.

   Press the **Enter** key without entering any value because this item is not used by VMware.

9. Set `Private_Key_File`.

   Press the **Enter** key without entering any value because this item is not used by VMware.

2.6.2 For Hyper-V

This subsection provides an example of setting up an instance environment to monitor Hyper-V.

(1) Assumed Hyper-V environment

The explanation in this subsection assumes that the user has the following Hyper-V environment:

Monitored host

- Virtual environment software: Hyper-V
- Physical server host name: `vm-host`
- Login user name: `user01`
- Login password: `pass01`
- Domain name: `domain01`

PFM - RM host

- Login user name: `vmuser`
- Login password: `vmpass`
- Domain name: `vmdomain`

(2) Example of setting up an instance environment

A command execution example for setting up an instance environment follows.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf inst setup -key RMVM -inst hype
rv1      ...1
 VM_Type          [vmware]        :hyperv<Enter>       ...2
 Interval         [300]           :<Enter>             ...3
 Offset           [10]            :<Enter>             ...4
```

```
 Std_Category     [Y]              :<Enter>           ...5
 Cpu_Category     [Y]              :<Enter>           ...6
 Memory_Category  [Y]              :<Enter>           ...7
 Disk_Category    [Y]              :<Enter>           ...8
 Network_Category [Y]              :<Enter>           ...9
 HostUserID       []               :vmuser<Enter>     ...10
 HostPassword     []               :vmpass<Enter>     ...11
                         Re-enter :vmpass<Enter>
 HostDomain       []               :vmdomain<Enter>   ...12
 SSH_Type         [putty]             :<Enter>        ...13
 SSH_Client       []                  :<Enter>        ...14
 Log_Size (MB)    [16]                :<Enter>        ...15
 UseVcpuMax       [N]                 :<Enter>        ...16
KAVE05080-I Creating an instance environment. (servicekey=RMVM, inst=hyperv1
)
KAVE05081-I An instance environment has been created. (servicekey=RMVM, inst
=hyperv1)
```

Legend:

<Enter>: **Enter** key must be pressed.

1 through *16*: Number of the corresponding step in the setup procedure.

To set up an instance environment:

1. Execute the `jpcconf inst setup` command.
   Specify any name as the instance name.
2. Set up `VM_Type`.
   Enter `hyperv` and press the **Enter** key.
3. Set up `Interval`.
   Since the default value (`300`) will be used, press the **Enter** key without entering any value.
4. Set up `Offset`.
   Since the default value (`10`) will be used, press the **Enter** key without entering any value.
5. Set up `Std_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
6. Set up `Cpu_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
7. Set up `Memory_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
8. Set up `Disk_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
9. Set up `Network_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
10. Set up `HostUserID`.

Enter the PFM - RM host's `vmuser` user name to be used for connection and press the **Enter** key.

11. Set up `HostPassword`.

    Enter `vmpass` as the password for the user ID set by `HostUserID`, and then press the **Enter** key. The system will ask you to re-enter the password for confirmation purposes. Re-enter `vmpass` and then press the **Enter** key.

    Note:

    The entered character strings are not displayed on the screen.

12. Set up `HostDomain`.

    Enter `vmdomain` as the name of the domain that the PFM - RM host belongs to, and then press the **Enter** key.

13. Set up `SSH_Type`.

    Press the **Enter** key without entering any value because this item is not used by Hyper-V.

14. Set up `SSH_Client`.

    Press the **Enter** key without entering any value because this item is not used by Hyper-V.

15. Set up `Log_Size`.

    Since the default value (16 megabytes) will be used, press the **Enter** key without entering any value.

16. Set up `UseVcpuMax`.

    Since the default value (N) will be used, press the **Enter** key without entering any value.

(3) Example of setting up monitoring targets

The following shows an example of executing a command to set up a monitoring target.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key RMVM -inst hy
perv1 -target vm-host    ...1
 Target Host      []         :vm-host <Enter>         ...2
 VM_Host          []         :<Enter>                 ...3
 Security         [1]        :<Enter>                 ...4
 Port             [0]        :<Enter>                 ...5
 UserID                      :user01<Enter>           ...6
 Password         []         :pass01<Enter>           ...7
                    Re-enter :pass01<Enter>
 Domain           []         :domain01<Enter>         ...8
 Private_Key_File []         :<Enter>                 ...9
KAVE05361-I The monitoring target is now being added. (servicekey=RMVM, inst
=hyperv1, target=vm-host)
KAVE05362-I The monitoring target has been added. (servicekey=RMVM, inst=hyp
erv1, target=vm-host)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *9*: Number of the corresponding step in the setup procedure.

To set up a monitoring target (as in the above example):

1. Execute the `jpcconf target setup` command.

   Specify `vm-host`, which is the host name of a physical server, as the name of a monitoring target.

2. Set `Target Host`.

   Enter `vm-host`, which is the host name of a physical server, and then press the **Enter** key.

3. Set `VM_Host`.

   Since the `Target Host` value will be used, press the **Enter** key without entering any value.

4. Set `Security`.

   Press the **Enter** key without entering any value because this item is not used by Hyper-V.

5. Set `Port`.

   Press the **Enter** key without entering any value because this item is not used by Hyper-V.

6. Set `UserID`.

   Enter `user01` as the user name that is used for connection, and then press the **Enter** key.

7. Set `Password`.

   Enter `pass01` as the password that is used for connection, and then press the **Enter** key. When you are prompted to enter the same password again, re-enter `pass01`, and then press the **Enter** key.

   Note:

   The password string that you enter is not displayed.

8. Set `Domain`.

   Enter `domain01` as the domain name that is used for connection, and then press the **Enter** key.

9. Set `Private_Key_File`.

   Press the **Enter** key without entering any value because this item is not used by Hyper-V.

2.6.3 For KVM

This subsection provides an example of setting up an instance environment to monitor KVM.

(1) Assumed KVM environment

The explanation in this subsection assumes that the user has the following KVM environment:

Monitored host

- Virtual environment software: KVM
- Physical server host name: `vm-host`
- Login user name: `user01`
- Login password: None
- Domain name: None

PFM - RM host

- Login user name: `vmuser`
- Login password: `vmpass`
- Domain name: `vmdomain`

- SSH client: `C:\Program Files\PuTTY\plink.exe`
- Private key file: `C:\Program Files\PuTTY\agt8.ppk`

(2) Example of setting up an instance environment

A command execution example for setting up an instance environment follows.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf inst setup -key RMVM -inst kvm1
    ...1
 VM_Type             [vmware]         :kvm<Enter>         ...2
 Interval            [300]            :<Enter>            ...3
 Offset              [10]            :<Enter>             ...4
 Std_Category        [Y]              :<Enter>            ...5
 Cpu_Category        [Y]              :<Enter>            ...6
 Memory_Category     [Y]              :<Enter>            ...7
 Disk_Category       [Y]              :<Enter>            ...8
 Network_Category    [Y]              :<Enter>            ...9
 HostUserID          []               :vmuser<Enter>      ...10
 HostPassword        []               :vmpass<Enter>      ...11
                             Re-enter :vmpass<Enter>
 HostDomain          []               :vmdomain<Enter>    ...12
 SSH_Type            [putty]          :<Enter>            ...13
 SSH_Client          []               :C:\Program Files\PuTTY\plink.exe<Enter
>                             ...14
 Log_Size (MB)       [16]             :<Enter>            ...15
 UseVcpuMax          [N]              :<Enter>            ...16
KAVE05080-I Creating an instance environment. (servicekey=RMVM, inst=kvm1)
KAVE05081-I An instance environment has been created. (servicekey=RMVM, inst
=kvm1)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *16*: Number of the corresponding step in the setup procedure.

To set up an instance environment:

1. Execute the `jpcconf inst setup` command.
   Specify any name as the instance name.

2. Set up `VM_Type`.
   Enter `kvm` and press the **Enter** key.

3. Set up `Interval`.
   Since the default value (`300`) will be used, press the **Enter** key without entering any value.

4. Set up `Offset`.
   Since the default value (`10`) will be used, press the **Enter** key without entering any value.

5. Set up `Std_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.

6. Set up `Cpu_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.

7. Set up `Memory_Category`.

   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.

8. Set up `Disk_Category`.

   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.

9. Set up `Network_Category`.

   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.

10. Set `HostUserID`.

    Enter `vmuser` as the PFM - RM host user name that is used for connection, and then press the **Enter** key.

11. Set up `HostPassword`.

    Enter `vmpass` as the password for the user ID set by `HostUserID`, and then press the **Enter** key. The system will ask you to re-enter the password for confirmation purposes. Re-enter the `vmpass` password and then press the **Enter** key.

    Note:

    The entered character strings are not displayed on the screen.

12. Set `HostDomain`.

    Enter `vmdomain` as the name of the domain that the PFM - RM host belongs to, and then press the **Enter** key.

13. Set up `SSH_Type`.

    Since the default value (`putty`) will be used, press the **Enter** key without entering any value.

14. Set up `SSH_Client`.

    Enter `C:\Program Files\PuTTY\plink.exe` as the executable file of the SSH command that is used for connection, and then press the **Enter** key.

15. Set `Log_Size`.

    Since the default value (16 MB) will be used, press the **Enter** key without entering any value.

16. Set up `UseVcpuMax`.

    Press the **Enter** key without entering any value because this item is not used by KVM.

(3) Example of setting up a monitoring target

The following shows an example of executing a command to set up a monitoring target.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key RMVM -inst kv
m1 -target vm-host    ...1
 Target Host       []             :vm-host <Enter>     ...2
 VM_Host           []             :<Enter>             ...3
 Security          [1]            :<Enter>             ...4
 Port              [0]            :<Enter>             ...5
 UserID                           :user01<Enter>       ...6
 Password          []             :<Enter>             ...7
 Domain            []             :<Enter>             ...8
 Private_Key_File  []             :C:\Program Files\PuTTY\agt8.ppk<Enter>
...9
KAVE05361-I The monitoring target is now being added. (servicekey=RMVM, inst
```

```
=kvm1, target=vm-host)
KAVE05362-I The monitoring target has been added. (servicekey=RMVM, inst=kvm
1, target=vm-host)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *9*: Number of the corresponding step in the setup procedure.

To set up a monitoring target (as in the above example):

1. Execute the `jpcconf target setup` command.
   Specify `vm-host`, which is the host name of a physical server, as the name of a monitoring target.
2. Set `Target Host`.
   Enter `vm-host`, which is the host name of a physical server, and then press the **Enter** key.
3. Set `VM_Host`.
   Since the `Target Host` value will be used, press the **Enter** key without entering any value.
4. Set `Security`.
   Press the **Enter** key without entering any value because this item is not used by KVM.
5. Set `Port`.
   Since the default value (`0`) will be used, press the **Enter** key without entering any value.
6. Set `UserID`.
   Enter `user01` as the user name that is used for connection, and then press the **Enter** key.
7. Set `Password`.
   Press the **Enter** key without entering any value because this item is not used by KVM.
8. Set `Domain`.
   Press the **Enter** key without entering any value because this item is not used by KVM.
9. Set `Private_Key_File`.
   Enter `C:\Program Files\PuTTY\agt8.ppk` as the name of the private key file that is used for SSH connection, and then press the **Enter** key.

2.6.4 For Docker environment

This subsection provides an example of setting up an instance environment to monitor Docker environment.

(1) Assumed Docker environment

The explanation in this subsection assumes that the user has the following Docker environment:

Monitored host

- Virtual environment software: Docker environment
- Physical server host name: `docker-host`

- Port number: 2376

PFM - RM host

- Login user name: `vmuser`
- Login password: `vmpass`
- Domain name: `vmdomain`

(2) Example of setting up an instance environment

A command execution example for setting up an instance environment follows.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf inst setup -key RMVM -inst dock
er1      ...1
 VM_Type          [vmware]        :docker<Enter>       ...2
 Interval         [300]           :<Enter>             ...3
 Offset           [10]            :<Enter>             ...4
 Std_Category     [Y]             :<Enter>             ...5
 Cpu_Category     [Y]             :<Enter>             ...6
 Memory_Category  [Y]             :<Enter>             ...7
 Disk_Category    [Y]             :<Enter>             ...8
 Network_Category [Y]             :<Enter>             ...9
 HostUserID       []              :vmuser<Enter>       ...10
 HostPassword     []              :vmpass<Enter>       ...11
                         Re-enter :vmpass<Enter>
 HostDomain       []              :vmdomain<Enter>     ...12
 SSH_Type         [putty]         :<Enter>             ...13
 SSH_Client       []              :<Enter>             ...14
 Log_Size (MB)    [16]            :<Enter>             ...15
 UseVcpuMax       [N]             :<Enter>             ...16
KAVE05080-I Creating an instance environment. (servicekey=RMVM, inst=docker1
)
KAVE05081-I An instance environment has been created. (servicekey=RMVM, inst
=docker1)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *16*: Number of the corresponding step in the setup procedure.

To set up an instance environment:

1. Execute the `jpcconf inst setup` command.
   Specify any name as the instance name.
2. Set up `VM_Type`.
   Enter `docker` and press the **Enter** key.
3. Set up `Interval`.
   Since the default value (`300`) will be used, press the **Enter** key without entering any value.
4. Set up `Offset`.
   Since the default value (`10`) will be used, press the **Enter** key without entering any value.
5. Set up `Std_Category`.

Since the default value (`Y`) will be used, press the **Enter** key without entering any value.

6. Set up `Cpu_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
7. Set up `Memory_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
8. Set up `Disk_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
9. Set up `Network_Category`.
   Since the default value (`Y`) will be used, press the **Enter** key without entering any value.
10. Set up `HostUserID`.
    Enter the PFM - RM host's `vmuser` user name to be used for connection and press the **Enter** key.
11. Set up `HostPassword`.
    Enter `vmpass` as the password for the user ID set by `HostUserID`, and then press the **Enter** key. The system will ask you to re-enter the password for confirmation purposes. Re-enter `vmpass` and then press the **Enter** key.
    Note:
    The entered character strings are not displayed on the screen.
12. Set up `HostDomain`.
    Enter `vmdomain` as the name of the domain that the PFM - RM host belongs to, and then press the **Enter** key.
13. Set up `SSH_Type`.
    Press the **Enter** key without entering any value because this item is not used by Docker environment.
14. Set up `SSH_Client`.
    Press the **Enter** key without entering any value because this item is not used by Docker environment.
15. Set up `Log_Size`.
    Since the default value (16 megabytes) will be used, press the **Enter** key without entering any value.
16. Set up `UseVcpuMax`.
    Since the default value (N) will be used, press the **Enter** key without entering any value.

(3) Example of setting up monitoring targets

The following shows an example of executing a command to set up a monitoring target.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key RMVM -inst do
cker1 -target docker-host    ...1
 Target Host      []         : docker-host <Enter>    ...2
 VM_Host          []         :<Enter>                 ...3
 Security         [1]        :<Enter>                 ...4
 Port             [0]        :2376<Enter>             ...5
 UserID           []         :<Enter>                 ...6
 Password         []         :<Enter>                 ...7
 Domain           []         :<Enter>                 ...8
```

```
 Private_Key_File []          :<Enter>                 ...9
KAVE05361-I The monitoring target is now being added. (servicekey=RMVM, inst
=docker1, target=docker-host)
KAVE05362-I The monitoring target has been added. (servicekey=RMVM, inst=doc
ker1, target=docker-host)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *9*: Number of the corresponding step in the setup procedure.

To set up a monitoring target:

1. Execute the `jpcconf target setup` command.
   Specify `docker-host`, which is the host name of a physical server, as the name of a monitoring target.
2. Set `Target Host`.
   Enter `docker-host`, which is the host name of a physical server, and then press the **Enter** key.
3. Set `VM_Host`.
   Since the `Target Host` value will be used, press the **Enter** key without entering any value.
4. Set `Security`.
   Since the default value (`1`) will be used, press the **Enter** key without entering any value.
5. Set `Port`.
   Enter a port number of Docker environment and press the **Enter** key.
6. Set `UserID`.
   Press the **Enter** key without entering any value because this item is not used by Docker environment.
7. Set `Password`.
   Press the **Enter** key without entering any value because this item is not used by Docker environment.
8. Set `Domain`.
   Press the **Enter** key without entering any value because this item is not used by Docker environment.
9. Set `Private_Key_File`.
   Press the **Enter** key without entering any value because this item is not used by Docker environment.

2.6.5 For Podman environment

This subsection provides an example of setting up an instance environment to monitor Podman environment.

(1) Assumed Podman environment

The explanation in this subsection assumes that the user has the following Podman environment:

Monitored host

- Virtual environment software: Podman environment

- Physical server host name: podmanhost
- Port number: 22

PFM - RM host

- Login user name: vmuser
- Login password: vmpass
- Domain name: vmdomain

(2) Example of setting up an instance environment

A command execution example for setting up an instance environment follows.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf inst setup -key agt8 -inst podm
an1      ...1
 VM_Type          [vmware]        :podman<Enter>       ...2
 Interval         [300]           :<Enter>             ...3
 Offset           [10]            :<Enter>             ...4
 Std_Category     [Y]             :<Enter>             ...5
 Cpu_Category     [Y]             :<Enter>             ...6
 Memory_Category  [Y]             :<Enter>             ...7
 Disk_Category    [Y]             :<Enter>             ...8
 Network_Category [Y]             :<Enter>             ...9
 HostUserID       []              :vmuser<Enter>       ...10
 HostPassword     []              :vmpass<Enter>       ...11
                         Re-enter :vmpass<Enter>
 HostDomain       []              :vmdomain<Enter>     ...12
 SSH_Type         [putty]         :<Enter>             ...13
 SSH_Client       []              :C:\Putty\plink.exe<Enter>...14
 Log_Size (MB)    [16]            :<Enter>             ...15
 UseVcpuMax       [N]             :<Enter>             ...16
KAVE05080-I Creating an instance environment. (servicekey=agt8, inst=podman1
)
KAVE05081-I An instance environment has been created. (servicekey=agt8, inst
=podman1)
```

Legend:

<Enter>: **Enter** key must be pressed.

1 through *16*: Number of the corresponding step in the setup procedure.

To set up an instance environment:

1. Execute the jpcconf inst setup command.
   Specify any name as the instance name.

2. Set up VM_Type.
   Enter podman and press the **Enter** key.

3. Set up Interval.
   Since the default value (300) will be used, press the **Enter** key without entering any value.

4. Set up Offset.
   Since the default value (10) will be used, press the **Enter** key without entering any value.

5. Set up Std_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.
6. Set up Cpu_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.
7. Set up Memory_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.
8. Set up Disk_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.
9. Set up Network_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.
10. Set up HostUserID.
    Enter the PFM - RM host's vmuser user name to be used for connection and press the **Enter** key.
11. Set up HostPassword.
    Enter vmpass as the password for the user ID set by HostUserID, and then press the **Enter** key. The system will ask you to re-enter the password for confirmation purposes. Re-enter vmpass and then press the **Enter** key.
    Note:
    The entered character strings are not displayed on the screen.
12. Set up HostDomain.
    Enter vmdomain as the name of the domain that the PFM - RM host belongs to, and then press the **Enter** key.
13. Set up SSH_Type.
    Since the default value (putty) will be used, press the **Enter** key without entering any value.
14. Set up SSH_Client.
    Enter C:\Program Files\PuTTY\plink.exe as the executable file of the SSH command that is used for connection, and then press the **Enter** key.
15. Set up Log_Size.
    Since the default value (16 megabytes) will be used, press the **Enter** key without entering any value.
16. Set up UseVcpuMax.
    Since the default value (N) will be used, press the **Enter** key without entering any value.

(3) Example of setting up monitoring targets

The following shows an example of executing a command to set up a monitoring target.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key agt8 -inst po
dman1 -target podmanhost    ...1
 Target Host      []         :podmanhost<Enter>       ...2
 VM_Host          []         :<Enter>                 ...3
 Security         [1]        :<Enter>                 ...4
 Port             [0]        :<Enter>                 ...5
 UserID           []         :root<Enter>             ...6
```

```
 Password         []          :<Enter>                  ...7
 Domain           []          :<Enter>                  ...8
 Private_Key_File []          :C:\Putty\agt8.ppk<Enter>...9
KAVE05361-I The monitoring target is now being added. (servicekey=agt8, inst
=podman1, target=podmanhost)
KAVE05362-I The monitoring target has been added. (servicekey=agt8, inst=pod
man1, target=podmanhost)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *9*: Number of the corresponding step in the setup procedure.

To set up a monitoring target:

1. Execute the `jpcconf target setup` command.
   Specify `podmanhost`, which is the host name of a physical server, as the name of a monitoring target.
2. Set `Target Host`.
   Enter `podmanhost`, which is the host name of a physical server, and then press the **Enter** key.
3. Set `VM_Host`.
   Since the `Target Host` value will be used, press the **Enter** key without entering any value.
4. Set `Security`.
   Press the **Enter** key without entering any value because this item is not used by Podman environment.
5. Set `Port`.
   Since the default value (`22`) will be used, press the **Enter** key without entering any value.
6. Set `UserID`.
   Enter the superuser (`root`), and then press the **Enter** key.
7. Set `Password`.
   Press the **Enter** key without entering any value because this item is not used by Podman environment.
8. Set `Domain`.
   Press the **Enter** key without entering any value because this item is not used by Podman environment.
9. Set `Private_Key_File`.
   Enter the name of the private key file, and the press the **Enter** key.

2.6.6 For logical partitioning feature

This subsection provides an example of setting up an instance environment to monitor logical partitioning feature.

(1) Assumed logical partitioning feature

The explanation in this subsection assumes that the user has the following logical partitioning feature environment:

Monitored host

- Virtual environment software: logical partitioning feature
- Physical server host name: `vm-host`
- Physical server IP address: 192.168.1.10
- Login user name: None
- Login password: None
- Domain name: None

PFM - RM host#

- Login user name: None
- Login password: None
- Domain name: None

#

It is not used with logical partitioning feature.

(2) Example of setting up an instance environment

A command execution example for setting up an instance environment follows.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf inst setup -key RMVM -inst virt
age1       ...1
 VM_Type          [vmware]        :virtage<Enter>      ...2
 Interval         [300]           :<Enter>             ...3
 Offset           [10]            :<Enter>             ...4
 Std_Category     [Y]             :<Enter>             ...5
 Cpu_Category     [Y]             :<Enter>             ...6
 Memory_Category  [Y]             :<Enter>             ...7
 Disk_Category    [Y]             :<Enter>             ...8
 Network_Category [Y]             :<Enter>             ...9
 HostUserID       []              :vmuser<Enter>       ...10
 HostPassword     []              :vmpass<Enter>       ...11
                         Re-enter :vmpass<Enter>
 HostDomain       []              :vmdomain<Enter>     ...12
 SSH_Type         [putty]         :<Enter>             ...13
 SSH_Client       []              :<Enter>             ...14
 Log_Size (MB)    [16]            :<Enter>             ...15
 UseVcpuMax       [N]             :<Enter>             ...16
KAVE05080-I Creating an instance environment. (servicekey=RMVM, inst=virtage
1)
KAVE05081-I An instance environment has been created. (servicekey=RMVM, inst
=virtage1)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *16*: Number of the corresponding step in the setup procedure.

To set up an instance environment:

1. Execute the `jpcconf inst setup` command.
   Specify any name as the instance name.

2. Set up VM_Type.
   Enter virtage and press the **Enter** key.

3. Set up Interval.
   Since the default value (300) will be used, press the **Enter** key without entering any value.

4. Set up Offset.
   Since the default value (10) will be used, press the **Enter** key without entering any value.

5. Set up Std_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.

6. Set up Cpu_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.

7. Set up Memory_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.

8. Set up Disk_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.

9. Set up Network_Category.
   Since the default value (Y) will be used, press the **Enter** key without entering any value.

10. Set HostUserID.
    Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.

11. Set up HostPassword.
    Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.

12. Set HostDomain.
    Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.

13. Set up SSH_Type.
    Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.

14. Set up SSH_Client.
    Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.

15. Set Log_Size.
    Since the default value (16 MB) will be used, press the **Enter** key without entering any value.

16. Set up UseVcpuMax.
    Since the default value (N) will be used, press the **Enter** key without entering any value.

(3) Example of setting up monitoring targets

The following shows an example of executing a command to set up a monitoring target.

```
C:\Program Files\Hitachi\jp1pc\tools>jpcconf target setup -key RMVM -inst vi
rtage1 -target vm-host   ...1
```

```
 Target Host         []            :vm-host <Enter>       ...2
 VM_Host             []            :192.168.1.10<Enter>   ...3
 Security            [1]           :<Enter>               ...4
 Port                [0]           :<Enter>               ...5
 UserID                            :<Enter>               ...6
 Password            []            :<Enter>               ...7
 Domain              []            :<Enter>               ...8
 Private_Key_File    []            :<Enter>               ...9
KAVE05361-I The monitoring target is now being added. (servicekey=RMVM, inst
=virtage1, target=vm-host)
KAVE05362-I The monitoring target has been added. (servicekey=RMVM, inst=vir
tage1, target=vm-host)
```

Legend:

`<Enter>`: **Enter** key must be pressed.

1 through *9*: Number of the corresponding step in the setup procedure.

To set up a monitoring target (as in the above example):

1. Execute the `jpcconf target setup` command.
   Specify `vm-host`, which is the host name of a physical server, as the name of a monitoring target.
2. Set `Target Host`.
   Enter `vm-host`, which is the host name of a physical server, and then press the **Enter** key.
3. Set `VM_Host`.
   Enter `192.168.1.10`, which is the IP address of a physical server, and then press the **Enter** key.
4. Set `Security`.
   Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.
5. Set `Port`.
   Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.
6. Set `UserID`.
   Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.
7. Set `Password`.
   Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.
8. Set `Domain`.
   Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.
9. Set `Private_Key_File`.
   Press the **Enter** key without entering any value because this item is not used by logical partitioning feature.

2.7 Backing up and restoring

This section explains how to back up and restore PFM - RM for Virtual Machine.

To be prepared for an error-induced system failure, back up the setup information of PFM - RM for Virtual Machine. Also, acquire a backup when you modify the system, such as during setup of PFM - RM for Virtual Machine.

For details about how to back up and restore the entire Performance Management system, see the chapter that explains backup and restore operations in the *JP1/Performance Management User's Guide*.

2.7.1 Backing up

You can obtain a backup using a procedure such as copying files. Note that before you obtain a backup, you must stop all PFM - RM for Virtual Machine services (of all PFM - RM for Virtual Machine instances).

The tables below describe the files of PFM - RM for Virtual Machine to back up.

For details about other files, see the section presenting a list of files to be backed up for PFM - RM for Virtual Machine (in Windows) in the *JP1/Performance Management User's Guide*.

Table 2–22: PFM - RM for Virtual Machine files to back up (For physical host)

No.	File name	Explanation
1	*installation-folder*`\agt8\agent\*.ini` files	Remote Monitor Collector service settings files
2	*installation-folder*`\agt8\agent\`*instance-name*#1`\*.ini` files	
3	*installation-folder*`\agt8\agent\`*instance-name*#1`\groups\*.ini` files	
4	*installation-folder*`\agt8\agent\`*instance-name*#1`\targets\*.ini` files	
5	*installation-folder*`\agt8\agent\`*instance-name*#1`\targets\`*monitoring-target-name*#2`\*.ini` files#3	
6	*installation-folder*`\agt8\plugin\*.ini` files	
7	*installation-folder*`\agt8\plugin\jpcagt5virtage.d\*.ini` files	
8	*installation-folder*`\agt8\store\*.ini` files	Remote Monitor Store service settings files
9	*installation-folder*`\agt8\store\`*instance-name*#1`\*.ini` files	

#1

These are folders used for operation in an instance environment. In the case of an instance configuration, as many folders as there are instances are created.

#2

These are created in an amount equal to the number of monitoring targets.

#3

The file exists only when logical partitioning feature is specified as a monitoring target.

Table 2–23: PFM - RM for Virtual Machine files to back up (For logical host)

No.	File name	Explanation
1	*installation-folder*\agt8\agent*.ini files	Remote Monitor Collector service settings files
2	*installation-folder*\agt8\plugin*.ini files	
3	*installation-folder*\agt8\plugin\jpcagt5virtage.d*.ini files	
4	*environment-folder*\jp1pc\agt8*instance-name*#1*.ini files	
5	*environment-folder*\jp1pc\agt8\agent*instance-name*#1\groups *.ini files	
6	*environment-folder*\jp1pc\agt8\agent*instance-name*#1\targets *.ini files	
7	*environment-folder*\jp1pc\agt8\agent*instance-name*#1\targets *monitoring-target-name*#2*.ini files#3	
8	*installation-folder*\agt8\store*.ini files	Remote Monitor Store service settings files
9	*environment-folder*\jp1pc\agt8\store*instance-name*#1*.ini files	

#1

These are folders used for operation in an instance environment. In the case of an instance configuration, as many folders as there are instances are created.

#2

These are created in an amount equal to the number of monitoring targets.

#3

The file exists only when logical partitioning feature is specified as a monitoring target.

Important

When you acquire a backup for PFM - RM for Virtual Machine, try to manage the product version number of the acquired environment. For details about product version numbers, see the release note.

When you acquire a backup, you must record the instance and monitoring target configurations (including logical host environments).

2.7.2 Restoring

To restore the settings information of PFM - RM for Virtual Machine, first check the prerequisite conditions listed below, and then copy the backup file to its original location. The backed-up settings information file overwrites the settings information file on the host.

If PFM - RM for Virtual Machine is running on a logical host, the service definition information file on the physical host and in the environment folder is overwritten.

Prerequisite conditions

- PFM - RM for Virtual Machine must already be installed.

- PFM - RM for Virtual Machine services must be stopped.
- The backed up instances and monitoring targets (including logical host environments) must have been set up previously.

> **Important**
>
> When you are restoring the settings information of PFM - RM for Virtual Machine, the product version number of the environment from which the backup was acquired must completely match that of the environment to be restored. For details about product version numbers, see the release note.

2.8 Settings for using a Web browser to reference manuals

In Performance Management, by copying the manual from the CD-ROM provided with the program product onto the host on which PFM - Web Console is installed, you can use a Web browser to view the manual. If PFM - Web Console is running in a cluster system, copy the manual onto both the executing and standby physical hosts.

2.8.1 Setup procedure

(1) To view the manual from Help on PFM - Web Console

To copy the manual for viewing from PFM - Web Console Help:

1. By following the PFM - Web Console setup procedure, register PFM - RM for Virtual Machine in PFM - Web Console (add and set up PFM - RM for Virtual Machine).
2. On the host on which PFM - Web Console is installed, create a directory in which to copy the manual.
    - In Windows: *PFM-Web-Console-installation-folder*`\doc\en\`*Help-ID-of-PFM-RM-for-Virtual-Machine*
    - In UNIX: `/opt/jp1pcwebcon/doc/en/`*Help-ID-of-PFM-RM-for-Virtual-Machine*
      For details about the Help ID of PFM - RM for Virtual Machine, see *B. List of Identifiers*.
3. Copy the following files and directory from the manual CD-ROM to the directory created in Step 2.
   HTML manual
   In Windows: All `.htm` files and the `GRAPHICS` folder under *CD-ROM-drive*`\MAN\3021\`*document-number* (e.g., `03004A0D`)
   In UNIX: All `.htm` files and the `GRAPHICS` directory under `/`*CD-ROM-mount-point*`/MAN/3021/`*document-number* (e.g., `03004A0D`)
   PDF manual
   In Windows: PDF file under *CD-ROM-drive*`\MAN\3021\`*document-number* (e.g., `03004A0D`)
   In UNIX: PDF file under `/`*CD-ROM-mount-point*`/MAN/3021/`*document-number* (e.g., `03004A0D`)
   When copying, place the `index.htm` file (in the case of the HTML manual) and the PDF file (in the case of the PDF manual) directly under the created file. For details about how to copy a manual, see `readme.txt` in the manual's CD-ROM.
4. Restart PFM - Web Console.

(2) Viewing the manual from the hard disk of the computer you are using

Either use `setup.exe` in the CD-ROM to install the manual, or directly copy the HTML files, PDF files, and GIF files to the desired directory. For the HTML manual, use the following directory configuration:

```
html (for storing HTML files and PDF files)
|-GRAPHICS (for storing GIF files)
```

2.8.2 Viewing procedure

To view the manual from PFM - Web Console:

1. Click **Help** in the menu bar frame of the Main window of PFM - Web Console.
   The Help window opens.

2. Click the manual name or **PDF** that follows the manual name.
   Clicking the manual name displays an HTML-format manual. Clicking **PDF** displays a PDF-format manual.

Notes on displaying characters in a Web browser

In Windows, if you display the online manual from the **Start** menu, the HTML manual may be displayed in a Web browser window that is already open.

2.9 Notes regarding operation

2.9.1 Notes on operating PFM - RM for Virtual Machine

- Any virtualization software products supported by PFM - RM for Virtual Machine need to be installed, set up, and started before you start PFM - RM for Virtual Machine.
- When you perform a new installation of this product, the first icon displayed might be replaced immediately by another icon. This problem concerns only what icon is displayed. Product operation is not affected.
- For notes on executing the `jpcspm start` command or the `jpcspm stop` command for PFM - RM for Virtual Machine, see the chapter that describes commands in the manual *JP1/Performance Management Reference*.
- For notes on executing the `jpctool service list` command for the Remote Monitor Collector service or Remote Monitor Store service, see the chapter that describes commands in the manual *JP1/Performance Management Reference*.
- PFM - RM for Virtual Machine does not support configurations where the same monitoring target is monitored from more than one instance of the product.
- When specifying an execution module in the Windows WOW64 system directory (SysWOW64) for an alarm action definition, see the chapter that notes on creating alarms in the manual *JP1/Performance Management User's Guide*.
- To use PFM - RM for Virtual Machine in an IPv6 environment, the operating environment must be an IPv4/IPv6 dual stack environment. This is because the IPv4 stack is used to communicate with JP1/SLM - Manager.

3 Operations in a Cluster System

This chapter explains the installation and setup procedures necessary for operating PFM - RM for Virtual Machine in a cluster system, and the processing flow for operating PFM - RM for Virtual Machine.

3.1 Overview of cluster systems

A *cluster system* is a system in which multiple server systems are linked together to operate as a single system. PFM - RM for Virtual Machine can be operated in the following cluster system.

- HA cluster system

This section explains the configuration needed to operate PFM - RM for Virtual Machine in a cluster system. For a cluster system overview and the system configuration for operating a Performance Management system in a cluster system, see the chapter explaining configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

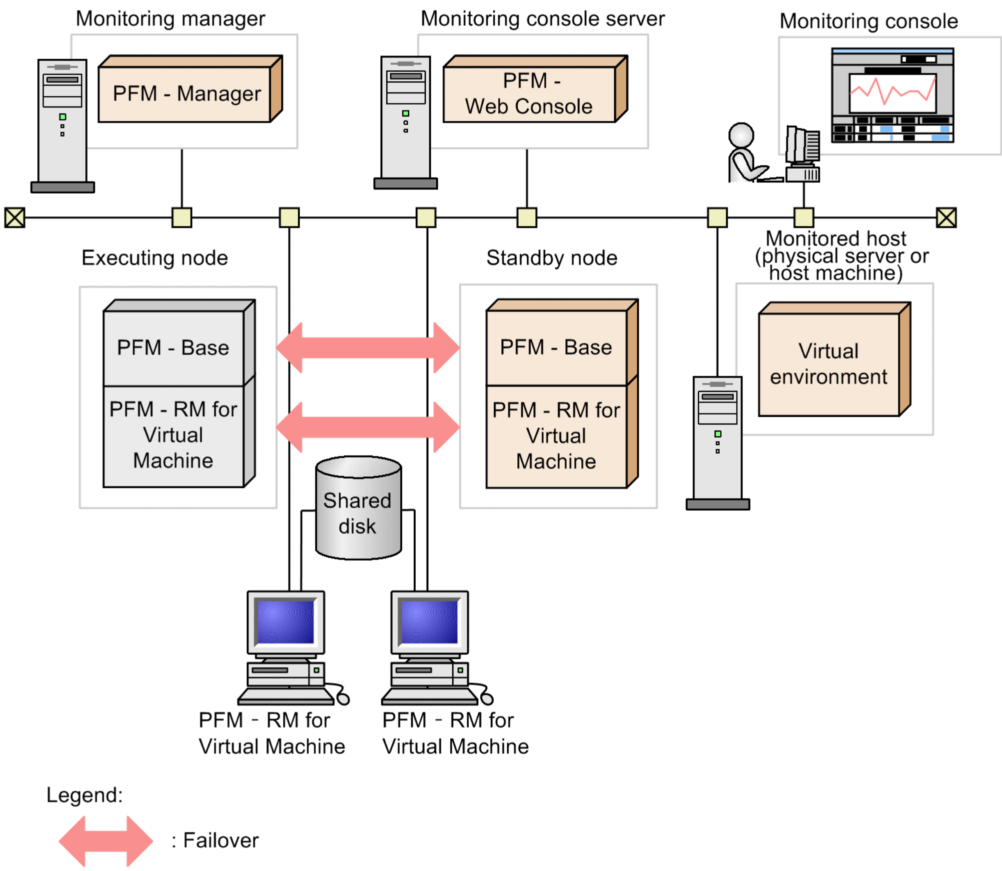
Note that in this chapter, a cluster system means an HA cluster system.

3.1.1 Configuring PFM - RM for Virtual Machine in an HA cluster system

When you operate PFM - RM for Virtual Machine in an HA cluster system, you can execute a failover when an error occurs, thereby improving system availability. To operate a PFM environment in an HA cluster system, you normally create an environment in which a PFM instance environment can be executed on both the executing node and a standby node, and use a configuration in which a set of data (e.g., data files, configuration files, and log files) are stored on a shared disk.

To operate a PFM - RM for Virtual Machine environment in an HA cluster system, use the configuration shown in the following figure.

Figure 3–1: Example of a PFM - RM for Virtual Machine configuration in an HA cluster system



As shown in Figure 3-1, PFM - RM for Virtual Machine runs in a logical host environment, and monitors a virtual environment that is running on a different host. Therefore, each host must be set up to be able to connect to the virtual environment under the same host name.

Definition information and performance information are stored on a shared disk, and are inherited during a failover. If multiple Performance Management programs are installed on a single logical host, they use the same shared folder.

Multiple instances of PFM - RM for Virtual Machine can be executed on a single node. If a configuration is being used that contains multiple cluster configurations (an active-active configuration), execute PFM - RM for Virtual Machine in each logical host environment. Each instance of PFM - RM for Virtual Machine runs independently and can execute a failover separately.

When using PFM - RM for Virtual Machine in a cluster environment, the support for the cluster environment depends on the virtual product being monitored. The following table shows the failover processing for each monitored virtualized product.

Monitoring target	Support for failover
VMware	active-active configuration
logical partitioning feature	active-active configuration
Hyper-V	active-standby configuration
KVM	active-standby configuration
Docker environment	active-standby configuration
Podman environment	active-standby configuration

3.2 Processing when a failover occurs

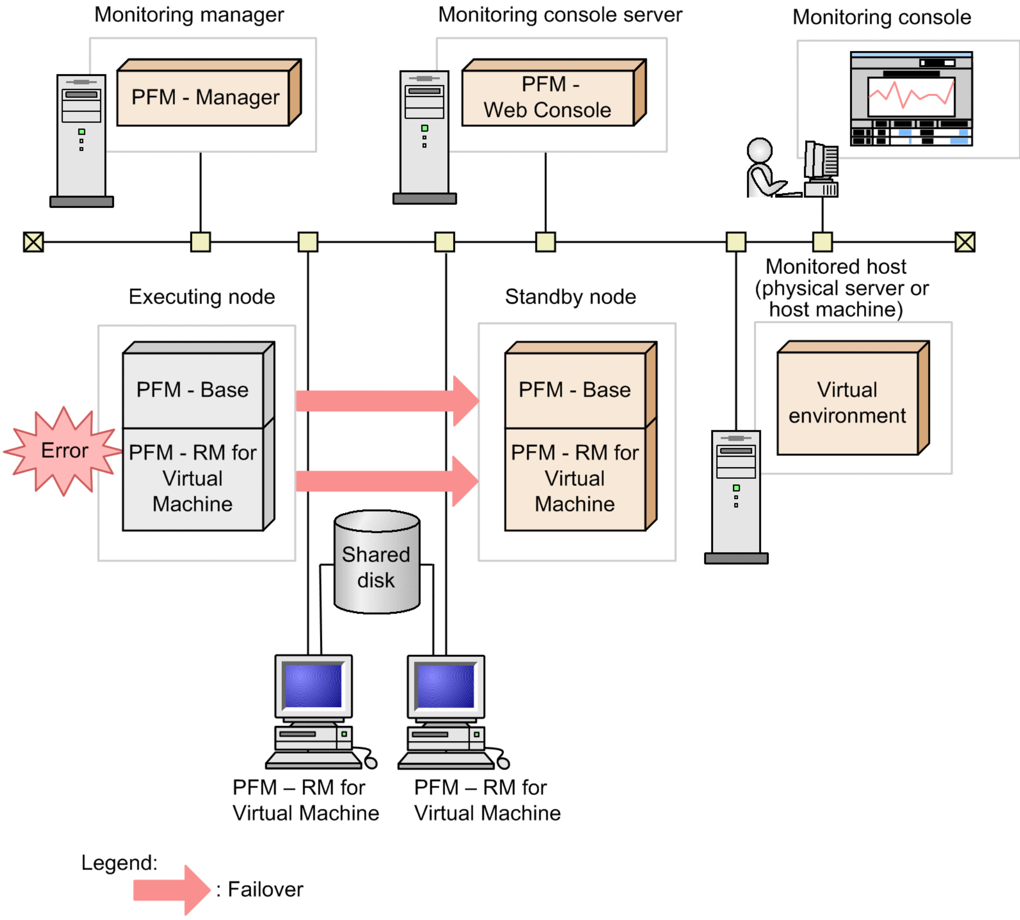
If an error occurs on the executing node, processing moves to a standby node.

This section explains the processing that occurs during a failover when an error occurs in PFM - RM for Virtual Machine. It also explains the effects on PFM - RM for Virtual Machine when an error occurs in PFM - Manager.

3.2.1 Failover when an error occurs on the PFM - RM host

The following figure shows how the PFM - RM host that runs PFM - RM for Virtual Machine performs failover if an error occurs.

Figure 3–2: Failover performed when an error occurs on the PFM - RM host



If you operate PFM - Web Console during a PFM - RM for Virtual Machine failover, the message `There was no answer(-6)` is displayed. In this case, wait until a failover is complete before operating PFM - Web Console.

If you operate PFM - Web Console after a PFM - RM for Virtual Machine failover, you will be connected to the PFM - RM for Virtual Machine instance that has started at the failover destination node.

3.2.2 Effects when PFM - Manager stops

When PFM - Manager stops, this affects the entire Performance Management system.

PFM - Manager centrally manages the agent information of the PFM - RM for Virtual Machine instances that are running at individual nodes. PFM - Manager also controls alarm event notification when PFM - RM for Virtual Machine exceeds a threshold during performance monitoring, and it controls the execution of actions that are triggered by alarms. Consequently, if PFM - Manager stops, the entire Performance Management system is affected as described in the following table.

Table 3–1: Effects on PFM - RM for Virtual Machine when PFM - Manager stops

Program name	Effects	Corrective action
PFM - RM for Virtual Machine	If PFM - Manager stops while PFM - RM for Virtual Machine is active, the following actions take place: • Performance data collection continues. • Since any alarm event that occurs cannot be reported to PFM - Manager, alarm events are held for each alarm definition, and notification is retried until PFM - Manager starts. When the number of alarm events held exceeds 3, the oldest alarm event is overwritten. Furthermore, stopping PFM - RM for Virtual Machine deletes the alarm events being held. • Alarm statuses that have already been reported to PFM - Manager are reset when PFM - Manager restarts. Then, after PFM - Manager checks the status of PFM - RM for Virtual Machine, the alarm statuses are refreshed. • If you try to stop PFM - RM for Virtual Machine, the operation will take a long time because it cannot be reported to PFM - Manager.	Start PFM - Manager. The PFM - RM for Virtual Machine instances that are running can be operated as is. However, alarms might not be reported as expected in some cases. Therefore, after PFM - Manager is restored, check the `KAVE00024-I` message, which is output to the common message log.

Before you decide on the corrective action to take, consider the effects that stoppage of PFM - Manager has. Also note that, in addition to trouble-induced stoppages, there are times that you need to stop PFM - Manager to change a configuration or perform maintenance tasks. Therefore, we recommend that you perform maintenance at a time when there will be little impact on system operations.

3.3 Installation and setup

This section explains how to install and set up PFM - RM for Virtual Machine in a cluster system.

For details about how to install and set up PFM - Manager, see the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

3.3.1 Preparation for installation and setup

This subsection explains the prerequisite conditions, required information, and notes you must be aware of before you install and set up PFM - RM for Virtual Machine.

(1) Prerequisite conditions

To use PFM - RM for Virtual Machine in a cluster system, the following prerequisite conditions must be satisfied.

(a) Cluster system

Confirm that the following conditions are met:

- The cluster system is controlled by the cluster software.
- The cluster software is configured to control the startup and termination of the PFM - RM for Virtual Machine running on logical hosts.
- Both the executing and standby systems are configured to suppress error reporting to Microsoft.

> **Important**
>
> If the occurrence of an error causes a dialog box for reporting the error to Microsoft to be displayed. Because this dialog box can prevent a failover from occurring, you need to suppress error reporting. If error reporting is currently enabled, suppress it by specifying the following setting:
>
> Windows Server 2012:
>
> 1. Select **Control Panel**, **System and Security**, **Action Center**, and then **Maintenance**.
> 2. Under **Check for solutions to problem reports**, select **Settings**.
> 3. In the Windows Error Reporting Configuration dialog box, select **I don't want to participate, and don't ask me again**.
> 4. Click the **OK** button.
>
> Windows Server 2016 or later:
>
> 1. Right-click the Windows **Start** menu and then choose **Run** from the displayed menu.
> 2. Enter `gpedit.msc`, and then click the **OK** button.
> The Local Group Policy Editor appears.
> 3. Click **Computer Configuration**, **Administrative Templates**, **Windows Components**, and then **Windows Error Reporting**.
> 4. In the right pane, right-click **Disable Windows Error Reporting**, and then from the displayed menu, choose **Edit**.
> The setting window appears.
> 5. In the setting window, select the **Enabled** check box.
> 6. Click the **OK** button.

(b) Shared disk

Make sure that the following conditions are satisfied:

- Each logical host must have a shared disk, and it must be possible to switch from the executing node to a standby node.
- The shared disk must be physically connected to each node via Fibre Channel or SCSI. Performance Management does not support a configuration in which a network drive or a disk replicated via a network is used as a shared disk.
- Even if a problem causes a process that is using the shared disk to remain when a failover occurs, it must be possible to forcibly take the shared disk offline and enable a failover using a means such as cluster software.
- If multiple PFM products are used on a single logical host, the folder name of the shared disks must be the same. You can change the storage destination of the Store database and store it in a different folder on the shared disk.

(c) Logical host names and logical IP addresses

Make sure that the following conditions are satisfied:

- Each logical host must have a logical host name, and each logical host name must have a corresponding logical IP address, and it must be possible to switch from the executing node to a standby node.
- The logical host names and logical IP addresses must be set in the `hosts` file and name server.
- If DNS is used, a host name excluding the domain name, not the FQDN name, must be used as the logical host name.
- Physical host names and logical host names must be unique within the system.

> **Important**
>
> - Do not specify a physical host name (host name displayed by the `hostname` command) for the logical host name. Otherwise, communication may not be processed correctly.
> - A logical host name can use from 1 through 32 bytes of alphanumeric characters. Spaces and the following symbols cannot be specified:
>   / \ : ; * ? ' " < > | & = , .
> - You cannot specify a logical host name that begins with the character string `localhost`, an IP address, or a hyphen (`-`).

(d) Settings for using IPv6

Performance Management supports network configurations in IPv4 and IPv6 environments. Performance Management is able to operate with a network configuration in which IPv4 and IPv6 environments coexist.

PFM - RM for Virtual Machine can communicate with PFM - Manager via IPv6.

Note that for Performance Management to operate with that network configuration, the OS of the host on which PFM - Manager is installed must be Windows or Linux.

For information about the applicable scope of communications in IPv4 and IPv6 environments, see *K. About Communication in IPv4 Environments and IPv6 Environments*.

To make it possible to communicate via IPv6, the setting for using IPv6 must be enabled on both the PFM - Manager host and the PFM - RM host. This setting can be enabled by using the `jpcconf ipv6 enable` command. However, there are cases in which execution of this command is necessary and others where it is unnecessary.

Cases in which the `jpcconf ipv6 enable` command needs to be executed:

- When switching from an IPv4 environment to an IPv6 environment on each host
- When switching the PFM - Manager environment from IPv4 to IPv6 in an environment in which both IPv4 and IPv6 are used

Cases in which the `jpcconf ipv6 enable` command does not need to be executed:

- When each host already operates in an IPv6 environment
- When PFM - Manager operates in an IPv6 environment in an environment in which both IPv4 and IPv6 are used

The following shows an example of executing the `jpcconf ipv6 enable` command:

```
jpcconf ipv6 enable
```

Make sure that you execute the `jpcconf ipv6 enable` command on both the executing node and the standby node.

For details about the `jpcconf ipv6 enable` command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*. For details about the conditions or occasions for executing the `jpcconf ipv6 enable` command, see the chapter that describes network configuration examples in an environment that includes IPv6 in the *JP1/Performance Management Planning and Configuration Guide*.

When you use IPv6 for communication between a monitored host and PFM - RM for Virtual Machine, specify the name of a monitored host where name resolution can be performed.

Communication between PFM - RM for Virtual Machine and a monitoring target is performed with an IP address that can be resolved. Also, if an IPv4 environment and an IPv6 environment are both used, and communication between PFM - RM for Virtual Machine and the monitoring target fails with an IP address that can be resolved, the communication is not retried by using another IP address.

For example, if communication fails when IPv4 is used, IPv6 is not used to retry communication. Similarly, if communication fails when IPv6 is used, IPv4 is not used to retry communication. Make sure that a connection can be established beforehand.

(2) Information necessary for setting up PFM - RM for Virtual Machine to run on a logical host

To set up PFM - RM for Virtual Machine to run on a logical host, you need the types of information listed in the table below in addition to the environment information normally necessary for setting up PFM - RM for Virtual Machine.

Table 3–2: Information necessary for setting up PFM - RM for Virtual Machine to run on a logical host

Item	Example
Logical host name	`jp1-halvm`
Logical IP address	`172.16.92.100`
Shared disk	`S:\jp1`

When multiple Performance Management programs are set to run on a single logical host, they must use the same folder on the shared disk.

To estimate the size necessary for the shared disk, see *A. Estimating System Requirements*.

(3) Notes on using PFM - RM for Virtual Machine to cause a failover on a logical host

If you are using a system configuration in which PFM - RM for Virtual Machine runs on a logical host, decide whether a failover should occur on the entire logical host when an error occurs in PFM - RM for Virtual Machine.

If an error in PFM - RM for Virtual Machine causes a failover on the entire logical host, business applications that are running on the same logical host as PFM - RM for Virtual Machine will also fail over. This may adversely affect operations.

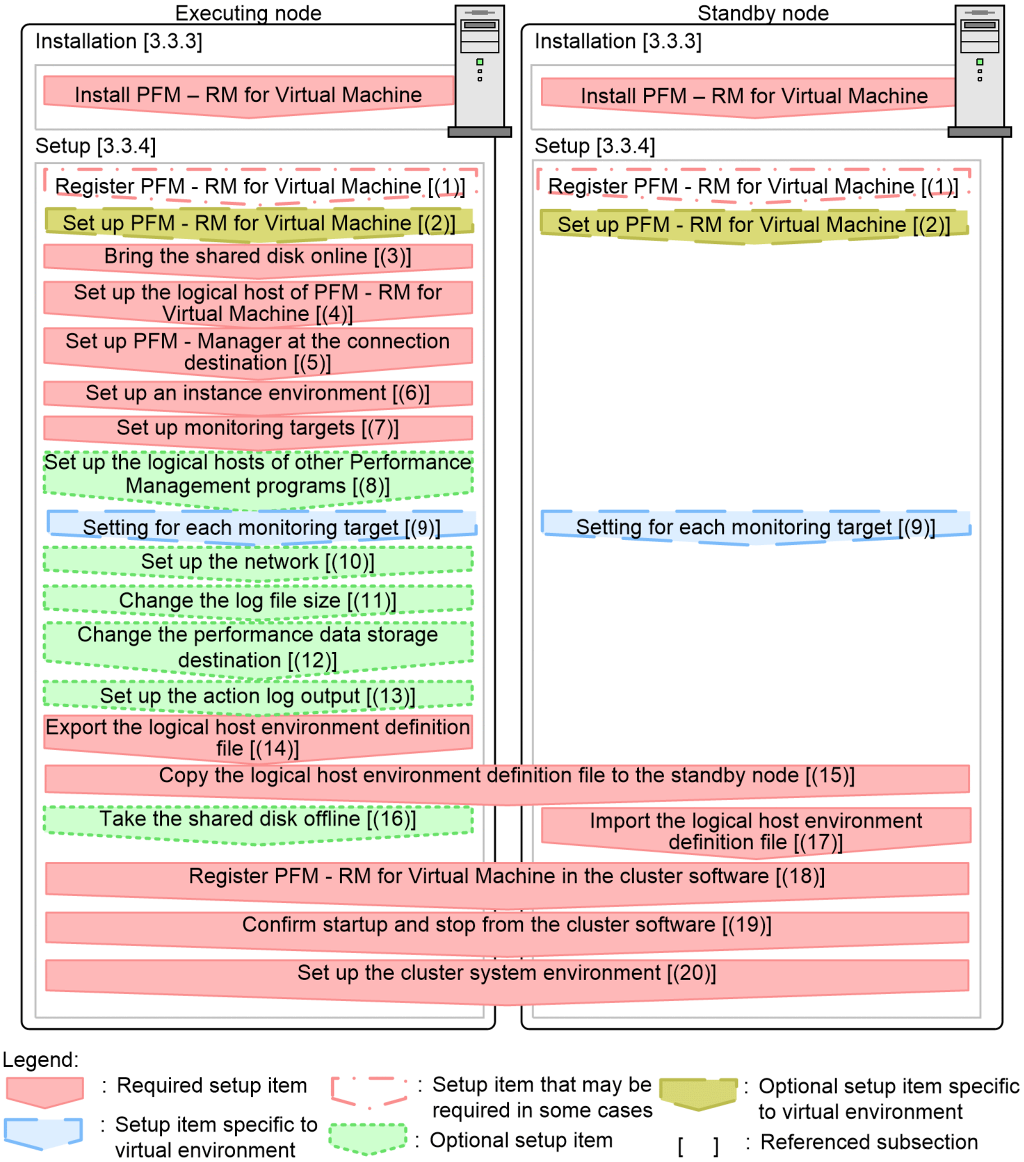
(4) Note on version upgrading of the logical host

When you upgrade the version of PFM - RM for Virtual Machine running on the logical host, you need to place the shared disk online on either the executing node or the standby node.

3.3.2 Installation and setup workflow

The following figure shows the flow for installing and setting up PFM - RM for Virtual Machine so that it runs on a logical host in a cluster system.

Figure 3–3: Flow for installing and setting up PFM - RM for Virtual Machine so that it runs on a logical host in a cluster system



Note:

Even when you set up PFM - RM for Virtual Machine in the logical host environment, the definition content of PFM - RM for Virtual Machine in the physical host environment is not inherited. For both the logical host environment and the physical host environment, a new environment is created when an instance environment is set up.

Note that you can select whether to execute a setup command requiring user entry interactively or non-interactively.

If you execute a setup command interactively, you need to enter a value in accordance with command directives.

If you execute a setup command non-interactively, user entry is not required because the operator entry required during command execution can be replaced by the specification of options or definition files. Also, batch processing or remote execution can automate setup operations to reduce administrator workload and operating costs. Non-interactive commands are useful in the following cases:

- You want to regularly change the password to be used for connection with the monitoring target.
- You want to improve operational efficiency when adding multiple monitoring targets.

For details about commands, see the manual *JP1/Performance Management Reference*.

3.3.3 Installation procedure

Install PFM - RM for Virtual Machine on both the executing node and the standby node.

> **Important**
>
> The installation destination is the local disk. Do not install PFM - RM for Virtual Machine on a shared disk.

The installation procedure is the same as that used for a non-cluster system. For details about the installation procedure, see *2.1.3 Installation procedure*.

3.3.4 Setup procedure

This subsection explains the setup for operating Performance Management in a cluster system.

The setup procedure depends on the virtual environment to be monitored. The icon VMware, Hyper-V, KVM , VMware , Hyper-V , logical partitioning feature , KVM , Docker environment , or Podman environment indicates a setup item required for the indicated virtual environment.

There are setup procedures for the executing node and for the standby node. Set up the executing node first and then set up the standby node.

The icon Executing system indicates an item that must be set up for the executing node, and the icon Standby system indicates an item that must be set up for the standby node. The icon Optional indicates an item that may be required depending on the environment that is used, or an optional setup item that is available for changing the default setting.

> **Important**
>
> Do not set `JPC_HOSTNAME` as an environment variable because it is used in Performance Management. If you do so, Performance Management will not operate correctly.

(1) Registering PFM - RM for Virtual Machine VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Executing system Standby system Optional

To use PFM - Manager and PFM - Web Console to centrally manage PFM - RM for Virtual Machine, you need to register PFM - RM for Virtual Machine in PFM - Manager and PFM - Web Console.

The conditions and procedure for registering PFM - RM for Virtual Machine are the same as that used for a non-cluster system. For details about the procedure, see *2.1.4(1) Registering PFM - RM for Virtual Machine*.

(2) Setting PFM - RM for Virtual Machine

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system | Optional

You need to specify the same settings on both the executing system and the standby system in PFM - RM for Virtual Machine. For details about the procedure, see *2.1.4(2) Setting PFM - RM for Virtual Machine*.

You need to specify the same settings on both the executing node and the standby node.

(3) Bringing the shared disk online

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

Confirm that the shared disk is online. If the shared disk is not online, bring it online using a cluster software operation or volume manager operation.

(4) Setting up the logical host of PFM - RM for Virtual Machine

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

Create a logical host environment by executing the `jpcconf ha setup` command. Executing this command copies the necessary data to the shared disk, sets up a definition for the logical host, and creates a logical host environment.

Note:

Before executing the command, stop all Performance Management programs and services in the entire Performance Management system. For details about how to stop services, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

To set up a logical host:

1. Create a logical host environment of PFM - RM for Virtual Machine by executing the `jpcconf ha setup` command.
   Execute the command as follows.
   ```
   jpcconf ha setup -key RMVM -lhost jp1-halvm -d S:\jp1
   ```
   Specify a logical host name in the `-lhost` option. Here, `jp1-halvm` is used as the logical host name. If DNS is used, specify a logical host name with the domain name omitted.
   Specify the folder name of the shared disk in the environment folder name of the `-d` option. For example, if `-d S:\jp1` is specified, `S:\jp1\jp1pc` is created and a logical host environment file is created.

   Important

   Make sure that the path name specified as the environment folder name does not include the following characters:

   ( )

   If at least one of the above characters is included, although the environment of the logical host might be successfully created, PFM - RM for Virtual Machine cannot start.

2. Confirm the logical host settings by executing the `jpcconf ha list` command.
   Execute the command as follows.

```
jpcconf ha list -key all
```

Confirm that the created logical host environment is correct.

(5) Setting up PFM - Manager at the connection destination

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

Set up the PFM - Manager that manages PFM - RM for Virtual Machine by executing the `jpcconf mgrhost define` command.

1. Set up PFM - Manager at the connection destination by executing the `jpcconf mgrhost define` command.
   Execute the command as follows.

   ```
   jpcconf mgrhost define -host jp1-hal -lhost jp1-halvm
   ```

   Specify the host name of PFM - Manager at the connection destination in the `-host`. If the PFM - Manager instance at the connection destination is running on a logical host, specify the logical host name of PFM - Manager at the connection destination in the `-host`. Here, `jp1-hal` is used as the logical host name of PFM - Manager.

   Specify the logical host name of PFM - RM for Virtual Machine in the `-lhost` option. Here, `jp1-halvm` is used as the logical host name of PFM - RM for Virtual Machine.

(6) Setting up an instance environment

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

Set up an instance environment for PFM - RM for Virtual Machine by executing the `jpcconf inst setup` command.

The setup procedure is the same as that used for a non-cluster system. Note, however, that in the case of a cluster system, you must specify a logical host name in the `-lhost` option when you execute the `jpcconf inst setup` command.

The procedure for specifying the `jpcconf inst setup` command for a cluster system is as follows.

```
jpcconf inst setup -key RMVM -lhost jp1-halvm -inst inst1
```

Although an example of interactive command execution is shown here, the `jpcconf inst setup` command can be also executed non-interactively. For details about the `jpcconf inst setup` command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*.

For other setting details and procedures, see *2.1.4(3) Setting up an instance environment*.

(7) Setting up monitoring targets

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

Set up the monitoring-target hosts of PFM - RM for Virtual Machine by executing the `jpcconf target setup` command.

The setup procedure is the same as that used for a non-cluster system. Note, however, that in the case of a cluster system, you must specify a logical host name with the `-lhost` option when you execute the `jpcconf target setup` command.

The following shows an example of the `jpcconf target setup` command that can be executed in a cluster system.

```
jpcconf target setup -key RMVM -lhost jp1-halvm -inst inst1 -target targetho
st1
```

Although an example of interactive command execution is shown here, the `jpcconf target setup` command can be also executed non-interactively. For details about the `jpcconf target setup` command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*.

For details about other setting and procedures, see *2.1.4(4) Setting up monitoring targets*.

(8) Setting up the logical hosts of other Performance Management programs

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment

Executing system | Optional

Besides PFM - RM for Virtual Machine, if there are other programs such as PFM - Manager and PFM - Agent or PFM - RM that must be set up on the same logical host, set them up at this stage.

For details about how to perform setup, see the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

(9) Settings for each monitoring target

Executing system | Standby system | Optional

This section describes the settings required for each monitored virtual environment.

(a) For VMware

VMware

Confirm that the following condition is satisfied:

- Encrypted communication necessary for SSL/TSL connections with monitored hosts is configured on both the executing node and the standby node.

For details about the SSL/TLS connection settings, see *2.5.1 For VMware*.

Use user defined records to monitor performance information that is not retrieved by PFM - RM for Virtual Machine. For details about the user defined records, *2.5.1(6) User defined records*.

(b) For Hyper-V

Hyper-V

Confirm that the following condition is satisfied:

- The same user account that can connect to the monitored host via WMI is created on both the executing node and the standby node.

For details about the WMI connection settings, see *2.5.2 For Hyper-V*.

(c) For KVM

KVM

Confirm that the following conditions are satisfied:

- A private key is stored at the same path on both the executing node and the standby node.
- The private key can be used to establish SSH connection with a monitored host.

- When SSH_Type is set to putty, PuTTY is installed in the same path on both the executing node and the standby node.
  If you use OpenSSH, which comes with Windows Server 2019, as the SSH client, you do not have to install PuTTY.

  Important

  Use either of the following methods to register the private key and public key:

  - After creating a private key on the active server and copying the private key to the standby server, pair the private key with the public key that is distributed from the active server to the monitored host.
  - After creating public keys separately on the active and standby servers, register both of these public keys on the monitored host so that either of the public keys can be paired with the private key.

For details about the SSH connection settings, see *2.5.7 SSH connection settings*.

(d) For Docker environment Docker environment

Confirm that the following conditions are satisfied:

- The CA and client certificates that are necessary for SSL/TLS connection with a monitored host are installed on both the executing node and the standby node.

For details about the SSL/TLS connection settings, see *2.5.4 For Docker environment*.

(e) For Podman environment Podman environment

The scenario for KVM also applies. See *(c) For KVM*.

(f) For logical partitioning feature logical partitioning feature

Confirm that the following conditions are satisfied:

- In the environment with logical partitioning feature to be monitored, set the IP address of the machine on which the monitoring agent is installed.

For details, see *2.5.6 For logical partitioning feature*.

(10) Setting up the network VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Executing system Optional

This is necessary only when you want to change the setup according to the network configuration that uses Performance Management.

During network setup, you can specify the following two items:

- IP address
  To specify the IP address to be used for operating Performance Management in a network that is connected to multiple LANs, directly edit the content of the jpchosts file.
  Copy the edited jpchosts file from the executing node to the standby node (place the jpchosts file under the *installation-folder*\ folder of the physical host).

For details about how to set IP addresses, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

- Port numbers

  To communicate among Performance Management programs via a firewall, set up port numbers by executing the `jpcconf port` command.

  For details about how to set up port numbers, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*, and the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

(11) Changing the log file size

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Optional

The operation status of Performance Management is output to a log file specific to Performance Management. This log file is called the *common message log*. This setting is necessary only when you want to change the size of this file.

For details, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*.

(12) Changing the performance data storage destination

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Optional

This setting is necessary only when you want to change the folder at the save destination, the backup destination, the export destination, or the import destination for the database that stores the performance data managed by PFM - RM for Virtual Machine.

For details about this setting, see *2.1.4(8) Changing the performance data storage destination*.

(13) Setting up the action log output

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Optional

This setting is necessary to output action logs when a PFM service starts up or stops, or when the connection status to PFM - Manager changes. An action log contains history information that is output in conjunction with the alarm function, which monitors items such as the system load threshold.

For details about this setting, see *I. Outputting Action Log Data*.

(14) Exporting the logical host environment definition file

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

When you have created a logical host environment for PFM - RM for Virtual Machine, export the environment definitions to a file. An export operation immediately outputs to a file the Performance Management program definition information that is set up on the logical host. If you are setting up other Performance Management programs on the same logical host, export the information after all of these programs have been set up.

To export the logical host environment definitions:

1. Export the logical host environment definitions by executing the `jpcconf ha export` command.

   The logical host environment definition information that has been created up until now is output to an export file. You can use any desired export file name.

For example, to export the logical host environment definitions to the lhostexp.txt file, execute the command as follows.

```
jpcconf ha export -f lhostexp.txt
```

(15) Copying the logical host environment definition file to the standby node

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

Copy the logical host environment definition file exported in *(14) Exporting the logical host environment definition file* from the executing node to the standby node.

(16) Taking the shared disk offline

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Optional

Using a cluster software operation or volume manager operation, take the shared disk offline to complete the task. If you plan to continue using the shared disk, there is no need to take it offline.

(17) Importing the logical host environment definition file

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Standby system

Import the export file copied from the executing node onto the standby node.

To specify the setting for executing the Performance Management program of the logical host created in the executing node on the standby node, use the `jpcconf ha import` command. If multiple Performance Management programs have been set up for a single logical host, all logical host environment definition files will be imported at once.

When you execute this command, there is no need to bring the shared disk online.

To import the logical host environment definition file:

1. Import the logical host environment definitions by executing the `jpcconf ha import` command.
   Execute the command as follows.

   ```
   jpcconf ha import -f lhostexp.txt
   ```

   When the command is executed, the settings of the standby node environment are changed to match the content of the export file. As a result, the settings for starting PFM - RM for Virtual Machine of the logical host are implemented.
   If a fixed port number was set by the `jpcconf port` command during setup, it will be set in the same way.

2. Confirm the logical host settings by executing the `jpcconf ha list` command.
   Execute the command as follows.

   ```
   jpcconf ha list -key all
   ```

   Confirm that the command displays the same content that was displayed when the `jpcconf ha list` command was executed at the executing node.

(18) Registering PFM - RM for Virtual Machine in cluster software

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

To operate Performance Management programs in a logical host environment, you need to register them in cluster software, and set up the environment such that the Performance Management programs can be started and stopped based on controls from the cluster software.

For details about how to register PFM - RM for Virtual Machine in cluster software, see the cluster software documentation.

The items registered in Windows WSFC are used as an example to describe the settings to be specified upon the registration of PFM - RM for Virtual Machine in the cluster software.

For PFM - RM for Virtual Machine, register in the cluster software the services described in the following table.

For the dependency settings specified if PFM - RM for Virtual Machine is on the PFM - Manager logical host, see the chapter on cluster system setup and operation in the *JP1/Performance Management User's Guide*.

Table 3–3: PFM - RM for Virtual Machine services to register in cluster software

No.	Name	Service name	Resource dependencies
1	`PFM - RM Store for Virtual Machine` *instance-name* `[LHOST]`	`JP1PCAGT_8S_`*instance-name* `[LHOST]`	IP address resource#1 Physical disk resource#2
2	`PFM - RM for Virtual Machine` *instance-name* `[LHOST]`	`JP1PCAGT_8A_`*instance-name* `[LHOST]`	Cluster resource for Item No. 1
3	`PFM - Action Handler [LHOST]`	`JP1PCMGR_PH [LHOST]`	IP address resource#1 Physical disk resource#2

#1

IP address resource defined in the cluster environment of the virtual environment

#2

Shared disk resource

Replace `[LHOST]` with a logical host name. If the instance name is `inst1` and the logical host name is `jp1-halvm`, the name becomes `PFM - RM Store for Virtual Machine inst1 [jp1-halvm]`, and the service name becomes `JP1PCAGT_8S_inst1 [jp1-halvm]`.

For WSFC, register these services as WSFC resources. Set each resource as follows:

- In **Resource Type**, register the resource as **Generic Service**.
- Set **Dependencies** as shown in *Table 3-3*.
- Do not set **Startup parameters**.
- Set the **Policies** tab in Properties by taking into account whether you want a failover to occur in the event of a Performance Management program failure.
  For example, to trigger a failover when a failure occurs in PFM - RM for Virtual Machine, set the **Policies** tab as follows:
  - Select the **If resource fails, attempt restart on current node** radio button.

- Select the **If restart is unsuccessful, fail over all resources in this Role** check box.
- In principle, set **Maximum restarts in the specified period:** to **3**.

Note:

- Since services that are registered in the cluster software are started and stopped from the cluster, set their **Startup type** to **Manual** so that they do not automatically start when the OS starts. Immediately after a service is set up by the `jpcconf ha setup` command, it is set to **Manual**.

  Do not perform a forced stop by executing the following command:

```
jpcspm stop -key all -lhost jp1-halvm -kill immediate
```

- When PFM - RM for Virtual Machine links with the integrated management product JP1/Integrated Management, set dependencies so that PFM - RM for Virtual Machine services stop before JP1/Base services stop.

(19) Confirming startup and stop from the cluster software

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

From the cluster software, try to start and stop the Performance Management programs at each node, and confirm that the starting and stopping operations work normally.

(20) Setting up the cluster system environment

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

Once you have finished setting up the Performance Management programs, set up the environment for these programs so that a report can be displayed from PFM - Web Console on the operation status of a monitored target according to the operating mode, and so that a user notification can be sent when a problem occurs on the monitored target.

For details about how to set up the environment for Performance Management programs, see the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

3.4 Uninstallation and unsetup

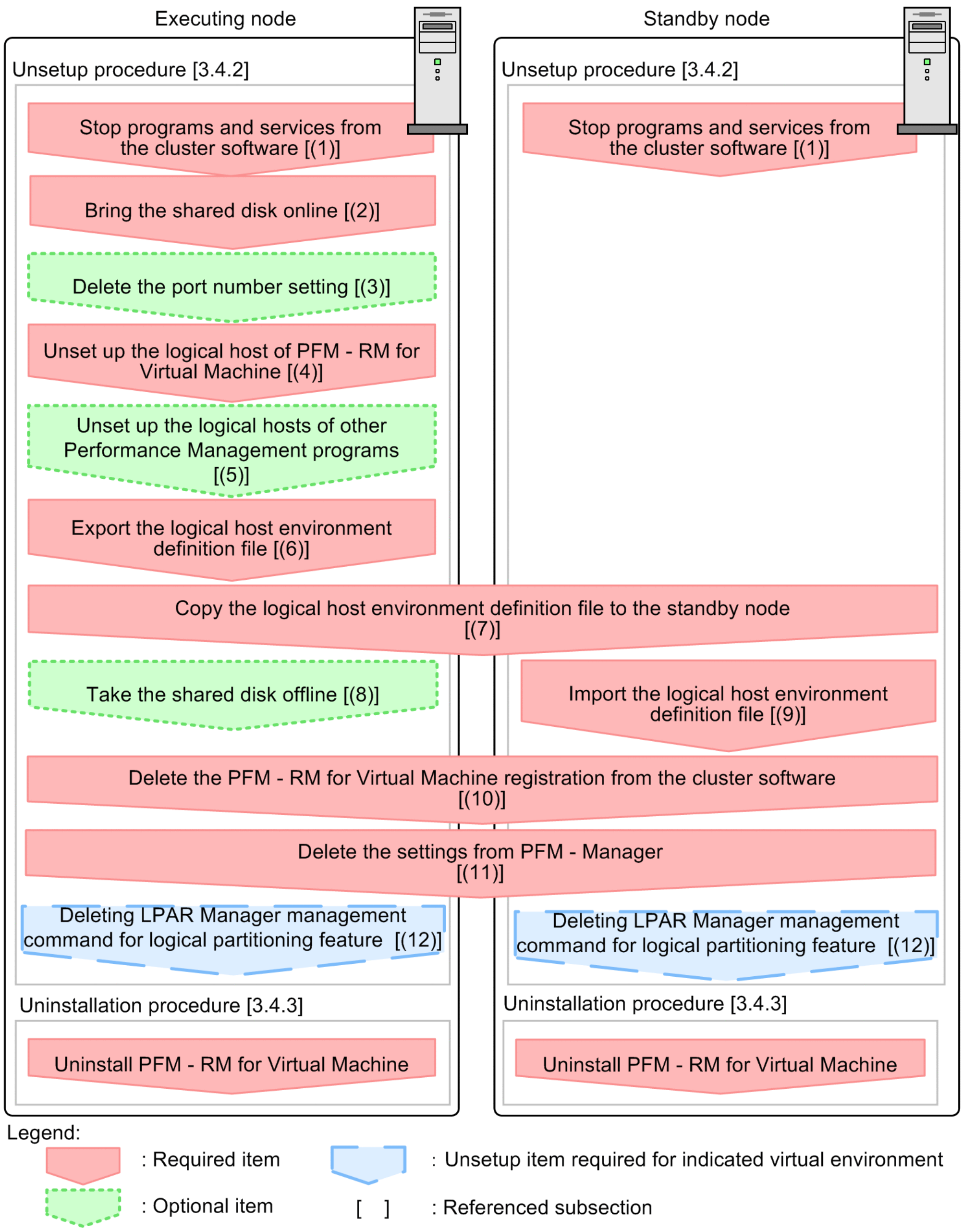
This section explains how to uninstall and unset up a PFM - RM for Virtual Machine instance that has been operating in a cluster system.

For details about how to uninstall and unset up PFM - Manager, see the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

3.4.1 Uninstallation and unsetup flow

The figure below shows the flow for uninstalling and unsetting up a PFM - RM for Virtual Machine instance that has been operating in a cluster system.

Figure 3–4: Flow for uninstalling and unsetting up a PFM - RM for Virtual Machine instance that runs on a logical host in a cluster system



3.4.2 Unsetup procedure

Unset up the logical host environment. The unsetup procedure depends on the virtual environment to be monitored. The icon VMware, Hyper-V, KVM logical partitioning feature Docker environment or Podman environment indicates an unsetup item required for the indicated virtual environment.

There are unsetup procedures for the executing node and for the standby node. Perform unsetup at the executing node first and then at the standby node.

The icon Executing system indicates an item that must be unset up from the executing node; the icon Standby system indicates an item that must be unset up from the standby node. The icon Optional indicates an item that may require unsetup depending on the environment that is used, or an optional unsetup item that is available for changing the default setting.

This subsection explains how to unset up PFM - RM for Virtual Machine.

(1) Stopping programs and services from the cluster software

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

From the cluster software, stop the Performance Management programs and services that are active at the executing node and the standby node. For details about how to stop programs and services, see the cluster software documentation.

(2) Bringing the shared disk online

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

Confirm that the shared disk is online. If the shared disk is not online, bring it online using a cluster software operation or volume manager operation.

(3) Deleting the port number setting

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Optional

This procedure is required only if you set up port numbers using the `jpcconf port` command during setup in an environment that uses a firewall.

For details about how to delete port numbers, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*, and the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*.

(4) Unsetting up the logical host of PFM - RM for Virtual Machine

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

The procedure for unsetting up the logical host follows.

Note:

If you delete the logical host environment while the shared disk is offline, only the settings of the logical host residing on the physical host are deleted, and folders and files on the shared disk are not deleted. In this case, you need to bring the shared disk online and manually delete the `jp1pc` folder from the environment folder.

To unset up the logical host:

1. Confirm the logical host settings by executing the `jpcconf ha list` command.

   Execute the command as follows.

   ```
   jpcconf ha list -key all -lhost jp1-halvm
   ```

   Before unsetting up the logical host environment, check the current settings. Check items such as the logical host name and the shared disk path.

2. Delete the information about the virtual environment monitored by PFM - RM for Virtual Machine.
   Execute the command as follows.
   ```
   jpcconf target unsetup -key RMVM -lhost jp1-halvm -inst inst1 -target targ
   ethost1
   ```
   When the `jpcconf target unsetup` command is executed, the virtual environment monitored by the logical host is no longer monitored.

3. Delete the instance environment of PFM - RM for Virtual Machine.
   Execute the command as follows.
   ```
   jpcconf inst unsetup -key RMVM -lhost jp1-halvm -inst inst1
   ```
   Executing the `jpcconf inst unsetup` command deletes the settings for starting the logical host instance. Additionally, the instance file is deleted from the shared disk.

4. Delete the logical host environment of PFM - RM for Virtual Machine by executing the `jpcconf ha unsetup` command.
   Execute the command as follows.
   ```
   jpcconf ha unsetup -key RMVM -lhost jp1-halvm
   ```
   Executing the `jpcconf ha unsetup` command deletes the settings for starting PFM - RM for Virtual Machine on the logical host. Also, the files for the logical host are deleted from the shared disk.

5. Use the `jpcconf ha list` command to confirm the logical host settings.
   Execute the command as follows.
   ```
   jpcconf ha list -key all
   ```
   Confirm that PFM - RM for Virtual Machine has been deleted from the logical host environment.

(5) Unsetting up the logical hosts of other Performance Management programs

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Optional

Besides PFM - RM for Virtual Machine, if there are other Performance Management programs that need to be unset up from the same logical host, unset them up at this time.

For details about the unsetup procedure, see the chapter that explains configuration and operations in a cluster system in the *JP1/Performance Management User's Guide*, or the chapter that explains operations in a cluster system in each PFM - RM manual.

(6) Exporting the logical host environment definition file

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system

After you have deleted PFM - RM for Virtual Machine from the logical host, export the environment definitions to a file.

Performance Management uses a method that matches the environment of the standby system to that of the executing system by exporting and importing the environment definitions. When the environment definitions (from which the Performance Management definition has been deleted) exported from the executing node are imported onto the standby node, they are compared with the standby node's existing environment definitions (which includes the Performance Management definition prior to its deletion). The difference (the part that was deleted from the executing node) is identified, and the environment definition for Performance Management is deleted.

To export the logical host environment definition file:

1. Export the logical host environment definitions by executing the `jpcconf ha export` command.

   Output the definition information of the logical host environment for Performance Management to a file. You can use any desired export file name. For example, to export the logical host environment definitions to the `lhostexp.txt` file, execute the command as follows.

   ```
   jpcconf ha export -f lhostexp.txt
   ```

(7) Copying the logical host environment definition file to the standby node VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Executing system Standby system

Copy the logical host environment definition file exported in *(6) Exporting the logical host environment definition file* from the executing node to the standby node.

(8) Taking the shared disk offline VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Executing system Optional

Using a cluster software operation or volume manager operation, take the shared disk offline to complete the task. If you plan to continue using the shared disk, there is no need to take it offline.

(9) Importing the logical host environment definition file VMware, Hyper-V, KVM logical partitioning feature Docker environment Podman environment Standby system

Import the export file copied from the executing node onto the standby node in order to commit the file to the standby node. At the standby node, there is no need to take the shared disk offline during the import operation.

To import the logical host environment definition file:

1. Import the logical host environment definitions by executing the `jpcconf ha import` command.

   Execute the command as follows.

   ```
   jpcconf ha import -f lhostexp.txt
   ```

   When the command is executed, the settings are changed to match the standby node environment to the content of the export file. As a result, the setting for starting the PFM - RM for Virtual Machine instance of the logical host is deleted. If the Performance Management programs of other logical hosts have been unset up, those settings are also deleted. If a fixed port number was set up using the `jpcconf port` command during setup, it is deleted as well.

2. Confirm the logical host settings by executing the `jpcconf ha list` command.

   Execute the command as follows.

   ```
   jpcconf ha list -key all
   ```

   Confirm that the command displays the same content that was displayed when the `jpcconf ha list` command was executed at the executing node.

(10) Deleting the PFM - RM registration from the cluster software

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

Delete the settings for the PFM - RM for Virtual Machine instance of the logical host from the cluster software. For details about how to delete the settings, see the cluster software documentation.

(11) Deleting the settings from PFM - Manager

VMware, Hyper-V, KVM | logical partitioning feature | Docker environment | Podman environment | Executing system | Standby system

From PFM - Web Console, log into PFM - Manager and delete the definition related to the PFM - RM for Virtual Machine instance that is being unset up.

To delete the settings:

1. If you have stopped the PFM - Manager services from the cluster software as described in *3.4.2(1) Stopping programs and services from the cluster software*, use the cluster software to start the PFM - Manager services. For details about how to start the services, see the cluster software documentation.

2. Delete the agent from PFM - Web Console.

3. Delete the agent information of PFM - Manager.
   For example, if PFM - Manager is running on the logical host `jp1-hal` and PFM - RM for Virtual Machine is running on the logical host `jp1-halvm`, specify the following and then execute the command:
   ```
   jpctool service delete -id service-ID -host jp1-halvm -lhost jp1-hal
   ```
   For *service-ID*, specify the service ID of the agent to be deleted.

4. Restart the PFM - Manager services.
   For details about how to start a service, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

5. Apply the service information of the PFM - Manager host.
   In order to update the PFM - Web Console host to reflect the deletion of service information, synchronize the agent information of the PFM - Manager host and that of the PFM - Web Console host. Use the `jpctool service sync` command to synchronize the agent information.

(12) Deleting the LPAR Manager management command for Hitachi Compute Blade logical partitioning feature

logical partitioning feature | Executing system | Standby system

Delete the `HvmSh` command (`HvmSh.exe`) of the management tool for Hitachi Compute Blade logical partitioning feature, which has been copied to the following folder during the setup:

```
installation-folder\agt8\plugin\jpcagt5virtage.d\
```

3.4.3 Uninstallation procedure

Uninstall PFM - RM for Virtual Machine from both the executing node and the standby node.

The uninstallation procedure is the same as that used for a non-cluster system. For the procedure, see *2.2.3 Uninstallation procedure*.

Notes:

- Before you uninstall PFM - RM for Virtual Machine, stop all Performance Management programs and services on the node from which you are uninstalling PFM - RM for Virtual Machine.
- If you uninstall PFM - RM for Virtual Machine without deleting the logical host environment, the environment folder may remain. In this case, delete the environment folder.

3.5 Changing the PFM - RM for Virtual Machine system configuration

In response to a change in the network configuration of the monitored system or its host name, you may need to modify the system configuration of PFM - RM for Virtual Machine.

If you modify the system configuration of PFM - RM for Virtual Machine, you must also modify the settings of PFM - Manager and PFM - Web Console. For details about how to modify the system configuration of Performance Management, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*. Note that when you are changing the logical host name, some PFM - RMs may require specific additional tasks. However, in the case of PFM - RM for Virtual Machine, no specific additional tasks are required.

3.6 Changing the PFM - RM for Virtual Machine operation method

This subsection explains how to change the operation of PFM - RM for Virtual Machine in a cluster system. For details about how to change the operation of Performance Management as a whole, see the chapter that explains cluster system configuration and operation in the *JP1/Performance Management User's Guide*.

3.6.1 Updating the instance environment settings

To update the instance environment in a cluster system, confirm the logical host name and the instance name, and then update the instance information. Set the instance information from the PFM - RM host on the executing node.

Check the information that needs to be updated by referring to *2.4.2 Updating an instance environment*.

To confirm the logical host name and the instance name, use the `jpcconf ha list` command. To update the instance environment, use the `jpcconf inst setup` command.

The procedure for updating the instance environment follows. To update multiple instance environments, repeat this procedure.

1. Confirm the logical host name and the instance name.
   Specify the service key indicating the PFM - RM for Virtual Machine instance that is running in the instance environment to be updated, and then execute the `jpcconf ha list` command.
   For example, to confirm the logical host name and the instance name of PFM - RM for Virtual Machine, specify the following and then execute the command.
   ```
   jpcconf ha list -key RMVM
   ```
   If the logical host name that has been set is `jp1_halvm` and the instance name is `inst1`, the following is displayed:
   ```
   Logical Host Name   Key    Environment Directory   Instance Name
   jp1_halvm           RMVM   logical-host-path       inst1
   ```
2. If there are any active PFM - RM for Virtual Machine services in the instance environment to be updated, stop these services from the cluster software.
3. If the shared disk goes offline in Step 2, bring it online using a cluster software operation or volume manager operation.
4. Specify both the service key and the instance name indicating the PFM - RM for Virtual Machine instance that is running in the instance environment to be updated, and then execute the `jpcconf inst setup` command.
   For example, to update an instance environment in which the logical host name of PFM - RM for Virtual Machine is `jp1_halvm` and the instance name is `inst1`, specify the following and then execute the command.
   ```
   jpcconf inst setup -key RMVM -lhost jp1_halvm -inst inst1
   ```
5. Update the instance information of PFM - RM for Virtual Machine.
   Enter the instance information of PFM - RM for Virtual Machine as instructed by the command. For details about the instance information of PFM - RM for Virtual Machine, see *2.4.2 Updating an instance environment*. The current settings are displayed (note that the password value will not be displayed). If you choose not to change the displayed value, simply press the **Enter** key. Once all information has been input, the instance environment is updated.
6. Restart the services of the updated instance environment from the cluster software.

For details about how to start a service, see the chapter that explains startup and termination of Performance Management in the *JP1/Performance Management User's Guide*.

Note:

If you wish to change the value of an item that cannot be updated, first delete the instance environment and then re-create one.

For details about the commands, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

3.6.2 Updating monitoring targets

If you want to update monitoring targets in a cluster system, first check the monitoring target names. Then, specify the new monitoring target settings on the PFM - RM host that is currently operating as the executing node.

Before you update monitoring targets, check what information will be updated. For the information that will be updated, see *2.4.3 Updating monitoring targets*.

To check the names of the monitoring targets, use the `jpcconf target list` command. To check the settings of the monitoring targets, use the `jpcconf target display` command. To update a monitoring target, use the `jpcconf target setup` command.

> **Note**
>
> You do not need to stop the PFM - RM for Virtual Machine service before updating the monitoring targets.

The following shows the procedure for updating a single monitoring target. Repeat the procedure if you want to update multiple monitoring targets. To update a monitoring target:

1. Check the names of the monitoring targets.
   Execute the `jpcconf target list` command by specifying a service key, logical host name, and instance name. This identifies the PFM - RM for Virtual Machine instance that is monitoring the monitoring targets that you want to update.
   For example, if you want to check the names of the hosts that are monitored by the PFM - RM for Virtual Machine instance whose logical host is `jp1_halvm` and instance name is `inst1`, execute the following command:
   ```
   jpcconf target list -key RMVM -lhost jp1_halvm -inst inst1
   ```
   The following shows an example of information displayed when the above command is executed.
   Output example:
   ```
   Targets:
   targethost1
   targethost2
   Groups:
   All
   ```
2. Check the settings of the monitoring target.
   Execute the `jpcconf target display` command by specifying the name of the monitoring-target host that you want to update, as well as the information that identifies the PFM - RM for Virtual Machine instance that is monitoring

the host. For the identification information, specify the service key, logical host name, and instance name of the PFM - RM for Virtual Machine instance.

For example, to display the settings of the host `targethost1` that is monitored by the instance `inst1` on the logical host `jp1_halvm`, execute the following command:

```
jpcconf target display -key RMVM -lhost jp1_halvm -inst inst1 -target targ
ethost1
```

3. If the shared disk is offline, place it online from the cluster software or volume manager.

4. Execute the `jpcconf target setup` command by specifying the name of the monitoring-target host that you want to update, as well as the information that identifies the PFM - RM for Virtual Machine instance that is monitoring the host. For the identification information, specify the service key, logical host name, and instance name of the PFM - RM for Virtual Machine instance.

   For example, to update the settings of host `targethost1` that is monitored by instance `inst1` on logical host `jp1_halvm`, execute the following command:

```
jpcconf target setup -key RMVM -lhost jp1_halvm -inst inst1 -target target
host1
```

   Although the above `jpcconf target setup` command is executed in interactive mode, it can also be executed in non-interactive mode.

   For how to execute the `jpcconf target setup` command in non-interactive mode, see *3.3.4(7) Setting up monitoring targets*.

   Note that if you execute the `jpcconf target setup` command in non-interactive mode, you can skip step 5.

5. Update the settings of the host monitored by PFM - RM for Virtual Machine.

   Enter necessary information in response to the command's prompt. For the information that can be entered, see *2.4.3 Updating monitoring targets*. The command sequentially displays the current values of settings (other than the password). For any value that you do not want to change, only press the **Enter** key. When all entries finish, the monitoring target is updated.

   Important

   There are settings whose values cannot be changed. To update the values of these settings, delete all monitoring-target information, and then re-create it with new values.

3.6.3 Exporting and importing the logical host environment definition file

You only need to export and import the logical host environment definition file if one of the following operations has been executed.

- The node configurations on the logical host were changed during setup of the logical host, instance environment, or monitoring target.

  For details about how to set up the logical host of PFM - RM for Virtual Machine, see the following subsection:

  - *3.3.4(4) Setting up the logical host of PFM - RM for Virtual Machine*

  For details about how to set up the instance environment, see the following subsection:

  - *3.3.4(6) Setting up an instance environment*

  For details about how to set up monitoring targets, see the following subsection:

    - *3.3.4(7) Setting up monitoring targets*
- An operation that requires the exportation of the logical host environment definition file was executed during setup of the logical host of other Performance Management programs.
  For details about how to set up the logical host of other Performance Management programs, see the following subsection:
    - *3.3.4(8) Setting up the logical hosts of other Performance Management programs*
- A port number was set up when the network was being set up.
  For details about how to set up the network, see the following subsection:
    - *3.3.4(10) Setting up the network*

For details about how to export and import the logical host environment definition file, see the following subsections:

- *3.3.4(14) Exporting the logical host environment definition file* through *3.3.4(17) Importing the logical host environment definition file*

If only the instance environment and monitoring target was updated, there is no need to export or import the logical host environment definition file.

For details about how to update the instance environment and monitoring target, see *3.6.1 Updating the instance environment settings* and *3.6.2 Updating monitoring targets*.

4 Monitoring Template

This chapter explains the monitoring templates of PFM - RM for Virtual Machine.

Overview of monitoring template

In Performance Management, you can define alarms and reports in the following ways:

- By using the alarms and reports defined in PFM - RM for Virtual Machine without modifying them
- By copying and customizing the alarms and reports defined in PFM - RM for Virtual Machine
- By using the wizard to define new alarms and reports

Alarms and reports provided by PFM - RM for Virtual Machine are called *monitoring templates*. Reports and alarms provided as monitoring templates already have the necessary information defined, and thus can be copied and used as is, or can be customized to suit the user environment. Therefore, without using the wizard to define new alarms and reports, you can easily prepare to monitor the operation status of the monitoring target.

This chapter explains the setting details of the alarms and reports that comprise the monitoring templates defined in PFM - RM for Virtual Machine.

For details on how to use monitoring templates, see the chapter that explains how to create reports for operation analysis or operation monitoring using alarms in the *JP1/Performance Management User's Guide*.

Format of alarm explanations

The following explains the alarm description format. Alarms are listed in alphabetical order.

Alarm name

Indicates the alarm name of a monitoring template.

Overview

Provides an overview of the target that can be monitored by this alarm.

Main settings

The main settings of this alarm are explained in a table. The table shows the correspondence between the alarm setting and the item that must be specified in the Properties window, which opens when you click first the alarm icon in the Alarms window of PFM - Web Console, and then the **Properties** method. To view detailed information about the settings for alarms, use the Properties window of that alarm in PFM - Web Console.

Two hyphens (--) in the *Setting* column signify that the setting is always invalid.

For *Conditional expression*, if the same setting is given for both an abnormal condition and a warning condition, an alarm event is issued only for the abnormal condition.

Action to be taken

Describes the action to take when this alarm is issued, or points to the location where the action to take is explained.

Alarm table

Indicates the alarm table in which this alarm's description is found.

Related report

Indicates the report in the monitoring template related to this alarm. You can display this report by clicking the agent icon from the Agents window of PFM - Web Console, and then clicking the  icon displayed under the **Display Alarm Status** method.

List of alarms

A table that lists and describes one or more alarms is called an *alarm table*. Alarms defined in a monitoring template of PFM - RM for Virtual Machine are stored in alarm table format in the `RM VirtualMachine` folder, which is displayed under the **Alarms** tab in PFM - Web Console.

An alarm table name is as expressed as follows.

- PFM RM VirtualMachine Template Alarms 12.50

12.50 at the end of the alarm table name

Indicates an alarm table version.

When you use an alarm that is defined in a monitoring template, check the version of the alarm table being used by the Performance Management system and its version compatibility. For details about the alarm table version and version compatibility, see *H. Version Compatibility*.

The table below shows the alarms defined in the monitoring templates of PFM - RM for Virtual Machine.

Table 4–1: Alarm list

Alarm name	Monitoring target	Monitoring purpose	Available virtual environment					
			VMware	Hyper-V	KVM	Docker environment	Podman environment	logical partitioning feature
Host Disk Usage	Logical disk usage in a physical server	Operation status monitoring	Y	Y	Y	N	N	N
Host Memory Usage	Memory usage in a physical server	Performance information monitoring	Y	Y	Y	N	N	Y
VM CPU Insufficient	CPU insufficiency in a virtual machine	Performance information monitoring	Y	N	N	N	N	Y
VM Disk Abort Cmds	Number of discarded disk commands in a virtual machine	Operation status monitoring	Y	N	N	N	N	N
VM Disk Usage	Logical disk usage in a virtual machine	Operation status monitoring	Y	N	N	N	N	N
VM Status	Virtual machine status	Operation status monitoring	Y	Y	Y	Y	Y	Y

Legend:

Y: Available

N: Not available

PFM - RM for Virtual Machine provides as monitoring templates an operating information monitoring alarm that monitors whether the system is operating normally, and a performance information monitoring alarm that monitors whether the system is providing sufficient services. Therefore, set the alarms that match your objectives.

If you use the monitoring templates, copy the provided monitoring template and customize them as appropriate for monitored environment.

Host Disk Usage

Overview

The Host Disk Usage alarm monitors the following item:

- For VMware, Hyper-V and KVM
  Logical disk usage (%) on the physical server
- For logical partitioning feature, Docker environment and Podman environment
  This alarm is not available because the Host Logical Disk Status (`PI_HLDI`) record is not supported.

To monitor a specific logical disk, copy this alarm and create an alarm in which `"*"` is replaced with the logical disk ID to be monitored.

Main settings

PFM - Web Console alarm properties		Setting
Item	Detailed item	
Main Information	**Product**	`RM VirtualMachine(9.0)`
	Alarm message	`Host logical disk(%CVS1) is at %CVS2% utilization`
	Enable alarm	Selected
	Alarm notification	Notify when the state changed
	Notification target	State changes for the alarm
	Evaluate all data	Not selected
	Monitoring time range	**Always monitor**
	Report alarm when the following damping condition is reached	Selected
	occurrence(s) during	`2`
	Interval(s)	`3`
Alarm Conditions	**Record**	`Host Logical Disk Status (PI_HLDI)`
	Field	• `Disk ID` • `Used %`
	Abnormal condition	`Disk ID = "*" AND` `Used % >= 90`
	Warning condition	`Disk ID = "*" AND` `Used % >= 80`
Action	**Email**	--
	Command	--
	SNMP	**Abnormal**, **Warning**, **Normal**

Action to be taken

- For VMware

See *1.4.5(2)(b) Example of monitoring the space usage of a physical server's logical disk*.

- For Hyper-V
  See *1.5.5(2)(a) Example of monitoring the space usage of a physical server's logical disk*.
- For KVM
  See *1.6.5(2)(a) Example of monitoring the space usage of a physical server's logical disk*.

Alarm table

PFM RM VirtualMachine Template Alarms 12.50

Related report

`Reports/RM VirtualMachine/Status Reporting/Real-Time/Host Disk Used(9.0)`

Host Memory Usage

Overview

The Host Memory Usage alarm monitors the following item:

- For VMware, Hyper-V and KVM
  Memory usage (%) on the physical server
- For logical partitioning feature
  Percentage of memory resources allocated to the host machine (%)
- For Docker environment and Podman environment
  This alarm is not available because the Host Memory Status (PI_HMI) record is not supported.

Main settings

PFM - Web Console alarm properties		Setting
Item	Detailed item	
Main Information	**Product**	`RM VirtualMachine(9.0)`
	Alarm message	`Host memory is at %CVS% utilization`
	Enable alarm	Selected
	Alarm notification	Notify when the state changed
	Notification target	State changes for the alarm
	Evaluate all data	Not selected
	Monitoring time range	**Always monitor**
	Report alarm when the following damping condition is reached	Selected
	occurrence(s) during	`2`
	Interval(s)	`3`
Alarm Conditions	**Record**	`Host Memory Status (PI_HMI)`
	Field	`Total Used %`
	Abnormal condition	`Total Used % >= 120`
	Warning condition	`Total Used % >= 100`
Actions	**Email**	--
	Command	--
	SNMP	**Abnormal**, **Warning**, **Normal**

Action to be taken

- For VMware
  See *1.4.4(2)(a) Example of monitoring the total memory usage of a physical server*.
- For Hyper-V
  See *1.5.4(2)(a) Example of monitoring the total memory usage of a physical server*.

- For logical partitioning feature
  The use of this alarm is not required because the Total Used % field indicates the memory resource allocation rate and the change in its value does not affect performance.
- For KVM
  See *1.6.4(2)(a) Example of monitoring the total memory usage of a physical server*.

Alarm table

PFM RM VirtualMachine Template Alarms 12.50

Related report

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/Host Memory Used`

VM CPU Insufficient

Overview

The VM CPU Insufficient alarm monitors the following item:

- For VMware and logical partitioning feature
  CPU insufficiency (%) on a virtual machine
- For Hyper-V, KVM, Docker environment and Podman environment
  This alarm is not available because the Insufficient % field is not supported.

This value approaches 0% if sufficient CPU resources are allocated to the virtual machine. To monitor a specific virtual machine, copy this alarm and create an alarm in which "*" is replaced with the name of the virtual machine to be monitored.

Main settings

PFM - Web Console alarm properties		Setting
Item	Detailed item	
Main Information	**Product**	`RM VirtualMachine(9.0)`
	Alarm message	`CPU insufficiency rate of %CVS1 is %CVS2%`
	Enable alarm	Selected
	Alarm notification	Notify when the state changed
	Notification target	State changes for the alarm
	Evaluate all data	Not selected
	Monitoring time range	**Always monitor**
	Report alarm when the following damping condition is reached	Selected
	occurrence(s) during	`2`
	Interval(s)	`3`
Alarm Conditions	**Record**	`VM Status (PI_VI)`
	Field	• `VM Name` • `Insufficient %`
	Abnormal condition	`VM Name = "*" AND` `Insufficient % >= 30`
	Warning condition	`VM Name = "*" AND` `Insufficient % >= 10`
Actions	**Email**	--
	Command	--
	SNMP	**Abnormal**, **Warning**, **Normal**

Action to be taken

- For VMware

See *1.4.3(2)(a) Example of monitoring CPU insufficiency in a virtual machine*.

- For logical partitioning feature
  See *1.9.3(2)(a) Example of monitoring CPU insufficiency in a LPAR*.

Alarm table

PFM RM VirtualMachine Template Alarms 12.50

Related report

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/VM CPU Insufficient (6.0)`

VM Disk Abort Cmds

Overview

The VM Disk Abort Cmds alarm monitors the following item:

- For VMware
  Number of discarded disk commands on a virtual machine
- For Hyper-V, KVM, logical partitioning feature, Docker environment and Podman environment
  This alarm is not available because the Abort Commands is not supported.

To monitor a specific virtual machine, copy this alarm and create an alarm in which "*" is replaced with the name of the virtual machine to be monitored.

Main settings

PFM - Web Console alarm properties		Setting
Item	Detailed item	
Main Information	**Product**	`RM VirtualMachine(9.0)`
	Alarm message	`Disk abort commands of %CVS1 is %CVS2`
	Enable alarm	Selected
	Alarm notification	Notify when the state changed
	Notification target	State changes for the alarm
	Evaluate all data	Not selected
	Monitoring time range	**Always monitor**
	Report alarm when the following damping condition is reached	Selected
	occurrence(s) during	`2`
	Interval(s)	`3`
Alarm Conditions	**Record**	`VM Physical Disk Status (PI_VPDI)`
	Field	• `VM Name` • `Abort Commands`
	Abnormal condition	`VM Name = "*" AND` `Abort Commands >= 10`
	Warning condition	`VM Name = "*" AND` `Abort Commands >= 1`
Actions	**Email**	--
	Command	--
	SNMP	**Abnormal**, **Warning**, **Normal**

Action to be taken

- For VMware

See *1.4.5(2)(a) Example of monitoring the disk command discarding rate of a physical disk being used by a virtual machine*.

Alarm table

PFM RM VirtualMachine Template Alarms 12.50

Related report

`Reports/RM VirtualMachine/Monthly Trend/VM Disk Abort Commands (6.0)`

VM Disk Usage

Overview

The VM Disk Usage alarm monitors the following item:

- For VMware
  Logical disk usage (%) on a virtual machine
- For Hyper-V, KVM, logical partitioning feature, Docker environment and Podman environment
  This alarm is not available because the VM Logical Disk Status (`PI_VLDI`) record is not supported.

To monitor a specific logical disk or virtual machine, copy this alarm and create an alarm in which "*" is replaced with the logical disk ID or the name of the virtual machine to be monitored.

Main settings

PFM - Web Console alarm properties		Setting
Item	Detailed item	
Main Information	**Product**	`RM VirtualMachine(9.0)`
	Alarm message	`Logical disk of %CVS1 is at %CVS3% utilization`
	Enable alarm	Selected
	Alarm notification	Notify when the state changed
	Notification target	State changes for the alarm
	Evaluate all data	Not selected
	Monitoring time range	**Always monitor**
	Report alarm when the following damping condition is reached	Selected
	occurrence(s) during	2
	Interval(s)	3
Alarm Conditions	**Record**	`VM Logical Disk Status (PI_VLDI)`
	Field	• `VM Name` • `Disk ID` • `Used %`

	Abnormal condition	`VM Name = "*" AND`
		`Disk ID = "*" AND`
		`Used % >= 90`

	Warning condition	`VM Name = "*" AND`
		`Disk ID = "*" AND`
		`Used % >= 80`

Actions	**Email**	--
	Command	--
	SNMP	**Abnormal**, **Warning**, **Normal**

Action to be taken

- For VMware
  See *1.4.5(2)(c) Example of monitoring the space usage of a virtual machine's logical disk*.

Alarm table

PFM RM VirtualMachine Template Alarms 12.50

Related report

`Reports/RM VirtualMachine/Status Reporting/Real-Time/VM Disk Used (9.0)`

VM Status

Overview

The VM Status alarm monitors the following item:

- For VMware, Hyper-V, KVM, logical partitioning feature, Docker environment and Podman environment
  Virtual machine status

To monitor a specific virtual machine, copy this alarm and create an alarm in which "*" is replaced with the name of the virtual machine to be monitored.

Main settings

PFM - Web Console alarm properties		Setting
Item	Detailed item	
Main Information	**Product**	`RM VirtualMachine(9.0)`
	Alarm message	`Virtual machine(%CVS1) not available`
	Enable alarm	Selected
	Alarm notification	Notify when the state changed
	Notification target	State changes for the alarm
	Evaluate all data	Not selected
	Monitoring time range	**Always monitor**
	Report alarm when the following damping condition is reached	Selected
	occurrence(s) during	`2`
	Interval(s)	`3`
Alarm Conditions	**Record**	`VM Status Detail (PD_VM)`
	Field	• `VM Name` • `Status`
	Abnormal condition	`VM Name = "*" AND` `Status <> "ON"`
	Warning condition	`VM Name = "*" AND` `Status <> "ON"`
Actions	**Email**	--
	Command	--
	SNMP	**Abnormal, Warning, Normal**

Action to be taken

- For VMware, Hyper-V, KVM, logical partitioning feature, Docker environment and Podman environment
  When this alarm is issued, check if the virtual machine is running.

Alarm table

PFM RM VirtualMachine Template Alarms 12.50

Related report

None

Format of report explanations

This section explains the report description format. Reports are presented in alphabetical order.

Report name

Indicates the report name of a monitoring template.

If a report name includes `(Multi-Agent)`, that report displays information about multiple instances.

If a report name does not include `(Multi-Agent)`, that report displays information about a single instance.

For details about data models, see *5. Records*.

Overview

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

Storage location

Indicates the location at which the report is stored.

Record

Indicates the record in which the performance data used by the report is stored. To display a historical report, you must specify collection of the record indicated in this section in advance. Before displaying the report, display the properties of the agent in the Agents window of PFM - Web Console, and make sure that `Log = Yes` is set for this record. This setting is not required for a real-time report.

Fields

The record fields used in the report are presented in a table.

Drilldown report (field level)

The reports in the monitoring template that are associated with the fields used in this report are presented in a table. To display this drilldown report, click the report graph displayed in the Report window of PFM - Web Console or the field name displayed at the bottom of the Report window. For a historical report, by clicking a time displayed in blue inside the report, you can display a report in more detailed time intervals. Note that some reports have a drilldown report while others do not.

For details about drilldown reports, see the chapter that explains creation of reports for operational analysis in the *JP1/Performance Management User's Guide*.

Organization of report folders

The folder structure of a report in PFM - RM for Virtual Machine is shown below. The item inside < > indicates a folder name.

```
<RM VirtualMachine>
 +-- <Monthly Trend>
 |     +-- <Host CPU Used Status (9.0)>
 |     +-- <Host Disk Used (9.0)>
 |     +-- <Host Memory Used>
 |     +-- <Host Network Data>
 |     +-- <VM CPU Insufficient (9.0)>
 |     +-- <VM CPU Trend (9.0)>
 |     +-- <VM Disk Abort Commands (6.0)>
 |     +-- <VM Disk Used (9.0)>
 |     +-- <VM Memory Trend>
 |     +-- <VM Network Data (6.0)>
 +-- <Status Reporting>
 |     +-- <Daily Trend>
 |     |     +-- <Host CPU Used Status (9.0)>
 |     |     +-- <Host Memory Used>
 |     |     +-- <VM CPU Insufficient (9.0)>
 |     +-- <Real-Time>
 |           +-- <Host Disk Used> (9.0)
 |           +-- <VM Disk Abort Commands (6.0)>
 |           +-- <VM Disk Used (9.0)>
 |           +-- <VM Virtual Disk Allocation Value (8.0)>
 +-- <Troubleshooting>
       +-- <Real-Time>
       |     +-- <Host CPU Used Status (9.0)>
       |     +-- <Host Disk I/O>
       |     +-- <Host Disk Used Status (9.0)>
       |     +-- <Host Memory Size>
       |     +-- <Host Memory Used>
       |     +-- <Host Memory Used Status>
       |     +-- <Host Network Data>
       |     +-- <VM CPU Allocation Value (9.0)>
       |     +-- <VM CPU Insufficient (9.0)>
       |     +-- <VM CPU Used (9.0)>
       |     +-- <VM Disk I/O (6.0)>
       |     +-- <VM Disk Used Status (9.0)>
       |     +-- <VM Memory Allocation Value (6.0)>
       |     +-- <VM Network Data (6.0)>
       |     +-- <VM Swap Used (6.0)>
       |     +-- <VM Working Size - Total (6.0)>
       +-- <Recent Past>
             +-- <Host CPU Used Status (9.0)>
             +-- <Host Disk I/O>
             +-- <Host Memory Size>
             +-- <Host Memory Used>
             +-- <Host Memory Used Status (6.0)>
             +-- <VM CPU Allocation Value (9.0)>
             +-- <VM CPU Insufficient (9.0)>
             +-- <VM CPU Used (9.0)>
             +-- <VM Disk I/O (6.0)>
             +-- <VM Memory Allocation Value (6.0)>
```

```
            +-- <VM Swap Used (6.0)>
            +-- <VM Working Size - Total (6.0)>
            +-- <Drilldown Only>
                 +-- <VM CPU Used Status (9.0)>
                 +-- <VM Memory Used (6.0)>
                 +-- <VM Memory Used Status (6.0)>
```

The following describes each folder.

- `Monthly Trend` folder

  This folder contains reports that display daily information for the past month. Use the reports in this folder to check monthly trends in the system.

- `Status Reporting` folder

  This folder contains reports that display daily information. Use the reports in this folder to check the overall status of the system. You can display real-time reports as well as historical reports.

  - `Daily Trend` folder

    This folder contains historical reports for displaying hourly information for the past 24 hours. Use the reports in this folder to check the daily status of the system.

  - `Real-Time` folder

    This folder contains real-time reports for checking the system status.

- `Troubleshooting` folder

  This folder contains reports for displaying information that is useful for resolving problems. In the event of a system problem, use the reports in this folder to check the cause of the problem.

  - `Real-Time` folder

    This folder contains real-time reports for checking the current system status.

  - `Recent Past` folder

    This folder contains historical reports for displaying minute-by-minute information for the past hour.

Additionally, these folders have the following folders under them. The folders that are located below them depend on the higher-order folder. These folders are explained below.

- `Drilldown Only` folder

  This folder stores the report that is displayed as a drilldown report (field level). Use this folder to display the detailed information related to that report field.

List of reports

The table below alphabetically lists the reports defined as monitoring templates.

Table 4–2: Report list

Category	Report name	Information to be displayed
CPU	*Host CPU Used Status (9.0)*	Displays the CPU usage status of a physical server.
	VM CPU Allocation Value (9.0)	Displays the CPU allocation upper limit of a virtual machine.
	VM CPU Insufficient (9.0)	Displays the CPU insufficiency of a virtual machine.
	VM CPU Trend (9.0)	Displays the virtual machine CPU usage.
	VM CPU Used (9.0)	Displays the CPU usage of a virtual machine.
	VM CPU Used Status (9.0)	Displays the CPU usage status of a physical server by a virtual machine.
Disk	*Host Disk I/O*	Displays the physical disk I/O of a physical server.
	Host Disk Used (9.0)	Displays the logical disk usage of a physical server.
	Host Disk Used Status (9.0)	Displays the logical disk usage status of a physical server.
	VM Disk Abort Commands (6.0)	Displays the disk command-discarding rate of a virtual machine.
	VM Disk I/O (6.0)	Displays the disk I/O of a virtual machine.
	VM Disk Used (9.0)	Displays the logical disk usage of a virtual machine.
	VM Disk Used Status (9.0)	Displays the logical disk usage status of a virtual machine.
	VM Virtual Disk Allocation Value (8.0)	Displays the amount of allocated virtual disk space.
Memory	*Host Memory Size*	Displays the total physical memory size of a physical server.
	Host Memory Used	Displays the memory resource usage of a physical server.
	Host Memory Used Status	Displays the memory resource usage status of a physical server.
	Host Memory Used Status (6.0)	Displays the memory resource usage status of a physical server.
	VM Memory Allocation Value (6.0)	Displays the memory allocation upper limit of a virtual machine.
	VM Memory Trend	Displays the memory usage of a virtual machine.
	VM Memory Used (6.0)	Displays the memory resources used by a virtual machine.
	VM Memory Used Status (6.0)	Displays the memory resource usage status of a virtual machine.
	VM Swap Used (6.0)	Displays the swap usage of a virtual machine.
	VM Working Size - Total (6.0)	Displays the working set size of all virtual machines.
Network	*Host Network Data*	Displays the volume of network data sent/received by a physical server.
	VM Network Data (6.0)	Displays the volume of network data sent/received by a virtual machine.

Note:

Some records and fields may not be supported, depending on the virtual environment being monitored. This means that a report that consists of such records or fields might not be available. For details about the support status of records and fields, see the section that explains each record in *5. Records*.

Host CPU Used Status (9.0) (Monthly Trend)

Overview

The `Host CPU Used Status (9.0)` report displays the CPU usage status of a physical server over the past month in daily summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Monthly Trend/`

Record

Host Status (`PI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Clocks	CPU resource clock frequency (MHz)
Count	Physical CPU core count
VM Used	CPU resources used by VM (MHz)
VMM Kernel Used	• For VMware and KVM CPU resources used by VMM kernel (MHz) • For logical partitioning feature CPU resources used by SYS1 (MHz)
VMM Console Used	CPU resources used by VMM console (MHz)
VMM Others Used	• For VMware and KVM CPU resources used by VMM and others (MHz) • For logical partitioning feature CPU resources used by SYS2 (MHz)
Unused	CPU resources unused (MHz)

Host CPU Used Status (9.0) (Status Reporting/Daily Trend)

Overview

The `Host CPU Used Status (9.0)` report displays the CPU usage status of a physical server over the past day in hourly summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Status Reporting/Daily Trend/`

Record

Host Status (`PI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Clocks	CPU resource clock frequency (MHz)
Count	Physical CPU core count
VM Used	CPU resources used by VM (MHz)
VMM Kernel Used	• For VMware and KVM CPU resources used by VMM kernel (MHz) • For logical partitioning feature CPU resources used by SYS1 (MHz)
VMM Console Used	CPU resources used by VMM console (MHz)
VMM Others Used	• For VMware and KVM CPU resources used by VMM and others (MHz) • For logical partitioning feature CPU resources used by SYS2 (MHz)
Unused	CPU resources unused (MHz)

Host CPU Used Status (9.0) (Troubleshooting/Real-Time)

Overview

The `Host CPU Used Status (9.0)` report displays the CPU usage status of a physical server on a real-time basis, in list and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

Host Status (`PI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Clocks	CPU resource clock frequency (MHz)
Count	Physical CPU core count
VM Used	CPU resources used by VM (MHz)
VMM Kernel Used	• For VMware and KVM CPU resources used by VMM kernel (MHz) • For logical partitioning feature CPU resources used by SYS1 (MHz)
VMM Console Used	CPU resources used by VMM console (MHz)
VMM Others Used	• For VMware and KVM CPU resources used by VMM and others (MHz) • For logical partitioning feature CPU resources used by SYS2 (MHz)
Unused	CPU resources unused (MHz)

Host CPU Used Status (9.0) (Troubleshooting/Recent Past)

Overview

The `Host CPU Used Status (9.0)` report displays the CPU usage status of a physical server over the past hour in minute-by-minute summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

Host Status (`PI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Clocks	CPU resource clock frequency (MHz)
Count	Physical CPU core count
VM Used	CPU resources used by VM (MHz)
VMM Kernel Used	• For VMware and KVM CPU resources used by VMM kernel (MHz) • For logical partitioning feature CPU resources used by SYS1 (MHz)
VMM Console Used	CPU resources used by VMM console (MHz)
VMM Others Used	• For VMware and KVM CPU resources used by VMM and others (MHz) • For logical partitioning feature CPU resources used by SYS2 (MHz)
Unused	CPU resources unused (MHz)

Drilldown report (field level)

Report name	Explanation
`VM CPU Used Status (9.0)`	Displays the CPU usage status of a physical server by the selected virtual machine. To display this report, click the VM Used field.

Host Disk I/O (Troubleshooting/Real-Time)

Overview

The `Host Disk I/O` report displays the physical disk I/O of a physical server on a real-time basis, in list and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

Host Physical Disk Status (`PI_HPDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Disk ID	Disk ID
Speed	Data transfer speed (KB/sec)
Requests	• For VMware, Hyper-V and KVM Processing count • For logical partitioning feature Number of interrupts from the HBA
Read Requests	Read request count
Read Speed	Read data transfer speed (KB/sec)
Write Requests	Write request count
Write Speed	Write data transfer speed (KB/sec)

Host Disk I/O (Troubleshooting/Recent Past)

Overview

The `Host Disk I/O` report displays the physical disk I/O of a physical server over the past hour in minute-by-minute summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

Host Physical Disk Status (`PI_HPDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Disk ID	Disk ID
Speed	Data transfer speed (KB/sec)
Requests	• For VMware, Hyper-V and KVM Processing count • For logical partitioning feature Number of interrupts from the HBA
Read Requests	Read request count
Read Speed	Read data transfer speed (KB/sec)
Write Requests	Write request count
Write Speed	Write data transfer speed (KB/sec)

Host Disk Used (9.0) (Monthly Trend)

Overview

The `Host Disk Used (9.0)` report displays the logical disk usage of a physical server over the past month in daily summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Monthly Trend/`

Record

Host Logical Disk Status (`PI_HLDI`) (`7.0`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Disk ID	Disk ID
Used	Size of disk space used (MB)
Used %	Disk usage (%)
Free	Size of disk space unused (MB)
Size	Disk size (MB)
Last Update	The time when the monitored host updated the values for disk space.

Host Disk Used (9.0) (Status Reporting/Real-Time)

Overview

The `Host Disk Used (9.0)` report displays the logical disk usage of a physical server on a real-time basis, in list and line graph formats.

Storage location

`Reports/RM VirtualMachine/Status Reporting/Real-Time/`

Record

Host Logical Disk Status (`PI_HLDI`) (`7.0`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Disk ID	Disk ID
Used	Size of disk space used (MB)
Used %	Disk usage (%)
Free	Size of disk space unused (MB)
Size	Disk size (MB)
Last Update	The time when the monitored host updated the values for disk space.

Host Disk Used Status (9.0) (Troubleshooting/Real-Time)

Overview

The `Host Disk Used Status (9.0)` report displays the logical disk usage status of a physical server on a real-time basis, in list and stacked column graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

Host Logical Disk Status (`PI_HLDI`) (`7.0`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Disk ID	Disk ID
Used	Size of disk space used (MB)
Used %	Disk usage (%)
Free	Size of disk space unused (MB)
Size	Disk size (MB)
Last Update	The time when the monitored host updated the values for disk space.

Host Memory Size (Troubleshooting/Real-Time)

Overview

The Host Memory Size report displays the total size of physical memory in a physical server on a real-time basis, in list and line graph formats.

Storage location

Reports/RM VirtualMachine/Troubleshooting/Real-Time/

Record

Host Memory Status (PI_HMI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Size	Memory resource size (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
VMM Used %	• For VMware and KVM Memory resources used by VMM (%) • For logical partitioning feature Percentage of memory resources allocated to the hypervisor (%)
VM Used %	• For VMware and KVM Memory resources used by VM (%) • For logical partitioning feature Percentage of memory resources allocated to the VM (%)

Host Memory Size (Troubleshooting/Recent Past)

Overview

The `Host Memory Size` report displays the total size of physical memory of a physical server over the past hour in minute-by-minute summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

Host Memory Status (`PI_HMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Size	Memory resource size (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
VMM Used %	• For VMware and KVM Memory resources used by VMM (%) • For logical partitioning feature Percentage of memory resources allocated to the hypervisor (%)
VM Used %	• For VMware and KVM Memory resources used by VM (%) • For logical partitioning feature Percentage of memory resources allocated to the VM (%)

Host Memory Used (Monthly Trend)

Overview

The Host Memory Used report displays the memory resource usage of a physical server over the past month in daily summaries, in table and stacked area graph formats.

Storage location

Reports/RM VirtualMachine/Monthly Trend/

Record

Host Memory Status (PI_HMI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Used	• For VMware, Hyper-V and KVM Memory resource usage (MB) • For logical partitioning feature Amount of memory resources allocated to the host machine (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
Unused	Memory resources unused (MB)

Host Memory Used (Status Reporting/Daily Trend)

Overview

The `Host Memory Used` report displays the memory resource usage of a physical server over the past day in hourly summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Status Reporting/Daily Trend/`

Record

Host Memory Status (`PI_HMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Used	• For VMware, Hyper-V and KVM Memory resource usage (MB) • For logical partitioning feature Amount of memory resources allocated to the host machine (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
Unused	Memory resources unused (MB)

Host Memory Used (Troubleshooting/Real-Time)

Overview

The `Host Memory Used` report displays the memory resource usage of a physical server on a real-time basis, in list and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

Host Memory Status (`PI_HMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Used	• For VMware, Hyper-V and KVM Memory resource usage (MB) • For logical partitioning feature Amount of memory resources allocated to the host machine (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
Unused	Memory resources unused (MB)

Host Memory Used (Troubleshooting/Recent Past)

Overview

The `Host Memory Used` report displays the memory resource usage of a physical server over the past hour in minute-by-minute summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

Host Memory Status (`PI_HMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Used	• For VMware, Hyper-V and KVM Memory resource usage (MB) • For logical partitioning feature Amount of memory resources allocated to the host machine (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
Unused	Memory resources unused (MB)

Host Memory Used Status (Troubleshooting/Real-Time)

Overview

The `Host Memory Used Status` report displays the memory resource usage status of a physical server on a real-time basis, in list and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

Host Memory Status (`PI_HMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Used	• For VMware, Hyper-V and KVM Memory resource usage (MB) • For logical partitioning feature Amount of memory resources allocated to the host machine (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Unused	Memory resources unused (MB)

Host Memory Used Status (6.0) (Troubleshooting/Recent Past)

Overview

The `Host Memory Used Status (6.0)` report displays the memory resource usage status of a physical server over the past hour in minute-by-minute summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

Host Memory Status (`PI_HMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Used	• For VMware, Hyper-V and KVM Memory resource usage (MB) • For logical partitioning feature Amount of memory resources allocated to the host machine (MB)
Used %	• For VMware, Hyper-V and KVM Memory resource usage (%) • For logical partitioning feature Percentage of memory resources allocated to the host machine (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Unused	Memory resources unused (MB)

Drilldown report (field level)

Report name	Explanation
`VM Memory Used`	Displays the memory resource usage by the selected virtual machine. To display this report, click the Used field.

Host Network Data (Monthly Trend)

Overview

The Host Network Data report displays the volume of network data sent/received by a physical server over the past month in daily summaries, in table and line graph formats.

Storage location

Reports/RM VirtualMachine/Monthly Trend/

Record

Host Network Status (PI_HNI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Net ID	Network ID
Rate	Speed at which the physical server sends data to or receives data from the network (KB/sec)
Recv Rate	Speed at which the physical server receives data from the network (KB/sec)
Send Rate	Speed at which the physical server sends data to the network (KB/sec)

Host Network Data (Troubleshooting/Real-Time)

Overview

The Host Network Data report displays the volume of network data sent/received by a physical server on a real-time basis, in list and line graph formats.

Storage location

Reports/RM VirtualMachine/Troubleshooting/Real-Time/

Record

Host Network Status (PI_HNI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
Net ID	Network ID
Rate	Speed at which the physical server sends data to or receives data from the network (KB/sec)
Recv Rate	Speed at which the physical server receives data from the network (KB/sec)
Send Rate	Speed at which the physical server sends data to the network (KB/sec)

VM CPU Allocation Value (9.0) (Troubleshooting/Real-Time)

Overview

The `VM CPU Allocation Value (9.0)` report displays the CPU allocation upper limit of a virtual machine on a real-time basis, in list and summary bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Expectation	CPU allocation balancing value (MHz)
Expectation %	CPU allocation balancing point (%)
Max	CPU allocation upper limit value (MHz)
Max %	CPU allocation upper limit rate (%)
Min	CPU allocation lower limit value (MHz)
Min %	CPU allocation lower limit rate (%)
Share	CPU allocation share

VM CPU Allocation Value (9.0) (Troubleshooting/Recent Past)

Overview

The `VM CPU Allocation Value (9.0)` report displays the CPU allocation upper limit of a virtual machine over the past hour in minute-by-minute summaries, in table and summary bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Expectation	CPU allocation balancing value (MHz)
Expectation %	CPU allocation balancing point (%)
Max	CPU allocation upper limit value (MHz)
Max %	CPU allocation upper limit rate (%)
Min	CPU allocation lower limit value (MHz)
Min %	CPU allocation lower limit rate (%)
Share	CPU allocation share

VM CPU Insufficient (9.0) (Monthly Trend)

Overview

The `VM CPU Insufficient (9.0)` report displays the CPU insufficiency of a virtual machine over the past month in daily summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Monthly Trend/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Request %	CPU request rate (%)
Used Per Request	CPU allocated rate (%)
Insufficient Per Request	CPU unallocated rate (%)

VM CPU Insufficient (9.0) (Status Reporting/Daily Trend)

Overview

The `VM CPU Insufficient (9.0)` report displays the CPU insufficiency of a virtual machine over the past day in hourly summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Status Reporting/Daily Trend/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Request %	CPU request rate (%)
Used Per Request	CPU allocated rate (%)
Insufficient Per Request	CPU unallocated rate (%)

VM CPU Insufficient (9.0) (Troubleshooting/Real-Time)

Overview

The `VM CPU Insufficient (9.0)` report displays the CPU insufficiency of a virtual machine on a real-time basis, in list and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Request %	CPU request rate (%)
Used Per Request	CPU allocated rate (%)
Insufficient Per Request	CPU unallocated rate (%)

VM CPU Insufficient (9.0) (Troubleshooting/Recent Past)

Overview

The `VM CPU Insufficient (9.0)` report displays the CPU insufficiency of a virtual machine over the past hour in minute-by-minute summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Request %	CPU request rate (%)
Used Per Request	CPU allocated rate (%)
Insufficient Per Request	CPU unallocated rate (%)

VM CPU Trend (9.0) (Monthly Trend)

Overview

The VM CPU Trend (9.0) report displays the CPU usage by a virtual machine over the past month in daily summaries, in table and line graph formats.

Storage location

Reports/RM VirtualMachine/Monthly Trend/

Record

VM Status (PI_VI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Request %	CPU request rate (%)
Used Per Request	CPU allocated rate (%)
Insufficient Per Request	CPU unallocated rate (%)

VM CPU Used (9.0) (Troubleshooting/Real-Time)

Overview

The `VM CPU Used (9.0)` report displays the CPU usage of a virtual machine on a real-time basis, in list and summary bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Expectation	CPU allocation balancing value (MHz)
Max	CPU allocation upper limit value (MHz)
Min	CPU allocation lower limit value (MHz)

VM CPU Used (9.0) (Troubleshooting/Recent Past)

Overview

The `VM CPU Used (9.0)` report displays the CPU usage of a virtual machine over the past hour in minute-by-minute summaries, in table and summary bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Used	CPU usage (MHz)
Used %	CPU usage rate (%)
Insufficient	CPU insufficiency (MHz)
Insufficient %	CPU insufficiency rate (%)
Expectation	CPU allocation balancing value (MHz)
Max	CPU allocation upper limit value (MHz)
Min	CPU allocation lower limit value (MHz)

VM CPU Used Status (9.0) (Troubleshooting/Recent Past/Drilldown Only)

Overview

The `VM CPU Used Status (9.0)` report displays the CPU usage status of a physical server by a virtual machine over the past hour in minute-by-minute summaries, in table and stacked column graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/Drilldown Only/`

Record

VM Status (`PI_VI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Used	CPU usage (MHz)
Used %	CPU usage rate (%)

VM Disk Abort Commands (6.0) (Monthly Trend)

Overview

The `VM Disk Abort Commands (6.0)` report displays the disk-command discarding rate of a virtual machine over the past month in daily summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Monthly Trend/`

Record

VM Physical Disk Status (`PI_VPDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Abort Commands	Number of disk commands discarded
Abort Commands %	Disk-command discarding rate (%)
Commands	Number of disk commands issued

VM Disk Abort Commands (6.0) (Status Reporting/Real-Time)

Overview

The VM Disk Abort Commands (6.0) report displays the disk-command discarding rate of a virtual machine on a real-time basis, in list and line graph formats.

Storage location

Reports/RM VirtualMachine/Status Reporting/Real-Time/

Record

VM Physical Disk Status (PI_VPDI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Abort Commands	Number of disk commands discarded
Abort Commands %	Disk-command discarding rate (%)
Commands	Number of disk commands issued

VM Disk I/O (6.0) (Troubleshooting/Real-Time)

Overview

The `VM Disk I/O (6.0)` report displays the disk I/O of a virtual machine on a real-time basis, in list and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Physical Disk Status (`PI_VPDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Speed	Data transfer speed (KB/sec)
Requests	Request count
Read Requests	Read request count
Read Speed	Read data transfer speed (KB/sec)
Write Requests	Write request count
Write Speed	Write data transfer speed (KB/sec)

VM Disk I/O (6.0) (Troubleshooting/Recent Past)

Overview

The `VM Disk I/O (6.0)` report displays the disk I/O operations of a virtual machine over the past hour in minute-by-minute summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Physical Disk Status (`PI_VPDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Speed	Data transfer speed (KB/sec)
Requests	Processing count
Read Requests	Read request count
Read Speed	Read data transfer speed (KB/sec)
Write Requests	Write request count
Write Speed	Write data transfer speed (KB/sec)

VM Disk Used (9.0) (Monthly Trend)

Overview

The VM Disk Used (9.0) report displays the logical disk usage of a virtual machine over the past month in daily summaries, in table and line graph formats.

Storage location

Reports/RM VirtualMachine/Monthly Trend/

Record

VM Logical Disk Status (PI_VLDI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Used	Size of disk space used (MB)
Used %	Disk usage (%)
Free	Size of disk space unused (MB)
Size	Disk size (MB)

VM Disk Used (9.0) (Status Reporting/Real-Time)

Overview

The `VM Disk Used (9.0)` report displays the logical disk usage of a virtual machine on a real-time basis, in list and line graph formats.

Storage location

`Reports/RM VirtualMachine/Status Reporting/Real-Time/`

Record

VM Logical Disk Status (`PI_VLDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Used	Size of disk space used (MB)
Used %	Disk usage (%)
Free	Size of disk space unused (MB)
Size	Disk size (MB)

VM Virtual Disk Allocation Value (8.0) (Status Reporting/Real-Time)

Overview

The `VM Virtual Disk Allocation Value (8.0)` report displays the amount of virtual disk space allocated to a virtual machine in table and summary bar graph formats.

Storage location

`Reports/RM VirtualMachine/Status Reporting/Real-Time/`

Record

VM Virtual Disk Detail (`PD_VDKD`)

Fields

Field name	Explanation
VM Name	Virtual machine name
Datastore ID	Datastore ID
Datastore Name	Datastore name
Controller Name	Controller name
Bus Number	Bus number associated with the controller
Unit Number	Unit number on the controller
Disk UUID	UUID of the virtual disk
Capacity	Capacity allocated to the virtual disk (MB)

VM Disk Used Status (9.0) (Troubleshooting/Real-Time)

Overview

The `VM Disk Used Status (9.0)` report displays the logical disk usage status of a virtual machine on a real-time basis, in list and stacked bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Logical Disk Status (`PI_VLDI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Disk ID	Disk ID
Used	Size of disk space used (MB)
Used %	Disk usage (%)
Free	Size of disk space unused (MB)
Size	Disk size (MB)

VM Memory Allocation Value (6.0) (Troubleshooting/Real-Time)

Overview

The VM Memory Allocation Value (6.0) report displays the memory allocation upper limit of a virtual machine on a real-time basis, in list and summary bar graph formats.

Storage location

Reports/RM VirtualMachine/Troubleshooting/Real-Time/

Record

VM Memory Status (PI_VMI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Expectation	Memory allocation balancing value (MB)
Expectation %	Memory allocation balancing point (%)
Max	Memory allocation upper limit value (MB)
Max %	Memory allocation upper limit rate (%)
Min	Memory allocation lower limit value (MB)
Min %	Memory allocation lower limit rate (%)
Size	Memory size (MB)
Share	Memory allocation share

VM Memory Allocation Value (6.0) (Troubleshooting/Recent Past)

Overview

The `VM Memory Allocation Value (6.0)` report displays the memory allocation upper limit of a virtual machine over the past hour in minute-by-minute summaries, in table and summary bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Expectation	Memory allocation balancing value (MB)
Expectation %	Memory allocation balancing point (%)
Max	Memory allocation upper limit value (MB)
Max %	Memory allocation upper limit rate (%)
Min	Memory allocation lower limit value (MB)
Min %	Memory allocation lower limit rate (%)
Size	Memory size (MB)
Share	Memory allocation share

VM Memory Trend (Monthly Trend)

Overview

The VM Memory Trend report displays the memory usage by a virtual machine over the past month in daily summaries, in table and line graph formats.

Storage location

Reports/RM VirtualMachine/Monthly Trend/

Record

VM Memory Status (PI_VMI)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Used	Memory usage volume (MB)
Used %	Memory usage rate (%)
Unused	Memory unused (MB)
Size	Memory size (MB)

VM Memory Used (6.0) (Troubleshooting/Recent Past/Drilldown Only)

Overview

The `VM Memory Used (6.0)` report displays the memory resource used by a virtual machine over the past hour in minute-by-minute summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/Drilldown Only/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Used	Memory usage volume (MB)
Used %	Memory usage rate (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Unused	Memory unused (MB)
Working Size	Working set size (MB)
Working Size %	Working set size rate (%)

Drilldown report (field level)

Report name	Explanation
`VM Memory Used Status`	Displays the memory resource usage status by the selected virtual machine. To display this report, click the VM Name field.

VM Memory Used Status (6.0) (Troubleshooting/Recent Past/Drilldown Only)

Overview

The `VM Memory Used Status (6.0)` report displays the memory resource usage status of a virtual machine over the past hour in minute-by-minute summaries, in table and stacked bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/Drilldown Only/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Used	Memory usage volume (MB)
Used %	Memory usage rate (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Working Size	Working set size (MB)
Working Size %	Working set size rate (%)
Unused	Memory unused (MB)

VM Network Data (6.0) (Monthly Trend)

Overview

The `VM Network Data (6.0)` report displays the volume of network data sent/received by a virtual machine over the past month in daily summaries, in table and line graph formats.

Storage location

`Reports/RM VirtualMachine/Monthly Trend/`

Record

VM Network Status (`PI_VNI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Net ID	Network ID
Rate	Speed at which the physical server sends data to or receives data from the network (KB/sec)
Recv Rate	Speed at which the physical server receives data from the network (KB/sec)
Send Rate	Speed at which the physical server sends data to the network (KB/sec)

VM Network Data (6.0) (Troubleshooting/Real-Time)

Overview

The `VM Network Data (6.0)` report displays the volume of network data sent/received by a virtual machine on a real-time basis, in list and line graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Network Status (`PI_VNI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Net ID	Network ID
Rate	Speed at which the physical server sends data to or receives data from the network (KB/sec)
Recv Rate	Speed at which the physical server receives data from the network (KB/sec)
Send Rate	Speed at which the physical server sends data to the network (KB/sec)

VM Swap Used (6.0) (Troubleshooting/Real-Time)

Overview

The `VM Swap Used (6.0)` report displays the swap usage of a virtual machine on a real-time basis, in list and stacked bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Resource Used	Memory resources used (MB)
Resource Used %	Memory resource usage (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
Unused	Memory unused (MB)

VM Swap Used (6.0) (Troubleshooting/Recent Past)

Overview

The `VM Swap Used (6.0)` report displays the swap usage of a virtual machine over the past hour in minute-by-minute summaries, in table and stacked bar graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Resource Used	Memory resources used (MB)
Resource Used %	Memory resource usage (%)
VM Swap Used	Internal swap used (MB)
VM Swap Used %	Internal swap usage (%)
Host Swap Used	External swap used (MB)
Host Swap Used %	External swap usage (%)
Unused	Memory unused (MB)

VM Working Size - Total (6.0) (Troubleshooting/Real-Time)

Overview

The `VM Working Size - Total (6.0)` report displays the working set size of all virtual machines on a real-time basis, in list and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Real-Time/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Working Size	Working set size (MB)
Working Size %	Working set size rate (%)

VM Working Size - Total (6.0) (Troubleshooting/Recent Past)

Overview

The `VM Working Size - Total (6.0)` report displays the working set size of all virtual machines over the past hour in minute-by-minute summaries, in table and stacked area graph formats.

Storage location

`Reports/RM VirtualMachine/Troubleshooting/Recent Past/`

Record

VM Memory Status (`PI_VMI`)

Fields

Field name	Explanation
Sampling Time	Performance information collection time on the monitored host
VM Name	Virtual machine name
Working Size	Working set size (MB)
Working Size %	Working set size rate (%)

5 Records

This chapter explains the records used in PFM - RM for Virtual Machine. For details about how to collect performance data for each record, see the chapter that explains Performance Management functions in the *JP1/Performance Management Planning and Configuration Guide*, or see the chapter that explains management of operation monitoring data in the *JP1/Performance Management User's Guide*.

Data model

The records and fields that each PFM - RM for Virtual Machine has are collectively called *data models*. The data models that these PFM - RM for Virtual Machine has are individually assigned unique version numbers. For details about the relationship between versions of PFM - RM for Virtual Machine and data model versions, see *H. Version Compatibility*.

To view the data model version of each PFM - RM for Virtual Machine, display the properties of the agent from the Agents window of PFM - Web Console.

For details about data models, see the chapter that explains Performance Management functions in the *JP1/Performance Management Planning and Configuration Guide*.

Format of record explanations

This chapter lists in alphabetical order the records used in PFM - RM for Virtual Machine. The explanation of each record consists of the following items.

Function

Provides an overview of the performance data stored in each record, and notes related to it.

Default values and values that can be changed

This item shows the default values of the performance data collection conditions that are set in each record and the values that can be changed by the user, in table format. The table below shows the items that are described as default values, the values that can be changed, and their meanings. For details about each item shown in this table, see the chapter that explains management of operation monitoring data in the *JP1/Performance Management User's Guide*.

Table 5–1: Default values and values that can be changed

Item	Meaning	Changeability
`Collection interval`	Performance data collection interval (seconds)	`Y`: Can be changed. `N`: Cannot be changed.
`Collection Offset`#1	Offset for starting performance data collection (seconds). For details about offsets, see the chapter that explains management of operation monitoring data in the *JP1/Performance Management User's Guide*. For details about the performance data collection start time, see the chapter that explains Performance Management functions in the *JP1/Performance Management Planning and Configuration Guide*.	
`Log`	Whether to record the collected performance data in the Store database Yes: Records the performance data. However, performance data will not be recorded if `Collection Interval=0`. No: Does not record the performance data.	
`LOGIF`	Condition for determining whether to record the collected performance data in the Store database	
`Over 10 Sec Collection Time`#2	Whether a certain system configuration can cause record collection to take 10 seconds or longer. `Yes`: Record collection might take 10 seconds or longer. `No`: Record collection does not take 10 seconds.	

#1

A value between 0 and 32,767 seconds (within the value specified in `Collection Interval`) can be specified. When multiple data items are being collected, the system load will increase if they are all collected at the same time. Therefore, a collection offset is used to distribute the collection load. Note that the data collection recording time is the same as that specified in `Collection Interval` regardless of the value specified in `Collection Offset`. When you change the `Collection Offset` value, specify a value by taking the collection load into consideration.

#2

This property is displayed if the collection of historical data is prioritized over the display processing of real-time reports. For details, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

ODBC key field

These fields display the primary keys that are necessary to use the data retrieved from records stored in the Store database on PFM - Manager. Some ODBC key fields are common to all records, while others are unique to each record. ODBC key fields unique to individual records are listed under this heading in each record description. Only multi-instance records have unique ODBC key fields.

For the ODBC key fields common to all records, see *List of ODBC key fields* in this chapter.

Lifetime

Lifetime indicates a period during which the integrity of the performance data collected in each record is guaranteed. For details about lifetime, see the chapter that explains Performance Management functions in the *JP1/Performance Management Planning and Configuration Guide*.

Record size

Record size indicates the size of the performance data that is stored in each record during a single collection operation.

Fields

The fields of each record are listed in a table. The various items in the table are explained below.

- PFM - View name (PFM - Manager name)
    - PFM - View name
      Indicates the field name (PFM - View name) displayed on PFM - Web Console.
    - PFM - Manager name
      Indicates the field name (PFM - Manager name) to be described in an SQL when you want to use the SQL from PFM - Manager to utilize the data of the field stored in the Store database.
      In the SQL statement, the field name is described in a format that has the record ID of each record at the beginning. For example, in the case of the VM Count `(VM_COUNT)` field of the Host Status Detail (`PD`) record, describe the field name as `PD_VM_COUNT`.
- Explanation
  Explains the performance data that is stored in each field.
  You can obtain the performance data of each field in any of the following ways:
    - Obtain the average or a percentage from the most recently collected data and the data collected during the previous interval
    - Obtain performance data only from the most recently collected data
    - Obtain data from other fields

  Unless otherwise specified, the value obtained from a data collection interval is used.
  In a historical report, the following types of values can be displayed when records of the PI record type are summarized by specifying a value other than minutes as the reporting interval:
    - The average value for the summary interval
    - The last collected value
    - The total value
    - The minimum value
    - The maximum value

  Unless otherwise specified, the average value for the summary interval is displayed for the field value.
- Summary rules
  Indicates the summary method (rule) used by Remote Monitor Store to summarize data. For details about summary rules, see *Summary rules* in this chapter.

- Grouping rules

  Grouping rules refers to a method that is used to consolidate the performance information of remote agents that belong to the same instance. For details about grouping rules, see *Grouping rules* in this chapter.

- Format

  Indicates the data type, such as `double`, of the value in each field. For details about data types, see *List of data types* in this chapter.

- Delta

  When data is collected as a cumulative value, a *delta* value is used to display the data as the changed amount. For details about delta values, see *Field values* in this chapter.

- When not collected

  Indicates the value that is stored in the field when performance data collection failed.

  - No: Indicates the case in which no data for the entire record can be collected when performance data collection fails.
  - --: Indicates a field in which data can always be collected.

- Unsupported

  Indicates a virtual environment that does not support the field.

  - --: Indicates that the field is supported by all virtual environments supported by PFM - RM for Virtual Machine.

List of ODBC key fields

Some ODBC key fields are common to all records, while others are specific to individual records. The ODBC key fields are necessary to use the data retrieved from records stored in the Store database on PFM - Manager.

The table below lists the ODBC key fields that are common to all records. For details about the ODBC key fields that are specific to each record, see the individual record descriptions.

Table 5–2: List of ODBC key fields common to all records

ODBC key field	ODBC format	Data	Explanation
*record-ID*_DATE	SQL_INTEGER	Internal	Record key indicating the date when the record was generated
*record-ID*_DATETIME	SQL_INTEGER	Internal	Combination of record ID_DATE field and record ID_TIME field
*record-ID*_DEVICEID	SQL_VARCHAR	Internal	Instance name [host name]
*record-ID*_DRAWER_TYPE	SQL_VARCHAR	Internal	Category. The values are valid: m: Minute H: Hour D: Day W: Week M: Month Y: Year
*record-ID*_PROD_INST	SQL_VARCHAR	Internal	PFM - RM for Virtual Machine instance name
*record-ID*_PRODID	SQL_VARCHAR	Internal	PFM - RM for Virtual Machine product ID
*record-ID*_RECORD_TYPE	SQL_VARCHAR	Internal	ID indicating a record type (4 bytes)
*record-ID*_TIME	SQL_INTEGER	Internal	Time at which the record was generated (Greenwich Mean Time)

Summary rules

For records of the `PI` record type, two types of data are stored in the Store database: The data collected at the interval set in Collection Interval, and the data summarized for a specific period of time (minute, hour, day, week, month, or year) according to a predefined rule. The type of summarization is defined for each field. This definition is called a *summarization rule*.

Depending on the summarization rule, intermediate data in the summarization period must be retained. In this case, a field for holding the intermediate data is added to a record in the Store database. This field is called an *added field*.

Part of an added field is displayed as a record field in PFM - Web Console. The added fields displayed on PFM - Web Console can be used as the fields to be displayed in a historical report.

The fields referred to in the record descriptions in this chapter are known as *record-specific fields* to distinguish them from additional fields generated when data is summarized.

Additional fields have the following field names:

- Additional field contained in the Store database
  PFM - Manager name of the record-specific field, plus a suffix
- Additional field displayed in PFM - Web Console
  PFM - View name of the record-specific field, plus a suffix

The following table shows the suffix added to the PFM - Manager name, the suffix added to the corresponding PFM - View name, and the data stored in that field.

Table 5–3: List of suffixes in additional field names

Suffix added to the PFM - Manager name	Suffix added to the PFM - View name	Field data
`_TOTAL`	(Total)	Sum of the field values in all records in the summary period
`_COUNT`	--	Number of records collected in the summary period
`_HI`	(Max)	Highest field value in the records in the summary period
`_LO`	(Min)	Lowest field value in the records in the summary period

Legend:
--: No additional field.

The table below lists the summary rules.

Table 5–4: List of summary rules

Summary rule name	Summary rules
`COPY`	Stores the actual field value of the most recent record in the summary period.
`AVG`	Stores the average field value of all field values in the summary period. The average value is calculated using the following expression: `(`*`sum-of-field-values`*`)/(`*`number-of-records-collected`*`)` Additional field (Store database) • `_TOTAL`

Summary rule name	Summary rules
AVG	• _COUNT Additional field (PFM - Web Console) • (Total)
HILO	Stores the highest value, lowest value, and average value of all field values in the summary period. A record-specific field stores the average value. The highest value, lowest value, and average value is calculated using the following expression: (*sum-of-field-values*) / (*number-of-records-collected*) Additional field (Store database) • _HI • _LO • _TOTAL • _COUNT Additional field (PFM - Web Console) • (Max) • (Min) • (Total)
--	No summarization

Grouping rules

The performance data of managed hosts in the same instance environment is consolidated as group agent data based on predefined rules. These rules are called *grouping rules*.

The following table lists the grouping rules.

Table 5–5: List of grouping rules

Grouping rule name	Grouping rules
`ADD`	Stores the sum of the performance data values for all monitored hosts in the same instance environment.
`AVG`	Stores the average of the performance data values for all monitored hosts in the same instance environment.
`COPY`	Stores the value of specific performance data for monitored hosts in the same instance environment.
`FIXED`	Stores a fixed value independently of performance data.

List of data types

The table below lists the data type for each field value and the corresponding C and C++ data types. The *Field* value under *Data type* shown in this table is indicated under the *Format* column in the Field table for each record.

Table 5–6: Data type list

Data type		Size (bytes)	Explanation
Field	C and C++		
char(*n*)	char()	1	Character data (0x20-0x7e)
double	double	8	Numeric value (1.7E±308 (15 digits))
long	long	4	Numeric value (-2,147,483,648 through 2,147,483,647)
short	short	2	Numeric value (-32,768 through 32,767)
string(*n*)	char[]	Number inside ( )	Character string that is *n*-byte long. The last characters are NULL.
time_t	unsigned long	4	Numeric value (0 through 4,294,967,295)
timeval	structure	8	Numeric value (The first 4 bytes and the next 4 bytes indicate seconds and microseconds, respectively.)
ulong	unsigned long	4	Numeric value (0 through 4,294,967,295)
ushort	unsigned short	2	Numeric value (0 through 65,535)
utime	structure	8	Numeric value (The first 4 bytes and the next 4 bytes indicate seconds and microseconds, respectively.)
word	unsigned short	2	Numeric value (0 through 65,535)
(Not applicable)	unsigned char	1	Numeric value (0 through 255)

Field values

This section explains the values stored in each field.

Delta

When data is collected as a cumulative value, *delta* is used to display the data as the amount by which the data has changed. For example, if the first performance data item collected is 3 and the second performance data item collected is 4, the cumulative value of 7 and a changed amount of 1 are stored. Whether the value in each field is a delta value is indicated in the *Delta* column of the Field table. Performance data collected by PFM - RM for Virtual Machine varies as shown in the table below.

Note that the delta value can be a negative value since it is a relative value that depends on the last data item.

Table 5–7: Performance data collected by PFM - RM for Virtual Machine

Record type	Delta	Type of data	Indicate delta value#	Record value
PI record type	Yes	Real-time data	Selected	Displays the changed amount.
			Not selected	Displays the changed amount.
		• Historical data • Alarm monitoring data	--	Displays the changed amount.
	No	Real-time data	Selected	Displays the value that was valid at the time of collection.
			Not selected	Displays the value that was valid at the time of collection.
		• Historical data • Alarm monitoring data	--	Displays the value that was valid at the time of collection.
PD record type	Yes	Real-time data	Selected	Displays the changed amount.
			Not selected	Displays the cumulative value.
		• Historical data • Alarm monitoring data	--	Displays the cumulative value.
	No	Real-time data	Selected	Displays the value that was valid at the time of collection.
			Not selected	Displays the value that was valid at the time of collection.
		• Historical data • Alarm monitoring data	--	Displays the value that was valid at the time of collection.

Legend:

--: Not applicable

#

Selected means that the following GUI item is selected in the dialog box of PFM - Web Console:

- **Indicate delta value** in the report wizard's Edit > Indication Settings(Realtime) window
- **Indicate delta value** in **Indication Settings(Realtime)** on the **Properties** page in the report window

Note the following when performance data is being collected:

- Before records of the PI record type can be saved, performance data must be collected at least twice.

  Records of the PI record type collect performance data for each collection interval that was set in PFM - Web Console. However, performance data is not stored in the Store database at the time performance data collection is set in PFM - Web Console.

  Since historical data of the PI record type contains data that requires the difference from the previously collected data (delta value), data from two collection intervals is required. Consequently, it takes a maximum of twice the set time before historical data is stored in the Store database.

  For example, if the performance data collection interval is set to 300 seconds (5 minutes) at 18:32 on PFM - Web Console, the first data collection starts at 18:35 and the next data collection starts at 18:40. Historical data is created based on the data collected at 18:35 and 18:40, and is stored in the Store database at 18:40 (8 minutes after the setting time of 18:32).
- A real-time report displays a value immediately when the first data is collected.

  However, if a report requires previous data, `0` is displayed as the first value. The second and subsequent data collection operations differ depending on the report.
- In the following cases, after the second data collection, the value in effect at the time of data collection is displayed:
  - **Indicate delta value** is not selected in the real-time report setting for the PI record type.
  - **Indicate delta value** is selected in the real-time report setting for the PD record type.
- In the following case, after the second data collection, the difference between the first and second data is displayed. After the third data collection, however, the value in effect at the time of data collection is displayed:
  - **Indicate delta value** is selected in the real-time report setting for the PI record type.
- If the monitored channel is rebooted while PFM - RM for Virtual Machine is starting, the collected data value may show a negative value. However, a value of `0` or greater will be displayed as the data difference for the second and subsequent data collections.

Fields that are added only when a record is recorded in the Store database

The table below shows the fields that are added only when data is being recorded in the Store database.

Table 5–8: Fields that are added only when data is being recorded in the Store database

PFM - View name (PFM - Manager name)	Explanation	Format	Delta	Supported version	Data source
Agent Host (`DEVICEID`)	Name of the host on which PFM - RM for Virtual Machine is running	`string` (256)	No	All	--
Agent Instance (`PROD_INST`)	PFM - RM for Virtual Machine instance name	`string` (256)	No	All	--
Agent Type (`PRODID`)	PFM - RM for Virtual Machine product ID, expressed as a 1-byte ID	`char`	No	All	--
Date (`DATE`)	Date on which the record was created. Greenwich Mean Time#1, #2	`char` (3)	No	All	--
Date and Time (`DATETIME`)	Combination of Date (`DATE`) field and Time (`TIME`) field#2	`char` (6)	No	All	--
Drawer Type (`DRAWER_TYPE`)	For a record of the PI record type, this is the category in which data is summarized.	`char`	No	All	--
GMT Offset (`GMT_ADJUST`)	Difference between Greenwich Mean Time and the local time, in seconds	`long`	No	All	--
Time (`TIME`)	Time at which the record was created. Greenwich Mean Time#1, #2	`char` (3)	No	All	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired from the monitoring-target environment.

#1

Since a record of the PI record type summarizes data, the time used as the basis for summarizing data is specified. The following table indicates the setting for each record category.

Table 5–9: Setting for each record category

Category	Setting for each record category
Minute	The 0th second of the time at which the record was created
Hour	The 0th minute and 0th second of the time at which the record was created
Day	The 0th hour, 0th minute, and 0th second of the time at which the record was created
Week	The 0th hour, 0th minute, and 0th second of the Monday of the week in which the record was created
Month	The 0th hour, 0th minute, and 0th second of the first day of the month in which the record was created
Year	The 0th hour, 0th minute, and 0th second of January 1st of the year in which the record was created

#2

When data is displayed in reports, the Date field is displayed in the format *YYYYMMDD*, the Date and Time field is displayed in the format *YYYYMMDD hh:mm:ss*, and the Time field is displayed in the format *hh:mm:ss*.

Notes on records

When you collect records, note the following.

Precautions for collecting performance data

The following provides precautions for collecting performance data.

About the registry

This note applies to only Hyper-V.

Operation of PFM - RM for Virtual Machine is only supported in an OS environment using the standard settings.

Performance data might not be collected correctly if special OS settings are specified in some way, such as using the registry editor to edit the registry directly. This is also true if the special settings have been published in the Microsoft Knowledge Base.

About collection of log data

PFM - RM for Virtual Machine stores log data for all monitoring targets or log data consolidated on a group agent basis in data files. A data file is created for each record. The maximum size of each data file is 2 GB. Therefore, if many hosts are monitored by a PFM - RM for Virtual Machine instance or if many instances are generated for a record, the Store database might be unable to store all of the log data.

In particular, log data consolidated by a group agent is likely to put pressure on the data file size. Therefore, if you use a group agent to perform monitoring, you must reduce the number of hosts monitored by the instance. Alternatively, you must use the `LOGIF` property to filter data that can be stored in the Store database.

For details about how to estimate the disk space requirements for the Store database, see *A. Estimating System Requirements*.

Notes on changing the system resources of a monitored host

The following provides notes on changing the system resources of a monitored host.

About performance data before and after system resources are changed

If the system resources of a monitored host are changed, continuity between data before the change and data after the change will be lost. Therefore, performance data before the system resources are changed and performance data after the system resources are changed must be handled separately.

Record generation result when data cannot be collected

The following explains the record generation result when data stored in a field cannot be collected.

Record is not generated

No record is generated in the following case:

- When PFM - RM for Virtual Machine cannot collect the performance data stored in a field defined as an ODBC key field

Note on monitoring VMware environment

- If a VMware instance is monitored that uses NFS or some other network file system as the storage system, the instance cannot be obtained by using the `PI_VPDI` or `PI_HPDI` record.

- If the monitoring target is VMware ESXi, Lockdown Mode is disabled. If Lockdown Mode is enabled, the collection of performance data fails.
- The virtual machine that has symbols such as / (slash), \ (backslash) and % (percent) in its name can not be monitored.

Note about virtual machine name for Hyper-V

- If there are multiple virtual machines with the same name in the same Hyper-V system, PFM - RM for Virtual Machine might not be able to collect the correct information. Make sure that each virtual machine has a unique name.

Notes on collecting KVM and Hyper-V records

For records of the PI record type, there are fields that require performance data to be collected at least twice.

Causes of the value of a field becoming 0

The value of a field becomes 0 in the following cases:

- The field requires performance data to be collected at least twice, but performance data has been collected only once.
- The field requires performance data to be collected at least twice, but performance data collection order is not chronologically correct.

Notes on collecting KVM records

- When the monitoring target is KVM, let PFM - RM for Virtual Machine refer the mounted remote file system (such as the df command be able to executed correctly). During the mounted remote file system does not reply, if you specify Disk_Category of an instance "Y", Remote Agent service cannot collect the performance data correctly. In this case, perform the following procedure:
  1. Change Disk_Category of an instance "N".
  2. Stop the df process on the remote host by using the following command:
     `# kill -TERM(or KILL) "process ID of df"`
  3. Recover the remote file system to be mounted correctly such as restarting the NFS daemon.
  4. Change Disk_Category of an instance "Y".
- If the monitored host is KVM, the value of the Used field in the Host Logical Disk Status (`PI_HLDI`) record does not include the free space of reserved blocks managed by Linux file system. Therefore, the value of the Size field is not equal to the total of the Used and Free fields.

Notes on collecting Docker environment records

For records of the PI record type, there are fields that require performance data to be collected at least twice.

Causes of the value of a field becoming 0

The value of a field becomes 0 in the following cases:

- The field requires performance data to be collected at least twice, but performance data has been collected only once.
- The field requires performance data to be collected at least twice, but start time of collected Docker containers has been changed.

Notes on collecting Podman environment records

For records of the PI record type, there are fields that require performance data to be collected at least twice.

Causes of the value of a field becoming 0

The value of a field becomes 0 in the following cases:

- The field requires performance data to be collected at least twice, but performance data has been collected only once.
- The field requires performance data to be collected at least twice, but the container is restarted before the latest performance data is collected after the previous performance data was collected.

Note on monitoring the environment with logical partitioning feature

When the HBA and an NIC in dedicated mode are assigned to a logical partition, the following information cannot be collected:

- `PI_VPDI` (VM Physical Disk Status)
- `PI_VNI` (VM Network Status)

Other notes

- The values of PI_V*** records are obtained from a virtual machine and are different from the performance data of the running guest OS actually.
- "PerfNet ID: 2006" is recorded in the event log of the monitored host when monitoring the Hyper-V environment where there are 32 or more CPUs on the monitored host, but there is no problem collecting performance data. Also, ignore the event of "PerfNet ID: 2006".
- If you specify the character strings other than ASCII to the image file name (and path name) of the guest OS, the value of the Disk ID field in the `PI_VPDI` record might not be displayed correctly.

List of records

The following table shows the records that can be collected by PFM - RM for Virtual Machine, and the information stored in these records.

Table 5–10: PFM - RM for Virtual Machine record list

Record name	Record ID	Information stored
Host CPU Status	*PI_HCI*	Performance data on a physical server's physical CPU over a unit of time
Host Logical Disk Status	*PI_HLDI*	Performance data on a physical server's logical disk over a unit of time
Host Memory Status	*PI_HMI*	Performance data on a physical server's physical memory over a unit of time
Host Network Status	*PI_HNI*	Performance data on a physical server's physical NIC over a unit of time
Host Physical Disk Status	*PI_HPDI*	Performance data on a physical server's physical disk over a unit of time
Host Status Detail	*PD*	Performance data that indicates the status of a physical server at a given point in time
Host Status	*PI*	Performance data on a physical server over a unit of time
VM CPU Status	*PI_VCI*	Performance data on a virtual CPU being used by a virtual machine over a unit of time
VM Logical Disk Status	*PI_VLDI*	Performance data on a logical disk being used by a virtual machine over a unit of time
VM Memory Status	*PI_VMI*	Performance data on virtual memory being used by a virtual machine over a unit of time
VM Network Status	*PI_VNI*	Performance data on a virtual NIC being used by a virtual machine over a unit of time
VM Physical Disk Status	*PI_VPDI*	Performance data on a physical disk being used by a virtual machine over a unit of time
VM Virtual Disk Status	*PI_VVDI*	Performance data on a virtual disk being used by a virtual machine over a unit of time
VM Status Detail	*PD_VM*	Performance data that indicates the status of a virtual machine at a given point in time
VM Status	*PI_VI*	Performance data on a virtual machine over a unit of time
Host Generic Data Detail	*PD_HGDD*	Performance data that indicates the status of a physical server at a given point in time
Host Generic Data Interval	*PI_HGDI*	Performance data on a physical server over a unit of time
VM Generic Data Detail	*PD_VGDD*	Performance data that indicates the status of a virtual machine at a given point in time
VM Generic Data Interval	*PI_VGDI*	Performance data on a virtual machine over a unit of time
VM Virtual Disk Detail	*PD_VDKD*	Performance data that indicates the status of a virtual machine at a given point in time
Pod Status Detail	*PD_PODD*	Performance data that indicates the status of a POD at a given point in time
Pod Status Interval	*PI_PODI*	Performance data on a POD over a unit of time
Pod Container Status Interval	*PI_POCI*	Performance data on a POD container over a unit of time

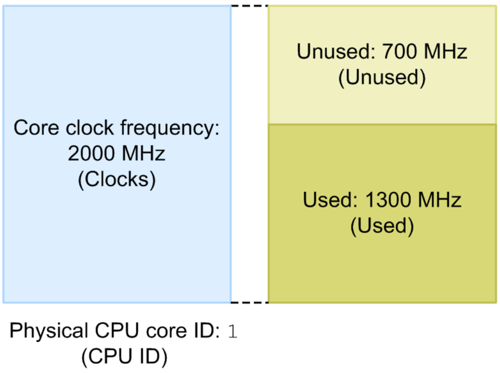
Host CPU Status (PI_HCI)

Function

This record stores performance data on a physical server's physical CPU over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as CPU usage for each CPU core. The CPU frequency can be divided into the part of the CPU that is being used and the part of the CPU that is not being used. The following figure shows an example of data that can be collected in this record.

Figure 5–1: Example of collected data



Note:

Information cannot be collected if an attempt to connect to the monitored physical server fails.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_HCI_CPU_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 473 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always "HCI".	COPY	COPY	char (8)	No	--	Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Docker environment
CPU ID (CPU_ID)	ID of the physical CPU.	COPY	FIXED	string (32)	No	No	Docker environment
CPU Name (CPU_NAME)	Name of the physical CPU.	COPY	FIXED	string (257)	No	Blank	Docker environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#1	COPY	FIXED	string (32)	No	Blank	Docker environment
Clocks (CLOCKS)	Clock frequency of the physical CPU. [Units: MHz]	COPY	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Used (USED)#2	Amount of used physical CPUs. The frequency used by the physical CPUs. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Unused (UNUSED)	Amount of unused physical CPUs. Unused frequency of the physical CPU. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Unused (UNUSED)	Amount of unused physical CPUs. Unused frequency of the physical CPU. [Units: MHz]	HILO	ADD	double	No	logical partitioning feature: -1 Podman environment: 0	Docker environment
Used % (USED_PERCENT)#2	Usage rate of physical CPUs. Percentage of physical CPUs that are being used. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Unused % (UNUSED_PERCENT)	Non-usage rate of physical CPUs. Percentage of physical CPUs that are not being used. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment

#1

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#2

For Hyper-V, KVM, or Podman environment this field requires performance data to be collected at least twice.

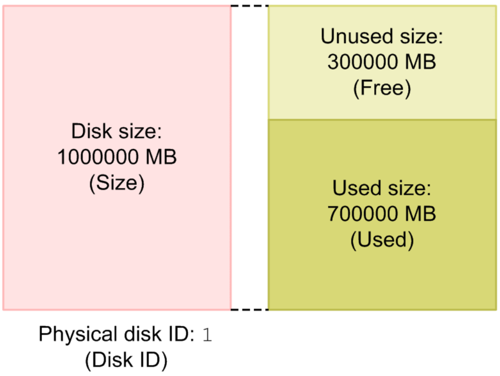
Host Logical Disk Status (PI_HLDI)

Function

This record stores performance data on a physical server's logical disk over a unit of time. This record is a multi-instance record.

In this record you can collect performance data, such as how much of a logical disk on a physical server is used. The disk space can be divided into used space and unused space. The figure below shows an example of data that can be collected in this record.

Figure 5–2: Example of collected data



Notes:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Returns the data store information.
- For Hyper-V, the record is created and performance data can be collected only if the disk is local disk.
- If the monitoring target uses VMware, the value of the `PI_HDLI` field is updated only when VMware ESX is managed with vCenter. For details, see *Important note* in *1.4.5(1) Overview*.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_HLDI_DISK_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 501 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `HLDI`.	`COPY`	`COPY`	`char` (8)	No	--	logical partitioning feature Docker environment
Record Time (`RECORD_TIME`)	The time when the record was created.	`COPY`	`COPY`	`time_t`	No	--	logical partitioning feature Docker environment
Interval (`INTERVAL`)	Interval during which the information is collected. [Units: seconds]	`COPY`	`FIXED`	`ulong`	No	--	logical partitioning feature Docker environment
VA DeviceID (`VADEVICEID`)	The device ID of a monitored host.	`COPY`	`COPY`	`string`(256)	No	--	logical partitioning feature Docker environment
Disk ID (`DISK_ID`)	Disk ID.	`COPY`	`FIXED`	`string` (257)	No	No	logical partitioning feature Docker environment
Sampling Time (`SAMPLING_TIME`)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#	`COPY`	`FIXED`	`string` (32)	No	0	logical partitioning feature Docker environment
Size (`SIZE`)	Disk size. [Units: MB]	`HILO`	`ADD`	`double`	No	0	logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Used (USED)	Size of used space. [Units: MB]	HILO	ADD	double	No	0	logical partitioning feature Docker environment
Free (FREE)	Size of free space. [Units: MB]	HILO	ADD	double	No	0	logical partitioning feature Docker environment
Used % (USED_PERCENT)	Usage rate. [Units: %]	HILO	AVG	double	No	0	logical partitioning feature Docker environment
Last Update (LAST_UPDATE)	The time when the monitored host updated the values for disk space.	COPY	FIXED	string (32)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Free % (FREE_PERCENT)	Non-usage rate of disk space. [Units: %]	HILO	AVG	double	No	0	logical partitioning feature Docker environment

#

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

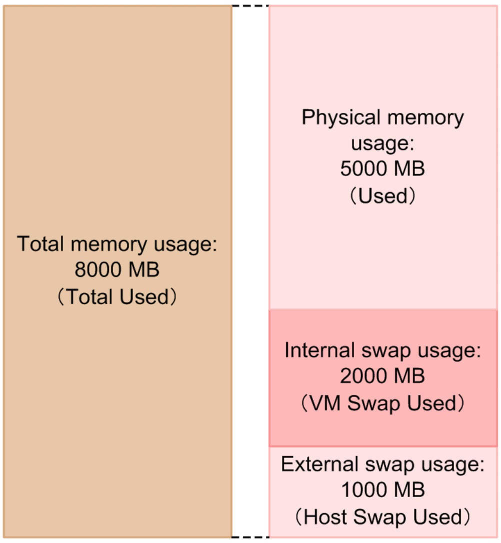
Host Memory Status (PI_HMI)

Function

This record stores performance data on a physical server's physical memory over a unit of time. This record is a single-instance record.

In this record, you can view performance data such as physical memory usage details, usage details by VMM, and swap usage. The figure below shows an example of data that can be collected in this record.

Figure 5–3: Example of collected data



Note:

Information cannot be collected if an attempt to connect to the monitored physical server fails.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

None

Lifetime

None

Record size

- Fixed portion: 1,553 bytes
- Variable portion: 0 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always HMI.	COPY	COPY	char (8)	No	--	Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Docker environment
Size (SIZE)	Memory resource size. Total size of the physical memory on the host. [Units: MB]	COPY	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#	COPY	FIXED	string (32)	No	Blank	Docker environment
Used (USED)	Amount of used memory resources. • For VMware, Hyper-V, KVM or Podman environment Amount of memory resources used on the physical server. • For logical partitioning feature Amount of memory resources allocated to the host machine. [Units: MB]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
VMM Used (VMM_USED)	Amount of memory resources used by the VMM. • For VMware and KVM Amount of memory resources used by VMM. • For logical partitioning feature Amount of memory resources allocated to the hypervisor.	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Hyper-V Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VMM Used (VMM_USED)	[Units: MB]	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Hyper-V Docker environment
VM Used (VM_USED)	Amount of memory resources used by the VM. • For VMware, KVM or Podman environment Amount of memory resources used by the virtual machine. • For logical partitioning feature Amount of memory resources allocated to the virtual machine. Only activated logical partitions are monitored. [Units: MB]	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Hyper-V Docker environment
Unused (UNUSED)	Amount of unused memory resources. Memory resources unused by the host. [Units: MB]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
VM Swap Used (VM_SWAP_USED)	Amount of internal swap areas usage. Amount of the internal swap area used in the virtual machines. [Units: MB]	HILO	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Host Swap Used (HOST_SWAP_USED)	Amount of external swap area usage. Total amount of the external swap area used in all virtual machines. [Units: MB]	HILO	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Total Used (TOTAL_USED)	Total amount of used memory. • For VMware, Hyper-V or KVM	HILO	ADD	double	No	VMware: 0 Hyper-V: 0	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Total Used (TOTAL_USED)	Total amount of memory resources used by the physical server added to the amount of used internal and external swap areas. • For logical partitioning feature Amount of memory resources allocated to the host machine. • For Podman environment Amount of memory resources used by the physical server. [Units: MB]	HILO	ADD	double	No	KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Used % (USED_PERCENT)	Usage rate of memory resources. • For VMware, Hyper-V, KVM or Podman environment Percentage of memory resources used at the physical server. • For logical partitioning feature Percentage of memory resources allocated to the host machine. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
VMM Used % (VMM_USED_PERCENT)	Usage rate of memory resources by the VMM. • For VMware and KVM Percentage of memory resources used by the virtual machine monitor. • For logical partitioning feature Percentage of memory resources allocated to the hypervisor. [Units: %]	HILO	AVG	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1	Hyper-V Docker environment Podman environment
VM Used % (VM_USED_PERCENT)	Usage rate of memory resources by the VM. • For VMware, KVM or Podman environment Percentage of memory resources used by the virtual machine. • For logical partitioning feature Percentage of memory resources allocated to the virtual machine. Only activated logical partitions are monitored. [Units: %]	HILO	AVG	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Hyper-V Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM Swap Used % (VM_SWAP_USED_PERCENT)	Usage rate of internal swap area. Percentage of the internal swap area, of all virtual machines, that is used. [Units: %]	HILO	AVG	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Host Swap Used % (HOST_SWAP_USED_PERCENT)	Usage rate of external swap area. Percentage of the external swap areas, of all virtual machines, that is used. [Units: %]	HILO	AVG	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Total Used % (TOTAL_USED_PERCENT)	Usage rate of all memory resources. • For VMware, Hyper-V, KVM or Podman environment Percentage of total memory resources used, including memory resources used by the physical server and the amount of internal and external swap areas that were used. • For logical partitioning feature Percentage of memory resources allocated to the host machine. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Swap IO (SWAP_IO)	Host swap I/O. Amount of operations, for swap resources, that were generated by the host machine. [Units: MB]	HILO	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment
Swap In IO (SWAP_IN_IO)	Host swap-in I/O. Amount of swap-in operations, for swap resources, that were generated by the host machine. [Units: MB]	HILO	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment
Swap Out IO (SWAP_OUT_IO)	Host swap-out I/O.	HILO	ADD	double	No	0	Hyper-V

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Swap Out IO (SWAP_OUT_IO)	Amount of swap-out operations, for swap resources, that were generated by the host machine. [Units: MB]	HILO	ADD	double	No	0	logical partitioning feature Docker environment

#

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

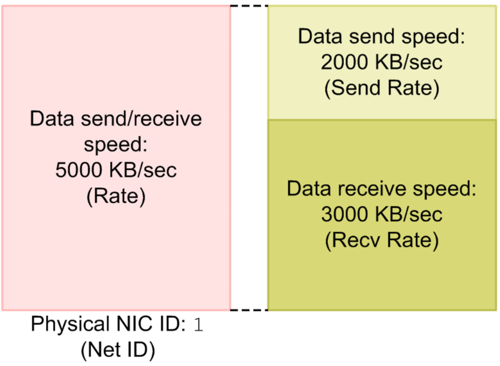
Host Network Status (PI_HNI)

Function

This record stores performance data on a physical server's physical NIC over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as data send/receive speed. The data send/receive speed can be divided into data receive speed and data send speed. The figure below shows an example of data that can be collected in this record.

Figure 5–4: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- For Hyper-V, specific symbols included in a network identifier are converted to other symbols when the network identifier is stored in the `Net ID` field. The conversion rules are as follows:

Before conversion	After conversion
/ and \	-
# and *	_
(	[
)	]

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_HNI_NET_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 397 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always HNI.	COPY	COPY	char (8)	No	--	Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Docker environment
Net ID (NET_ID)	Network ID.	COPY	FIXED	string (257)	No	No	Docker environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*][#1]	COPY	FIXED	string (32)	No	Blank	Docker environment
Rate (RATE)[#2]	Speed of transmission to and from the network by the host machine. [Units: KBps] For logical partitioning feature, when the associated NIC is running in dedicated mode, the value is always set to -1 and cannot be obtained.	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Send Rate (SEND_RATE)[#2]	Speed of sending to the network by the host machine. [Units: KBps] For logical partitioning feature, when the associated NIC is running in dedicated mode, the value is always set to -1 and cannot be obtained.	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Recv Rate (RECV_RATE)#2	Speed of receiving from the network by the host machine. [Units: KBps] For logical partitioning feature, when the associated NIC is running in dedicated mode, the value is always set to -1 and cannot be obtained.	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment

#1

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#2

For Hyper-V, KVM or Podman environment this field requires performance data to be collected at least twice.

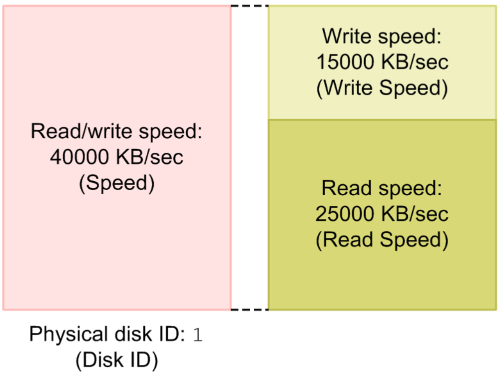
Host Physical Disk Status (PI_HPDI)

Function

This record stores performance data on a physical disk of a physical server over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as data read/write speed on a physical disk of a physical server. For the data read/write speed, you can collect further details, that is, data read speed and data write speed. The figure below shows an example of data that can be collected into this record.

Figure 5–5: Example of collected data



Note:

Information cannot be collected if an attempt to connect to the monitored physical server fails.

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_HPDI_DISK_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 973 bytes

Field

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always HPDI.	COPY	COPY	char (8)	No	--	Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Docker environment
Disk ID (DISK_ID)	Disk ID. ID of the physical disk connected to the physical server.	COPY	FIXED	string (257)	No	No	Docker environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*][#1]	COPY	FIXED	string (32)	No	Blank	Docker environment
Speed (SPEED)[#2]	Data transfer speed. Speed of reads from or writes to the physical disk, made by the host machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature Docker environment
Read Speed (READ_SPEED)[#2]	Reading data transfer speed. Speed of reads from the physical disk, made by the host machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature Docker environment
Write Speed (WRITE_SPEED)[#2]	Writing data transfer speed. Speed of writes to the physical disk, made by the host machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature Docker environment
Requests (REQUESTS)[#2]	Number of requests. • For VMware, Hyper-V, KVM or Podman environment Number of times that read/write requests from the physical server to the physical disk are processed.	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Requests (REQUESTS)#2	• For logical partitioning feature Number of interrupts from the HBA	HILO	ADD	double	No	Podman environment: 0	Docker environment
Read Requests (READ_REQUESTS)#2	Number of read requests. Number of times that read requests, made by the host machine to the physical disk, are processed.	HILO	ADD	double	No	0	logical partitioning feature Docker environment
Write Requests (WRITE_REQUESTS)#2	Number of write requests. Number of times that write requests, made by the host machine to the physical disk, are processed.	HILO	ADD	double	No	0	logical partitioning feature Docker environment
Commands (COMMANDS)	Number of issued disk commands. Number of commands issued by the host machine to the physical disk.	HILO	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Abort Commands (ABORT_COMMANDS)	Number of aborted disk commands. Number of aborted commands made by the host machine to the physical disk.	HILO	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Abort Commands % (ABORT_COMMANDS_PERCENT)	Aborted disk command rate. Percentage of all commands, made by the host machine to a physical disk, that are aborted.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Bus Resets (BUS_RESETS)	Number of bus resets. Number of times the bus was reset for a disk on the host machine.	HILO	ADD	double	No	0	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Bus Resets (`BUS_RESETS`)	Number of bus resets. Number of times the bus was reset for a disk on the host machine.	`HILO`	`ADD`	`double`	No	0	logical partitioning feature Docker environment Podman environment
Device Latency (`DEVICE_LATENCY`)	Average amount of physical device latency. Average amount of time to complete a command from the physical device. [Unit: milliseconds]	`HILO`	`AVG`	`double`	`No`	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Device Read Latency (`DEVICE_READ_LATENCY`)	Average amount of physical device read latency. Average amount of time to complete read from the physical device. [Unit: milliseconds]	`HILO`	`AVG`	`double`	`No`	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Device Write Latency (`DEVICE_WRITE_LATENCY`)	Average amount of physical device write latency. Average amount of time to complete write from the physical device. [Unit: milliseconds]	`HILO`	`AVG`	`double`	`No`	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Kernel Latency (`KERNEL_LATENCY`)	Average Amount of `VMkernel` processing latency. Average amount of spent by `VMkernel` processing each command. [Unit: milliseconds]	`HILO`	`AVG`	`double`	`No`	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Kernel Read Latency (KERNEL_READ_LATENCY)	Average amount of VMkernel processing read latency. Average amount of spent by VMkernel processing each read command. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Kernel Write Latency (KERNEL_WRITE_LATENCY)	Average amount of VMkernel processing write latency. Average amount of spent by VMkernel processing each write command. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Queue Latency (QUEUE_LATENCY)	Average amount of time to VMkernel queue latency. Average amount of spent in the VMkernel queue command. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Queue Read Latency (QUEUE_READ_LATENCY)	Average amount of time to VMkernel queue read latency. Average amount of spent in the VMkernel queue read command. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Queue Write Latency (QUEUE_WRITE_LATENCY)	Average amount of time to VMkernel queue write latency. Average amount of spent in the VMkernel queue write command. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup- ported
Queue Write Latency (QUEUE_WRITE_LATENCY)	Average amount of time to VMkernel queue write latency. Average amount of spent in the VMkernel queue write command. [Unit: milliseconds]	HILO	AVG	double	No	0	Podman environment

#1

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#2

For Hyper-V only, this field requires performance data to be collected at least twice.

Host Status Detail (PD)

Function

This record stores performance data that indicates the status of a physical server at a given point in time. This record is a single-instance record.

Notes:

Information, unless it is on one of the following fields, cannot be collected if an attempt to connect to the monitored physical server fails:

- Record Type
- Record Time
- Interval
- Status
- Reason

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`Yes`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

None

Lifetime

None

Record size

- Fixed portion: 1,595 bytes
- Variable portion: 0 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `PD`.	--	`COPY`	`char` (8)	No	--	--
Record Time (`RECORD_TIME`)	The time when the record was created.	--	`COPY`	`time_t`	No	--	--

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Interval (INTERVAL)	This is always 0.	--	FIXED	ulong	No	--	--
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string(256)	No	--	--
Status (STATUS)	Connection status. SUCCESS: Now executing ERROR: Connection failed	--	FIXED	string (8)	No	Blank	--
Host Name (HOST_NAME)	Connection destination host name.	--	FIXED	string (257)	No	Blank	--
Reason (REASON)	Cause when the "Status" field is ERROR: Connection failed: Connection failed. Authorization failed: Authorization failed. Response invalid: There was an unintended response from the server. Timeout: Collection of the performance data did not end within a given time period. Collection error: Collection failed. When the "Status" field is SUCCESS, this character string is empty.	--	FIXED	string (128)	No	Blank	--
Product (PRODUCT)	Product name of the virtual environment.	--	FIXED	string (257)	No	Blank	--
VM Count (VM_COUNT)	Number of virtual machines or Docker containers that exist on the connection destination hosts. • For VMware, Hyper-V, KVM, or logical partitioning feature Number of Virtual Machines • For Docker environment or Podman environment Number of Docker containers#1	--	ADD	long	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 or 0#2 Docker environment: 0 Podman environment: 0	--
VM Active (VM_ACTIVE)	Number of virtual machines or Docker containers starting up on the connection destination hosts.	--	ADD	long	No	VMware: 0 Hyper-V: 0 KVM: 0	--

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM Active (VM_ACTIVE)	• For VMware, Hyper-V, KVM, or logical partitioning feature Number of Virtual Machines • For Docker environment or Podman environment Number of Docker containers#1	--	ADD	long	No	logical partitioning feature: -1 or 0# Docker environment: 0 Podman environment: 0	--

#1

It does not contain the number of Infra containers in the Podman environment.

#2

If the value of the Status field is set to SUCCESS, this field will be -1. If the value of the Status field is set to ERROR, this field will be -1 or 0.

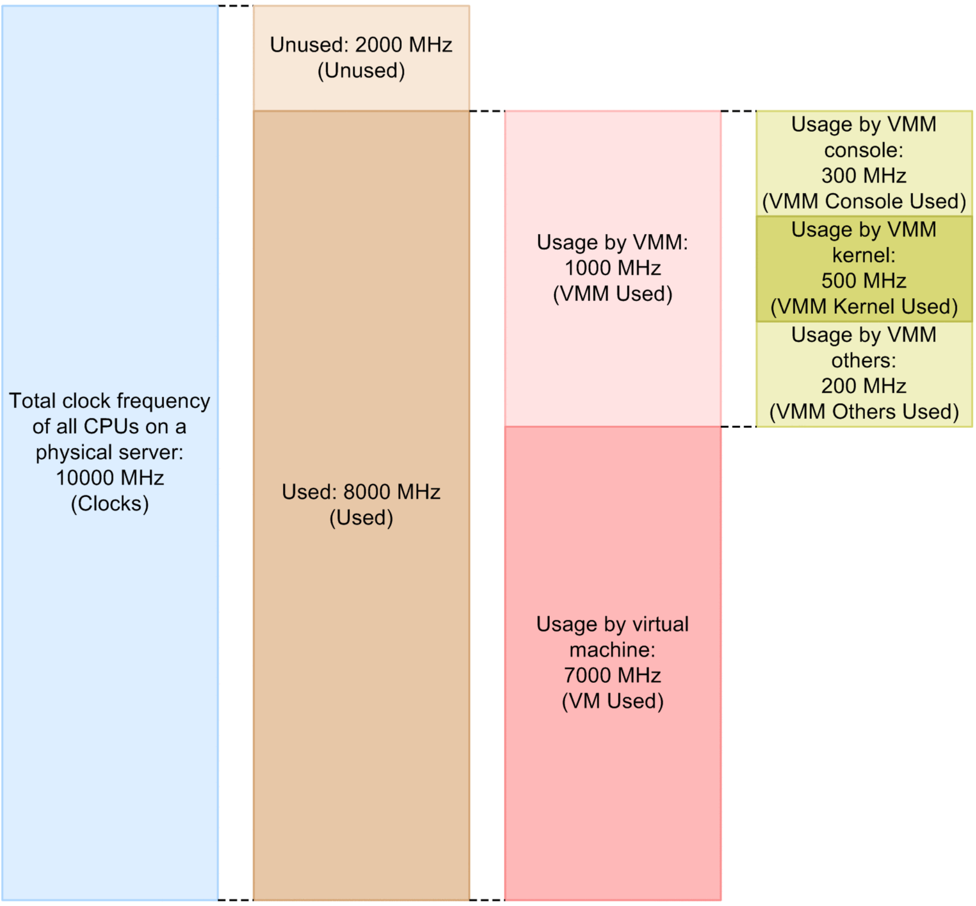
Host Status (PI)

Function

This record stores performance data on a physical server over a unit of time. This record is a single-instance record.

In this record, you can collect performance data such as CPU usage and details about a physical server, and CPU usage details by VMM. The figure below shows an example of data that can be collected in this record.

Figure 5–6: Example of collected data



Notes:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- When VMware information is collected, the timing for collecting the usage, VMM usage, and virtual machine usage differs in VMware, and as a result the following relationship may not hold true in some cases:
  Usage (Used) = VMM usage (VMM Used) + virtual machine usage (VM Used)
- If the monitored host is VMware ESXi 6.7, cannot be collect the Resource CPU usage value from the System(host/system/kernel). Therefore, the value of the VMM Kernel Used and VMM Kernel Used % field is always 0 in the Host Status (PI) record.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes

Item	Default value	Can it be changed?
Log	Yes	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

None

Lifetime

None

Record size

- Fixed portion: 1,629 bytes
- Variable portion: 0 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always PI.	COPY	COPY	char (8)	No	--	Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Docker environment
Clocks (CLOCKS)	Clock frequency of the CPU resource. The total value of the clock frequencies of the physical CPUs installed on the host machine.	COPY	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Count (COUNT)	Number of physical CPU cores in the CPU resources. Number of physical CPU cores installed on the host machine.	COPY	ADD	long	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*][#1]	COPY	FIXED	string (32)	No	Blank	Docker environment
Used (USED)[#2,#3]	Amount of used CPU resources. Amount of CPU resources used by the host machine. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
VMM Used (VMM_USED)[#2,#3]	Amount of CPU resources used by the VMM. • For VMware, Hyper-V and KVM Amount of CPU resources used by the virtual machine monitor. • For logical partitioning feature Amount of CPU resources used by the hypervisor. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment
VM Used (VM_USED)[#2,#3]	Amount of CPU resources used by the VM. Amount of CPU resources used by the virtual machine. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
VMM Console Used (VMM_CONSOLE_USED)[#3]	Amount of CPU resources used by the VMM console. Amount of CPU resources used by the virtual system console. [Units: MHz]	HILO	ADD	double	No	0	VMware Hyper-V logical partitioning feature Docker environment Podman environment
VMM Kernel Used (VMM_KERNEL_USED)[#3]	Amount of CPU resources used by the VMM kernel. • For VMware and KVM	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1	VMware (ESXi 6.7 or later) Hyper-V

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VMM Kernel Used (VMM_KERNEL_USED)#3	Amount of CPU resources used by the virtual machine monitor kernel. • For logical partitioning feature Amount of CPU resources used by SYS1 as one part of the hypervisor. [Units: MHz]	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment
VMM Others Used (VMM_OTHERS_USED)#3	Amount of CPU resources used by non-VMM processes. • For VMware and KVM Amount of CPU resources used in the virtual machine monitor by something other than the virtual machine monitor console and the virtual machine monitor kernel. • For logical partitioning feature Amount of CPU resources used by SYS2 as one part of the hypervisor. [Units: MHz]	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1	Hyper-V Docker environment Podman environment
Unused (UNUSED)	Amount of unused CPU resources. Amount of unused host CPU resources. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Used % (USED_PERCENT)#2,#3	Usage rate of CPU resources. Percentage of CPU resources used by the host machine. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
VMM Used % (VMM_USED_PERCENT)#2,#3	Usage rate of CPU resources by the VMM. • For VMware, Hyper-V and KVM Percentage of CPU resources used by the virtual machine monitor. • For logical partitioning feature Percentage of CPU resources used by the hypervisor.	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VMM Used % (VMM_USED_PERCENT)#2,#3	[Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment
VM Used % (VM_USED_PERCENT)#2,#3	Usage rate of CPU resources by the VM. Percentage of CPU resources used by virtual machines. [Units: %]	HILO	AVG	double	No	0	Docker environment
VMM Console Used % (VMM_CONSOLE_USED_PERCENT)#3	Usage rate of CPU resources by the VMM console. Percentage of CPU resources used by the virtual system console. [Units: %]	HILO	AVG	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)#3	Usage rate of CPU resources by the VMM kernel. • For VMware and KVM Percentage of CPU resources used by the virtual machine monitor kernel. • For logical partitioning feature Percentage of CPU resources used by SYS1 as one part of the hypervisor. [Units: %]	HILO	AVG	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1	Hyper-V Docker environment Podman environment
VMM Others Used % (VMM_OTHERS_USED_PERCENT)#3	Usage rate of CPU resources by non-VMM processes. • For VMware and KVM Percentage of CPU resources used in the virtual machine monitor by something other than the virtual machine monitor console and the virtual machine monitor kernel. • For logical partitioning feature Percentage of CPU resources used by SYS2 as one part of the hypervisor. [Units: %]	HILO	AVG	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1	Hyper-V Docker environment Podman environment
Unused % (UNUSED_PERCENT)	Non-usage rate of CPU resources. Percentage of unused CPU resources.	HILO	AVG	double	No	VMware: 0 Hyper-V: 0	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Unused % (UNUSED_PERCENT)	[Units: %]	HILO	AVG	double	No	KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Insufficient (INSUFFICIENT)	CPU insufficiency. CPU resources that could not be run by the virtual machine. [Units: MHz]	HILO	ADD	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Insufficient % (INSUFFICIENT_PERCENT)	CPU insufficiency rate. Percentage of CPU resources that are unallocated to a CPU. [Units: %]	HILO	AVG	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Co-Stop (CO_STOP)	CPU conflict. CPU resources that the virtual machine could not run due to CPU conflict. [Units: MHz]	HILO	ADD	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Co-Stop % (CO_STOP_PERCENT)	CPU conflict rate. Percentage of CPU resources that the virtual machine could not run due to CPU conflict. [Units: %]	HILO	AVG	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#1

The value used in `[±hh:mm]` varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, `+09:00` is displayed. If the monitored virtual environment is operating according to UTC, `Z` is displayed.

#2

For Hyper-V, this field requires performance data to be collected at least twice.

#3

For KVM or Podman environment, this field requires performance data to be collected at least twice.

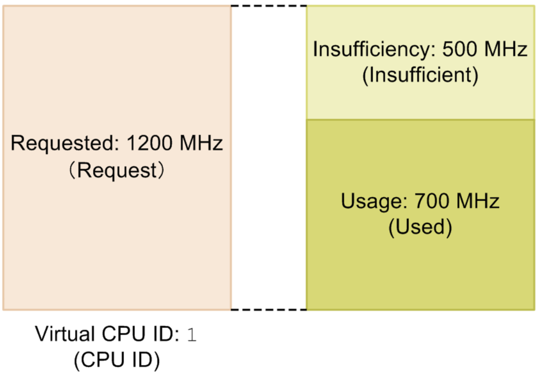
VM CPU Status (PI_VCI)

Function

This record stores performance data on a virtual CPU being used by a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as CPU usage by a virtual machine. The CPU usage can be divided into CPU usage and CPU insufficiency. The figure below shows an example of data that can be collected in this record.

Figure 5–7: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.
- For Hyper-V, if there are multiple virtual machines with the same virtual machine name, PFM - RM for Virtual Machine might not be able to collect the correct information.
- For Hyper-V, any special character contained in the virtual machine name is converted to a different character and then stored in the VM Name field. The conversion rules are as follows:

Before conversion	After conversion
/ and \	-
# and *	_
(	[
)	]

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_VCI_VM_ID

PI_VCI_CPU_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,003 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VCI.	COPY	COPY	char (8)	No	--	Docker environment (Windows)
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Docker environment (Windows)
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Docker environment (Windows)
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Docker environment (Windows)
VM ID (VM_ID)	ID of the virtual machine. For Hyper-V, the root partition is set to Root.	COPY	FIXED	string (65)	No	No	Docker environment (Windows)
CPU ID (CPU_ID)	ID of the virtual CPU.	COPY	FIXED	string (32)	No	No	Docker environment (Windows)
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string(257)	No	Blank	Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string(257)	No	Blank	Docker environment (Windows)
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	Docker environment (Windows)
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*][#1]	COPY	FIXED	string (32)	No	Blank	Docker environment (Windows)
Used (USED)[#2]	Amount of used virtual CPU resources. CPU resources that could run a virtual machine on a virtual CPU. [Units: MHz]	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Docker environment
Insufficient (INSUFFICIENT)	Amount of virtual CPU insufficiency. CPU resources that could not run a virtual machine on a virtual CPU. [Units: MHz]	HILO	ADD	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Request (REQUEST)	Amount of virtual CPU requests. Resources required to run virtual machines. (This is the total of the amount of allocated CPUs added to the amount of unallocated CPUs.) [Units: MHz]	HILO	ADD	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Used % (USED_PERCENT)[#2]	Usage rate of virtual CPUs. Percentage of CPU resources that are allocated to a virtual CPU. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Docker environment (Linux): 0	Docker environment (Windows)

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Used % (USED_PERCENT)#2	Usage rate of virtual CPUs. Percentage of CPU resources that are allocated to a virtual CPU. [Units: %]	HILO	AVG	double	No	Podman environment: 0	Docker environment (Windows)
Insufficient % (INSUFFICIENT_PERCENT)	Insufficiency rate of virtual CPUs. Percentage of CPU resources that are unallocated to a virtual CPU. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Request % (REQUEST_PERCENT)	Request rate of virtual CPUs. Percentage of CPU resources for which virtual CPUs are requested. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Used Per Request (USED_PER_REQUEST)	Usage ratio of virtual CPUs. Percentage of the CPU resources, required to run a virtual machine, that are CPU resources that could run a virtual machine. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	Insufficiency rate for virtual CPUs. Percentage of the CPU resources, required to run a virtual machine, that are CPU resources that could not run a virtual machine. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Co-Stop (CO_STOP)	CPU conflict. CPU resources that the virtual machine could not run due to CPU conflict. [Units: MHz]	HILO	ADD	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Co-Stop % (CO_STOP_PERCENT)	CPU conflict rate. Percentage of CPU resources that the virtual machine could not run due to CPU conflict. [Units: %]	HILO	AVG	double	No	VMware: 0	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Co-Stop % (CO_STOP_PERCENT)	CPU conflict rate. Percentage of CPU resources that the virtual machine could not run due to CPU conflict. [Units: %]	HILO	AVG	double	No	VMware: 0	logical partitioning feature Docker environment Podman environment

#1

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#2

For Hyper-V, KVM, Docker environment, or Podman environment, this field requires performance data to be collected at least twice.

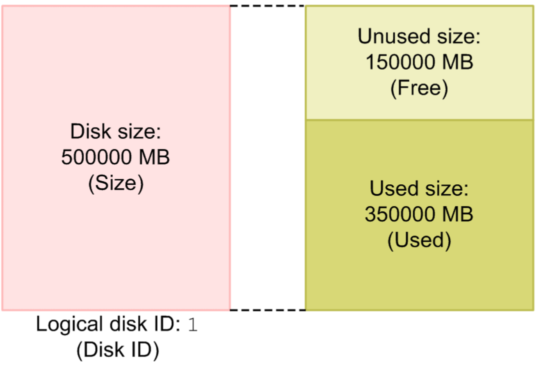
VM Logical Disk Status (PI_VLDI)

Function

This record stores performance data on a logical disk being used by a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as the size of the logical disk that can be used by a virtual machine. The disk space can be divided into used space and unused space. The figure below shows an example of data that can be collected in this record.

Figure 5–8: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.
- Information cannot be collected for a virtual machine on which `vmware-tools` is not running.
- Information cannot be collected for a logical volume that was created by using LVM (Logical Volume Manager) functions.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_VLDI_VM_ID`

`PI_VLDI_DISK_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,048 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup- ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VLDI.	COPY	COPY	char (8)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string(256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM ID (`VM_ID`)	ID of the virtual machine.	`COPY`	`FIXED`	`string` (65)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Disk ID (`DISK_ID`)	Disk ID.	`COPY`	`FIXED`	`string` (257)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Host Name (`VM_HOST_NAME`)	Host name of the virtual machine.	`COPY`	`FIXED`	`string`(257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Name (`VM_NAME`)	Name of the virtual machine.	`COPY`	`FIXED`	`string` (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Sampling Time (`SAMPLING_TIME`)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#	`COPY`	`FIXED`	`string` (32)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Sampling Time (`SAMPLING_TIME`)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#	`COPY`	`FIXED`	`string` (32)	No	Blank	Podman environment
Size (`SIZE`)	Disk size. [Units: MB]	`HILO`	`ADD`	`double`	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Used (`USED`)	Size of used space. [Units: MB]	`HILO`	`ADD`	`double`	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Free (`FREE`)	Size of free space. [Units: MB]	`HILO`	`ADD`	`double`	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Used % (`USED_PERCENT`)	Usage rate. [Units: %]	`HILO`	`AVG`	`double`	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Free % (FREE_PERCENT)	Non-usage rate of disk space. [Units: %]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

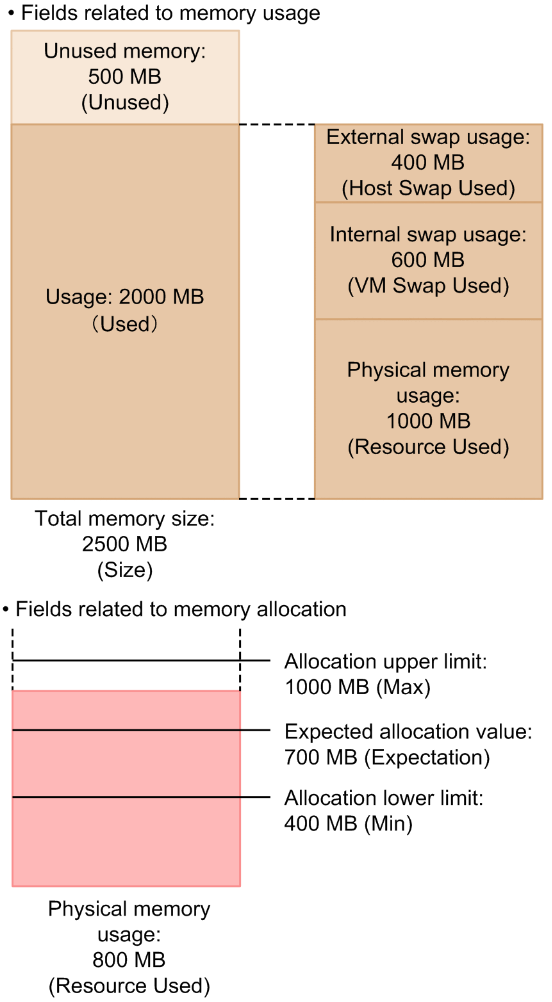
VM Memory Status (PI_VMI)

Function

This record stores performance data on virtual memory being used by a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as details about the memory being used by the virtual machine and memory allocation. You can also collect performance data on the swap that the virtual machine is using. The figure below shows an example of data that can be collected in this record.

Figure 5–9: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes

Item	Default value	Can it be changed?
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_VMI_VM_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 787 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VMI.	COPY	COPY	char (8)	No	--	logical partitioning feature
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	logical partitioning feature
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	logical partitioning feature
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string (256)	No	--	logical partitioning feature
VM ID (VM_ID)	ID of the virtual machine.	COPY	FIXED	string (65)	No	No	logical partitioning feature
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	logical partitioning feature
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format:	COPY	FIXED	string (32)	No	Blank	logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Sampling Time (SAMPLING_TIME)	*yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*][#]	COPY	FIXED	string (32)	No	Blank	logical partitioning feature
Size (SIZE)	Memory size. Memory size on the virtual machine. [Units: MB]	COPY	ADD	double	No	0	logical partitioning feature
Used (USED)	Amount of used memory. Amount of memory used by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	logical partitioning feature
Resource Used (RESOURCE_USED)	Amount of used memory resources. Memory resources used by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	logical partitioning feature Docker environment Podman environment
VM Swap Used (VM_SWAP_USED)	Amount of internal swap usage. Amount of swaps used by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Host Swap Used (HOST_SWAP_USED)	Amount of used external swaps. Amount of the swap area, on the host machine, used by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Unused (UNUSED)	Amount of unused memory. Size of free memory in the virtual machine. [Units: MB]	COPY	ADD	double	No	0	logical partitioning feature Docker environment
Used % (USED_PERCENT)	Memory usage rate. Percentage of used memory in the memory of the virtual machine. [Units: %]	COPY	AVG	double	No	0	logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Resource Used % (RESOURCE_USED_PERCENT)	Usage rate of memory resources. Percentage of memory resources used in the memory of the virtual machine. [Units: %]	COPY	AVG	double	No	0	logical partitioning feature Docker environment Podman environment
VM Swap Used % (VM_SWAP_USED_PERCENT)	Usage rate of internal swaps. Percentage of memory used for internal swaps in the memory of the virtual machine. [Units: %]	COPY	AVG	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Host Swap Used % (HOST_SWAP_USED_PERCENT)	Usage rate of external swaps. Percentage of memory used for external swaps in the memory of the virtual machine. [Units: %]	COPY	AVG	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
VM Swap IO (VM_SWAP_IO)	Internal swap I/Os. Amount of data internally swapped by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Swap In (VM_SWAP_IN)	Internal swap-in. Amount of data internally swapped in by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Swap Out (VM_SWAP_OUT)	Internal swap-out.	COPY	ADD	double	No	0	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM Swap Out (VM_SWAP_OUT)	Amount of data internally swapped out by the virtual machine. [Units: MB]	COPY	ADD	double	No	0	logical partitioning feature Docker environment Podman environment
Working Size (WORKING_SIZE)	Working set size. Amount for statistics about memory recently accessed within the virtual machine. [Units: MB]	COPY	ADD	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Working Size % (WORKING_SIZE_PERCENT)	Working set size rate. Percentage of all memory, on the virtual machine, that was recently accessed. [Units: %]	COPY	AVG	double	No	0	Hyper-V logical partitioning feature Docker environment Podman environment
Share (SHARE)	Memory allocation ratio. Reference for the balance point for memory allocation when multiple virtual machines require memory resources at the same time.	COPY	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Max (MAX)	Maximum value for memory allocation. The maximum value for memory resources allocated to a virtual machine. [Units: MB] -1: No limit.	COPY	ADD	double	No	0	Hyper-V logical partitioning feature
Min (MIN)	Minimum value for memory allocation. The minimum value for memory resources allocated to a virtual machine. [Units: MB]	COPY	ADD	double	No	0	Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Expectation (EXPECTATION)	Balance value for memory allocation. The memory resources allocated to memory when multiple virtual machines require memory resources at the same time. [Units: MB]	COPY	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Max % (MAX_PERCENT)	Maximum % for memory allocation. Percentage of the memory size, on the virtual machine, that is the maximum value for memory allocation. [Units: %]	COPY	AVG	double	No	0	Hyper-V logical partitioning feature
Min % (MIN_PERCENT)	Minimum % for memory allocation. Percentage of the memory size, on the virtual machine, that is the minimum value for memory allocation. [Units: %]	COPY	AVG	double	No	0	Hyper-V KVM logical partitioning feature
Expectation % (EXPECTATION_PERCENT)	Balance point for memory allocation. Percentage of memory resources that is the balance point for memory allocation when multiple virtual machines require memory resources at the same time. [Units: %]	COPY	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#

The value used in `[±`*hh*`:`*mm*`]` varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, `+09:00` is displayed. If the monitored virtual environment is operating according to UTC, `Z` is displayed.

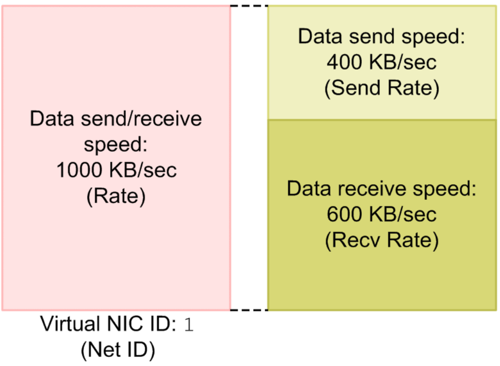
VM Network Status (PI_VNI)

Function

This record stores performance data on a virtual NIC being used by a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as the data send/receive speed of the virtual NIC. The data send/receive speed can be divided into data receive speed and data send speed. The figure below shows an example of data that can be collected in this record.

Figure 5–10: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.
- For Hyper-V, specific symbols included in a network identifier are converted to other symbols when the network identifier is stored in the `Net ID` field. The conversion rules are as follows:

Before conversion	After conversion
/ and \	-
# and *	_
(	[
)	]

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_VNI_VM_ID`

PI_VNI_NET_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 976 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VNI.	COPY	COPY	char (8)	No	--	--
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	--
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	--
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string (256)	No	--	--
VM ID (VM_ID)	ID of the virtual machine. For Hyper-V, the root partition is set to Root.	COPY	FIXED	string (65)	No	No	--
Net ID (NET_ID)	Network ID.	COPY	FIXED	string (257)	No	No	--
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	--
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#1	COPY	FIXED	string (32)	No	Blank	--
Rate (RATE)#2	Speed of transmission to and from the network by a virtual machine. [Units: KBps]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	--

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Rate (RATE)#2	Speed of transmission to and from the network by a virtual machine. [Units: KBps]	HILO	AVG	double	No	Docker environment: 0 Podman environment: 0	--
Send Rate (SEND_RATE)#2	Speed of sending to the network by a virtual machine. [Units: KBps]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Docker environment: 0 Podman environment: 0	--
Recv Rate (RECV_RATE)#2	Speed of receiving from the network by a virtual machine. [Units: KBps]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Docker environment: 0 Podman environment: 0	--

#1

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#2

For Hyper-V, KVM, Docker environment, or Podman environment this field requires performance data to be collected at least twice.

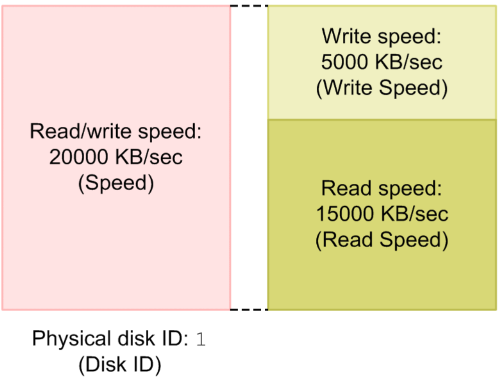
VM Physical Disk Status (PI_VPDI)

Function

This record stores performance data on a physical disk being used by a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as the data read/write speed of the physical disk being used by the virtual machine. For the data read/write speed, you can collect further details, that is, data write speed and data read speed. The figure below shows an example of data that can be collected in this record.

Figure 5–11: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.
- For Hyper-V, any special character contained in the disk name is converted to a different character and then stored in the Disk ID field. The conversion rules are as follows:

Before conversion	After conversion
# and \	_
(	[
)	]

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_VPDI_VM_ID

PI_VPDI_DISK_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,228 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VPDI.	COPY	COPY	char (8)	No	--	--
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	--
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	--
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string (256)	No	--	--
VM ID (VM_ID)	ID of the virtual machine.	COPY	FIXED	string (65)	No	No	--
Disk ID (DISK_ID)	ID of the physical disk (or file) used by the virtual machine. If the monitoring target is a Docker environment (Windows), the value is n/a.	COPY	FIXED	string (257)	No	No	--
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	--
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#1	COPY	FIXED	string (32)	No	Blank	--
Speed (SPEED)#2,#3	Data transfer speed. Speed of reads from or writes to the physical disk, made by the virtual machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Read Speed (READ_SPEED)#2,#3	Reading data transfer speed. Speed of reads from the physical disk, made by the virtual machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature
Write Speed (WRITE_SPEED)#2,#3	Writing data transfer speed. Speed of writes to the physical disk, made by the virtual machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature
Requests (REQUESTS)#3	Number of requests. • For VMware, KVM, Docker environment, or Podman environment Number of times that read/write requests from the virtual machine to the physical disk are processed. • For logical partitioning feature Number of interrupts from the HBA	HILO	ADD	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1 Docker environment: 0 Podman environment: 0	Hyper-V
Read Requests (READ_REQUESTS)#3	Number of read requests. Number of times that read requests, made by the virtual machine to the physical disk, are processed.	HILO	ADD	double	No	0	Hyper-V logical partitioning feature
Write Requests (WRITE_REQUESTS)#3	Number of write requests. Number of times that write requests, made by the virtual machine to the physical disk, are processed.	HILO	ADD	double	No	0	Hyper-V logical partitioning feature
Commands (COMMANDS)	Number of issued disk commands. Number of commands issued by the virtual machine to the physical disk.	HILO	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Abort Commands (ABORT_COMMANDS)	Number of aborted disk commands. Number of aborted commands made by the virtual machine to the physical disk.	HILO	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Abort Commands (ABORT_COMMANDS)	Number of aborted disk commands. Number of aborted commands made by the virtual machine to the physical disk.	HILO	ADD	double	No	0	Podman environment
Abort Commands % (ABORT_COMMANDS_PERCENT)	Aborted disk command rate. Percentage of all commands, made by the virtual machine to a physical disk, that are aborted.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Bus Resets (BUS_RESETS)	Number of bus resets. Number of times the bus was reset for a disk on the virtual machine.	HILO	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#1

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#2

For Hyper-V, this field requires performance data to be collected at least twice.

#3

For KVM, Docker environment, or Podman environment this field requires performance data to be collected at least twice.

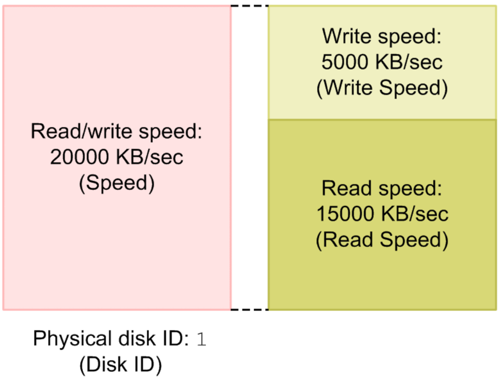
VM Virtual Disk Status (PI_VVDI)

Function

This record stores performance data on a virtual disk being used by a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as the data read/write speed of the virtual disk being used by the virtual machine. For the data read/write speed, you can collect further details, that is, data write speed and data read speed. The figure below shows an example of data that can be collected in this record.

Figure 5–12: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_VVDI_VM_ID`

`PI_VVDI_DISK_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,300 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsupported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VVDI.	COPY	COPY	char(8)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (VADEVICEID)	Device ID of a monitored host.	COPY	COPY	string(256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM ID (VM_ID)	ID of the virtual machine.	COPY	FIXED	string(65)	No	No	Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM ID (VM_ID)	ID of the virtual machine.	COPY	FIXED	string(65)	No	No	Docker environment Podman environment
Disk ID (DISK_ID)	Disk ID. ID of the virtual disk connected to the virtual machine.	COPY	FIXED	string(257)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string(257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string(257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#	COPY	FIXED	string(32)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Speed (SPEED)	Data transfer speed.	HILO	AVG	double	No	0	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Speed (SPEED)	Speed of reads from or writes to the virtual disk, made by the virtual machine. [Units: KBps]	HILO	AVG	double	No	0	logical partitioning feature Docker environment Podman environment
Read Speed (READ_SPEED)	Reading data transfer speed. Speed of reads from the virtual disk, made by the virtual machine. [Units: KBps]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Write Speed (WRITE_SPEED)	Writing data transfer speed. Speed of writes to the virtual disk, made by the virtual machine. [Units: KBps]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Requests Per Sec (REQUESTS_PER_SEC)	Number of requests per second. Number of times that read/write requests, made by the virtual machine to the virtual disk, are processed.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	Number of read requests per second. Number of times that read requests, made by the virtual machine to the virtual disk, are processed. Read request count.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	Number of write requests per second. Number of times that write requests, made by the virtual machine to the virtual disk, are processed.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Total Latency (TOTAL_LATENCY)	Average amount of total latency. Average amount of time for a read/write operation from the virtual disk. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Total Read Latency (TOTAL_READ_LATENCY)	Average amount of total read latency. Average amount of time for a read operation from the virtual disk. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Total Write Latency (TOTAL_WRITE_LATENCY)	Average amount of total write latency. Average amount of time for a write operation from the virtual disk. [Unit: milliseconds]	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Outstanding Requests (OUTSTANDING_REQUESTS)	Average amount of outstanding requests. Average number of outstanding read/write requests to the virtual disk.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Outstanding Requests (OUTSTANDING_REQUESTS)	Average amount of outstanding requests. Average number of outstanding read/write requests to the virtual disk.	HILO	AVG	double	No	0	Podman environment
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	Average amount of outstanding read requests. Average number of outstanding read requests to the virtual disk.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	Average amount of outstanding write requests. Average number of outstanding write requests to the virtual disk.	HILO	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

VM Status Detail (PD_VM)

Function

This record stores performance data that indicates the status of a virtual machine at a given point in time. This record is a multi-instance record.

Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- In the case of logical partitioning feature, information on LPARs that are not activated can not be collected.
- You cannot collect the `VM Host Name` field from a virtual machine that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PD_VM_VM_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 663 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `VM`.	--	`COPY`	`char` (8)	No	--	--
Record Time (`RECORD_TIME`)	The time when the record was created.	--	`COPY`	`time_t`	No	--	--
Interval (`INTERVAL`)	This is always `0`.	--	`FIXED`	`ulong`	No	--	--
VA DeviceID (`VADEVICEID`)	The device ID of a monitored host.	--	`COPY`	`string` (256)	No	--	--

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM ID (VM_ID)	ID of the virtual machine. For Hyper-V, the root partition is set to Root.	--	FIXED	string (65)	No	No	--
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	--	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature
VM Name (VM_NAME)	Name of the virtual machine.	--	FIXED	string (257)	No	Blank	--
Information (INFORMATION)	Information about the virtual machine. For logical partitioning feature, when Status of the associated logical partition is set to anything other than ON (activated), or when a program whose OS type cannot be identified is running, the value is always set to blank and cannot be obtained. Information about the virtual machine such as OS type. For VMware Version information of the Guest OS specified for the virtual machine. For logical partitioning feature When Status of the associated logical partition is set to anything other than ON (activated), or when a program whose OS type cannot be identified is running, the value is always set to blank and cannot be obtained. For Docker environment The command name of the container. For Podman environment The command name of the container.	--	FIXED	string (64)	No	Blank	Hyper-V
Status (STATUS)	Virtual machine status. For VMware: • ON: Running • OFF: Stopped • SUSPENDED: Stopped temporarily • UNKNOWN: Unknown status	--	FIXED	string (16)	No	Blank	--

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Status (STATUS)	For Hyper-V: • ON: Running • OFF: Stopped • PAUSED: Paused • SUSPENDED: Stopped temporarily • STARTING: Under start processing • SNAPSHOTTING: Obtaining a snapshot • SAVING: Save processing underway • STOPPING: Stop processing underway • PAUSING: Paused • RESUMING: Resuming • UNKNOWN: Unknown status For KVM: • ON: Running • OFF: Stopped • PAUSED: Paused • STOPPING: Stop processing underway • CRASHED: Faulty • UNKNOWN: Unknown status For logical partitioning feature: • ON: Activated • OFF: Deactivated • ACTPEND: Now activating • DEACTPEND: Now deactivating • STANDBY: On standby • MIGRATION: Now migrating • FAIL: Blocked For Docker environment • ON: Running • OFF: Stopped • PAUSED: Paused • STARTING: Under start processing • DEAD: Abnormally terminated • OOMKILLED: OOM terminated • UNKNOWN: Unknown status	--	FIXED	string (16)	No	Blank	--

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Status (STATUS)	For Podman environment • ON: Running • OFF: Stopped • PAUSED: Paused • STARTING: Under start processing • DEAD: Abnormally terminated • OOMKILLED: OOM terminated • UNKNOWN: Unknown status	--	FIXED	string (16)	No	Blank	--
Snapshot (SNAPSHOT)	Whether or not snapshots are taken. For VMware • 0: No snapshots • 1: Snapshots taken	--	FIXED	ulong	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

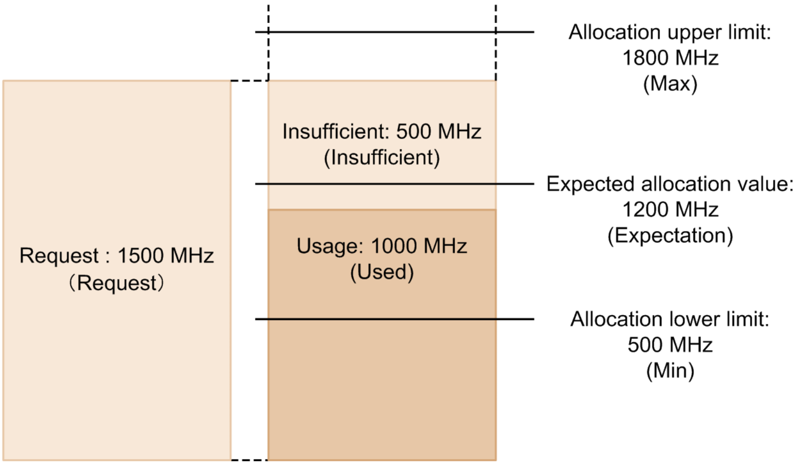
VM Status (PI_VI)

Function

This record stores performance data on a virtual machine over a unit of time. This record is a multi-instance record.

In this record, you can collect performance data such as CPU usage by the virtual machine, CPU insufficiency, and CPU allocation. The figure below shows an example of data that can be collected in this record.

Figure 5–13: Example of collected data



Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a virtual machine that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`Yes`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_VI_VM_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,223 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VI.	COPY	COPY	char (8)	No	--	--
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	--
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	--
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string (256)	No	--	--
VM ID (VM_ID)	ID of the virtual machine. For Hyper-V, the root partition is set to Root.	COPY	FIXED	string (65)	No	No	--
Clocks (CLOCKS)#1	Clock frequency of the CPU resources. The value that is the total of the clock frequencies of the physical CPUs allocated to the virtual machine. [Units: MHz]	COPY	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment
Count (COUNT)	Number of physical CPU cores in the CPU resources. The number of cores of the physical CPUs allocated to the virtual machine.	COPY	ADD	long	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Docker environment: 0	Podman environment
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string(257)	No	Blank	Hyper-V KVM logical partitioning feature
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	--
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*]#2	COPY	FIXED	string (32)	No	Blank	--
Used (USED)#3	CPU usage amount. CPU resources that could run a virtual machine on a physical CPU.	HILO	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0	Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Used (USED)#3	[Units: MHz]	HILO	ADD	double	No	logical partitioning feature: -1 Podman environment: 0	Docker environment
Insufficient (INSUFFICIENT)	Amount of CPU insufficiency. CPU resources that could not run a virtual machine on a physical CPU. [Units: MHz]	HILO	ADD	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Request (REQUEST)	Amount of CPU requests. The amount of CPU resources required to run a virtual machine. (This is the total allocated resources added to the total unallocated resources.) [Units: MHz]	HILO	ADD	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Host Used % (HOST_USED_PERCENT)	Usage rate of host CPUs. Percentage of total physical CPU resources that were executed on a physical CPU. [Units: %]	HILO	AVG	double	No	VMware: 0 KVM: 0 logical partitioning feature: -1 Podman environment: 0	Hyper-V Docker environment
Used % (USED_PERCENT)#1,#3,#4#5	CPU usage rate. Percentage of CPU resources that could run a virtual machine on a physical CPU. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1 Docker environment: 0 Podman environment: 0	--
Insufficient % (INSUFFICIENT_PERCENT)#1	CPU insufficiency rate. Percentage of CPU resources that could not run a virtual machine on a physical CPU. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Request % (REQUEST_PERCENT)#1	CPU request rate. Percentage of CPU resources required to run a virtual machine. (This is the total of the CPU	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Request % (REQUEST_PERCENT)#1	allocation rate added to the CPU unallocation rate.) [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Podman environment
Used Per Request (USED_PER_REQUEST)	% of allocatable CPU resources (= amount of CPU usage/amount of CPU requests). Percentage of the CPU resources required to run a virtual machine, that could run a virtual machine. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	Rate of unallocated CPUs (= amount of CPU insufficiency/ amount of CPU requests). Percentage of the CPU resources required to run a virtual machine, that could not run a virtual machine. [Units: %]	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Affinity (AFFINITY)	Physical CPU that can be moved. The Affinity setting specifies that a physical CPU can be allocated to an individual virtual machine. (Ex: If a virtual machine can be allocated to physical CPU 1 and 2, the value of this field is 1,2.)	COPY	FIXED	string (32)	No	Blank	Hyper-V logical partitioning feature Podman environment
Share (SHARE)	CPU allocation ratio. A value used to request the CPU resources to be allocated to virtual machines. If two or more virtual machines require a large amount of CPU resources at the same time, the CPU resources are allocated according to this ratio. For Hyper-V, the root partition information cannot be obtained. For logical partitioning feature, when the CPU of the associated logical partition is running in dedicated mode, the value is always set to -1 and cannot be obtained.	COPY	AVG	double	No	VMware: 0 Hyper-V: 0 logical partitioning feature: -1 Docker environment: 0	KVM Podman environment
Max (MAX)	Maximum value for CPU allocation. The maximum value for CPUs allocatable to a virtual machine. [Units: MHz] -1: No limit. For Hyper-V, the root partition information cannot be obtained.	COPY	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Max (MAX)	For logical partitioning feature, -1 indicates that the value cannot be obtained.	COPY	ADD	double	No	VMware: 0 Hyper-V: 0 KVM: 0 logical partitioning feature: -1	Docker environment Podman environment
Min (MIN)	Minimum value for CPU allocation. The minimum value for CPUs allocatable to a virtual machine. [Units: MHz] For Hyper-V, the root partition information cannot be obtained.	COPY	ADD	double	No	0	KVM logical partitioning feature Docker environment Podman environment
Expectation (EXPECTATION)	Balance value for CPU allocation. If two or more virtual machines require a large amount of resources at the same time, this value estimates what CPU resources can be expected to be available for the virtual machines. [Units: MHz] For logical partitioning feature, when the CPU of the associated logical partition is running in dedicated mode, the value is always set to -1 and cannot be obtained.	HILO	ADD	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Docker environment Podman environment
Max % (MAX_PERCENT)	Maximum % for CPU allocation. Percentage of CPU resources that is the maximum allocatable to a virtual machine. [Units: %]	HILO	AVG	double	No	VMware: 0 Hyper-V: 0 logical partitioning feature: -1	KVM Docker environment Podman environment
Min % (MIN_PERCENT)	Minimum % for CPU allocation. The minimum percentage of CPU resources that are allocatable to a virtual machine. [Units: %]	HILO	AVG	double	No	0	KVM logical partitioning feature Docker environment Podman environment
Expectation % (EXPECTATION_PERCENT)#4	Balance point rate for CPU allocation. Percentage of CPU resources that can be expected to be used for virtual machines.	HILO	AVG	double	No	VMware: 0 logical partitioning feature: -1	Hyper-V KVM Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Expectation % (EXPECTATION_PERCENT)#4	[Units: %] For logical partitioning feature, when the CPU of the associated logical partition is running in dedicated mode, the value is always set to -1 and cannot be obtained.	HILO	AVG	double	No	Docker environment: 0	Hyper-V KVM Podman environment
Snapshot (SNAPSHOT)	Whether or not snapshots are taken. For VMware • 0: No snapshots • 1: Snapshots taken	--	FIXED	ulong	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Co-Stop (CO_STOP)	CPU conflict. CPU resources that the virtual machine could not run due to CPU conflict. [Units: MHz]	HILO	ADD	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Co-Stop % (CO_STOP_PERCENT)	CPU conflict rate. Percentage of CPU resources that the virtual machine could not run due to CPU conflict. [Units: %]	HILO	AVG	double	No	VMware: 0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#1

The value changes in accordance with the UseVcpuMax value set as instance information as follows:

N: Clock frequency of the physical CPU **x** number of cores

Y: Clock frequency of the CPU allocated to the virtual machine

#2

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

#3

For Hyper-V or KVM, this field requires performance data to be collected at least twice.

#4

For Docker environment, this field requires performance data to be collected at least twice.

#5

For Podman environment, this field requires performance data to be collected at least twice.

Host Generic Data Detail (PD_HGDD)

Function

This record stores performance data that indicates the status of a physical server at a given point in time. This record is a multi-instance record.

This record collects performance data defined by the user.

Notes:

- If an attempt to connect to the monitored physical server fails, information cannot be collected.
- If the counter name for which data is collected does not exist, the record will not be created.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PD_HGDD_SECTION_NAME`

`PD_HGDD_DATA_NAME`

`PD_HGDD_OBJECT_NAME`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 620 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `HGDD`.	--	`COPY`	`char` (8)	No	--	Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always HGDD.	--	COPY	char (8)	No	--	Docker environment Podman environment
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (INTERVAL)	This is always 0.	--	FIXED	ulong	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string (256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Section Name (SECTION_NAME)	The section name. The section name specified in the user-defined record definition file.	--	COPY	string (33)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Data Name (DATA_NAME)	The data name.	--	COPY	string (65)	No	--	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Data Name (DATA_NAME)	The counter name specified in the user-defined record definition file.	--	COPY	string (65)	No	--	logical partitioning feature Docker environment Podman environment
Object Name (OBJECT_NAME)	The object name. The object name of the collected performance data.	--	COPY	string (257)	No	n/a	Hyper-V KVM logical partitioning feature Docker environment Podman environment
String Data (STRING_DATA)	Performance data of a string type.	--	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Double Data (DOUBLE_DATA)	Performance data of a numeric type.	--	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

Host Generic Data Interval (PI_HGDI)

Function

This record stores performance data on a physical server over a unit of time. This record is a multi-instance record.

This record collects performance data defined by the user.

Notes:

- If an attempt to connect to the monitored physical server fails, information cannot be collected.
- If the counter name for which data is collected does not exist, the record will not be created.

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_HGDI_SECTION_NAME

PI_HGDI_DATA_NAME

PI_HGDI_OBJECT_NAME

Lifetime

None

Record size

- Fixed portion: 969 bytes
- Variable portion: 664 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always HGDI.	COPY	COPY	char (8)	No	--	Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `HGDI`.	`COPY`	`COPY`	`char` (8)	No	--	Podman environment
Record Time (`RECORD_TIME`)	The time when the record was created.	`COPY`	`COPY`	`time_t`	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (`INTERVAL`)	Interval during which the information is collected. [Units: seconds]	`COPY`	`FIXED`	`ulong`	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (`VADEVICEID`)	The device ID of a monitored host.	`COPY`	`COPY`	`string` (256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Section Name (`SECTION_NAME`)	The section name. The section name specified in the user-defined record definition file.	`COPY`	`COPY`	`string` (33)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Data Name (`DATA_NAME`)	The data name. The counter name specified in the user-defined record definition file.	`COPY`	`COPY`	`string` (65)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Object Name (`OBJECT_NAME`)	The object name. The object name of the collected performance data.	`COPY`	`COPY`	`char` (257)	No	n/a	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Object Name (`OBJECT_NAME`)	The object name. The object name of the collected performance data.	`COPY`	`COPY`	`char` (257)	No	n/a	logical partitioning feature Docker environment Podman environment
Sampling Time (`SAMPLING_TIME`)	Sampling time. The time when performance data is collected on the monitored host. Displayed in the following format: *yyyy*`-`*mm*`-`*dd*`T`*hh*`:`*mm*`[`±*hh*`:`*mm*`]`#	`COPY`	`FIXED`	`string` (32)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
String Data (`STRING_DATA`)	Performance data of a string type.	`COPY`	`FIXED`	`string` (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Double Data (`DOUBLE_DATA`)	Performance data of a numeric type.	`AVG`	`AVG`	`double`	No	`0`	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#

The value used in `[`±*hh*`:`*mm*`]` varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, `+09:00` is displayed. If the monitored virtual environment is operating according to UTC, `Z` is displayed.

VM Generic Data Detail (PD_VGDD)

Function

This record stores performance data that indicates the status of a virtual machine at a given point in time. This record is a multi-instance record.

This record collects performance data of a virtual machine.

Notes:

- If an attempt to connect to the monitored physical server fails, information cannot be collected.
- Information cannot be collected for a virtual machine that is not running.
- If the counter name for which data is collected does not exist, the record will not be created.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PD_VGDD_VM_ID`

`PD_VGDD_SECTION_NAME`

`PD_VGDD_DATA_NAME`

`PD_VGDD_OBJECT_NAME`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,199 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `VGDD`.	--	`COPY`	`char` (8)	No	--	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VGDD.	--	COPY	char (8)	No	--	logical partitioning feature Docker environment Podman environment
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (INTERVAL)	This is always 0.	--	FIXED	ulong	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string (256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM ID (VM_ID)	ID of the virtual machine.	--	FIXED	string (65)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	--	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM Name (VM_NAME)	Name of the virtual machine.	--	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Section Name (SECTION_NAME)	The section name. The section name specified in the user-defined record definition file.	--	COPY	string (33)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Data Name (DATA_NAME)	The data name. The counter name specified in the user-defined record definition file.	--	COPY	string (65)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Object Name (OBJECT_NAME)	The object name. The object name of the collected performance data.	--	COPY	char (257)	No	n/a	Hyper-V KVM logical partitioning feature Docker environment Podman environment
String Data (STRING_DATA)	Performance data of a string type.	--	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Double Data (DOUBLE_DATA)	Performance data of a numeric type.	--	AVG	double	No	0	Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsupported
Double Data (DOUBLE_DATA)	Performance data of a numeric type.	--	AVG	double	No	0	Docker environment Podman environment

VM Generic Data Interval (PI_VGDI)

Function

This record stores performance data on a virtual machine over a unit of time. This record is a multi-instance record.

This record collects performance data defined by the user.

Notes:

- If an attempt to connect to the monitored physical server fails, information cannot be collected.
- Information cannot be collected for a virtual machine that is not running.
- If the counter name for which data is collected does not exist, the record will not be created.

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_VGDI_VM_ID

PI_VGDI_SECTION_NAME

PI_VGDI_DATA_NAME

PI_VGDI_OBJECT_NAME

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,243 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VGDI.	COPY	COPY	char (8)	No	--	Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VGDI.	COPY	COPY	char (8)	No	--	Docker environment Podman environment
Record Time (RECORD_TIME)	The time when the record was created.	COPY	COPY	time_t	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	COPY	FIXED	ulong	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	COPY	COPY	string (256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM ID (VM_ID)	ID of the virtual machine.	COPY	FIXED	string (65)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Name (VM_NAME)	Name of the virtual machine.	COPY	FIXED	string (257)	No	Blank	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM Name (`VM_NAME`)	Name of the virtual machine.	`COPY`	`FIXED`	`string` (257)	No	Blank	logical partitioning feature Docker environment Podman environment
Sampling Time (`SAMPLING_TIME`)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*`T`*hh*:*mm*[±*hh*:*mm*]#	`COPY`	`FIXED`	`string` (32)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Section Name (`SECTION_NAME`)	The section name. The section name specified in the user-defined record definition file.	`COPY`	`COPY`	`string` (33)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Data Name (`DATA_NAME`)	The data name. The counter name specified in the user-defined record definition file.	`COPY`	`COPY`	`string` (65)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Object Name (`OBJECT_NAME`)	The object name. The object name of the collected performance data.	`COPY`	`COPY`	`string` (257)	No	n/a	Hyper-V KVM logical partitioning feature Docker environment Podman environment
String Data (`STRING_DATA`)	Performance data of a string type.	`COPY`	`FIXED`	`string` (257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsupported
Double Data (DOUBLE_DATA)	Performance data of a numeric type.	AVG	AVG	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

#

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

VM Virtual Disk Detail (PD_VDKD)

Function

This record stores performance data concerning the status of a virtual machine at a certain point in time. This record is a multiple-instance record.

This record allows you to acquire performance data, such as the size of the virtual disk space allocated to the virtual machine.

Notes:

- If an attempt to connect to the physical server to be monitored fails, no information can be collected.
- You cannot collect the `VM Host Name` field from a virtual machine that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	`300`	Yes
Collection Offset	`0`	Yes
Log	`No`	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	`No`	No

ODBC key field

`PD_VDKD_DATASTORE_ID`

`PD_VDKD_DATASTORE_NAME`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,110 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `VDKD`.	--	`COPY`	`char` (8)	No	--	Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always VDKD.	--	COPY	char (8)	No	--	Podman environment
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Interval (INTERVAL)	This is always 0.	--	FIXED	ulong	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string (256)	No	--	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Datastore ID (DATASTORE_ID)	Identifier of the datastore.	--	COPY	string (128)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Datastore Name (DATASTORE_NAME)	Name of the datastore.	--	COPY	string (257)	No	No	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM ID (VM_ID)	Identifier of the virtual machine.	--	FIXED	string(65)	No	Blank	Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
VM ID (VM_ID)	Identifier of the virtual machine.	--	FIXED	string(65)	No	Blank	logical partitioning feature Docker environment Podman environment
VM Host Name (VM_HOST_NAME)	Host name of the virtual machine.	--	FIXED	string(257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
VM Name (VM_NAME)	Name of the virtual machine.	--	FIXED	string(257)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Controller Name (CONTROLLER_NAME)	Name of the controller.	--	FIXED	string(65)	No	Blank	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Bus Number (BUS_NUMBER)	Bus number associated with the controller.	--	FIXED	long	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment
Unit Number (UNIT_NUMBER)	Unit number on the controller.	--	FIXED	long	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Disk UUID (DISK_UUID)	UUID of the virtual disk.	--	FIXED	string(65)	No	Blank	VMware (ESXi 5.5, 6.0) Hyper-V KVM logical partitioning feature Docker environment Podman environment
Capacity (CAPACITY)	Capacity allocated to the virtual disk [Units: MB]	--	ADD	double	No	0	Hyper-V KVM logical partitioning feature Docker environment Podman environment

Pod Status Detail (PD_PODD)

Function

This record stores performance data that indicates the status of a Pod at a given point in time. This record is a multi-instance record.

Note:

Information cannot be collected if an attempt to connect to the monitored physical server fails.

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PD_PODD_POD_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 342 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always PODD.	--	COPY	char (8)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	VMware Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	logical partitioning feature Docker environment
Interval (INTERVAL)	This is always 0.	--	FIXED	ulong	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string (256)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
POD ID (POD_ID)	ID of the Pod.	--	FIXED	string (65)	No	No	VMware Hyper-V KVM logical partitioning feature Docker environment
POD Name (POD_NAME)	Name of the Pod.	--	FIXED	string (257)	No	Blank	VMware Hyper-V KVM logical partitioning feature Docker environment
Status (STATUS)	Pod status. • CREATE: Create • EXITED: Exited • PAUSED: Paused • ON: Running • OFF: Stopped • UNKNOWN: Unknown status	--	FIXED	string (16)	No	Blank	VMware Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Container Count (CONTAINER_COUNT)	Number of containers that exist in the pod.	--	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment

Pod Status Interval (PI_PODI)

Function

This record stores performance data on a Pod over a unit of time. This record is a multi-instance record.

Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a Pod that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
Collection Interval	300	Yes
Collection Offset	0	Yes
Log	No	Yes
LOGIF	Blank	Yes
Over 10 Sec Collection Time	No	No

ODBC key field

PI_PODI_POD_ID

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 822 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (INPUT_RECORD_TYPE)	The record name. This is always PODI.	--	COPY	char (8)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	VMware Hyper-V KVM

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	logical partitioning feature Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	--	FIXED	ulong	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string (256)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
POD ID (POD_ID)	ID of the Pod.	--	FIXED	string (65)	No	No	VMware Hyper-V KVM logical partitioning feature Docker environment
POD Name (POD_NAME)	Name of the Pod.	--	FIXED	string (257)	No	Blank	VMware Hyper-V KVM logical partitioning feature Docker environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*] #1	COPY	FIXED	string (32)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
CPU Used (CPU_USED)	CPU usage. CPU resources that the container in the pod could execute on the physical CPU. [Units: MHz]	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
CPU Used % (CPU_USED_PERCENT)	CPU usage rate. Percentage of CPU resources that the container in the pod could execute on the physical CPU. [Units: %]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Memory Used (MEMORY_USED)	Memory usage volume. Memory usage by the container in the pod. [Units: MB]	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Memory Used % (MEMORY_USED_PERCENT)	Memory usage rate. Percentage of memory resources that the container in the pod uses on the physical server. [Units: %]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Speed (DISK_SPEED)	Data transfer speed. Speed of reads from or writes to the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Read Speed (DISK_READ_SPEED)	Reading data transfer speed. Speed of reads from the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Disk Read Speed (DISK_READ_SPEED)	Reading data transfer speed. Speed of reads from the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	Docker environment
Disk Write Speed (DISK_WRITE_SPEED)	Writing data transfer speed. Speed of writes to the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Requests (DISK_REQUESTS)	Number of requests. Number of times that read/write requests from the container in the pod to the physical disk are processed.	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Read Requests (DISK_READ_REQUESTS)	Number of read requests. Number of times that read requests by the container in the pod from the physical disk are processed.	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Write Requests (DISK_WRITE_REQUESTS)	Number of write requests. Number of times that write requests by the container in the pod to the physical disk are processed.	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Network Rate (NETWORK_RATE)	Speed of transmission to and from the network by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Network Send Rate (NETWORK_SEND_RATE)	Speed of sending to the network by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Network Recv Rate (NETWORK_RECV_RATE)	Speed of receiving from the network by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment

#

The value used in [±*hh*:*mm*] varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, +09:00 is displayed. If the monitored virtual environment is operating according to UTC, Z is displayed.

Pod Container Status Interval (PI_POCI)

Function

This record stores performance data on a Container over a unit of time. This record is a multi-instance record.

Note:

- Information cannot be collected if an attempt to connect to the monitored physical server fails.
- Information cannot be collected for a Container that is not running.

Default values and values that can be changed

Item	Default value	Can it be changed?
`Collection Interval`	`300`	Yes
`Collection Offset`	`0`	Yes
`Log`	`No`	Yes
`LOGIF`	Blank	Yes
`Over 10 Sec Collection Time`	`No`	No

ODBC key field

`PI_POCI_POD_ID`

`PI_POCI_CONTAINER_ID`

Lifetime

None

Record size

- Fixed portion: 937 bytes
- Variable portion: 1,144 bytes

Fields

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Type (`INPUT_RECORD_TYPE`)	The record name. This is always `POCI`.	--	`COPY`	`char` (8)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
Record Time (`RECORD_TIME`)	The time when the record was created.	--	`COPY`	`time_t`	No	--	VMware Hyper-V

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Record Time (RECORD_TIME)	The time when the record was created.	--	COPY	time_t	No	--	KVM logical partitioning feature Docker environment
Interval (INTERVAL)	Interval during which the information is collected. [Units: seconds]	--	FIXED	ulong	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
VA DeviceID (VADEVICEID)	The device ID of a monitored host.	--	COPY	string (256)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
POD ID (POD_ID)	ID of the Pod.	--	FIXED	string (65)	No	No	VMware Hyper-V KVM logical partitioning feature Docker environment
POD Name (POD_NAME)	Name of the Pod.	--	FIXED	string (257)	No	Blank	VMware Hyper-V KVM logical partitioning feature Docker environment
Container ID (CONTAINER_ID)	ID of the Container.	COPY	FIXED	string (65)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Container Name (CONTAINER_NAME)	Name of the Container.	COPY	FIXED	string (257)	No	Blank	VMware Hyper-V KVM logical partitioning feature Docker environment
Sampling Time (SAMPLING_TIME)	Sampling time. The time when the performance information is collected on the host machine. Displayed in the following format: *yyyy*-*mm*-*dd*T*hh*:*mm*[±*hh*:*mm*][#1]	COPY	FIXED	string (32)	No	--	VMware Hyper-V KVM logical partitioning feature Docker environment
CPU Used (CPU_USED)	CPU usage. CPU resources that the container could execute on the physical CPU. [Units: MHz]	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
CPU Used % (CPU_USED_PERCENT)	CPU usage rate. Percentage of CPU resources that the container in the pod could execute on the physical CPU. [Units: %]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Memory Used (MEMORY_USED)	Memory usage volume. Memory usage by the container in the pod. [Units: MB]	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Memory Used % (MEMORY_USED_PERCENT)	Memory usage rate. Percentage of memory resources that the container in the pod uses on the physical server. [Units: %]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
Memory Used % (MEMORY_USED_PERCENT)	Memory usage rate. Percentage of memory resources that the container in the pod uses on the physical server. [Units: %]	HILO	AVG	double	No	0	Docker environment
Disk Speed (DISK_SPEED)	Data transfer speed. Speed of reads from or writes to the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Read Speed (DISK_READ_SPEED)	Reading data transfer speed. Speed of reads from the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Write Speed (DISK_WRITE_SPEED)	Writing data transfer speed. Speed of writes to the physical disk, made by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Requests (DISK_REQUESTS)	Number of requests. Number of times that read/write requests from the container in the pod to the physical disk are processed.	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Read Requests (DISK_READ_REQUESTS)	Number of read requests. Number of times that read requests by the container in the pod from the physical disk are processed.	HILO	ADD	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Disk Write Requests	Number of write requests. Number of times that write	HILO	ADD	double	No	0	VMware Hyper-V

PFM - View name (PFM - Manager name)	Explanation	Summary	Grouping	Format	Delta	When not collected	Unsup-ported
(DISK_WRITE_REQUESTS)	requests by the container in the pod to the physical disk are processed.	HILO	ADD	double	No	0	KVM logical partitioning feature Docker environment
Network Rate (NETWORK_RATE)	Speed of transmission to and from the network by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Network Send Rate (NETWORK_SEND_RATE)	Speed of sending to the network by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment
Network Recv Rate (NETWORK_RECV_RATE)	Speed of receiving from the network by the container in the pod. [Units: KB/sec]	HILO	AVG	double	No	0	VMware Hyper-V KVM logical partitioning feature Docker environment

#

The value used in `[±`*hh*`:`*mm*`]` varies depending on the time zone of the monitored host. For example, if the monitored virtual environment is operating according to JST, `+09:00` is displayed. If the monitored virtual environment is operating according to UTC, `Z` is displayed.

6 Messages

This chapter explains the format of PFM - RM for Virtual Machine messages, the output list, the Windows event log list, and the message list.

6.1 Message format

This section explains the format of the messages that are output by PFM - RM for Virtual Machine, and the format this manual uses to describe them.

6.1.1 Message output format

The following explains the format of the messages that are output by PFM - RM for Virtual Machine. A message consists of a message ID followed by message text. The format is shown below.

```
KAVLnnnnn-Y message-text
```

The parts of the message ID indicate the following:

`K`

Indicates a system identifier.

`AVL`

Indicates a PFM - RM message.

nnnnn

Message serial number. The message number for PFM - RM for Virtual Machine has the `20`*xxx* format.

Y

Indicates a message type.

- `E`: Error
  Processing is interrupted.
- `W`: Warning
  Following the message output, processing continues.
- `I`: Information
  Indicates information for the user.
- `Q`: Response
  Prompts the user to respond.

The correspondence between message type and Windows Event Log type is shown below.

`-E`

- Level: Error
- Meaning: Error message

`-W`

- Level: Warning
- Meaning: Warning message

`-I`

- Level: Information
- Meaning: Additional information message

`-Q`

(Not output)

6.1.2 Message description format

This subsection explains the message description format used in this manual. The bolded area within message text means that the content to be displayed varies depending on the situation in which the message is issued. Messages are listed in numerical order by message ID. A description format example follows.

Message ID

Message text

Message explanation

(S)

Indicates an action taken by the system.

(O)

Indicates the action to be taken by the operator when the message is displayed.

> **Note**
>
> When notified by an operator, the system administrator must collect log information by referencing *7. Error Handling Procedures*, and must perform an initial investigation.
>
> During an initial investigation of the problem cause, refer to the OS log information (Windows Event Log) and the various types of log information that are output by PFM - RM for Virtual Machine. Use this log information to view the details for the time period in which the problem occurred, and then take action to solve the problem or avoid its recurrence. Also record the operating procedures that led to the problem. At the same time, check for reproducibility as much as possible.

The < > symbols within the message text indicate embedded characters. For example, if the host name is `hostA` and the message text shows `host=`<*host-name*>, `host= hostA` is output.

6.2 Message output destinations

This section lists the output destinations of the messages that are output by PFM - RM for Virtual Machine.

The following legend applies to the table below:

Legend:
- stdout: Standard output
- stderr: Standard error
- Y: The message is output.
- N: The message is not output.

Table 6–1: List of output destinations for PFM - RM for Virtual Machine messages

Message ID	Output destination					
	Windows Event Log	Common message log	stdout	stderr	JP1 system event#1	Agent event#2
KAVL20000-I	Y	Y	N	N	N	N
KAVL20001-E	Y	Y	N	N	N	N
KAVL20002-I	Y	Y	N	N	N	N
KAVL20004-E	Y	Y	N	N	N	N
KAVL20005-E	Y	Y	N	N	N	N
KAVL20006-E	Y	Y	N	N	Y	Y
KAVL20007-W	N	Y	N	N	Y	Y
KAVL20008-W	N	Y	N	N	Y	Y
KAVL20009-W	N	Y	N	N	Y	Y
KAVL20010-E	N	Y	N	N	N	N
KAVL20011-W	N	Y	N	N	Y	Y
KAVL20013-W	N	Y	N	N	Y	Y
KAVL20014-W	N	Y	N	N	Y	Y
KAVL20015-I	N	N	N	N	N	N
KAVL20016-W	N	Y	N	N	Y	Y
KAVL20017-W	N	Y	N	N	Y	Y
KAVL20018-W	N	Y	N	N	Y	Y
KAVL20201-W	N	Y	N	N	Y	Y
KAVL20202-W	N	Y	N	N	Y	Y
KAVL20203-W	N	Y	N	N	Y	Y
KAVL20204-W	N	Y	N	N	Y	Y
KAVL20205-W	N	Y	N	N	Y	Y
KAVL20206-W	N	Y	N	N	Y	Y
KAVL20300-W	N	Y	N	N	N	N

Message ID	Output destination					
	Windows Event Log	Common message log	stdout	stderr	JP1 system event#1	Agent event#2
KAVL20301-W	N	Y	N	N	N	N
KAVL20505-E	Y	Y	N	N	N	N
KAVL20513-W	N	Y	N	N	N	N
KAVL20516-W	N	Y	N	N	N	N
KAVL20517-W	N	Y	N	N	N	N
KAVL20521-I	N	Y	N	N	N	N
KAVL20524-W	N	Y	N	N	N	N
KAVL20526-W	N	Y	N	N	N	N
KAVL20527-W	N	Y	N	N	N	N
KAVL20528-I	N	Y	N	N	N	N
KAVL20529-W	N	Y	N	N	N	N

#1

A *JP1 system event* is an event for notifying JP1/Integrated Management about a change in the agent's status. For details about JP1 system events, see the chapter that explains operation monitoring via linkage to the integrated management product JP1/Integrated Management in the *JP1/Performance Management User's Guide*.

Note that when a message is issued as a JP1 system event, only the first 255 bytes of the message text are output. Beginning with the 256th character, the text is not output.

The table below shows the prerequisite programs for issuing JP1 system events.

Table 6–2: Prerequisite programs for issuing JP1 system events

Host type	Prerequisite program	Version
PFM - Manager host	PFM - Manager	09-00 or later
PFM - Web Console host	PFM - Web Console	08-00 or later
PFM - RM host	PFM - RM for Virtual Machine	10-00 or later
	PFM - Manager or PFM - Base	09-00 or later
	JP1/Base	08-50 or later

#2

An *agent event* is an event for notifying PFM - Manager about a change in the agent's status. For details about agent events, see the chapter that explains event display in the *JP1/Performance Management User's Guide*.

The table below shows the prerequisite programs for issuing agent events.

Table 6–3: Prerequisite programs for issuing agent events

Host type	Prerequisite program	Version
PFM - Manager host	PFM - Manager	09-00 or later
PFM - Web Console host	PFM - Web Console	08-00 or later
PFM - RM host	PFM - RM for Virtual Machine	10-00 or later
	PFM - Manager or PFM - Base	09-00 or later

6.3 List of messages output to the Windows event log

This section lists the message information that is output by PFM - RM for Virtual Machine to the Windows Event Log.

The Windows Event Log is displayed in the Application Log in the Event Viewer window.

To open the Event Viewer window, from the **Start** menu in Windows, choose **Administrative Tools**, and then **Event Viewer**.

For an event that is output by PFM - RM for Virtual Machine, the identifier `PFM-Virtual Machine` is displayed under **Source** in the Event Viewer window.

The table below lists the message information that is output by PFM - RM for Virtual Machine to the Windows Event Log.

Table 6–4: List of Windows Event Log output message information

Message ID	Windows Event Log	
	Event ID	Type
KAVL20000-I	20000	Information
KAVL20001-E	20001	Error
KAVL20002-I	20002	Information
KAVL20004-E	20004	Error
KAVL20005-E	20005	Error
KAVL20006-E	20006	Error
KAVL20505-E	20505	Error

6.4 Messages

This section explains the messages that are output by PFM - RM for Virtual Machine and the action to be taken by the user. The list of PFM - RM for Virtual Machine messages follows.

KAVL20000-I

Remote Monitor Collector has stopped. (host=<*host-name*>, service=<*service-ID*>)

The Remote Monitor Collector service terminated normally.

(S)

Terminates processing of the Remote Monitor Collector service.

KAVL20001-E

Remote Monitor Collector failed to start.

The Remote Monitor Collector service failed to start.

(S)

Terminates processing of the Remote Monitor Collector service.

(O)

Check the previous message that was output to the common message log, and take the action indicated in that message.

KAVL20002-I

Remote Monitor Collector started. (host=<*host-name*>, service=<*service-ID*>)

Startup of the Remote Monitor Collector service is complete.

(S)

Starts the performance data collection process of the Remote Monitor Collector service.

KAVL20004-E

An attempt to read the service startup information file has failed.

During startup processing of the Remote Monitor Collector service, the service startup initialization file could not be loaded.

(S)

Terminates processing of the Remote Monitor Collector service.

(O)

If this message is output repeatedly, delete the instance environment and then re-create the instance. If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20005-E

> Remote Monitor Collector will now stop because an error occurred.

The Remote Monitor Collector service is about to stop because an error has occurred.

(S)

Terminates processing of the Remote Monitor Collector service.

(O)

Check the previous message that was output to the common message log, and take the action indicated in that message. If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20006-E

> Memory allocation failed.

Memory allocation failed.

(S)

Terminates processing of the Remote Monitor Collector service.

(O)

Increase the amount of free memory.

KAVL20007-W

> Memory allocation failed. (RecordType=<*record-type*>)

Memory allocation failed. If `UNKNOWN` is output in *record-type*, this means that memory allocation failed for multiple record IDs.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Increase the amount of free memory.

KAVL20008-W

> An attempt to collect the record failed. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, RecordType=<*record-type*>)

The record indicated by *record-type* could not be collected.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20009-W

An invalid value or a value outside the range was specified for the property of the Remote Monitor Collector service. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, property=<*property-name*>, value=<*range-value*>)

An invalid value or a value outside the range was specified for a property of the Remote Monitor Collector service.

(S)

Invalidates the specified value and continues processing of the Remote Monitor Collector service. The item value remains unchanged.

(O)

Check whether the value that is specified is acceptable. If not, specify an appropriate value.

KAVL20010-E

An error occurred in the function. (function=<*function-name*>, en=<*error-code*>, arg1=<*argument-1*>, arg2=<*argument-2*>, arg3=<*argument-3*>)

An error occurred during execution of the function indicated by *function-name*.

(S)

Terminates processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20011-W

The collector process failed to start.

A collection process failed to start.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20013-W

A performance data file is invalid. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>)

The content of a performance data storage file is invalid.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20014-W

A collector process was forcefully terminated because performance data collection did not end in the specified period of time. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>)

A collection process was forcibly terminated because performance data collection did not end within the specified period of time.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Check the following items:

- Whether the monitoring-target host has started
- Whether the connection settings on the virtualization system side are correct
- Whether the PFM - RM host and the virtualization system are in a temporary high load state
- Whether the following settings (specified during setup of the monitoring target) are correct

If the monitoring target uses VMware:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address

- Whether a certificate has been installed

- Whether the `Port` value is correct

- Whether the `UserID` and `Password` values are correct

- If the default certificate of VMware is used, confirm whether the Update Root Certificate functions affect the operation. For details, see *2.5.1 For VMware*.

If the monitoring target uses Hyper-V:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address

- Whether the WMI service has started on the monitoring-target host

- Whether the WMI connection setup procedure has been performed correctly

- Whether the `UserID`, `Password`, and `Domain` values are correct

If the monitoring target uses logical partitioning feature:

- Whether the IP address (`VM_Host`) is correct

If the monitoring target uses KVM:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address

- Whether the SSH server is started on the monitoring-target host

- Whether the SSH connection setup procedure has been performed correctly

- Whether the `SSH_Type` and `SSH_Client` values are correct

- Whether the `Port` value is correct

- Whether the `UserID` and `Private_Key_File` values are correct

If the monitoring target uses Docker environment:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address

- Whether a certificate has been installed

- Whether `Port` has been set to a correct value other than `0`

If the monitoring target uses Podman environment:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address

- Whether the SSH server is started on the monitoring-target host

- Whether the SSH connection setup procedure has been performed correctly

- Whether the `SSH_Type` and `SSH_Client` values are correct
- Whether the `Port` value is correct
- Whether the `UserID` and `Private_Key_File` values are correct

If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20015-I

The records were successfully saved onto the Store database. (Instance=<*instance-name*>, RecordType=<*record-type*>, count=<*record-count*>)

The process of saving the record(s) indicated by *record-type* in the Store database is complete.

(S)

Continues processing of the Remote Monitor Collector service.

KAVL20016-W

The initialization of interprocess communication failed. (Instance=<*instance-name*>)

Initialization of communications between the Remote Monitor Collector service and a collection process failed.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Opening of or writing to a work file may have failed. Check to make sure that there is no disk space shortage.

If disk space is not a problem, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20017-W

An error occurred during collection of the record. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>)

An error occurred during collection of the record.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20018-W

Account authentication failed. (Instance=<*instance-name*>)

Account authentication failed.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Check whether the following items that were specified during setup of the instance environment are correct:

- `HostUserID`
- `HostPassword`
- `HostDomain`

KAVL20201-W

The system could not connect to the monitored virtual environment. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, datetime=<*date-time*>, message=<*message*>)

The system could not connect to the monitored virtual environment.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Check the following items:

- Whether the monitoring-target host has started
- Whether the connection settings on the virtualization system side are correct
- Whether the PFM - RM host and the virtualization system are in a temporary high load state
- (If there is a firewall between PFM - RM for Virtual Machine and the monitoring-target host) Whether the ports used for communication are properly set in the firewall
- Whether the following settings (specified during setup of the monitoring target) are correct
  If the monitoring target uses VMware:
  - Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
  - Whether a certificate has been installed
  - Whether the `Port` value is correct
  - Whether the `UserID` and `Password` values are correct
  If the monitoring target uses Hyper-V:
  - Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
  - Whether the WMI service has started on the monitoring-target host
  - Whether the WMI connection setup procedure has been performed correctly
  - Whether the `UserID`, `Password`, and `Domain` values are correct
  If the monitoring target uses logical partitioning feature:
  - Whether the IP address (`VM_Host`) is correct
  If the monitoring target uses KVM:
  - Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
  - Whether the SSH server is started on the monitoring-target host
  - Whether the SSH connection setup procedure has been performed correctly
  - Whether the `SSH_Type` and `SSH_Client` values are correct
  - Whether the `Port` value is correct
  - Whether the `UserID` and `Private_Key_File` values are correct

If the monitoring target uses Docker environment:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
- Whether a certificate has been installed
- Whether `Port` has been set to a correct value other than `0`

If the monitoring target uses Podman environment:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
- Whether the SSH server is started on the monitoring-target host
- Whether the SSH connection setup procedure has been performed correctly
- Whether the `SSH_Type` and `SSH_Client` values are correct
- Whether the `Port` value is correct
- Whether the `UserID` and `Private_Key_File` values are correct

If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20202-W

An attempt to authenticate the monitored virtual environment has failed. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, datetime=<*date-time*>, message=<*message*>)

The monitored virtual environment could not be authenticated.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Make sure that the host of the virtual environment is active. Furthermore, make sure that the following items, which were specified during setup of the monitoring-target, are correct:

- `UserID`
- `Password`

KAVL20203-W

Initialization of the collector process log file failed. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, datetime=<*date-time*>, message=<*message*>)

Initialization of a collection process log failed.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Opening of or writing to a work file may have failed. Check to make sure there is no disk space shortage.

If disk space is not a problem, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20204-W

Initialization of the collector process failed. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, datetime=<*date-time*>, message=<*message*>)

Initialization of a collection process failed.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20205-W

A certificate is not installed or a certificate is not correct. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, datetime=<*date-time*>, message=<*message*>)

No certificate is installed or an incorrect certificate is installed.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Install a CA-signed certificate.

If the problem is not resolved even after a CA-signed certificate is installed, check the following items in the installed certificate:

- Effective period
- Issuing destination (whether it is the same as the host name)
- Validity of the certificate (has verification been completed successfully?)

If the monitoring target uses VMware:

If there is a problem with the certificate, re-create a certificate and then re-install it. For details about how to create a certificate, see the VMware documentation.

If the default certificate of VMware is used for operation, ignore this message.

For details about a certificate, see *2.5.1 For VMware*.

If the monitoring target uses Docker environment:

For details about a certificate, see *2.5.4 For Docker environment*.

KAVL20206-W

A certificate is not correct. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, datetime=<*date-time*>, message=<*message*>)

The certificate is not trusted.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Configure the certificate installed on the monitoring target so that it can be trusted.

If the problem is not resolved even after the certificate is configured to be trusted, check the following items in the certificate:

If the problem is not resolved even after the certificate is configured to be trusted, check the following items in the certificate:

- Effective period
- Issuing destination (whether it is the same as the host name)
- Validity of the certificate (has verification been completed successfully?)

If there is a problem with the certificate, re-create a certificate and then re-install it.

If the monitoring target uses VMware:

For details about how to create a certificate, see the VMware documentation.

For details about a certificate, see *2.5.1 For VMware*.

If the monitoring target uses Docker environment:

For details about a certificate, see *2.5.4 For Docker environment*.

KAVL20300-W

It failed to occur JP1 system event or Agent event.

A JP1 system event or agent event could not be issued.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20301-W

It failed to issue JP1 system event or Agent event, because Memory is insufficient.

A JP1 system event or agent event could not be issued.

(S)

Continues processing of the Remote Monitor Collector.

(O)

Increase the amount of free memory.

KAVL20505-E

An attempt to read the target information file has failed. (Target=<*monitoring-target-name*>)

The monitoring target information file could not be loaded while the Remote Monitor Collector service was being started.

(S)

Stops the Remote Monitor Collector service.

(O)

Delete the relevant monitoring target, and then re-create it.

KAVL20513-W

The collector process failed to start.

The collection process could not be started.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20516-W

Performance data was not saved to the Store database because it is the same as previous performance data. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>, RecordType=<*record-type*>)

Performance data was not saved in the Store database because the data was the same as the previously collected data.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Make sure that the collection process execution interval does not exceed the record collection interval.

If this problem frequently occurs when the collection process execution interval does not exceed the record collection interval, increase the value of the record collection interval or decrease the number of hosts monitored in the instance environment.

> **Note**
>
> If this message is output repeatedly, see *Appendix N. What to Do When the Message KAVL20516-W Is Output Repeatedly*.

KAVL20517-W

The record build failed because there is no performance data. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>)

An attempt to build records failed because there was no performance data.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

This alert might occur immediately after startup because no performance data has been collected.

If this alert continues to occur after the first collection finishes, check the following items:

- Whether the monitoring-target host has started
- Whether the following settings (specified during setup of the monitoring target) are correct
  If the monitoring target uses VMware:
  - Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
  - Whether a certificate has been installed

- Whether the `Port` value is correct
- Whether the `UserID` and `Password` values are correct

If the monitoring target uses Hyper-V:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
- Whether the WMI service is started on the monitoring-target host
- Whether the WMI connection setup procedure has been performed correctly
- Whether the `UserID`, `Password`, and `Domain` values are correct

If the monitoring target uses logical partitioning feature:

- Whether the IP address (`VM_Host`) is correct

If the monitoring target uses KVM:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
- Whether the SSH server is started on the monitoring-target host
- Whether the SSH connection setup procedure has been performed correctly
- Whether the `SSH_Type` and `SSH_Client` values are correct
- Whether the `Port` value is correct
- Whether the `UserID` and `Private_Key_File` values are correct
- Whether the `SSH_Client` value specified during instance setup is correct

If the monitoring target uses Docker environment:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
- Whether a certificate has been installed
- Whether `Port` has been set to a correct value other than `0`

If the monitoring target uses Podman environment:

- Whether the host name (specified for `Target Host` and `VM_Host`) can resolve to an IP address
- Whether the SSH server is started on the monitoring-target host
- Whether the SSH connection setup procedure has been performed correctly
- Whether the `SSH_Type` and `SSH_Client` values are correct
- Whether the `Port` value is correct
- Whether the `UserID` and `Private_Key_File` values are correct

If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

KAVL20521-I

A collector process will restart because the system detected that it stopped. (Instance=<*instance-name*>)

The collection process will be restarted because its stoppage was detected.

(S)

Continues processing of the Remote Monitor Collector service.

KAVL20524-W

There are no SSH client execution modules. (Instance=<*instance-name*>)

The value set as the SSH client execution module (`SSH_Client`) is incorrect. Make sure that you specify a correct value when monitoring KVM or Podman environment.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Check whether the value of the following property that was set during setup of the instance environment is correct:

- SSH_Client

KAVL20526-W

There are no private keys. (Instance=<*instance-name*>, Target=<*monitoring-target-name*>)

The private key file to be used for SSH public key encryption is not correctly set (`Private_Key_File`). Make sure that you specify a correct value when monitoring KVM or Podman environment.

(S)

Continues processing of the Remote Monitor Collector service.

(O)

Check whether the value of the following property that was set during setup of the monitoring target is correct:

- Private_Key_File

KAVL20527-W

An attempt to read the user record define information file has failed. (Instance=<*instance-name*>, Line=<*line-number*>, message=<*detailed-message*>)

During startup processing of the Remote Monitor Collector service, the user-defined record definition file could not be loaded.

(S)

Continues processing of the Remote Monitor Collector service.
Does not collect the user defined record of the record type with the error.

(O)

Check the detailed message to make sure the user-defined record definition file is correct.

- File open error.
  The user-defined record definition file exists but failed to open.
- Ignoring duplicate [[SECTION NAME]].
  `[[SECTION NAME]]` specified for `[RECORD ID]` is duplicated.
- *XXXXX* is required.
  XXXXX needs to be specified.
- *XXXXX* is not correct.
  The specified *XXXXX* is incorrect.
- The specified order is not correct.
  The items are specified in the wrong order.
- Unknown parameter.

An unknown parameter is specified.

KAVL20528-I

User record collection enabled. (Instance=<instance-name>, RecordType=<*record-type*>)

Performance data is collected as specified in the user-defined record definition file.

(S)

Continues processing of the Remote Monitor Collector service.

KAVL20529-W

An attempt to read the settings information file has failed.

Failed to read the PFM - RM for Virtual Machine setup file while the Remote Monitor Collector service was starting up.

(S)

Continues processing of the Remote Monitor Collector service using default settings.

(O)

If this message is output with changes made to the default settings, first enable reading of the PFM - RM for Virtual Machine setup file (save the file and ensure it can be read) then restart the service of the instance environment.

If operating with the default settings, you do not have to restart the service of the instance environment. Continue processing using the default settings.

7 Error Handling Procedures

This chapter explains the actions that you can take when problems occur during Performance Management operations, especially with regard to PFM - RM for Virtual Machine. For details about troubleshooting for the entire Performance Management system, see the chapter that explains error detection in the *JP1/Performance Management User's Guide*.

7.1 Error handling procedures

This section describes the procedure to follow when a problem occurs in Performance Management.

Checking the event

Check the following details:

- What was happening when the problem occurred
- Message content (if a message has been output)
- Log information, such as the common message log

For the cause of each message and the action to take, see *6. Messages*. For details about the log information that is output by Performance Management, see *7.3 Log information*.

Collecting data

You need to collect data to investigate the cause of the problem. See *7.4 Data collected when a problem occurs* and *7.5 How to collect data*, and collect the necessary data.

Investigating the problem

Based on the collected data, investigate the cause of the problem, and identify the area where the problem occurred and its scope.

7.2 Troubleshooting

This section describes troubleshooting related to the use of Performance Management. If a problem occurs when you are using Performance Management, first check whether any of the events explained in this section has occurred.

The table below describes the main problems that can occur in Performance Management.

Table 7–1: Troubleshooting

Classification	Problem description	Described in
Setup and service startup	• A Performance Management program service does not start. • It takes a long time for a service to start after a service startup request is issued. • When another program begins a service after a Performance Management program service has stopped, communication cannot be established correctly. • After an `The disk capacity is insufficient` message is output, the Master Store service or Remote Monitor Store service stops. • The Remote Monitor Collector service of PFM - RM does not start.	See the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.
Command execution	• When the `jpctool service list` command is executed, a service name that is not active is output. • When the `jpctool db dump` command is executed, data that is not from the specified Store database is output.	
Report definition	• Some time bands are not displayed in the historical report.	
Alarm definition	• A program defined in the action definition is not running correctly. • No alarm event is displayed.	
Performance data collection and management	• Even though a short data retention period was set, the size of the PFM - RM for Virtual Machine Store database is not reduced. • The message `Illegal data was detected in the Store database` is output to the common message log.	
	• Even though PFM - RM for Virtual Machine has started, no performance data is being collected. • There is a large time difference between Record Time and Sampling Time.	*7.2.1*

7.2.1 Performance data collection and management

This subsection describes troubleshooting related to performance data collection and management procedures of Performance Management. For details about how to correct other types of errors, see the *JP1/Performance Management User's Guide*.

(1) No performance data is collected even when PFM - RM for Virtual Machine is started

The items to check and the actions to be taken are described below for each monitored virtual environment.

(a) For VMware

- If you are monitoring information of a virtual machine running on VMware, check the following items:
  - Make sure that VMware Tools is running on the monitored virtual machine.
  - If you change the time on a physical server on which VMware is running. performance data may not be collected in some cases. In such cases, start the mgmt-vmware service.
- If the Status field value of a PD record shows `ERROR`, take an action based on the value in the Reason field. The table below shows the actions to be taken.

Table 7–2: Reason field value and actions to be taken

Reason field value	Explanation	Action
Connection failed	An attempt to connect to the target virtual environment failed.	• Check whether the host of the virtual environment is active. • Check whether the following items, which were specified during setup of the instance environment, are valid: # - `VM_Type` • Check whether the following items, which were specified during setup of the monitoring target, are valid:# - `Target Host` - `VM_Host` - `Security` - `Port` - `UserID` - `Password` • Check whether the host name (`Target Host` or `VM_Host`), which was specified during setup of the monitoring target, can be resolved. • Check whether a valid certificate is installed in the PFM - RM for Virtual Machine host. When the default certificate of VMware is used in an environment in which a PFM - RM for Virtual Machine host cannot connect to the Windows Update site, check the Windows settings. For details, see *2.5.1 For VMware*. • HTTPS connection must be permitted on the virtual environment side. For details about how to set a communication method for a virtual environment, see the VMware documentation. • Make sure that the user ID that was specified for `UserID` during setup of the monitoring target has at least read-only permissions for VMware roles. For details about how to check permissions, see the VMware documentation.
Authorization failed	Authorization failed at the target virtual environment.	• Check whether the host of the virtual environment is active. • Check whether the following items, which were specified during setup of the monitoring target, are valid: # - `UserID` - `Password` - `Domain`
Timeout	Performance data collection did not end within the specified time.	• Check whether the host of the virtual environment is active. • Check whether the following items, which were specified during setup of the instance environment, are valid: # - `VM_Type` • Check whether the following items, which were specified during setup of the monitoring target, are valid: # - `Target Host` - `VM_Host` - `Security` - `Port`

Reason field value	Explanation	Action
Timeout	Performance data collection did not end within the specified time.	- `UserID` - `Password` • Check whether the host name (`Target Host` or `VM_Host`) which was specified during setup of the monitoring target, can be resolved. • Make sure that a certificate has been embedded. For PFM - RM for Virtual Machine, a certificate must be installed for each monitoring target. When the default certificate of VMware is used check the Windows settings. For details, see *2.5.1 For VMware*. • If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.
Collection Error	A collection error occurred.	• Check whether the following items that were specified during setup of the instance environment are correct:# - `HostUserID` - `HostPassword` - `HostDomain` • If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

#

To check an item that has been set, execute the `jpcconf` command. Alternatively, on PFM - Web Console, view the Remote Monitor Configuration properties from the Remote Monitor Collector service of PFM - RM for Virtual Machine.

- In all other cases, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

(b) For Hyper-V

- If the Status field value of a PD record shows `ERROR`, take action as indicated by the value in the Reason field. The table below shows the actions to be taken.

Table 7–3: Reason field value and actions to be taken

Reason field value	Explanation	Action
Connection failed	Attempt to connect to the target virtual environment failed.	• Check whether the host of the virtual environment is active. • Check whether the following items that were specified during setup of the instance environment are correct:# - `VM_Type` • Check whether the following items, which were specified during setup of the monitoring target, are valid: # - `Target Host` - `VM_Host` - `UserID` - `Password` - `Domain` • Check whether the host name (`Target Host` or `VM_Host`) that was specified during setup of the monitoring target can be resolved. • Check whether the WMI connection has been set up correctly. For PFM - RM for Virtual Machine, WMI connection settings must be specified for each monitoring target.

Reason field value	Explanation	Action
Authorization failed	Authorization failed at the target virtual environment.	• Check whether the host of the virtual environment is active. • Check whether the following items that were specified during setup of the monitoring target are correct:# - `UserID` - `Password` - `Domain`
Timeout	Performance data collection did not end within the specified time.	• Check whether the host of the virtual environment is active. • Check whether the following items that were specified during setup of the instance environment are correct:# - `VM_Type` • Check whether the following items, which were specified during setup of the monitoring target, are valid: # - `Target Host` - `VM_Host` - `UserID` - `Password` - `Domain` • Check whether the host name (`Target Host` or `VM_Host`) that was specified during setup of the monitoring target can be resolved. • Check whether the WMI connection has been set up correctly. For PFM - RM for Virtual Machine, WMI connection settings must be specified for each monitoring target. • If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.
Collection Error	A collection error occurred.	• Check whether the following items that were specified during setup of the instance environment are correct:# - `HostUserID` - `HostPassword` - `HostDomain` • If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

#

Use the `jpcconf` command to check what has been set for an item. Alternatively, the Remote Monitor Configuration properties from the Remote Monitor Collector service of PFM - RM for Virtual Machine can be viewed on PFM - Web Console.

- In all other cases, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.
- If the OS of the monitoring-target host is Windows, check the application event log.

(c) For KVM

- If the `Status` field of the PD record stores the value `ERROR`, take action according to the value of the `Reason` field. The following table lists the values of the `Reason` field and describes the action to be taken for each value.

Table 7–4: Action to be taken according to the value of the Reason field

Reason field value	Explanation	Action
Connection failed	Connection to the destination virtual environment failed.	Check whether the host in the virtual environment has started. Make sure that the following settings specified during instance environment setup are correct:# - `VM_Type` - `SSH_Type` - `SSH_Client` Make sure that the following settings specified during monitoring target setup are correct:# - `Target Host` - `VM_Host` - `Port` - `UserID` - `Private_Key_File` Make sure that the host name specified (for `Target Host` and `VM_Host`) during monitoring target setup can resolve to an IP address. Make sure that the SSH connection settings are specified. For PFM - RM for Virtual Machine, these settings must be specified for each monitoring target.
Authorization failed	Authentication failed in the destination virtual environment.	Check whether the host in the virtual environment has started. Make sure that the following settings specified during instance environment setup are correct:# - `HostUserID` - `HostPassword` - `HostDomain` Make sure that the following settings specified during monitoring target setup are correct:# - `UserID`
Timeout	Performance data collection did not end within the specified length of time.	Check whether the host in the virtual environment has started. Make sure that the following settings specified during instance environment setup are correct:# - `VM_Type` - `HostUserID` - `HostPassword` - `HostDomain` Make sure that the following settings specified during monitoring target setup are correct:# - `Target Host` - `VM_Host` - `Port` - `UserID` - `Private_Key_File` Make sure that the host name specified (for `Target Host` and `VM_Host`) during monitoring target setup can resolve to an IP address. Make sure that the SSH connection settings are specified. For PFM - RM for Virtual Machine, these settings must be specified for each monitoring target. If the cause of the problem cannot be determined, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

Reason field value	Explanation	Action
Collection Error	A collection error occurred.	Make sure that the following settings specified during instance environment setup are correct:# - HostUserID - HostPassword - HostDomain If the cause of the problem cannot be determined, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

#

To check the settings that have been specified, execute the `jpcconf` command. You can also check them from PFM - Web Console by referencing the Remote Monitor Configuration properties of the Remote Monitor Collector service of PFM - RM for Virtual Machine.

- In any conditions other than the above condition, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

(d) For Docker environment

- If the `Status` field of the PD record stores the value `ERROR`, take action according to the value of the `Reason` field. The following table lists the values of the `Reason` field and describes the action to be taken for each value.

Table 7–5: Action to be taken according to the value of the Reason field

Reason field value	Explanation	Action
Connection failed	Connection to the destination virtual environment failed.	• Check whether the destination server is running. • Check whether the Docker Engine is configured to allow TCP connection on the destination server. • Check whether the following items, which were specified during setup of the instance environment, are valid: # - VM_Type • Check whether the following items, which were specified during setup of the monitoring target, are valid:# - Target Host - VM_Host - Security - Port • Check whether the host name (`Target Host` or `VM_Host`), which was specified during setup of the monitoring target, can be resolved. • Check whether root certificate of the certificate authority and client certificates have been embedded. For details, see *2.5.4 For Docker environment*. • Whether Port has been set to a correct value other than 0.
Timeout	Performance data collection did not end within the specified length of time.	• Check whether the destination server is running. • Check whether the Docker Engine is configured to allow TCP connection on the destination server. • Check whether Docker environment running on the destination server uses API version 1.19 or later. • Check whether the following items, which were specified during setup of the instance environment, are valid: # - VM_Type - HostUserID

Reason field value	Explanation	Action
Timeout	Performance data collection did not end within the specified length of time.	- `HostPassword` - `HostDomain` • Check whether the following items, which were specified during setup of the monitoring target, are valid: # - `Target Host` - `VM_Host` - `Security` - `Port` • Check whether the host name (`Target Host` or `VM_Host`) which was specified during setup of the monitoring target, can be resolved. • Check whether root certificate of the certificate authority and client certificates have been embedded. For details, see *2.5.4 For Docker environment*. • Whether Port has been set to a correct value other than 0. • If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.
Collection Error	A collection error occurred.	• Check whether the following items that were specified during setup of the instance environment are correct:# - `HostUserID` - `HostPassword` - `HostDomain` • If the cause cannot be determined, collect maintenance data and contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

#

To check an item that has been set, execute the `jpcconf` command. Alternatively, on PFM - Web Console, view the Remote Monitor Configuration properties from the Remote Monitor Collector service of PFM - RM for Virtual Machine.

- In all other cases, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

(e) For Podman environment

- If the `Status` field of the PD record stores the value `ERROR`, take action according to the value of the `Reason` field. The following table lists the values of the `Reason` field and describes the action to be taken for each value.

Table 7–6: Action to be taken according to the value of the Reason field

Reason field value	Explanation	Action
Connection failed	Connection to the destination virtual environment failed.	• Check whether the host in the virtual environment has started. • Make sure that the following settings specified during instance environment setup are correct:# - `VM_Type` - `SSH_Type` - `SSH_Client` • Make sure that the following settings specified during monitoring target setup are correct:# - `Target Host` - `VM_Host` - `Port`

Reason field value	Explanation	Action
Connection failed	Connection to the destination virtual environment failed.	- `UserID` - `Private_Key_File` • Make sure that the host name specified (for `Target Host` and `VM_Host`) during monitoring target setup can resolve to an IP address. • Make sure that the SSH connection settings are specified. For PFM - RM for Virtual Machine, these settings must be specified for each monitoring target.
Authorization failed	Authentication failed in the destination virtual environment.	• Check whether the host in the virtual environment has started. • Make sure that the following settings specified during instance environment setup are correct:# - `HostUserID` - `HostPassword` - `HostDomain` • Make sure that the following settings specified during monitoring target setup are correct:# - `UserID`
Timeout	Performance data collection did not end within the specified length of time.	• Check whether the host in the virtual environment has started. • Make sure that the following settings specified during instance environment setup are correct:# - `VM_Type` - `HostUserID` - `HostPassword` - `HostDomain` - `SSH_Type` - `SSH_Client` • Make sure that the following settings specified during monitoring target setup are correct:# - `Target Host` - `VM_Host` - `Port` - `UserID` - `Private_Key_File` • Make sure that the host name specified (for `Target Host` and `VM_Host`) during monitoring target setup can resolve to an IP address. • Make sure that the SSH connection settings are specified. For PFM - RM for Virtual Machine, these settings must be specified for each monitoring target. • If the cause of the problem cannot be determined, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.
Collection Error	A collection error occurred.	• Make sure that the following settings specified during instance environment setup are correct:# - `HostUserID` - `HostPassword` - `HostDomain` • If the cause of the problem cannot be determined, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

#

To check the settings that have been specified, execute the `jpcconf` command. You can also check them from PFM - Web Console by referencing the Remote Monitor Configuration properties of the Remote Monitor Collector service of PFM - RM for Virtual Machine.

- In any conditions other than the above condition, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

(f) For logical partitioning feature

- If the `Status` field of the PD record stores the value `ERROR`, take action according to the value of the `Reason` field. The following table lists the values of the `Reason` field and describes the action to be taken for each value.

Table 7–7: Action to be taken according to the value of the Reason field

Reason field value	Explanation	Action
Connection failed	Connection to the destination virtual environment failed.	• Check whether the `HvmSh` command (`HvmSh.exe`) has been copied to *installation-folder* `\agt8\plugin\jpcagt5virtage.d\`. For details about how to copy the `HvmSh` command (`HvmSh.exe`), see *2.5.6 For logical partitioning feature*. • Make sure that the following setting specified during monitoring target setup is correct:# - `VM_Host` • It is possible that after performance data was obtained last time, the state of logical partitions or the configuration of Hitachi Compute Blade logical partitioning feature was changed. This is not an error. For details, see the *Action* column for the Message ID `KAVL20201-W` in *Table 7-8*. • If the cause cannot be determined, collect maintenance data and the LPAR Manager dump, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*. Also, for details about how to capture the LPAR Manager dump, see the topic that describes the LPAR Manager dump capture command in the User's Guide for each Hitachi Compute Blade device.
Response invalid	There was an unintended response from the server.	• The version of Hitachi Compute Blade logical partitioning feature or `HvmSh` command (`HvmSh.exe`) is not supported. For details about the version of Hitachi Compute Blade logical partitioning feature and `HvmSh` command, see *2.1.1(4) Prerequisite programs*. For details about how to upgrade the version of Hitachi Compute Blade logical partitioning feature, see the topic that describes the upgrade procedures of Hitachi Compute Blade logical partitioning feature in the User's Guide for each Hitachi Compute Blade device. For details about the `HvmSh` command (`HvmSh.exe`), see *2.5.6 For logical partitioning feature*.
Timeout	Performance data collection did not end within the specified length of time.	• Check whether Hitachi Compute Blade logical partitioning feature is working properly. • Make sure that the following setting specified during instance environment setup is correct:# - `VM_Type` • Make sure that the following settings specified during monitoring target setup are correct:# - `Target Host` - `VM_Host` • Check whether the IP address for PFM - RM for Virtual Machine specified during the environment setup of Hitachi Compute Blade logical partitioning feature is correct. For details about the environment settings of Hitachi Compute Blade logical partitioning feature, see *2.5.6 For logical partitioning feature*. • Check the state of the network for any problem. • If there is a firewall between PFM - RM for Virtual Machine and Hitachi Compute Blade logical partitioning feature, check whether the ports used for communication are properly set in the firewall. For details about the settings of the firewall, see *D.2(7) Firewall passage direction during communication between PFM - RM for Virtual Machine and logical partitioning feature*. • If the cause cannot be determined, collect maintenance data and the LPAR Manager dump, and then contact the system administrator. For details about how to collect maintenance data,

Reason field value	Explanation	Action
Timeout	Performance data collection did not end within the specified length of time.	see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*. Also, for details about how to capture the LPAR Manager dump, see the topic that describes the LPAR Manager dump capture command in the User's Guide for each Hitachi Compute Blade device.
Collection error	A collection error occurred.	• Opening of or writing to a work file may have failed. Check to make sure that there is no disk space shortage. • Make sure that the work file used by PFM - RM for Virtual Machine is not open. If the work file is not open, check the memory usage with Task Manager. If free memory is not sufficient, exit some applications to free up memory. • If the cause cannot be determined, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

#

To check the settings that have been specified, execute the `jpcconf` command. You can also check them from PFM - Web Console by referencing the Remote Monitor Configuration properties of the Remote Monitor Collector service of PFM - RM for Virtual Machine.

- When one of the following messages is output to the common message log, take the corresponding action for each message ID. The following table shows the actions to take.

Table 7–8: Message IDs and actions

Message ID	Explanation	Action
`KAVL20201-W`	The system could not connect to the monitored virtual environment.	If the message in the message text is `message=<Virtage(`*instance-name*`) Connect failed.>`: • Check whether the `HvmSh` command (`HvmSh.exe`) has been copied to *installation-folder*`\agt8\plugin\jpcagt5virtage.d\`. For details about how to copy the `HvmSh` command (`HvmSh.exe`), see *2.5.6 For logical partitioning feature*. • Check the memory usage with Task Manager. If free memory is not sufficient, exit some applications to free up memory. • If the problem is not solved, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.
		If the message in the message text is `message=<Virtage(`*instance-name*`) No data.` *record-type*`>`: See *5. Records* to check whether the record type of the record is supported by Hitachi Compute Blade logical partitioning feature. • If the record is not supported, performance data cannot be collected. Configure the system so that the data is not collected. • If the record is supported, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.
		If the message in the message text is `message=<Virtage(`*instance-name*`) Connect failed. Return:`*exit-code*`>`: • When the exit code is one of `0x01000000`, `0x11000000`, and `0xFFFFFFFF`: According to the subsection *2.1.1(4) Prerequisite programs*, check whether the version of the firmware of your Hitachi Compute Blade logical partitioning feature is consistent with that of the `HvmSh` command (`HvmSh.exe`). For details about how to upgrade the version of Hitachi Compute Blade logical partitioning feature, see the topic that describes the upgrade procedures of Hitachi Compute Blade logical partitioning feature in the User's Guide for each Hitachi Compute Blade device. • When the exit code is none of the above: Based on the exit code, see the action described in the topic on error messages in the *Hitachi Compute Blade User's Guide*.

Message ID	Explanation	Action
`KAVL20201-W`	The system could not connect to the monitored virtual environment.	If the cause cannot be determined, collect maintenance data and the LPAR Manager dump, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*. Also, for details about how to capture the LPAR Manager dump, see the topic that describes the LPAR Manager dump capture command in the User's Guide for each Hitachi Compute Blade device.
`KAVL20203-W`	Initialization of the collector process log file failed.	In the message in the message text, check the details shown in `message=<Virtage(`*instance-name*`) Log failed.` *detailed-information*>. • If the detailed information is either `(Initialization error=`*%d*`)` or `(Level setting error=`*%d*`)`: Opening of or writing to a work file may have failed. Check to make sure that there is no disk space shortage. If disk space is not a problem, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*. • If the detailed information is none of the above: Check the memory usage with Task Manager. If free memory is not sufficient, exit some applications to free up memory. If free memory is sufficient, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.
`KAVL20204-W`	Initialization of the collector process failed.	If the message in the message text is `message=<Virtage(`*instance-name*`) Initialization failed. Param error=`*detailed-information*>: • If the detailed information is `VM_Host`, check the following items: - For the setting items of the monitoring target, if an IP address is specified for `VM_Host`, check whether the IP address is correct. - For the setting items of the monitoring target, if a host name is specified for `VM_Host`, check whether the host name can be resolved. • If the detailed information is none of the above, check the memory usage with Task Manager. If free memory is not sufficient, exit some applications to free up memory. • If the problem is not solved, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.
		If the message in the message text is `message=<Virtage(`*instance-name*`) Initialization failed.>`: • Make sure that the work file used by PFM - RM for Virtual Machine is not open. Also, check to make sure that there is no disk space shortage. If the problem is not resolved, check the memory usage with Task Manager. If free memory is not sufficient, exit some applications to free up memory. • If the cause cannot be determined, collect maintenance data and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

#

To check the settings that have been specified, execute the `jpcconf` command. You can also check them from PFM - Web Console by referencing the Remote Monitor Configuration properties of the Remote Monitor Collector service of PFM - RM for Virtual Machine.

- In any conditions other than the above condition, collect maintenance data, and then contact the system administrator. For details about how to collect maintenance data, see the chapter on troubleshooting in the *JP1/Performance Management User's Guide*.

(2) There is a large time difference between Record Time and Sampling Time

Because Sampling Time is the time in the monitored virtual environment, a slight time deviation may occur in normal operations. If the time deviation is large enough to cause an operational problem, check the following items:

- Is the time set on the RM host different from that set in the monitored virtual environment?
  If they are different, they need to be synchronized. Before changing the time settings, carefully read the notes related to the OS of the RM host and the virtual environment software.
- If SSL/TLS is being used for communication with VMware, has a certificate been embedded?
  If a certificate has not been embedded correctly, information collection may be delayed. See *2.5.1 For VMware*, and correctly embed a certificate in the RM host. For details about how to embed certificates into VMware, see the VMware manuals.
- If SSL/TLS is being used for communication with Docker environment, has a certificate been embedded?
  If a certificate has not been embedded correctly, information collection may be delayed. See *2.5.4 For Docker environment* and correctly embed a certificate in the RM host.

7.2.2 Other problems

Check what was happening when the problem occurred. If a message was output, check its content. For details about the log information that is output by Performance Management, see *7.3 Log information*.

If the problem cannot be solved even after you have taken the actions prescribed in the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide* and *7.2.1 Performance data collection and management*, or if other problems occur, collect data for the purpose of investigating the cause of the problems, and then contact the system administrator.

For details about the data that must be collected and how to collect it, see *7.4 Data collected when a problem occurs* and *7.5 How to collect data*.

7.3 Log information

When a problem occurs in Performance Management, check the log information to determine which corrective action to take. The following five types of log information are output when Performance Management is operating:

- System log
- Common message log
- Operation status log
- Trace log
- Collected log

Each of these types of log information is explained below.

7.3.1 Log information types

(1) System log

The system log contains information that indicates the system status and reports problems. In Windows, this information is output to the event log file.

For details about the output format, see the chapter that explains log information in the manual *JP1/Performance Management Reference*.

Note about operations in a logical host environment

In addition to the system log of Performance Management, you need the cluster software's log in order to check the control of Performance Management by the cluster software.

(2) Common message log

The common message log contains information that indicates the system status and reports problems. This log contains more detailed information than the system log. For details about the name of the output destination file for the common message log and the file size, see *7.3.2 Log file and folder list*. For details about the output format, see the chapter that explains log information in the manual *JP1/Performance Management Reference*.

Note about operations in a logical host environment

When Performance Management is running on a logical host, the common message log is output to a shared disk. Because log files located on the shared disk are inherited along with the system when a failover occurs, messages are recorded in the same log file. If you want to check a history of operations on Performance Management performed from the cluster software in addition to the Performance Management system log, the log of the cluster software is required.

(3) Operation status log

The operation status log contains the log information that is output by PFM - Web Console. For details about the name of the output destination file for the operation status log and the file size, see the chapter that explains error detection in the *JP1/Performance Management User's Guide*. For details about the output format, see the chapter that explains log information in the manual *JP1/Performance Management Reference*.

(4) Trace log

The trace log is used to collect log information that can be used for investigating a problem's history, and for measuring the processing time each process uses.

The trace log is output to the log file that each service of the Performance Management program has.

Note about operations in a logical host environment

When Performance Management is running on a logical host, the trace log is output to a shared disk. Because log files located on the shared disk are inherited along with the system when a failover occurs, messages are recorded in the same log file.

(5) Collected logs

Collected logs contain log information about processes related to record collection, and are output by PFM - RM for Virtual Machine. Logs are collected for the purpose of obtaining detailed information about these processes when a problem occurs. For details about the output destination, see *7.3.2(3) Collected log*.

7.3.2 Log file and folder list

This subsection explains the log information that is output by Performance Management programs. For details about the name of the output destination file for the operation status log and the file size, see the chapter that explains error detection in the *JP1/Performance Management User's Guide*.

(1) Common message log

For details about the common message log, see the chapter that describes details on log information in the *JP1/Performance Management User's Guide*.

(2) Trace log

Of the five types of log information that are output by Performance Management, the table below shows the service name or control name that outputs the trace log of PFM - RM for Virtual Machine, and the folder name at the destination where the log is saved.

Table 7–9: Trace log storage folder name

Log information type	Output source	Folder name
Trace log	Action Handler service	*installation-folder*`\bin\action\log\`
	Performance Management command	*installation-folder*`\tools\log\`
	Remote Monitor Collector service	*installation-folder*`\agt8\agent\`*instance-name*`\log\`
	Remote Monitor Store service	*installation-folder*`\agt8\store\`*instance-name*`\log\`
	Status Server service	*installation-folder*`\bin\statsvr\log\`

Log information type	Output source	Folder name
Trace log (in logical host environment)	Action Handler service	*environment-folder*# `\jp1pc\bin\action\log\`
	Performance Management command	*environment-folder*# `\jp1pc\tools\log\`
	Remote Monitor Collector service	*environment-folder*# `\jp1pc\agt8\agent\`*instance-name*`\log\`
	Remote Monitor Store service	*environment-folder*# `\jp1pc\agt8\store\`*instance-name*`\log\`

#

The environment folder is a folder that was specified on the shared disk when the logical host was created.

(3) Collected log

Of the five types of log information that are output by Performance Management, the table below shows log file names, disk usage, and the service name or control name that outputs the collected log of PFM - RM for Virtual Machine.

A collected log is created for each monitoring target.

Table 7–10: Collected log files

Log information type	Output source	Output destination	File name	Default disk usage#1 (megabytes)
Regular log	PFM - RM for Virtual Machine	*installation-folder*`\agt8\agent\`*instance-name*`\log\`	*VM_Type_monitoring-target-name*`{1|2|3|4|5|6|7|8}.log`#2	128
		installation-folder`\agt8\agent\`*instance-name*`\targets\`*monitoring-targetname*`\log\`	*jpcagt5virtage*`{1|2|3|4|5|6|7|8}.log`#2	128
Regular log (in logical host environment)	PFM - RM for Virtual Machine	*environment-folder*#3 `\jp1pc\agt8\agent\`*instance-name*`\log\`	*VM_Type_monitoring-target-name*`{1|2|3|4|5|6|7|8}.log`#2	128
		environment-folder#3 `\jp1pc\agt8\agent\`*instance-name*`\targets\`*monitoring-targetname*`\log\`	*jpcagt5virtage*`{1|2|3|4|5|6|7|8}.log`#2	128

#1

The maximum file size of the collected log can be checked or modified using the following procedures:

- Execute the `jpcconf inst setup` command.
- View the Remote Monitor Configuration property in the PFM - Web Console window.

For details about the procedure that uses the `jpcconf inst setup` command, see *2.4.2 Updating an instance environment*.

#2

The collected log uses the wrap-around file method. This method adds data to the latest log, without deleting the existing log. When the file size of a log exceeds the specified size, a new file with the log file number incremented by one is created. When the maximum number of logs that can be generated is reached (fixed at 8), the first file is overwritten.

#3

The environment folder is a folder that was specified on the shared disk when the logical host was created.

7.4 Data collected when a problem occurs

If a problem cannot be solved even after you have taken the actions prescribed in *7.2 Troubleshooting*, you need to collect data to investigate the cause of the problem, and then contact the system administrator. This section explains the data that needs to be collected when a problem occurs.

Performance Management provides a command for batch-collecting the necessary data. To collect PFM - RM for Virtual Machine data, use the `jpcras` command. Data that can be collected using the `jpcras` command is indicated in the tables below.

Note:

Data that can be collected using the `jpcras` command differs depending on the option specified for command execution. For details about the options that can be specified for the command and the types of data that can be collected, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

Notes about operations in a logical host environment

Note the following points when Performance Management is running on a logical host:

- When Performance Management is running on a logical host, the Performance Management logs are output to a shared disk. If the shared disk is online, you can use the `jpcras` command to batch-collect the logs from the shared disk.
- To investigate a problem that occurred during a failover, you need the data that was present before and after the failover. Therefore, you need data from both the active server and the standby server.
- To investigate Performance Management when it is running on a logical host, you need data from the cluster software. Because the operations of starting and stopping Performance Management are controlled by the cluster software in this case, you need to compare the behavior of the cluster software with that of Performance Management.

7.4.1 In Windows

(1) OS log information

The table below shows the types of OS log information that need to be collected.

Table 7–11: OS log information

Information type	Overview	Default file name	Collectable using the jpcras command?
System log	Windows event log	N/A	Y
	WMI log	*system-folder*`\system32\WBEM\Logs\*` #	Y
Process information	Process list	N/A	Y
System file	`hosts` file	*system-folder*`\system32\drivers\etc\hosts`	Y
	`services` file	*system-folder*`\system32\drivers\etc\services`	Y
OS information	System information	N/A	Y
	Network status	N/A	Y
	Host name	N/A	Y

Information type	Overview	Default file name	Collectable using the jpcras command?
OS information	Windows Firewall information	N/A	Y

Dump information	Log file for problem reports and solutions	`user-mode-process-dump-output-folder\program-name.process-ID.dmp`
		Example:
		`jpcagt8.exe.2420.dmp`
			N

Legend:

Y: Can be collected

N: Cannot be collected

N/A: Not applicable

#

If log files are set to be output to a different folder, collect data from the applicable folder.

(2) Performance Management information

The following types of information related to Performance Management need to be collected. If a problem occurred with a network connection, you also need to collect files from the target machine. The table below shows the Performance Management information.

Table 7–12: Performance Management information

Information type	Overview	Default file name	Collectable using the jpcras command?
Common message log	Message log output by Performance Management (Sequential file method)	*installation-folder*`\log\jpclog{01|02}`[#1]	Y
	Message log output by Performance Management (Wrap-around file method)	*installation-folder*`\log\jpclogw{01|02}`[#1]	Y
Configuration information	Each configuration information file	N/A	Y
	`jpctool service list` (`jpcctrl list`) command output result	N/A	Y
Version information	Product version	N/A	Y
	History information	N/A	Y
Database information	Remote Monitor Store service	`*.DB` and `*.IDX` files in the following folders: *installation-folder*`\agt8\store\`*instance-name*`\STPD` *installation-folder*`\agt8\store\`*instance-name*`\STPI`	Y

Information type	Overview	Default file name	Collectable using the jpcras command?
Trace log	Trace information of each Performance Management program service	N/A#2	Y
Collected log	Information when performance information was collected	*installation-folder*\agt8\agent*instance-name*\log*VM_Type_monitoring-target-name**.log *installation-folder*\agt8\agent*instance-name*\targets*monitoring-target-name*\log\jpcagt5virtage*.log	Y
Work data	Work data when performance information was collected	*installation-folder*\agt8\agent*instance-name*\targets\monitoring-target-name\work* *installation-folder*\agt8\agent*instance-name*\targets\monitoring-target-name\data*	Y
Collector plug-in data	Data for each collector plug-in	*installation-folder*\agt8\plugin\jpcagt5*.d* *installation-folder*\agt8\plugin\jpcagt8*.d*	Y
Installation log#3	Message log during installation	*system-folder*\TEMP\HCDINST*.LOG	N

Legend:

Y: Can be collected

N: Cannot be collected

N/A: Not applicable

#1

For the log file output method, see the chapter that explains error detection in the *JP1/Performance Management User's Guide*.

#2

For the trace log storage folder, see *7.3.2(2) Trace log*.

#3

Collect this information if installation fails.

(3) Registry information

Collect registry information related to HNTRLib2. The table below shows the information to collect.

Table 7–13: HNTRLib2 information (in Windows)

Information type	Overview	Registry key name	Collectable using the jpcras command?
Configuration information	Each type of configuration information	HKEY_LOCAL_MACHINE\SOFTWARE\HITACHI\HNTRLIB2\HNTR1 (other than the x64 environment) HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\HITACHI\HNTRLIB2\HNTR1 (in the x64 environment)	Y

Legend:

Y: Can be collected

(4) Collector plug-in information of Hitachi Compute Blade logical partitioning feature (logical partitioning feature only)

When the virtual environment to be monitored is one with logical partitioning feature, the information listed in the table below is required.

Table 7–14: Collector plug-in information of Hitachi Compute Blade logical partitioning feature

Overview	File name	Collectable using the jpcras command?
Temporary file for the LPAR Manager management command	*installation-folder*\agt8\agent*instance-name*\targets *monitoring-targetname*\log\HvmPerMon.bin	Y
Definition file for the LPAR Manager management command (master file)	*installation-folder*\agt8\plugin\jpcagt5virtage.d \jpcagt5virtageSetup.ini	Y
LPAR Manager management command	*installation-folder*\agt8\plugin\jpcagt5virtage.d \HvmSh.exe	Y
Definition file for the LPAR Manager management command (copy of the file)	*installation-folder*\agt8\agent*instance-name*\targets *monitoring-targetname*\jpcagt5virtageSetup.ini	Y

Legend:

Y: Can be collected

(5) LPAR Manager dump information (logical partitioning feature only)

If an error occurs during a connection to Hitachi Compute Blade logical partitioning feature, or there is a problem with performance data obtained from the environment with Hitachi Compute Blade logical partitioning feature, capture the LPAR Manager dump. For details about how to capture an LPAR Manager dump, see the topic that describes the LPAR Manager dump capture command in the User's Guide for each Hitachi Compute Blade device.

(6) Operation content

You need the following types of information about the operation that was taking place when the problem occurred:

- Detailed operation content
- Time at which the problem occurred
- Machine configuration (including the version of each OS, host name, and configuration of PFM - Manager and PFM - RM for Virtual Machine)
- Whether the problem is reproducible
- If the user logged on from PFM - Web Console, the Performance Management user name used for the logon

(7) Error information on the screen

Collect hard copies of the following:

- Operation window if an application error occurred
- Error message dialog box (including the description displayed when the **Details** button is clicked, if available)
- The Command Prompt window if the problem occurred during command execution

(8) User-mode process dump

If a Performance Management process stops due to an application error, obtain a user-mode process dump.

(9) Collecting problem reports

If a Performance Management process stops due to an application error, obtain a problem report.

(10) Other information

In addition to the information described above, the following types of information are needed:

- Argument specified for a command if the problem occurred during command execution

7.5 How to collect data

This section explains how to collect data when a problem occurs.

7.5.1 In Windows

(1) Collecting dump information

To collect dump information:

1. Open Task Manager.
2. Select the **Processes** tab.
3. Right-click the process name for which you want to collect dump information, and then select **Create Dump File**.
   Dump files are stored in the following folder:
   ```
   system-drive\Users\user-name\AppData\Local\Temp
   ```
4. Obtain the dump file from the folder created in step 3.
   If you have changed the environment variable settings so that dump files are output to a different folder, obtain the dump file from that folder.

(2) Executing the data collection command

To collect data for troubleshooting, use the `jpcras` command. The procedure for executing the data collection command follows. The operations described here must be executed by an OS user who has Administrator permissions.

To execute the data collection command:

1. Log on to the host on which the service is installed whose data you want to collected.
2. At the command prompt, execute the following command to enable the command extension function of the command interpreter:
   ```
   cmd /E:ON
   ```
3. Specify the data you want to collect and the storage folder, and then execute the `jpcras` command.
   In the following command specification example, all information that can be collected using the `jpcras` command is saved in the `c:\tmp\jpc\agt` folder:
   ```
   jpcras c:\tmp\jpc\agt all all
   ```

When the `jpcras` command is executed, the `jpctool service list -id * -host *` command is internally executed in order to obtain a PFM service list and check the startup status. If a firewall is installed between the command execution host and the host of another Performance Management system, or if the system configuration is large, execution of the `jpctool service list -id * -host *` command may take a long time. In such a case, by setting the environment variable `JPC_COLCTRLNOHOST` to 1, you can suppress the processing of the `jpctool service list -id * -host *` command, thereby shortening the command execution time.

For details about the `jpcras` command, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

Note on executing the command

If the user account control functionality (UAC) is enabled in the operating system, the User Account Control dialog box might be displayed during command execution. If it is displayed, click the **Continue** button to continue data collection, or click the **Cancel** button to cancel data collection.

(3) Executing the data collection command (in a logical host environment)

When Performance Management is running on a logical host, data is stored on a shared disk, and must therefore be collected from both the active server and the standby server.

To collect data for troubleshooting, use the `jpcras` command. The procedure for executing the data collection command follows. The operations described here must be executed by an OS user who has Administrator permissions.

To execute the data collection command in a logical host operation environment:

1. Bring the shared disk online.

   The data of the logical host is stored on the shared disk. On the executing node, make sure that the shared disk is online and then collect the data.

2. On both the active server and standby server, specify the data you want to collect and the storage folder, and then execute the `jpcras` command.

   In the following command specification example, all information that can be collected using the `jpcras` command is saved in the `c:\tmp\jpc\agt` folder:

   ```
   jpcras c:\tmp\jpc\agt all all
   ```

   If you execute the `jpcras` command without specifying the `lhost` argument, all Performance Management data is collected from the node's physical host and logical host. If a Performance Management instance is running in a logical host environment, log files on the shared disk are collected.

   If you execute the `jpcras` command on a node at which the shared disk is offline, files on the shared disk cannot be collected, but the process terminates normally without causing an error.

   Note:

   Collect data from both the executing node and the standby node by executing the data collection command. To investigate what happened before and after a failover, you need data from both the active server and the standby server.

   For details about the `jpcras` command, see the chapter that explains commands in the manual *JP1/Performance Management Reference*.

3. Collect cluster software data.

   This data is required in order to determine whether the problem occurred in the cluster software or in Performance Management. Collect data that can be used to investigate start, stop, and other control requests issued by the cluster software to Performance Management, and the results.

(4) Collecting a Windows event log

In the Windows Event Viewer window, collect the content of **System** and **Applications**.

(5) Checking the operation content

Check the operational details having to do with the problem and record them. You need to check the following types of information:

- Detailed operation content
- Time at which the problem occurred
- Machine configuration (including the version of each OS, host name, and configuration of PFM - Manager and PFM - RM for Virtual Machine)
- Whether the problem is reproducible
- If the user logged on from PFM - Web Console, the Performance Management user name used for the logon

(6) Collecting error information from windows

Obtain a hardcopy of the following information:

- The window operation when the application error occurred
- The error message dialog box
  If there is detailed information, also make a hardcopy of that information.
- The Command Prompt window or the Administrator Console window, if the error occurred during command execution
  To print the Command Prompt window or the Administrator Console window, specify the following settings for the Command Prompt Properties window:
  - **Edit options** on the **Options** page
    Select **Quick Edit Mode**.
  - **Layout** page
    For **Screen buffer size**, set **Height** to `500`.

(7) Collecting other information

Collect other necessary information.

- Content of **Accessories** > **System Tools** > **System Information**

7.6 Detecting problems within Performance Management

Using the health check function, Performance Management can detect errors that occur in Performance Management itself. The health check function monitors the operating status of the monitoring agent and the host on which the monitoring agent is running, and displays the monitoring result on PFM - Web Console as changes in the operating status of the monitoring agent.

Furthermore, using the PFM service automatic restart function, Performance Management can automatically restart a PFM service if it abnormally terminates due to a problem, and it can restart a PFM service at regular intervals.

To have the health check function monitor the operating status of the monitoring agent or to have the PFM service automatic restart function automatically restart a PFM service, you use the status management function, which checks the detailed status of Performance Management services. Therefore, the monitoring agent to be monitored must be a version that is compatible with the status management function, and this function must be enabled. To monitor the operating status of a host, there are no prerequisite conditions.

Errors in Performance Management can also be detected using JP1/Base, which is an integrated system monitoring product, to monitor Performance Management log files. Using these monitoring results, the system administrator can detect problems when they occur, identify their causes, and take the necessary corrective actions.

For details about how to detect errors in Performance Management, see the chapter that explains error detection in Performance Management in the *JP1/Performance Management User's Guide*.

7.7 Recovering from Performance Management system errors

When an error occurs on the Performance Management server, you need to restore it to the normal state that existed before the error occurred, based on backup files.

For details about how to restore the Performance Management server to the state that existed before the error, see the chapter that explains troubleshooting in the *JP1/Performance Management User's Guide*.

Appendixes

A. Estimating System Requirements

Before you create a system that uses PFM - RM for Virtual Machine, confirm that the machine on which that system will be created is capable of running PFM - RM for Virtual Machine.

A.1 Memory requirement

Memory requirements depend on how PFM - RM for Virtual Machine is set up and used. For details about the formula for estimating memory requirements, see the *Release Notes*.

A.2 Disk occupancy

Disk space requirements depend on the number of records used to collect performance data.

To estimate how much disk space PFM - RM for Virtual Machine requires, you must estimate the disk space requirements for either the entire system or for the Store database. For the estimation formulas for these disk space requirements, see the *Release Notes*.

B. List of Identifiers

When you operate PFM - RM for Virtual Machine or extract performance data from the Store database of PFM - RM for Virtual Machine, an identifier indicating PFM - RM for Virtual Machine is required in some cases. The table below shows the PFM - RM for Virtual Machine identifiers.

Table B–1: List of identifiers for PFM - RM for Virtual Machine

Application	Name	Identifier	Explanation
Commands, etc.	Product ID	`8`	A product ID is part of a service ID. A service ID is required when you use a command to check the system configuration of Performance Management or to back up performance data. For details about the service ID, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.
	Service key	`RMVM`	A service key is required when you use a command to start or stop PFM - RM for Virtual Machine. For details about the service key, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.
Help	Help ID	`pca8`	Help ID indicates a help item for PFM - RM for Virtual Machine.

C. List of Processes

This appendix lists the PFM - RM for Virtual Machine processes.

The PFM - RM for Virtual Machine processes are listed in the table below. The value following the process name indicates the number of processes that can run concurrently.

Note:

The processes that can run and the number of processes that can run concurrently also apply to PFM - RM running on a logical host.

Table C–1: PFM - RM for Virtual Machine process list

Process name (process count)	Function
`jpcagt8.exe` (*n*)	Remote Monitor Collector service process. This process starts for each instance of PFM - RM for Virtual Machine.
`jpc8collect.exe` (*n*)#1	Remote Monitor Collector collection process. This process starts for each instance of PFM - RM for Virtual Machine. This process runs as a subprogram of `jpcagt8.exe`.
`jpcagt8vmware.exe` (*n*)#2	Collection process for VMware. This process starts during collection for each monitoring target of PFM - RM for Virtual Machine that monitors VMware, and stops when collection is completed.
`jpcagt8hyperv.exe` (*n*)#25	Collection process for Hyper-V. This process starts during collection for each monitoring target of PFM - RM for Virtual Machine that monitors Hyper-V, and stops when collection is completed.
`jpcagt8kvm.exe` (*n*) #2	Collection process for KVM. This process starts during collection for each monitoring target of PFM -RM for Virtual Machine that monitors KVM, and stops when collection is completed.
`jpcagt8docker.exe` (*n*)#2	Collection process for Podman environment. This process starts during collection for each monitoring target configured on the Podman container instance of PFM - RM for Virtual Machine, and stops when collection is completed.
`jpcagt8podman.exe` (*n*)#2	Collection process for Docker environment. This process starts during collection for each monitoring target of PFM - RM for Virtual Machine that monitors Docker containers, and stops when collection is completed.
`jpcagt5virtage.exe` (*n*) #2	Collection process for logical partitioning feature. This process starts during collection for each monitoring target of PFM - RM for Virtual Machine that monitors logical partitioning feature, and stops when collection is completed.
`jpcsto.exe` (*n*)	Remote Monitor Store service process. This process starts for each instance of PFM - RM for Virtual Machine.
`stpqlpr.exe` (1)#3	Store database backup/export execution process
`hntr2srv.exe` (1)#4	Integrated trace startup service
`hntr2mon.exe` (1)#4	Integrated trace service

#1

This is a child process of the `jpcagt8.exe` process.

#2

This is a child process of the `jpc8collect.exe` process.

It starts 10 or less processes for each an instance. If there are 10 or less monitoring targets in an instance, it starts number of the processes depending on the monitoring targets.

#3

This is a child process of the `jpcsto.exe` process.

#4

This process is shared by programs that are installed on the same machine and that use the integrated trace log.

D. List of Port Numbers

This appendix describes the port numbers used by PFM - RM for Virtual Machine.

For details about the port numbers of PFM - Manager and PFM - Base, and the firewall passage direction, see the appendix section of the manual *JP1/Performance Management Reference*.

You can also change the port numbers to the desired numbers according to the user environment.

For details about how to change port numbers, see the chapter that explains installation and setup in the *JP1/Performance Management Planning and Configuration Guide*. The protocol used is TCP/IP.

Note:

- Performance Management supports static NAT (Basic NAT), in which addresses are translated on a 1-to-1 basis.
- Performance Management does not support dynamic NAT or NAPT (IP Masquerade or NAT+) that includes a port translation function.

D.1 Port numbers of PFM - RM for Virtual Machine

The following table shows the port numbers used by PFM - RM for Virtual Machine.

Table D–1: Port numbers used by PFM - RM for Virtual Machine

Port number	Service name	Parameter	Application
Automatic#1	Remote Monitor Store service	`jp1pcsto8[`*nnn*`]`#2	Used for recording performance data and acquiring historical reports
Automatic#1	Remote Monitor Collector service	`jp1pcagt8[`*nnn*`]`#2	Used for binding alarms and acquiring real-time reports

#1

Each time the service is restarted, a port number not being used by the system is automatically assigned.

#2

When multiple instances are created, serial numbers (*nnn*) are added to the second and subsequent instances. No serial number is added to the first instance created.

D.2 Firewall passage direction

(1) Setting up the firewall passage direction

When PFM - Manager and PFM - RM for Virtual Machine are installed across a firewall, set up fixed port numbers for all services of PFM - Manager and PFM - RM for Virtual Machine Furthermore. For more details, see the section describing the firewall passage direction in the manual *JP1/Performance Management Reference*.

(2) Firewall passage direction during communication between PFM - RM for Virtual Machine and VMware

To collect VMware information, PFM - RM for Virtual Machine needs to communicate with VMware. Therefore, if there is a firewall between PFM - RM for Virtual Machine and VMware, set the firewall to permit communication over the port of the monitoring target set on the PFM - RM for Virtual Machine host. The communication direction between PFM - RM for Virtual Machine and VMware is shown below.

Passage direction
PFM - RM for Virtual Machine (Remote Monitor Collector service) → VMware

Legend:

→: Direction for starting communication (connection) from the item on the left to the item on the right

The table below shows port numbers that can be used for communication with a monitoring target. For details, see *2.1.4(4) Setting up monitoring targets*.

Table D–2: Port numbers that can be used for communication with a monitoring target

Description	Setting item	Value that can be set	Default
VMware target port number	`Port`	0-65,535	Port = `0`#

\#

When `Port` = `0`, the system will actually use port number `443`, which is the default port number for HTTPS communication.

(3) Firewall passage direction during communication between PFM - RM for Virtual Machine and Hyper-V

To collect Hyper-V information, it is necessary for PFM - RM for Virtual Machine to use WMI to communicate with Hyper-V. Therefore, when PFM - RM for Virtual Machine and Hyper-V are installed across a firewall, passage through the firewall must be enabled.

Passage direction
PFM - RM for Virtual Machine (Remote Monitor Collector service) → Hyper-V

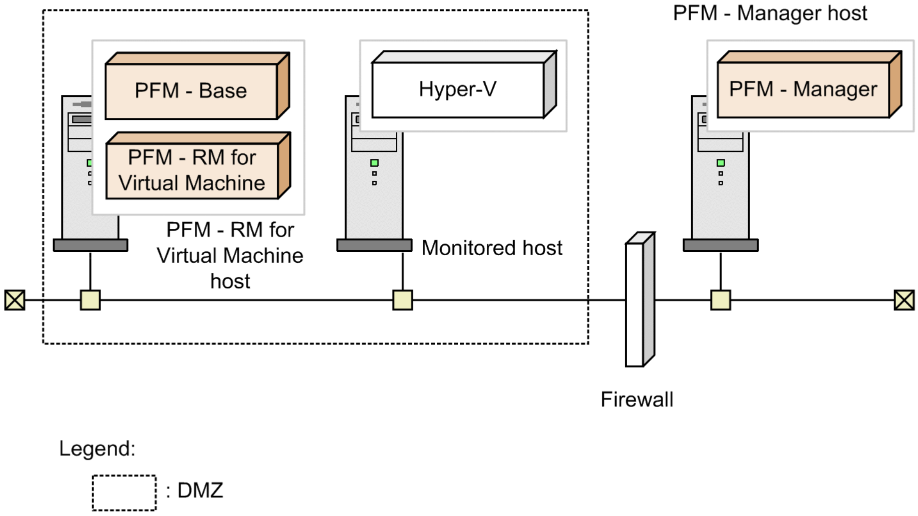
Legend:

→: Direction for starting communication (for connecting) from the item on the left to the item on the right

WMI uses DCOM. Because DCOM uses dynamic port allocation, the port used for DCOM must pass through the firewall. For details about the setup method, see the firewall product's documentation or check with the firewall product's developer.

Operation via a firewall is not suitable because individual WMI and DCOM requests cannot be separated. The following figure shows a recommended configuration.

Figure D–1: Example of configuration where the port used for DCOM passes through a firewall



(4) Firewall passage direction during communication between PFM - RM for Virtual Machine and KVM

To collect KVM information, it is necessary for PFM - RM for Virtual Machine to communicate via SSH. Therefore, when PFM - RM for Virtual Machine and KVM are installed across a firewall, set the firewall to permit communication over the port of the monitoring target set on the PFM - RM for Virtual Machine host. The communication direction between PFM - RM for Virtual Machine and KVM is shown below.

Passage direction
PFM - RM for Virtual Machine (Remote Monitor Collector service) → KVM

Legend:

→: Direction for starting communication (for connecting) from the item on the left to the item on the right

The table below shows port numbers that can be used for communication with a monitoring target. For details, see *2.1.4(4) Setting up monitoring targets*.

Table D–3: Port numbers that can be used for communication with a monitoring target

Description	Setting item	Value that can be set	Default
KVM port number for an SSH connection	`Port`	0-65,535	`Port =0`#

#

When `Port` = `0`, the system will actually use port number `22`, which is the default port number for SSH communication.

(5) Firewall passage direction during communication between PFM - RM for Virtual Machine and Docker Engine

To collect Docker environment information, it is necessary for PFM - RM for Virtual Machine to communicate via Docker Engine. Therefore, when PFM - RM for Virtual Machine and Docker environment are installed across a firewall, set the firewall to permit communication over the port of the monitoring target set on the PFM - RM for Virtual Machine host. The communication direction between PFM - RM for Virtual Machine and Docker Engine is shown below.

Passage direction
PFM - RM for Virtual Machine (Remote Monitor Collector service) → Docker Engine

Legend:

→: Direction for starting communication (for connecting) from the item on the left to the item on the right

The table below shows port numbers that can be used for communication with a monitoring target. For details, see *2.1.4(4) Setting up monitoring targets*.

Table D–4: Port numbers that can be used for communication with a monitoring target

Description	Setting item	Value that can be set	Default
Docker Engine target port number	Port	0~65,535	Port= `0`#

#

When `Port=0`, the Docker environment information cannot be collected. The port number of Docker Engine must be specified.

(6) Firewall passage direction during communication between PFM - RM for Virtual Machine and Podman environment

To collect Podman environment information, it is necessary for PFM - RM for Virtual Machine to communicate via SSH. Therefore, when PFM - RM for Virtual Machine and Podman environment are installed across a firewall, set the firewall to permit communication over the port of the monitoring target set on the PFM - RM for Virtual Machine host. The communication direction between PFM - RM for Virtual Machine and Podman environment is shown below.

Passage direction
PFM - RM for Virtual Machine (Remote Monitor Collector service) → Podman environment

Legend:

→: Direction for starting communication (for connecting) from the item on the left to the item on the right

The table below shows port numbers that can be used for communication with a monitoring target. For details, see *2.1.4(4) Setting up monitoring targets*.

Table D–5: Port numbers that can be used for communication with a monitoring target

Description	Setting item	Value that can be set	Default
Podman environment port number for an SSH connection	`Port`	0-65,535	`Port = 0`#

#

When `Port` = `0`, the system will actually use port number `22`, which is the default port number for SSH communication.

(7) Firewall passage direction during communication between PFM - RM for Virtual Machine and logical partitioning feature

To collect information from logical partitioning feature, PFM - RM for Virtual Machine communicates with hosts with logical partitioning feature through the UDP protocol. Therefore, if PFM - RM for Virtual Machine and hosts with logical partitioning feature are deployed across a firewall, the firewall must be configured to allow passage of such communication.

Port Numbers	Protocol type	Transmission type	Passage direction
623	UDP	Unicast	LPAR Manager management command for logical partitioning feature → logical partitioning feature
Automatic (Any port)			LPAR Manager management command for logical partitioning feature ← logical partitioning feature

Legend:

→: Direction for starting communication (for connecting) from the item on the left to the item on the right

←: Direction for starting communication (connection) from the item on the right to the item on the left

E. Properties of PFM - RM for Virtual Machine

This appendix describes the Remote Monitor Store service property list and Remote Monitor Collector service property list of PFM - RM for Virtual Machine. These lists are displayed on PFM - Web Console.

E.1 Remote Monitor Store service property list

The table below shows the Remote Monitor Store service property list of PFM - RM for Virtual Machine.

Table E–1: Remote Monitor Store service property list of PFM - RM for Virtual Machine

Folder name	Property name	Explanation
--	First Registration Date	Shows the first date/time on which the service was recognized by PFM - Manager.
	Last Registration Date	Shows the last date/time on which the service was recognized by PFM - Manager.
General	--	Stores information such as host names and folder names. The properties stored in this folder cannot be modified.
	Directory	Shows the name of the current folder in which the service is running.
	Host Name	Shows the name of the host on which the service is running.
	Process ID	Shows the process ID of the service.
	Physical Address	Shows the IP address and port number of the host on which the service runs if IPv6 communication is disabled.
	Physical Address(IPv4)	Shows the IPv4-format IP address of the host on which the service runs if IPv6 communication is enabled.
	Physical Address(IPv6)	Shows the IPv6-format IP address of the host on which the service runs if IPv6 communication is enabled.
	Port Number	Shows the number of the port that is used by the service if IPv6 communication is enabled.
	User Name	Shows the name of the user that executed the service process.
	Time Zone	Shows the time zone used by the service.
System	--	Stores information on the OS under which the service was started. The properties stored in this folder cannot be modified.
	CPU Type	Shows the CPU type.
	Hardware ID	Shows the hardware ID.
	OS Type	Shows the OS type.
	OS Name	Shows the OS name.
	OS Version	Shows the OS version.
Network Services	--	Stores information about the communication common library of Performance Management. The properties stored in this folder cannot be modified.

Folder name	Property name	Explanation
`Network Services`	`Build Date`	Shows the Remote Monitor Store service creation date.
	`INI File`	Shows the name of the folder that stores the `jpcns.ini` file.
`Network Services` / `Service`	--	Stores information about the service. The properties stored in this folder cannot be modified.
	`Description`	Shows additional service information, such as the host name and service type. Shows the host name in the following format: *instance-name_host-name*
	`Local Service Name`	Shows the service ID.
	`Remote Service Name`	Shows the service ID of the Master Manager service of the PFM - Manager host at the connection destination.
	`EP Service Name`	Shows the service ID of the Correlator service of the PFM - Manager host at the connection destination.
`RetentionEx`	--	When the Store version is 2.0, a data retention period is set in this folder. For details, see the chapter that explains management of operation monitoring data in the *JP1/Performance Management User's Guide*.
`RetentionEx` / `Product Interval -` *record-ID-of-PI-record-type*	--	Specifies the PI record type record retention period.
	`Period - Minute Drawer (Day)`	Specifies the minute-by-minute PI record type record retention period. The value that can be specified is 0-366 days, in 1-day increments.
	`Period - Hour Drawer (Day)`	Specifies the hourly PI record type record retention period. The value that can be specified is 0-366 days, in 1-day increments.
	`Period - Day Drawer (Week)`	Specifies the daily PI record type record retention period. The value that can be specified is 0-522 weeks, in 1-week increments.
	`Period - Week Drawer (Week)`	Specifies the weekly PI record type record retention period. The value that can be specified is 0-522 weeks, in 1-week increments.
	`Period - Month Drawer (Month)`	Specifies the monthly PI record type record retention period. The value that can be specified is 0-120 months, in 1-month increments.
	`Period - Year Drawer (Year)`	Specifies the yearly PI record type record retention period. Although `10` will be displayed as the fixed value, there is no limit.
`RetentionEx` / `Product Detail -` *record-ID-of-PD-record-type*	`Period (Day)`	Specifies a performance data retention period for each PD record type record ID. Specify an integer between 0 and 366 for the retention period (number of days).
`Disk Usage`	--	Stores the disk size being used by each database. The properties stored in this folder show the disk usage when the properties are displayed. The properties stored in this folder cannot be modified.
	`Product Interval`	Shows the disk size being used by PI record type records.

Folder name	Property name	Explanation
Disk Usage	Product Detail	Shows the disk size being used by PD record type records.
	Product Alarm	Shows the disk size being used by PA record type records. This property is not used by PFM - RM for Virtual Machine.
	Product Log	Shows the disk size being used by PL record type records. This property is not used by PFM - RM for Virtual Machine.
	Total Disk Usage	Shows the disk size being used in the entire database.

Legend:

--: Not applicable

E.2 Remote Monitor Collector service property list

The table below shows the Remote Monitor Collector service property list of PFM - RM for Virtual Machine.

Table E–2: Remote Monitor Collector service property list of PFM - RM for Virtual Machine

Folder name	Property name	Explanation
--	First Registration Date	Shows the first date/time on which the service was recognized by PFM - Manager.
	Last Registration Date	Shows the last date/time on which the service was recognized by PFM - Manager.
	Data Model Version	Shows the data model version.
General	--	Stores information such as host names and folder names. The properties stored in this folder cannot be modified.
	Directory	Shows the name of the current folder in which the service is running.
	Host Name	Shows the name of the host on which the service is running.
	Process ID	Shows the process ID of the service.
	Physical Address	Shows the IP address and port number of the host on which the service runs if IPv6 communication is disabled.
	Physical Address(IPv4)	Shows the IPv4-format IP address of the host on which the service runs if IPv6 communication is enabled.
	Physical Address(IPv6)	Shows the IPv6-format IP address of the host on which the service runs if IPv6 communication is enabled.
	Port Number	Shows the number of the port that is used by the service if IPv6 communication is enabled.
	User Name	Shows the name of the user that executed the service process.
	Time Zone	Shows the time zone used by the service.
System	--	Stores information on the OS under which the service was started. The properties stored in this folder cannot be modified.

Folder name		Property name	Explanation
System		CPU Type	Shows the CPU type.
		Hardware ID	Shows the hardware ID.
		OS Type	Shows the OS type.
		OS Name	Shows the OS name.
		OS Version	Shows the OS version.
Network Services		--	Stores information about the communication common library of Performance Management. The properties stored in this folder cannot be modified.
		Build Date	Shows the Remote Monitor Collector service creation date.
		INI File	Shows the name of the folder that stores the jpcns.ini file.
Network Services	Service	--	Stores information about the service. The properties stored in this folder cannot be modified.
		Description	Shows the host name in the following format: *instance-name_host-name*
		Local Service Name	Shows the service ID.
		Remote Service Name	Shows the service ID of the Remote Monitor Store service to which Remote Monitor Collector service connects.
		EP Service Name	Shows the service ID of the Correlator service of the PFM - Manager host at the connection destination.
		AH Service Name	Shows the service ID of the Action Handler service that is on the same host.
JP1 Event Configurations		--	Stores the properties of JP1 event generation conditions.
		Each service	From the list items in the Remote Monitor Collector service, Remote Monitor Store service, Action Handler service, and Status Server service, select Yes or No to specify whether each service should generate JP1 system events.
		JP1 Event Send Host	Specifies the target event server name for JP1/Base. You can specify only an event server that is running on the logical or physical host on the same machine as the Action Handler service. The values that can be specified are alphanumeric characters, the period (.), and the dash (-) for a total of between 0 and 255 bytes. If a value outside the range is specified, it is assumed that no value is specified. If value specification is omitted, the host on which the Action Handler service is running is used as the event-generation source host. If localhost is specified, it is assumed that the physical host is specified.
		Monitoring Console Host	When the browser of PFM - Web Console is to be started during the startup of the monitor of JP1/IM - Manager, this property specifies the PFM - Web Console host to be started. The values that can be specified are alphanumeric characters, the period (.), and the dash (-) for a total of between 0 and 255 bytes. If a value outside the range is specified, it is assumed that no value is specified. If value specification is omitted, the PFM - Manager host at the connection destination is assumed.

Folder name		Property name	Explanation
JP1 Event Configurations		Monitoring Console Port	Specifies the port number (HTTP request port number) of PFM - Web Console to be started. A value between 1 and 65,535 can be specified. If a value outside the range is specified, it is assumed that no value is specified. If value specification is omitted, 20,358 is set.
		Monitoring Console Https	Specifies whether to use HTTPS-encrypted communication to access PFM - Web Console when PFM - Web Console is started by JP1/IM - Manager monitor startup. By default, this property is set to **No**. • **Yes**: Use encrypted communication. • **No**: Do not use encrypted communication.
JP1 Event Configurations	Alarm	JP1 Event Mode	Specifies whether a JP1 system event or JP1 user event will be generated when an alarm status changes. • JP1 User Event: Generates a JP1 user event. • JP1 System Event: Generates a JP1 system event.
Detail Records		--	Stores PD record type record properties. The record IDs of collected records are shown in bold characters.
Detail Records	*record-ID*#1	--	Stores record properties.
		Description	Shows record descriptions. This property cannot be modified.
		Log	From the list items, select Yes or No to specify whether to record records in the Remote Monitor Store database. If this value is Yes and Collection Interval is greater than 0, records are recorded in the database.
		Log(ITSLM)	Shows Yes or No to indicate whether records are to be saved in the Store database of PFM - RM for Virtual Machine from JP1/SLM - Manager. This value is fixed to No.
		Monitoring(ITSLM)	Shows the value (Yes or No) that was set in JP1/SLM - Manager to indicate whether records are to be sent to JP1/SLM - Manager. This value is fixed to No.
		Collection Interval	Specifies the data collection interval. The value that can be specified is 0-2,147,483,647 seconds, in 1-second increments. If 0 is specified, no data is collected.
		Collection Offset	Specifies the offset for starting data collection. The value that can be specified is 0-32,767 seconds, in 1-second increments, and must be within the value range specified in Collection Interval. The data collection recording time is the same as Collection Interval and does not depend on the Collection Offset value.
		Over 10 Sec Collection Time	This property is displayed only when historical data collection takes priority over real-time report display processing. (The history collection priority function is enabled.) #2 Whether collection of records takes 10 seconds or longer is displayed as Yes or No. • Yes: It sometimes takes 10 seconds or longer. • No: It takes less than 10 seconds. You cannot change this property.
		LOGIF	Specifies the condition for recording records in the database. Only those records that satisfy the condition are recorded in the database. Shows the conditional expression (character

Folder name		Property name	Explanation
Detail Records	*record-ID*#1	LOGIF	string) created in the LOGIF Expression Editor window, which opens when **LOGIF** is clicked in the lower frame of the Properties window of the service displayed from the **Services** page of PFM - Web Console.
Interval Records		--	Stores PI record type record properties. The record IDs of collected records are shown in bold characters.
Interval Records	*record-ID*#1	--	Stores record properties.
		Description	Shows record description. This property cannot be modified.
		Log	From the list items, select Yes or No to specify whether to record records in the Remote Monitor Store database. If this value is Yes and Collection Interval is greater than 0, records are recorded in the database.
		Log(ITSLM)	Shows Yes or No to indicate whether records are to be saved in the Store database of PFM - RM for Virtual Machine from JP1/SLM - Manager. This value is fixed to No.
		Monitoring(ITSLM)	Shows the value (Yes or No) that was set in JP1/SLM - Manager to indicate whether records are to be sent to JP1/SLM - Manager. This value is fixed to No.
		Collection Interval	Specifies the data collection interval. The value that can be specified is 0-2,147,483,647 seconds, in 1-second increments. If 0 is specified, no data is collected.
		Collection Offset	Specifies the offset for starting data collection. The value that can be specified is 0-32,767 seconds, in 1-second increments, and must be within the value range specified in Collection Interval. The data collection recording time is the same as Collection Interval and does not depend on the Collection Offset value.
		Over 10 Sec Collection Time	This property is displayed only when historical data collection takes priority over real-time report display processing. (The history collection priority function is enabled.) #2 Whether collection of records takes 10 seconds or longer is displayed as Yes or No. • Yes: It sometimes takes 10 seconds or longer. • No: It takes less than 10 seconds. You cannot change this property.
		LOGIF	Specifies the condition for recording records in the database. Only those records that satisfy the condition are recorded in the database. Shows the conditional expression (character string) created in the LOGIF Expression Editor window, which opens when **LOGIF** is clicked in the lower frame of the Properties window of the service displayed from the **Services** page of PFM - Web Console.
Log Records		--	Stores PL record type record properties. This folder is not used by PFM - RM for Virtual Machine because it does not support PL records.
Monitoring Targets		--	Stores the properties of hosts that are monitored by PFM - RM for Virtual Machine.
Monitoring Targets	*monitoring-target-names*	--	Shows a description of each monitoring target.

Folder name		Property name	Explanation
Monitoring Targets	*monitoring-target-names*	Target Name	Shows the name of a monitoring target. This property cannot be changed.
		Target Host	Shows the name of a monitoring-target host. This property cannot be changed.
Health Check Configurations		Health Check for Target Hosts	Specifies whether to perform polling for monitoring-target hosts. This setting is applied to all monitoring-target hosts within the instance.
Restart Configurations		--	Specifies the PFM service automatic restart condition. For details about the PFM service automatic restart function, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.
		Restart when Abnormal Status	Specifies whether to automatically restart a service when the Status Server service cannot normally collect the status of the Action Handler service, Remote Monitor Collector service, or Remote Monitor Store service.
		Restart when Single Service Running	Specifies whether to automatically restart a service when either the Remote Monitor Store service or Remote Monitor Collector service has not been started.
Restart Configurations	Remote Monitor Collector	Auto Restart	Specifies whether to use the automatic restart function for the Remote Monitor Collector service.
		Auto Restart - Interval (Minute)	Specifies the interval for checking the operating status of services (in minutes) when the automatic restart function is used.
		Auto Restart - Repeat Limit	Specifies the number of times auto restart should be consecutively tried when the automatic restart function is used.
		Scheduled Restart	Specifies whether to use the scheduled restart function for the Remote Monitor Collector service.
		Scheduled Restart - Interval	Specifies the restart interval when the scheduled restart function is used.
		Scheduled Restart - Interval Unit	Specifies the restart interval unit when the scheduled restart function is used.
		Scheduled Restart - Origin - Year	Specifies an integer between 1971 and 2035 as the year for restart.
		Scheduled Restart - Origin - Month	Specifies an integer between 1 and 12 as the month for restart.
		Scheduled Restart - Origin - Day	Specifies an integer between 1 and 31 as the day for restart.
		Scheduled Restart - Origin - Hour	Specifies an integer between 0 and 23 as the time (hour) for restart.
		Scheduled Restart - Origin - Minute	Specifies an integer between 0 and 59 as the time (minutes) for restart.

Folder name		Property name	Explanation
Restart Configurations	Remote Monitor Store	Auto Restart	Specifies whether to use the automatic restart function for the Remote Monitor Store service.
		Auto Restart - Interval (Minute)	Specifies the interval for checking the operating status of services (in minutes) when the automatic restart function is used.
		Auto Restart - Repeat Limit	Specifies the number of times auto restart should be consecutively tried when the automatic restart function is used.
		Scheduled Restart	Specifies whether to use the scheduled restart function for the Remote Monitor Store service.
		Scheduled Restart - Interval	Specifies the restart interval when the scheduled restart function is used.
		Scheduled Restart - Interval Unit	Specifies the restart interval unit when the scheduled restart function is used.
		Scheduled Restart - Origin - Year	Specifies an integer between 1971 and 2035 as the year for restart.
		Scheduled Restart - Origin - Month	Specifies an integer between 1 and 12 as the month for restart.
		Scheduled Restart - Origin - Day	Specifies an integer between 1 and 31 as the day for restart.
		Scheduled Restart - Origin - Hour	Specifies an integer between 0 and 23 as the time (hour) for restart.
		Scheduled Restart - Origin - Minute	Specifies an integer between 0 and 59 as the time (minutes) for restart.
	Action Handler	Auto Restart	Specifies whether to use the automatic restart function for the Action Handler service.
		Auto Restart - Interval (Minute)	Specifies the interval for checking the operating status of services (in minutes) when the automatic restart function is used.
		Auto Restart - Repeat Limit	Specifies the number of times auto restart should be consecutively tried when the automatic restart function is used.
		Scheduled Restart	Specifies whether to use the scheduled restart function for the Action Handler service.
		Scheduled Restart - Interval	Specifies the restart interval when the scheduled restart function is used.
		Scheduled Restart - Interval Unit	Specifies the restart interval unit when the scheduled restart function is used.

Folder name		Property name	Explanation
Restart Configurations	Action Handler	Scheduled Restart - Origin - Year	Specifies an integer between 1971 and 2035 as the year for restart.
		Scheduled Restart - Origin - Month	Specifies an integer between 1 and 12 as the month for restart.
		Scheduled Restart - Origin - Day	Specifies an integer between 1 and 31 as the day for restart.
		Scheduled Restart - Origin - Hour	Specifies an integer between 0 and 23 as the time (hour) for restart.
		Scheduled Restart - Origin - Minute	Specifies an integer between 0 and 59 as the time (minutes) for restart.
ITSLM Connection Configuration		--	Shows information about the JP1/SLM - Manager to link with.
ITSLM Connection Configuration	ITSLM Connection	--	Shows information about the JP1/SLM - Manager to connect to.
		ITSLM Host	Shows the host name of the connected JP1/SLM - Manager. This item is displayed only when a connection to JP1/SLM - Manager is established.
		ITSLM Port	Shows the port number of the connected JP1/SLM - Manager. This item is displayed only when a connection to JP1/SLM - Manager is established.
	MANAGE ITSLM CONNECTION	--	Shows whether to disconnect from the JP1/SLM - Manager.
		DISCONNECT ITSLM CONNECTION	From a list, select the name of the JP1/SLM - Manager host that you want to disconnect from. If you select a blank item, no action occurs. If no JP1/SLM - Manager hosts are connected, the list displays only a blank item.
Remote Monitor Configuration		--	Stores configuration properties specific to PFM - RM for Virtual Machine.
Remote Monitor Configuration	Remote Monitor	--	Shows an overview of the Remote Monitor Collector service.
		Product	Shows the product ID 8. This property cannot be modified.
		Instance	Shows the instance name specified by the jpcconf inst setup command. This property cannot be modified.
		VM_Type	Shows the type of the virtual environment to be monitored. This property cannot be modified. The following character string is displayed: • vmware (indicating VMware ESX or VMware ESXi) • hyperv (indicating Hyper-V) • virtage (indicating logical partitioning feature) • kvm (indicating KVM) • docker (indicating Docker environment) • podman (indicating Podman environment)
		Interval	Specifies the interval at which the collection process collects information. The value of this property can be changed.

Folder name		Property name	Explanation
Remote Monitor Configuration	Remote Monitor	Offset	Specifies the offset value to apply before the first collection cycle. The value of this property can be changed.
		Std_Category	Specifies whether the collection process outputs the basic information (PI, VI, and VM records) to a temporary performance information file. The value of this property can be changed.
		Cpu_Category	Specifies whether the collection process outputs the CPU information (HCI and VCI records) to a temporary performance information file. The value of this property can be changed.
		Memory_Category	Specifies whether the collection process outputs the memory information (HMI and VMI records) to a temporary performance information file. The value of this property can be changed.
		Disk_Category	Specifies whether the collection process outputs the disk information (HPDI, VPDI, HLDI, and VLDI records) to a temporary performance information file. The value of this property can be changed.
		Network_Category	Specifies whether the collection process outputs the network information (HNI and VNI records) to a temporary performance information file. The value of this property can be changed.
		HostUserID	Displays the user ID of the PFM - RM host. The value of this property can be changed.
		HostPassword	Displays **** (fixed). This property cannot be changed.
		HostDomain	Displays the domain name of the PFM - RM host. The value of this property can be changed.
		SSH_Type	Displays the value specified for SSH_Type during setup of the instance environment. This property cannot be changed.
		SSH_Client	Specifies the absolute path of the execution module (plink.exe or ssh.exe) of the SSH client (PuTTY or OpenSSH) specified in SH_Type. The value of this property can be changed.
		Log_Size	Specifies the maximum size of each file in the collected log (megabytes). The value of this property can be changed.
		UseVcpuMax	Specifies which is to be used as the CPU resource clock frequency: the frequency assigned to the virtual machine or the clock frequency of the physical CPU. The value of this property can be changed.

Legend:

--: Not applicable

#1

For the folder name, a database ID excluding the record ID is displayed. For the record ID of each record, see *5. Records*.

#2

For details, see the chapter on troubleshooting in the JP1/Performance Management User's Guide.

E.3 Remote agent and group agent property list

The following table lists the properties of PFM - RM for Virtual Machine remote agents and group agents.

Table E–3: List of properties of PFM - RM for Virtual Machine remote agents and group agents

Folder name		Property name	Explanation	Remote Agent	Group Agent
--		First Registration Date	Shows the time at which the service was first recognized by PFM - Manager.	D	D
		Last Registration Date	Shows the time at which the service was last recognized by PFM - Manager.	D	D
		Data Model Version	Shows the data model version.	D	D
Remote Monitoring		--	Stores the properties for remote agents and group agents.	D	D
		Agent Type	Shows the agent type: • Remote Agent The agent is a remote agent. • Group Agent The agent is a group agent.	D	D
		Remote Monitor Name	Shows the service ID of PFM - RM for Virtual Machine.	D	D
		Target Name	Shows the monitoring target name.	D	N
		Target Host	Shows the monitoring-target host name.	D	N
		Group Name	Shows the group name.	N	D
		Primary Host	Shows the primary host name.	N	D
		Grouping Targets	Lists the names of targets to be consolidated.	N	D
Detail Records		--	Stores the properties of records of the PD record type. The IDs of records that have been collected are indicated in bold type.	D	D
Detail Records	*record-ID*#2	--	Stores record properties.	D	D
		Description	Shows a description of the record.	D	D
		Log#1	Shows Yes or No to indicate whether to save records in the Store database of PFM - RM for Virtual Machine. If the value of this property is Yes and the Collection Interval value is greater than 0, records are saved in the database.	U	U
		Log(ITSLM)#1	Shows Yes or No to indicate whether to save records received from JP1/SLM - Manager in the Store database of PFM - RM for Virtual Machine. If the value of this property is Yes and the Collection Interval value is greater than 0, records are saved in the database. The value of this property cannot be changed.	D	D

Folder name		Property name	Explanation	Remote Agent	Group Agent
Detail Records	*record-ID*#2	Monitoring(ITSLM)	Shows the value (Yes or No) that was set from JP1/SLM - Manager to indicate whether records are to be sent to JP1/SLM - Manager. The value of this property cannot be changed.	D	D
		Collection Interval	Specifies the data collection interval. You can specify a value from 0 through 2,147,483,647 in seconds. If you specify 0, data collection is disabled.	D#3	D#3
		Collection Offset	Specifies the offset value to apply before the first collection cycle. The value is in seconds, and can be from 0 through 32,767, but must be less than the value specified in Collection Interval. The time at which the collected data is recorded matches the collection interval time, regardless of the offset value.	D#3	D#3
		Over 10 Sec Collection Time	This property is only displayed if collection of historical data takes precedence over the display processing of real-time reports (if the functionality that prioritizes the collection of historical data is enabled).#5 Whether record collection might require 10 seconds or more is indicated by Yes or No. • Yes: Might require 10 seconds or more. • No: Does not require 10 seconds. The value of this property cannot be changed.	D	D
		Realtime Report Data Collection Mode	Specifies the display mode of the real-time report. • Reschedule: Reschedule mode is used. • Temporary Log: Temporary save mode is used. Note that temporary save mode (Temporary Log) must be specified for records whose Over 10 Sec Collection Time value is Yes.	U	U
		LOGIF	Specifies the condition for recording records in the database. Only those records that satisfy the condition are recorded in the database. The conditional expression (string) created in the LOGIF Expression Editor window is displayed. This window opens when **LOGIF** is clicked in the lower frame of the Properties window of the service displayed from the **Services** page of PFM - Web Console.	D#3	D#3

Folder name		Property name	Explanation	Remote Agent	Group Agent
`Interval Records`		--	Stores the properties of records of the PI record type. The IDs of records that have been collected are indicated in bold type.	D	D
`Interval Records`	*record-ID*#2	--	Stores record properties.	D	D
		`Description`	Shows a description of the record. The value of this property cannot be changed.	D	D
		`Log`#1	Shows `Yes` or `No` to indicate whether to save records in the Store database of PFM - RM for Virtual Machine. If the value of this property is `Yes` and the Collection Interval value is greater than 0, records are saved in the database.	U	U
		`Log(ITSLM)`#1	Shows `Yes` or `No` to indicate whether to save records received from JP1/SLM - Manager in the Store database of PFM - RM for Virtual Machine. If the value of this property is `Yes` and the Collection Interval value is greater than 0, records are saved in the database. The value of this property cannot be changed.	D	D
		`Monitoring(ITSLM)`	Shows the value (`Yes` or `No`) that was set from JP1/SLM - Manager to indicate whether records are to be sent to JP1/SLM - Manager. The value of this property cannot be changed.	D	D
		`Collection Interval`	Specifies the data collection interval. You can specify a value from 0 through 2,147,483,647 in seconds. If you specify `0`, data collection is disabled.	D#3	D#3
		`Collection Offset`	Specifies the offset value to apply before the first collection cycle. The value is in seconds, and can be from 0 through 32,767, but must be less than the value specified in Collection Interval. The time at which the collected data is recorded matches the collection interval time, regardless of the offset value.	D#3	D#3
		`Over 10 Sec Collection Time`	This property is only displayed if collection of historical data takes precedence over the display processing of real-time reports (if the functionality that prioritizes the collection of historical data is enabled).#5 Whether record collection might require 10 seconds or more is indicated by `Yes` or `No`. • `Yes`: Might require 10 seconds or more. • `No`: Does not require 10 seconds. The value of this property cannot be changed.	N	N

Folder name			Property name	Explanation	Remote Agent	Group Agent
Interval Records	*record-ID*#2		Realtime Report Data Collection Mode	Specifies the display mode of the real-time report. • Reschedule: Reschedule mode is used. • Temporary Log: Temporary save mode is used. Note that temporary save mode (Temporary Log) must be specified for records whose Over 10 Sec Collection Time value is Yes.	U	U
			LOGIF	Specifies the condition for recording records in the database. Only those records that satisfy the condition are recorded in the database. The conditional expression (string) created in the LOGIF Expression Editor window is displayed. This window opens when **LOGIF** is clicked in the lower frame of the Properties window of the service displayed from the **Services** page of PFM - Web Console.	D#3	D#3
Log Records			--	Stores the properties of records of the PL record type. This property is not used in PFM - RM for Virtual Machine.	D	D
Remote Monitor Configuration			--	Stores configuration properties specific to the monitoring target.	D	N
Remote Monitor Configuration	Target		--	Shows an overview of the remote agent service.	D	N
	Target	*monitoring-target-names*	--	Shows the monitoring target name.	D	N
			VM_Host	Shows the host name of the monitoring-target physical server.	U	N
			Security	For VMware and Docker environment#4: Specifies whether SSL/TLS communication is used for the monitored physical server. Either of the following values can be specified: • Security = 0 (SSL/TLS is not used.) 0 cannot be specified, must use SSL/TLS connection. • Security = 1 (SSL/TLS is used.) If there is a problem with the server certificate, performance data continues to be collected, and a warning message is output to the common message log. When the warning message is output, you must change the invalid certificate to a valid one.	U	N

Folder name			Property name	Explanation	Remote Agent	Group Agent
Remote Monitor Configuration	Target	*monitoring-target-names*	Security	• Security = 2 (SSL/TLS is used.) If there is a problem with the server certificate, performance data continues to be collected, but a warning message is not output to the common message log. If the default certificate of VMware is used for operation, output of warning messages can be suppressed by specifying the value 2. If you want operation without using the server certificate of Docker environment, specify the value 2 so that a warning message can be prevented from outputting. • Security = 3 (SSL/TLS is used.) If there is a problem with the server certificate, performance data is not collected, and a warning message is output to the common message log. If you want to use a valid certificate for operation and collect performance data only from trusted monitoring targets, specify the value 3. For Hyper-V, logical partitioning feature, KVM and Podman environment: This item is always assumed to be 0.	U	N
			Port	Specifies the number of the port that is to be used for communication with the monitoring-target physical server. If VMware is being used and the port number is set to 0, the system sets HTTPS default port number 443. For logical partitioning feature, the fixed port number 623 is used. This port number cannot be changed. Even if a value other than 623 is specified for this property, the system uses the port number 623. If KVM is being used and the value of this property is 0, the system uses the SSH default port number of 22. If Docker environment is being used and the value 0 is specified for this property, monitoring is not performed because no port number is specified. If Podman environment is being used and the value of this property is 0, the system uses the SSH default port number of 22.	U	N

Folder name			Property name	Explanation	Remote Agent	Group Agent
Remote Monitor Configuration	Target	*monitoring-target-names*	UserID	Specifies the user ID that is used to connect to the monitoring-target physical server. Note that this property is valid only when the monitoring-target host is a VMware, Hyper-V, or KVM host.	U	N
			Password	Shows the fixed string `****` as the password that is used to connect to the monitoring-target physical server. The value of this property cannot be changed. Note that this property is valid only when the monitoring-target host is a VMware or Hyper-V host.	D	N
			Domain	Specifies the name of the domain that the monitoring-target physical server belongs to. Note that this property is valid only when the monitoring-target host is a Hyper-V host.	U	N
			Private_Key_File	Specifies the absolute path name of the private key file that is used for SSH public key based authentication. Note that this property is valid only when the monitoring-target host is a KVM or Podman environment host.	U	N

Legend:

--: Not applicable

U: This property is displayed and can be updated.

D: This property is displayed but cannot be updated.

N: This property is not displayed (and cannot be updated).

#1

If either of these properties is `Yes`, records are saved in the Store database.

#2

The record ID excluding the database ID is displayed as the folder name. For the ID of each record, see *5. Records*.

#3

The value set in PFM - RM for Virtual Machine is displayed.

#4

If there is a problem with the client certificate, the connection is rejected on the Docker environment side. Thus, performance data cannot be collected.

#5

For details, see the chapter on troubleshooting in the JP1/Performance Management User's Guide.

F. List of Files and Folders

This appendix describes the file and folder list of PFM - RM for Virtual Machine.

F.1 PFM - RM for Virtual Machine file and folder list

The table below shows the PFM - RM for Virtual Machine file and folder list.

Table F–1: PFM - RM for Virtual Machine file and folder list

Folder name	File name	Explanation
installation-folder`\`	`instagt8.ini`	Intermediate file for internal processing
installation-folder`\agt8\`	--	PFM - RM for Virtual Machine base folder
	`insrules.dat`	Intermediate file for internal processing
	`jpcagtras.bat`	Maintenance data collection program
	`PATCHLOG.TXT`	Intermediate file for internal processing#1
	`Readme_`*language code*`.txt`	README text file
	`VERSION.TXT`	Version information file
installation-folder`\agt8\agent\`	--	Remote Monitor Collector service base folder
	`jpcagt.ini.instmpl`	Intermediate file for internal processing
	`jpcagt8.exe`	Remote Monitor Collector service execution program
	`jpc8collect.exe`	Remote Monitor Collector record collection program
	`jpcagt8hcc.dll`	Hitachi common library file
	`inssetup.bat.instmpl`	Intermediate file for internal processing
	`agtlist.ini`	Intermediate file for internal processing#2
	`GARULES.DAT`	Group agent creation rule file
	`group.ini.tmpl`	Group agent settings template file

Folder name	File name	Explanation
installation-folder`\agt8\agent\`	`target.ini.tmpl`	Monitoring-target settings template file
	`targetrules.dat`	Monitoring-target creation rule file
	`jpcagt8cfg.ini`	PFM - RM for Virtual Machine setup file
	`jpcagt8cfg.ini.model`	Model file for the PFM - RM for Virtual Machine setup file
installation-folder`\agt8\plugin\`	--	Folder for the Collector plug-ins
	`jpcagt8hyperv.exe`	Execution program of the Collector plug-in for Hyper-V
	`jpcagt8hyperv.ini`	Configuration file for the Collector plug-in for Hyper-V
	`jpcagt5virtage.exe`	Execution program of the Collector plug-in for logical partitioning feature
	`jpcagt5virtage.ini`	Configuration file for the Collector plug-in for logical partitioning feature
	`jpcagt8vmware.exe`	Execution program of the Collector plug-in for VMware
	`jpcagt8vmware.ini`	Configuration file for the Collector plug-in for VMware
	`jpcagt8kvm.exe`	Execution program of the Collector plug-in for KVM
	`jpcagt8kvm.ini`	Configuration file for the Collector plug-in for KVM
	`jpcagt8docker.exe`	Execution program of the Collector plug-in for Docker environment
	`jpcagt8docker.ini`	Configuration file for the Collector plug-in for Docker environment
	`jpcagt8podman.exe`	Execution program of the Collector plug-in for Podman environment

Folder name	File name	Explanation
installation-folder\agt8\plugin\	jpcagt8podman.ini	Configuration file for the Collector plug-in for Podman environment
installation-folder\agt8\plugin\jpcagt8hyperv.d\	--	Data folder for the Collector plug-in for Hyper-V#3
	jpcagt8hyperv.dat	Empty file
installation-folder\agt8\plugin\jpcagt5virtage.d\	--	Data folder for the Collector plug-in for logical partitioning feature#3
	jpcagt5virtageSetup.ini	Definition file for the Collector plug-in for logical partitioning feature
installation-folder\agt8\plugin\jpcagt8vmware.d\	--	Data folder for the Collector plug-in for VMware#3
	*.xml	XML for SOAP request (parameter can be embedded)
installation-folder\agt8\plugin\jpcagt8kvm.d\	--	Data folder for the Collector plug-in for KVM#3
	command.dat	KVM request definition file
	procname.dat	KVM request definition file
installation-folder\agt8\plugin\docker.d\	--	Docker environment request definition file#3
	dockerdef.dat	Docker environment request definition file#3
	recorddef.dat	Docker environment record definition file#3
installation-folder\agt8\plugin\jpcagt8podman.d\	--	Podman environment request definition file#3
	command.dat	Podman environment request definition file
installation-folder\agt8\lib\	--	Message catalog storage folder
	jpcagt8msg.dll	Message catalog file

Folder name	File name	Explanation
installation-folder\agt8\store\	--	Store service base folder in which Remote Monitor is also located
	STDICT.DAT STRULES.DAT	Data model definition file
	jpcsto.ini.instmpl	Intermediate file for internal processing
	stolist.ini	Intermediate file for internal processing#2
installation-folder\patch_files\agt8\	--	Folder for storing files for patches (for Agent)
installation-folder\setup\	--	Setup file storage folder
	jpcagt8u.Z	PFM - RM setup archive file (UNIX)
	jpcagt8w.EXE	PFM - RM setup archive file (Windows)
installation-folder\agt8\agent*instance-name*\	--	Remote Monitor Collector service base folder (all files under this folder are created for each instance)#2
	jpcagt.ini	Remote Monitor Collector service startup initialization file#2
	jpcagt.ini.lck	Lock file for the Remote Monitor Collector service startup initialization file (this file is created for each instance)
	jpcagt.ini.model	Model file for Remote Monitor Collector service startup initialization file#2
	status.dat	Intermediate file for internal processing#4
	inssetup.bat	jpcconf inst setup command extension processing for PFM - RM for Virtual Machine#2

Folder name	File name	Explanation
installation-folder\agt8\agent\ *instance-name*\	plugin.ini	Virtual environment connection definition file[#2]
	GARULES.DAT	Consolidation rule file
	grouplist.ini	Group list file
	targetlist.ini	Monitoring-target list file
	tstatuses.dat	Virtual-agent status information[#5]
	recorddef.ini	user-defined record definition file [#8]
installation-folder\agt8\agent\ *instance-name*\targets\	--	Folder for group agent
	monitoring-target-name.ini	Monitoring-target settings file
	monitoring-target-name.ini.model	Model file for the monitoring-target settings file
installation-folder\agt8\agent\ *instance-name*\targets\ *monitoring-target-name*\data\	--	Remote Monitor Collector service data folder[#2]
	records.dat	Performance data storage file[#3, #6]
	error.dat	For error messages[#3, #6]
	records.tmp	Intermediate file for internal processing[#3, #6]
	error.tmp	Intermediate file for internal processing[#3, #6]
installation-folder\agt8\agent\ *instance-name*\groups\	--	Group agent folder[#2]
	group-name.ini	Group agent settings file[#3]
installation-folder\agt8\agent\ *instance-name*\log\	--	Folder for storing Remote Monitor Collector service internal log files[#2]
	*x*_*yN*.log (*x*: VM_Type value, *y*: *monitoring-target-name*)	Internal log file for Hyper-V collection processing[#3] Internal log file for VMware collection processing[#3]

Folder name	File name	Explanation
installation-folder\agt8\agent*instance-name*\log\	*x*_*yN*.log (*x*: VM_Type value, *y*: *monitoring-target-name*)	Internal log file for KVM collection processing#3 Internal log file for Docker environment collection processing#3 Internal log file for Podman environment collection processing#3 (*N*: 1 through 8)
	msglog01 msglog02 msglog03 msglog04	Internal log files#6
	nslog01 nslog02	Internal log files#6
	collect_01 collect_02 collect_03	Internal log files#6
	timer_01 timer_02 timer_03	Internal log files#6
installation-folder\agt8\agent*instance-name**monitoring-target-name*\work\	--	Work data storage folder for the Collector plug-in#2
	*	Intermediate file for internal processing#3, #6
installation-folder\agt8\agent*instance-name*\targets*monitoring-target-name*\log\	jpcagt5virtage1.log jpcagt5virtage2.log jpcagt5virtage3.log jpcagt5virtage4.log jpcagt5virtage5.log jpcagt5virtage6.log jpcagt5virtage7.log jpcagt5virtage8.log	Internal log files for collection processing of logical partitioning feature#3
installation-folder\agt8\store*instance-name*\	--	Remote Monitor Store service base folder (all files under this folder are created for each instance)#2
	*.DB	Performance data file#7
	*.IDX	Index file of performance data file#7

Folder name	File name	Explanation
installation-folder\agt8\store*instance-name*\	*.LCK	Lock file of performance data file[#7]
	jpcsto.ini	Remote Monitor Store service startup initialization file[#2]
	jpcsto.ini.model	Model for Remote Monitor Store service startup initialization file[#2]
	*.DAT	Data model definition file[#2]
	status.dat	Intermediate file for internal processing[#4]
installation-folder\agt8\store*instance-name*\STPD\	--	Performance data file for PD records[#7]
installation-folder\agt8\store*instance-name*\STPI\	--	Performance data file for PI records[#7]
installation-folder\agt8\store*instance-name*\STPL\	--	Performance data file for PL records[#7]
installation-folder\agt8\store*instance-name*\backup\	--	Default database backup folder[#2]
installation-folder\agt8\store*instance-name*\dump\	--	Default database export folder[#2]
installation-folder\agt8\store*instance-name*\import\	--	Default database import folder[#2]
installation-folder\agt8\store*instance-name*\log\	--	Folder for storing Remote Monitor Store service internal log files[#2]
	msglog01 msglog02	Internal log files[#7]
	nslog01 nslog02	Internal log files[#7]
installation-folder\agt8\store*instance-name*\partial\	--	Default database partial backup folder[#2]
<Program Files>\Hitachi\HNTRLib2\	--	Folder for storing HNTRLib2 execution files and include files
	*	HNTRLib2 execution files and include files
<Program Files>\Common Files\Hitachi\	--	Folder for storing HNTRLib2's public DLL

Folder name	File name	Explanation
`<Program Files>\Common Files\Hitachi\`	*	HNTRLib2's public DLL

Legend:

--: Not applicable

#1

Created when a patch is applied.

#2

Created when the `jpcconf inst setup` command is executed.

#3

This file is used internally by PFM - RM for Virtual Machine. Do not modify or delete it.

#4

May be temporarily created in some cases.

#5

Generated when the health check function is enabled.

#6

Created when the Remote Monitor Collector service is started or when a record is collected.

#7

Created when the Remote Monitor Store service is started.

#8

If you use user defined records (applicable only when the monitoring target is a VMware environment), create the file and add it to the folder.

G. Migration from PFM - Agent for Virtual Machine to PFM - Remote Monitor for Virtual Machine

Because PFM - Agent for Virtual Machine and PFM - RM for Virtual Machine cannot be installed on the same host due to a difference in supported OSs, PFM - Remote Monitor for Virtual Machine version 12-00 or later does not support the migration command.

If you want to migrate from PFM - Agent for Virtual Machine to PFM - Remote Monitor for Virtual Machine version 12-10, use the migration command of PFM - Remote Monitor for Virtual Machine version 10-00, 11-00, or 11-50 to perform migration first and then upgrade to version 12-10.

H. Version Compatibility

In PFM - RM, there are multiple product versions as well as data model versions.

Because data models maintain upward compatibility, report definitions and alarm definitions created using an earlier version can also be used by a later version.

The table below shows PFM - RM for Virtual Machine version compatibility.

Table H–1: PFM - RM for Virtual Machine version compatibility

PFM - RM for Virtual Machine version	Data model version	Alarm table version of monitoring template
10-00	4.0	10.00
11-00	5.0	11.00
11-50	6.0	11.50
12-00	7.0	12.00
12-10	8.0	12.10
12-50	9.0	12.50

For details about version compatibility, see the version compatibility described in the appendix section of the *JP1/Performance Management Planning and Configuration Guide*.

I. Outputting Action Log Data

Performance Management action logs contain historical information that is output during linkage to alarm functions that are activated when a threshold, such as the system load, is exceeded.

For example, an action log is output when a PFM service starts up or stops, or when the connection status to PFM - Manager changes.

An action log can be output when the version of PFM - Manager or PFM - Base is 08-11 or later.

An action log is a text file that is output in CSV format. By periodically saving the action log and processing it using a spreadsheet program, you can utilize it as analysis data.

An action log is output based on the setting in the `jpccomm.ini` file. This appendix explains the contents of the action logs that are output by PFM - RM for Virtual Machine and PFM - Base, and the settings necessary for outputting action logs.

I.1 Types of events that are output to action logs

The table below shows the types of events that are output to action logs and the triggers used by PFM - RM for Virtual Machine and PFM - Base for outputting action logs. Types of events refer to the identifiers in action logs that categorize the items that are output to action logs.

Table I–1: Types of events that are output to action logs

Type of event	Explanation	Triggers used by PFM - RM for Virtual Machine and PFM - Base for outputting action logs
`StartStop`	Event indicating software start or stop	• PFM service start/stop • Start/stop of the Stand-alone mode
`ExternalService`	Event indicating the communication result between a JP1 product and an external service Event indicating an occurrence of abnormal communication	Change in the connection status with PFM - Manager
`ManagementAction`	Event indicating the execution of an important action of a program Event indicating that an action has been executed using another audit category as the trigger	Execution of an automatic action

I.2 Format for saving action logs

This subsection explains the file format for saving action logs.

There are two types of action log files. The file to which action log data is output is called the *current output file*. If this file becomes full, it is saved to a different file called the *shift file*. The flow for switching action log files follows.

1. Action logs are sequentially output to the current output file `jpcaudit.log`.
2. When the current output file becomes full, the action logs in it are saved in a shift file.

The shift file name consists of the current output file name with a numerical suffix added to it. Whenever the current output file becomes full, the numerical suffix of the shift file name is incremented by 1. In other words, the larger the numerical suffix, the older the log file.

Example:

When the current output file `jpcaudit.log` becomes full, its content is saved in the shift file `jpcaudit1.log`.

When the current output file becomes full again, the logs in it are moved to `jpcaudit1.log` and the existing shift file `jpcaudit1.log` is renamed `jpcaudit2.log`.

When the number of log files exceeds the save limit (specified in the `jpccomm.ini` file), the oldest log file is deleted.

3. The current output file is initialized and new action logs are written into it.

To specify whether to output action logs, their output destination, and the number of logs that can be saved, use the `jpccomm.ini` file. For details about how to set up the `jpccomm.ini` file, see *I.4 Settings for outputting action logs*.

I.3 Action log output format

Information related to monitored events is output to Performance Management action logs. An action log file is output for each host (physical host or logical host). The output destination hosts for action logs are as follows.

- When a service is executed: Host on which the execution source service is running
- When a command is executed: Host that executed the command

The output format of action logs, their output destination, and output items are explained below.

(1) Output format

```
CALFHM x.x,output-item-1 = value-1,output-item-2 = value-2,...,output-item-n
 = value-n
```

(2) Output destination

```
installation-folder\auditlog\
```

You can change the output destination for action logs in the `jpccomm.ini` file. For details about how to set up the `jpccomm.ini` file, see *I.4 Settings for outputting action logs*.

(3) Output items

Output items can be divided into the following two categories:

- Common output items
  These are common items that are output by all JP1 products that output action logs.
- Fixed output items
  These are optional items that are output by JP1 products that output action logs.

(a) Common output items

The table below shows the values that are output to common output items and descriptions of these items. This table also includes items that are output by PFM - Manager and their descriptions.

Table I–2: Common output items in action logs

No.	Output item		Value	Description
	Item name	Attribute name that is output		
1	Common specification identifier	--	`CALFHM`	Identifier that indicates the action log format
2	Common specification revision number	--	*x*.*x*	Revision number for managing action logs
3	Serial number	`seqnum`	*serial-number*	Action log record serial number
4	Message ID	`msgid`	`KAVE`*xxxxx*-*x*	Product message ID
5	Date/time	`date`	*YYYY*-*MM*-*DD*`T`*hh*:*mm*:*ss*.*sssTZD*#	Output date/time and time zone of action log
6	Program name	`progid`	`JP1PFM`	Name of the program in which the event occurred
7	Component name	`compid`	*service-ID*	Name of the component in which the event occurred
8	Process ID	`pid`	*process-ID*	Process ID for which the event occurred
9	Location	`ocp:host`	• *host-name* • *IP-address*	Location where the event occurred
10	Event type	`ctgry`	• `StartStop` • `Authentication` • `ConfigurationAccess` • `ExternalService` • `AnomalyEvent` • `ManagementAction`	Category name for categorizing the events that are output to action logs
11	Event result	`result`	• `Success` (success) • `Failure` (failure) • `Occurrence` (occurrence)	Event result
12	Subject identification information	`subj:pid`	*process-ID*	One of the following types of information: • Process ID that runs based on user operation • Process ID that caused the event • User name that caused the event • Identification information
		`subj:uid`	*account-identifier* (PFM user/JP1 user)	
		`subj:euid`	*effective-user-ID* (OS user)	

No.	Output item		Value	Description
	Item name	Attribute name that is output		
12	Subject identification information	`subj:euid`	*effective-user-ID* (OS user)	assigned to users on a 1:1 basis

Legend:

--: None

#

`T` is a delimiter between date and time.

TZD is a time zone specifier and one of the following is output:

`+`*hh*`:`*mm*: Indicates that the time is *hh*`:`*mm* ahead of UTC.

`-`*hh*`:`*mm*: Indicates that the time is *hh*`:`*mm* behind UTC.

`Z`: Indicates that the time is the same as UTC.

(b) Fixed output items

The table below shows the values that are output to fixed output items, and descriptions of these items. This table also includes items that are output by PFM - Manager and their descriptions.

Table I–3: Fixed output items in action logs

No.	Output item		Value	Description
	Item name	Attribute name that is output		
1	Object information	`obj`	• *PFM-RM-service-ID* • *added-deleted-or-updated-user-name* (PFM user)	Target of operation
		`obj:table`	*alarm-table-name*	
		`obj:alarm`	*alarm-name*	
2	Action information	`op`	• `Start` (start) • `Stop` (stop) • `Add` (add) • `Update` (update) • `Delete` (delete) • `Change Password` (password change) • `Activate` (activate) • `Inactivate` (inactivate) • `Bind` (bind) • `Unbind` (unbind)	Information on the action that caused the event

3	Permissions information	`auth`	• Management user
			`Management`
			• Ordinary user
			`Ordinary`
			• Windows
			`Administrator`
			• UNIX
			`SuperUser`
				Permissions information about the user who performed the operation

		`auth:mode`	• PFM authentication mode `pfm`	Authentication mode of the user who

No.	Output item		Value	Description
	Item name	Attribute name that is output		

3	Permissions information	`auth:mode`	• JP1 authentication mode
			`jp1`
			• OS user
			`os`
				performed the operation

4	Output source location	`outp:host`	*PFM-Manager-host-name*	Action log output source host
5	Location that issued the instruction	`subjp:host`	• *login-source-host-name* • *target-host* (only when the `jpctool alarm` (`jpcalarm`) command is executed)	Host that issued the operation instruction
6	Free description	`msg`	*message*	Message that is output when an alarm occurs or an automatic action is executed

Depending on the output trigger, some fixed output items are output while others are not, and the content of the output items also varies. Message IDs and the content of fixed output items are explained below for each output trigger.

■ PFM service start/stop (StartStop)

- Output host: Host on which the service is running
- Output component: Service that executes start/stop

Item name	Attribute name	Value
Message ID	`msgid`	Start: `KAVE03000-I` Stop: `KAVE03001-I`
Action information	`op`	Start: `Start` Stop: `Stop`

■ Stand-alone mode start/stop (StartStop)

- Output host: PFM - RM host
- Output component: Remote Monitor Collector service or Remote Monitor Store service

Item name	Attribute name	Value
Message ID	`msgid`	When the stand-alone mode starts: `KAVE03002-I` When the stand-alone mode stops: `KAVE03003-I`

Note 1: Fixed output items are not output.

Note 2: When a service of PFM - RM for Virtual Machine starts, it connects to the PFM - Manager host to register node information and collect the latest alarm definition information. If the service cannot connect to the PFM - Manager host, the service starts in the mode that enables only some functions such as operating information collection (stand-alone mode). In such a case, to indicate the stand-alone mode, message `KAVE03002-I` is issued. Thereafter, the service tries to connect to PFM - Manager at specified intervals, and if it succeeds in registering the node information and collecting the definition information, it recovers from the stand-alone mode, and message `KAVE03003-I` is issued. Therefore, based on this action log, you can know that PFM - RM for Virtual Machine has started in an incomplete state as long as messages `KAVE03002-I` and `KAVE03003-I` are being issued.

■ Change in the status of connection with PFM - Manager (ExternalService)

- Output host: PFM - RM host
- Output component: Remote Monitor Collector service or Remote Monitor Store service

Item name	Attribute name	Value
Message ID	msgid	Sending of an event to PFM - Manager failed (queuing has started): KAVE03300-I Resending of an event to PFM - Manager was completed: KAVE03301-I

Note 1: Fixed output items are not output.

Note 2: If the Remote Monitor Store service fails to send an event to PFM - Manager, it starts event queuing, and from then on, a maximum of 3 cases are accumulated in the queue for each event. Message KAVE03300-I is issued when sending of the event fails and queuing begins. After the connection with PFM - Manager is restored, when sending of the queued event is completed, message KAVE03301-I is issued. Based on this action log, you can know that sending of the event to PFM - Manager did not succeed on a real-time basis as long as messages KAVE03300-I and KAVE03301-I are being issued.

Note 3: The Remote Monitor Collector service normally sends an event to PFM - Manager via the Remote Monitor Store service. Only when the Remote Monitor Store service is stopped for some reason, the event is sent directly to PFM - Manager. But if sending fails, message KAVE03300-I is issued. In this case, queuing does not start, and therefore message KAVE03301-I is not issued. Based on this action log, you know that there is an event that was not able to be sent to PFM - Manager.

■ Execution of automatic action (ManagementAction)

- Output host: Host that executed the action
- Output component: Action Handler service

Item name	Attribute name	Value
Message ID	msgid	Command execution process generation succeeded: KAVE03500-I Command execution process generation failed: KAVE03501-W Email transmission succeeded: KAVE03502-I Email transmission failed: KAVE03503-W
Free description	msg	Command execution: cmd = *executed-command-line* Email transmission: mailto = *destination-mail-address*

Note: When a command execution process is successfully generated, message KAVE03500-I is issued. Thereafter, neither a log indicating whether the command was executed nor an execution result log is output to the action log.

(4) Output example

An action log output example follows.

```
CALFHM 1.0, seqnum = 1, msgid = KAVE03000-I, date = 2007-01-18T22:46:49.682
+ 09:00,
progid = JP1PFM, compid = 8A1host01, pid = 2076,
ocp:host = host01, ctgry = StartStop, result = Occurrence,
subj:pid = 2076,op = Start
```

I.4 Settings for outputting action logs

Settings for outputting action logs are defined in the jpccomm.ini file. If no settings are defined, no action log is output. The settings necessary for outputting action logs and the procedure are described below.

(1) Setting procedure

To specify the settings for outputting an action log:

1. Stop all PFM services on the host.

2. Use a text editor, for example, to edit the `jpccomm.ini` file.

3. Save and close the `jpccomm.ini` file.

(2) Details about the jpccomm.ini file

Details about the `jpccomm.ini` file follow.

(a) Storage folder

installation-folder

(b) Format

For the `jpccomm.ini` file, define the following items:

- Whether to output action logs
- Action log output destination
- Number of action logs to be saved
- Action log file size

The specification format follows.

`"`*item-name*`"  =  `*value*

The table below shows the items to be set.

Table I–4: Items to be set in the jpccomm.ini file and their initial values

No.	Item	Explanation
1	`[Action Log Section]`	Section name. This cannot be changed.
2	`Action Log Mode`	Specifies whether to output action logs. This item must be specified. • Initial value `0` (Do not output action logs.) • Values that can be specified `0` (Do not output action logs.), `1` (Output action logs.) If any other value is specified, an error message is issued, and no action log is output.
3	`Action Log Dir`	Specify the output destination for action logs as an absolute path. In a logical host environment, specify a directory on the shared disk. If the directory you specify is not on the shared disk, Performance Management will output action logs to each physical host that forms the basis for the logical host. If you specify a path that exceeds the maximum allowable length or the system is unable to access the directory, an error message is output to the common message log and Performance Management will not output action logs. • Initial value: Omitted • Value used when this item is omitted (default): On physical hosts: *installation-folder*`\auditlog` On logical hosts: *environment-directory*`\jp1pc\auditlog` • Available values: Character strings from 1 to 185 bytes in length

No.	Item	Explanation
4	Action Log Num	Specifies the upper limit for the total number of log files (number of files to be saved). Specify the total for the current output file and shift files. • Initial value Omitted • Value that is used when specification is omitted (default value) 5 • Value that can be specified An integer between 2 and 10 If a non-numeric character string is specified, an error message is output and the default value of 5 is set. If a value outside the range is specified, an error message is output and an integer between 2 and 10 that is closest to the specified value is set.
5	Action Log Size	Specifies the log file size in kilobytes. • Initial value Omitted • Value that is used when specification is omitted (default value) 2048 • Value that can be specified An integer between 512 and 2,096,128 If a non-numeric character string is specified, an error message is output and the default value of 2,048 is set. If a value outside the range is specified, an error message is output and an integer between 512 and 2,096,128 that is closest to the specified value is set.

J. Linkage with JP1/SLM

The ability to monitor the operating status of PFM - RM for Virtual Machine can be enhanced by linkage with JP1/SLM.

PFM - RM for Virtual Machine provides some default monitoring items for PFM - Manager. These items are required to enable monitoring with JP1/SLM.

The table below lists and describes the default monitoring items that PFM - RM for Virtual Machine provides for PFM - Manager.

For multi-instance records, the records whose key values match specified values are collected. For the collection-target keys, see the collection results of the relevant record.

Table J–1: Default monitoring items that PFM - RM for Virtual Machine provides for PFM - Manager

Name displayed in JP1/SLM	Description	Record (record ID)	Key (PFM - Manager name)	Field name
CPU resource used on the physical server (MHz)	CPU resource used on the physical server (units: MHz)	`Host Status(PI)` #3	None	`Used`
CPU resource available on the physical CPU for executing virtual machines (MHz)	CPU resource available on the physical CPU for executing virtual machines (units: MHz)	`VM Status(PI_VI)` #3	`VM_ID`	`Used`
Total memory usage (MB)	Total memory usage on the physical server (units: MB)	`Host Memory Status(PI_HMI)` #3	None	`Total Used`
Virtual-machine memory usage (MB)	Virtual-machine memory usage (units: MB)	`VM Memory Status(PI_VMI)` #1	`VM_ID`	`Used`
Disk usage (MB)	Logical disk usage on the physical server (units: MB)	`Host Logical Disk Status(PI_HLDI)` #1#3	`DISK_ID`	`Used`
Virtual-machine disk space usage (MB)	Virtual-machine disk space usage (units: MB)	`VM Logical Disk Status(PI_VLDI)` #1#2#3#4	`VM_ID` `DISK_ID`	`Used`

#1

If the monitoring target is either logical partitioning feature, a hyphen (–) is displayed as the value.

#2

If the monitoring target is either Hyper-V or KVM, a hyphen (–) is displayed as the value.

#3

If the monitoring target is either Docker environment, a hyphen (–) is displayed as the value.

#4

If the monitoring target is either Podman environment, a hyphen (–) is displayed as the value.

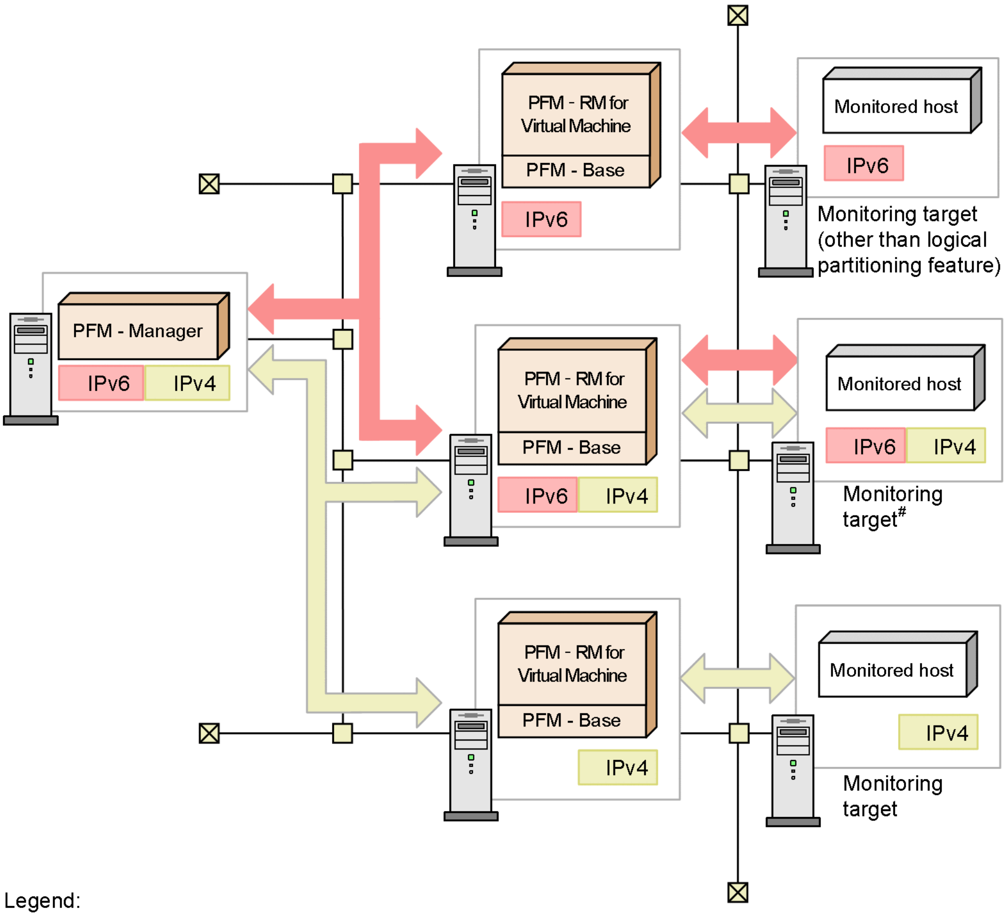
To provide the default monitoring items for PFM - Manager, you must copy a setup file and execute a setup command. For details, see *2.1.4(1) Registering PFM - RM for Virtual Machine*.

K. About Communication in IPv4 Environments and IPv6 Environments

Performance Management supports network configurations in IPv4 and IPv6 environments. Performance Management is able to operate with a network configuration in which IPv4 and IPv6 environments coexist.

Note that for Performance Management to operate with that network configuration, the OS of the host on which PFM - Manager is installed must be Windows or Linux.

Figure K–1: Scope of communication when an IPv4 environment and an IPv6 environment are used



Legend:

: Indicates a program provided by Performance Management

: IPv4 environment

: IPv6 environment

: IPv4 communication

: IPv6 communication

#: Logical pertitioning feature communicates only with IPv4.

To enable communication in an IPv6 environment, you must execute the `jpcconf ipv6 enable` command. For details about the `jpcconf ipv6 enable` command, see the chapter that describes commands in the manual *JP1/Performance Management Reference*. For the conditions and timing for executing the `jpcconf ipv6 enable` command, see the chapter that describes network configuring examples in an IPv6 environment in the *JP1/Performance Management Planning and Configuration Guide*.

L. Data Sources of the Record

In each fields of the record, the values are stored which have been acquired from the Performance Management or the monitored programs, and calculated these values in some formula. This appendix provides the list of acquisition targets or calculation methods in each fields of the record.

We call these acquisition targets and calculation methods as the data source.

L.1 In VMware

This section describes the field values of the data sources in VMware.

VMware vSphere Client indicates a GUI that is based on Windows applications. *VMware vSphere Web Client* indicates a GUI that is based on HTML5 of vCenter Server. *VMware vSphere Web Client (ESXi Host)* indicates a GUI that is based on HTML5 that manages a single ESXi host.

(1) Host CPU Status(PI_HCI)

The following table lists the data sources in each fields of the Host CPU Status(PI_HCI) record.

Table L–1: The data sources in each fields of the Host CPU Status(PI_HCI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
CPU ID (CPU_ID)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: CPU Objects: Description VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: CPU Target Objects
CPU Name (CPU_NAME)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Processors General: Model VMware vSphere Web Client Select machine: Host TAB: Summary Processor Type

PFM - View name (PFM - Manager name)	Data Sources
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Clocks (CLOCKS)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Processors General: Processor Speed VMware vSphere Web Client Select machine: Host TAB: Summary Hardware: CPU CPU: Cores
Used (USED)	CLCOKS * (USED_PERCENT / 100)
Unused (UNUSED)	CLOCKS - USED
Used % (USED_PERCENT)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: CPU Counters: Description: Usage Internal Name: usage VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Usage Internal Name: usage
Unused % (UNUSED_PERCENT)	100 - USED_PERCENT

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(2) Host Logical Disk Status(PI_HLDI)

The following table lists the data sources in each fields of the Host Logical Disk Status(PI_HLDI) record.

Table L–2: The data sources in each fields of the Host Logical Disk Status(PI_HLDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	VMware vSphere Client Select machine: Host

PFM - View name (PFM - Manager name)	Data Sources
Disk ID (DISK_ID)	TAB: Configuration Hardware: Storage Datastores: Identification VMware vSphere Web Client Select machine: Host TAB: Datastores Name
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Storage Datastores: Capacity VMware vSphere Web Client Select machine: Host TAB: Datastores Capacity
Used (USED)	SIZE - FREE
Free (FREE)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Storage Datastores: Free VMware vSphere Web Client Select machine: Host TAB: Datastores Free
Used % (USED_PERCENT)	(USED / SIZE) * 100
Last Update (LAST_UPDATE)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Storage Datastores: Last Update VMware vSphere Web Client None
Free % (FREE_PERCENT)	(FREE / SIZE) * 100

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(3) Host Memory Status(PI_HMI)

The following table lists the data sources in each fields of the Host Memory Status(PI_HMI) record.

Table L–3: The data sources in each fields of the Host Memory Status(PI_HMI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Size (SIZE)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Memory Physical: Total VMware vSphere Web Client Select machine: Host TAB: Summary Hardware: Memory
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Memory Counters: Description: Consumed Internal Name: consumed VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Consumed Internal Name: consumed
VMM Used (VMM_USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Memory Counters: Description: Used by VMkernel Internal Name: sysUsage VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: VMKernel consumed Internal Name: sysUsage
VM Used (VM_USED)	USED - VMM_USED
Unused (UNUSED)	SIZE - USED

PFM - View name (PFM - Manager name)	Data Sources
VM Swap Used (VM_SWAP_USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Memory Counters: Description: Balloon Internal Name: vmmemctl VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Ballooned memory Internal Name: vmmemctl
Host Swap Used (HOST_SWAP_USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Memory Counters: Description: Swap Used Internal Name: swapused VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Swap consumed Internal Name: swapused
Total Used (TOTAL_USED)	USED + VM_SWAP_USED + HOST_SWAP_USED
Used % (USED_PERCENT)	(USED / SIZE) * 100
VMM Used % (VMM_USED_PERCENT)	(VMM_USED / SIZE) * 100
VM Used % (VM_USED_PERCENT)	(VM_USED / SIZE) * 100
VM Swap Used % (VM_SWAP_USED_PERCENT)	(VM_SWAP_USED / SIZE) * 100
Host Swap Used % (HOST_SWAP_USED_PERCENT)	(HOST_SWAP_USED / SIZE) * 100
Total Used % (TOTAL_USED_PERCENT)	(TOTAL_USED / SIZE) * 100
Swap IO (SWAP_IO)	SWAP_IN_IO + SWAP_OUT_IO
Swap In IO (SWAP_IN_IO)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Memory Counters: Description: Swap in Internal Name: swapin

PFM - View name (PFM - Manager name)	Data Sources
Swap In IO (SWAP_IN_IO)	VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Swap in Internal Name: swapin
Swap Out IO (SWAP_OUT_IO)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Memory Counters: Description: Swap Out Internal Name: swapout VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Swap out Internal Name: swapout

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(4) Host Network Status(PI_HNI)

The following table lists the data sources in each fields of the Host Network Status(PI_HNI) record.

Table L–4: The data sources in each fields of the Host Network Status(PI_HNI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Net ID (NET_ID)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Network Objects: Description VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Network Target Objects
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE

PFM - View name (PFM - Manager name)	Data Sources
Send Rate (SEND_RATE)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Network Counters: Description: Data transmit rate Internal Name: transmitted VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Network Counters: Data transmit rate Internal Name: transmitted
Recv Rate (RECV_RATE)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Network Counters: Description: Data receive rate Internal Name: received VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Network Counters: Data receive rate Internal Name: received

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(5) Host Physical Disk Status(PI_HPDI)

The following table lists the data sources in each fields of the Host Physical Disk Status(PI_HPDI) record.

Table L–5: The data sources in each fields of the Host Physical Disk Status(PI_HPDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Objects: Description

PFM - View name (PFM - Manager name)	Data Sources
Disk ID (DISK_ID)	VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Target Objects
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Read rate Internal Name: read VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Read rate Internal Name: read
Write Speed (WRITE_SPEED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Write rate Internal Name: write VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Write rate Internal Name: write
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Read requests Internal Name: numberRead VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Read request Internal Name: numberRead

PFM - View name (PFM - Manager name)	Data Sources
Write Requests (WRITE_REQUESTS)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Write requests Internal Name: numberWrite VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Write request Internal Name: numberWrite
Commands (COMMANDS)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Commands issued Internal Name: commands VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Commands issued Internal Name: commands
Abort Commands (ABORT_COMMANDS)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Command aborted Internal Name: commandsAborted VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Commands aborted Internal Name: commandsAborted
Abort Commands % (ABORT_COMMANDS_PERCENT)	(ABORT_COMMANDS / COMMANDS) * 100
Bus Resets (BUS_RESETS)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Bus resets Internal Name: busResets

PFM - View name (PFM - Manager name)	Data Sources
Bus Resets (BUS_RESETS)	VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Bus resets Internal Name: busResets
Device Latency (DEVICE_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Physical device command latency Internal Name: deviceLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Physical device command latency Internal Name: deviceLatency
Device Read Latency (DEVICE_READ_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Physical device read latency Internal Name: deviceReadLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Physical device read latency Internal Name: deviceReadLatency
Device Write Latency (DEVICE_WRITE_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Physical device write latency Internal Name: deviceWriteLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Physical device write latency Internal Name: deviceWriteLatency
Kernel Latency (KERNEL_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance

PFM - View name (PFM - Manager name)	Data Sources
Kernel Latency (KERNEL_LATENCY)	Chart Options: Disk Counters: Description: Kernel command latency Internal Name: kernelLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Kernel command latency Internal Name: kernelLatency
Kernel Read Latency (KERNEL_READ_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Kernel read latency Internal Name: kernelReadLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Kernel read latency Internal Name: kernelReadLatency
Kernel Write Latency (KERNEL_WRITE_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Kernel write latency Internal Name: kernelWriteLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Kernel write latency Internal Name: kernelWriteLatency
Queue Latency (QUEUE_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Queue command latency Internal Name: queueLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Queue command latency Internal Name: queueLatency

PFM - View name (PFM - Manager name)	Data Sources
Queue Read Latency (QUEUE_READ_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Queue read latency Internal Name: queueReadLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Queue read latency Internal Name: queueReadLatency
Queue Write Latency (QUEUE_WRITE_LATENCY)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Disk Counters: Description: Queue write latency Internal Name: queueWriteLatency VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Queue write latency Internal Name: queueWriteLatency

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(6) Host Status Detail(PD)

The following table lists the data sources in each fields of the Host Status Detail(PD) record.

Table L–6: The data sources in each fields of the Host Status Detail(PD) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Status (STATUS)	--
Host Name (HOST_NAME)	Setting for monitored VM_Host parameter
Reason (REASON)	--

PFM - View name (PFM - Manager name)	Data Sources
Product (PRODUCT)	VMware vSphere Client Select machine: Host Main Window: Upper side of tab view VMware vSphere Web Client Select machine: Host TAB: Summary Hypervisor
VM Count (VM_COUNT)	VMware vSphere Client Select machine: Host TAB: Virtual Machines Number of Virtual Machines VMware vSphere Web Client Select machine: Host TAB: Summary Virtual Machines
VM Active (VM_ACTIVE)	VMware vSphere Client Select machine: Host TAB: Virtual Machines Number of Virtual Machines which State is Powered On. VMware vSphere Web Client Select machine: Host TAB: VMs Number of Virtual Machines which State is Powered On.

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(7) Host Status(PI)

The following table lists the data sources in each fields of the Host Status(PI) record.

Table L–7: The data sources in each fields of the Host Status(PI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Clocks (CLOCKS)	[In case of `UseHTPhysicalClocks=N` in jpcagt8cfg.ini] VMware vSphere Client Select machine: Host TAB: Summary General: Logical Processors **x** Frequency of CPU Cores VMware vSphere Web Client Select machine: Host TAB: Summary

PFM - View name (PFM - Manager name)	Data Sources
Clocks (CLOCKS)	Logical Processors **x** Frequency of CPU Cores [In case of `UseHTPhysicalClocks=Y` in jpcagt8cfg.ini] VMware vSphere Client Select machine: Host TAB: Summary Resources: Capacity VMware vSphere Web Client Select machine: Host TAB: Summary CPU: Capacity
Count (COUNT)	VMware vSphere Client Select machine: Host TAB: Summary General: CPU Cores VMware vSphere Web Client Select machine: Host TAB: Summary Hardware: CPU CPU Cores
Sampling Time (SAMPLING_TIME)	VMware vSphere Client Select machine: Host TAB: Performance Graph: Time VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Graph: Time
Used (USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: CPU Counters: Description: Usage in MHz Internal Name: usagemhz VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Usage in MHz Internal Name: usagemhz
VMM Used (VMM_USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: System Object: host/system and host/vim Counters: Description: Resource CPU usage (Average) Internal Name: resourceCpuUsage

PFM - View name (PFM - Manager name)	Data Sources
VMM Used (VMM_USED)	VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: System Target Objects: host/system and host/vim Counters: Resource CPU usage (Average) Internal Name: resourceCpuUsage
VM Used (VM_USED)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: System Object: host/user Counters: Description: Resource CPU usage (Average) Internal Name: resourceCpuUsage VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: System Target Objects: host/user Counters: Resource CPU usage (Average) Internal Name: resourceCpuUsage
VMM Console Used (VMM_CONSOLE_USED)	Not supported (always 0)
VMM Kernel Used (VMM_KERNEL_USED)	ESXi 6.7 is not supported (always 0) VMware vSphere Client Select machine: Host TAB: Performance Chart Options: System Object: host/system/kernel Counters: Description: Resource CPU usage (Average) Internal Name: resourceCpuUsage VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: System Target Objects: host/system/kernel Counters: Resource CPU usage (Average) Internal Name: resourceCpuUsage
VMM Others Used (VMM_OTHERS_USED)	VMM_USED - VMM_CONSOLE_USED - VMM_KERNEL_USED
Unused (UNUSED)	CLOCKS - USED
Used % (USED_PERCENT)	USED / CLOCKS * 100
VMM Used % (VMM_USED_PERCENT)	VMM_USED / CLOCKS * 100
VM Used % (VM_USED_PERCENT)	VM_USED / CLOCKS * 100

PFM - View name (PFM - Manager name)	Data Sources
VMM Console Used % (VMM_CONSOLE_USED_PERCEN T)	VMM_CONSOLE_USED / CLOCKS * 100
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)	VMM_KERNEL_USED / CLOCKS * 100
VMM Others Used % (VMM_OTHERS_USED_PERCENT)	VMM_OTHERS_USED / CLOCKS * 100
Unused % (UNUSED_PERCENT)	UNUSED / CLOCKS * 100
Insufficient (INSUFFICIENT)	USED / USED_PERCENT * INSUFFICIENT_PERCENT
Insufficient % (INSUFFICIENT_PERCENT)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: CPU Counters: Description: Ready Internal Name: ready VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Ready Internal Name: ready
Co-Stop (CO_STOP)	USED / USED_PERCENT * CO_STOP_PERCENT
Co-Stop % (CO_STOP_PERCENT)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: CPU Counters: Description: Co-stop Internal Name: costop VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Co-stop Internal Name: costop

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(8) VM CPU Status(PI_VCI)

The following table lists the data sources in each fields of the VM CPU Status(PI_VCI) record.

Table L–8: The data sources in each fields of the VM CPU Status(PI_VCI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
CPU ID (CPU_ID)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: CPU Objects: Description VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: CPU Target Objects
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: CPU Counters: Description: Usage in MHz Internal Name: usagemhz VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Usage in MHz Internal Name: usagemhz
Insufficient (INSUFFICIENT)	USED / USED_PERCENT * INSUFFICIENT_PERCENT
Request (REQUEST)	USED + INSUFFICIENT
Used % (USED_PERCENT)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: CPU Counters: Description: Used Internal Name: used VMware vSphere Web Client Select machine: Virtual Machine

PFM - View name (PFM - Manager name)	Data Sources
Used % (USED_PERCENT)	TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Used Internal Name: used
Insufficient % (INSUFFICIENT_PERCENT)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: CPU Counters: Description: Ready Internal Name: ready VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Ready Internal Name: ready
Request % (REQUEST_PERCENT)	USED_PERCENT + INSUFFICIENT_PERCENT
Used Per Request (USED_PER_REQUEST)	USED / REQUEST * 100
Insufficient Per Request (INSUFFICIENT_PER _REQUEST)	INSUFFICIENT / REQUEST * 100
Co-Stop (CO_STOP)	USED / USED_PERCENT * CO_STOP_PERCENT
Co-Stop % (CO_STOP_PERCENT)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: CPU Counters: Description: Co-stop Internal Name: costop VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Co-stop Internal Name: costop

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(9) VM Logical Disk Status(PI_VLDI)

The following table lists the data sources in each fields of the VM Logical Disk Status(PI_VLDI) record.

Table L–9: The data sources in each fields of the VM Logical Disk Status(PI_VLDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Disk ID (DISK_ID)	VMware vSphere Client Select machine: Virtual Machine TAB: Console A drive name or a mount position displayed on the Guest OS VMware vSphere Web Client (ESXi Host) Select machine: Virtual Machine General Information / Storage / Guest Disks Disk path
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	VMware vSphere Client Select machine: Virtual Machine TAB: Console Capacity of the drive displayed on the Guest OS VMware vSphere Web Client (ESXi Host) Select machine: Virtual Machine General Information / Storage / Guest Disks Capacity
Used (USED)	SIZE - FREE
Free (FREE)	VMware vSphere Client Select machine: Virtual Machine TAB: Console Free space of the drive displayed on the Guest OS VMware vSphere Web Client (ESXi Host) Select machine: Virtual Machine General Information / Storage / Guest Disks Free
Used % (USED_PERCENT)	(USED / SIZE) * 100
Free % (FREE_PERCENT)	(FREE / SIZE) * 100

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(10) VM Memory Status(PI_VMI)

The following table lists the data sources in each fields of the VM Memory Status(PI_VMI) record.

Table L–10: The data sources in each fields of the VM Memory Status(PI_VMI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary General: Memory VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware Memory
Used (USED)	RESOURCE_USED + VM_SWAP_USED + HOST_SWAP_USED
Resource Used (RESOURCE_USED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Memory Counters: Description: Consumed Internal Name: consumed VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Consumed Internal Name: consumed
VM Swap Used (VM_SWAP_USED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Memory Counters: Description: Balloon Internal Name: vmmemctl VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Ballooned memory

PFM - View name (PFM - Manager name)	Data Sources
VM Swap Used (VM_SWAP_USED)	Internal Name: vmmemctl
Host Swap Used (HOST_SWAP_USED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Memory Counters: Description: Swapped Internal Name: swapped VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Swapped Internal Name: swapped
Unused (UNUSED)	SIZE - USED
Used % (USED_PERCENT)	(USED / SIZE) * 100
Resource Used % (RESOURCE_USED_PERCENT)	(RESOURCE_USED / SIZE) * 100
VM Swap Used % (VM_SWAP_USED_PERCENT)	(VM_SWAP_USED / SIZE) * 100
Host Swap Used % (HOST_SWAP_USED_PERCENT)	(HOST_SWAP_USED / SIZE) * 100
VM Swap IO (VM_SWAP_IO)	VM_SWAP_IN + VM_SWAP_OUT
VM Swap In (VM_SWAP_IN)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Memory Counters: Description: Swap in Internal Name: swapin VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Swap in Internal Name: swapin
VM Swap Out (VM_SWAP_OUT)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Memory Counters: Description: Swap out Internal Name: swapout VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced

PFM - View name (PFM - Manager name)	Data Sources
VM Swap Out (VM_SWAP_OUT)	Chart Options: Memory Counters: Swap out Internal Name: swapout
Working Size (WORKING_SIZE)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Memory Counters: Description: Active Internal Name: active VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Memory Counters: Active Internal Name: active
Working Size % (WORKING_SIZE_PERCENT)	(WORKING_SIZE / SIZE) * 100
Share (SHARE)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation Memory: Resource Settings: Shares VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware Memory Shares
Max (MAX)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation Memory: Resource Settings: Limit VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware Memory Limit
Min (MIN)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation Memory: Resource Settings: Reservation VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary

PFM - View name (PFM - Manager name)	Data Sources
Min (MIN)	VM Hardware Memory Reservation
Expectation (EXPECTATION)	(share/ ∑ share) * PI_HMI.SIZE
Max % (MAX_PERCENT)	(MAX / SIZE) * 100
Min % (MIN_PERCENT)	(MIN / SIZE) * 100
Expectation % (EXPECTATION_PERCENT)	(EXPECTATION / PI_HMI.SIZE) * 100

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(11) VM Network Status(PI_VNI)

The following table lists the data sources in each fields of the VM Network Status(PI_VNI) record.

Table L–11: The data sources in each fields of the VM Network Status(PI_VNI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Net ID (NET_ID)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Network Objects: Description VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Network Target Objects
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE
Send Rate (SEND_RATE)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Network

PFM - View name (PFM - Manager name)	Data Sources
Send Rate (SEND_RATE)	Counters: Description: Data transmit rate Internal Name: transmitted VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Network Counters: Data transmit rate Internal Name: transmitted
Recv Rate (RECV_RATE)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Network Counters: Description: Data receive rate Internal Name: received VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Network Counters: Data receive rate Internal Name: received

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(12) VM Physical Disk Status(PI_VPDI)

The following table lists the data sources in each fields of the VM Physical Disk Status(PI_VPDI) record.

Table L–12: The data sources in each fields of the VM Physical Disk Status(PI_VPDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Disk ID (DISK_ID)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Objects: Description VMware vSphere Web Client Select machine: Virtual Machine

PFM - View name (PFM - Manager name)	Data Sources
Disk ID (DISK_ID)	TAB: Monitor / Performance / Advanced Chart Options: Disk Target Objects
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Read rate Internal Name: read VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Read rate Internal Name: read
Write Speed (WRITE_SPEED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Write rate Internal Name: write VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Write rate Internal Name: write
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Read requests Internal Name: numberRead VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Read requests

PFM - View name (PFM - Manager name)	Data Sources
Read Requests (READ_REQUESTS)	Internal Name: numberRead
Write Requests (WRITE_REQUESTS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Write requests Internal Name: numberWrite VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters:Write requests Internal Name: numberWrite
Commands (COMMANDS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Commands issued Internal Name: commands VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Commands issued Internal Name: commands
Abort Commands (ABORT_COMMANDS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Commands aborted Internal Name: commandsAborted VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: Commands aborted Internal Name: commandsAborted
Abort Commands % (ABORT_COMMANDS_PERCENT)	ABORT_COMMANDS / COMMANDS * 100
Bus Resets (BUS_RESETS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Counters: Description: Bus resets

PFM - View name (PFM - Manager name)	Data Sources
Bus Resets (BUS_RESETS)	Internal Name: busResets VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Disk Counters: bus resets Internal Name: busResets

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(13) VM Virtual Disk Status(PI_VVDI)

The following table lists the data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record.

Table L–13: The data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Disk ID (DISK_ID)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Disk Objects: Description VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Target Objects
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Read rate Internal Name: read

PFM - View name (PFM - Manager name)	Data Sources
Read Speed (READ_SPEED)	VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Read rate Internal Name: read
Write Speed (WRITE_SPEED)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Write rate Internal Name: write VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Write rate Internal Name: write
Requests Per Sec (REQUESTS_PER_SEC)	READ_REQUESTS_PER_SEC + WRITE_REQUESTS_PER_SEC
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Average read requests per second Internal Name: numberReadAveraged VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Average read requests per second Internal Name: numberReadAveraged
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Average write requests per second Internal Name: numberWriteAveraged VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Average write requests per second Internal Name: numberWriteAveraged
Total Latency (TOTAL_LATENCY)	TOTAL_READ_LATENCY + TOTAL_WRITE_LATENCY

PFM - View name (PFM - Manager name)	Data Sources
Total Read Latency (TOTAL_READ_LATENCY)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Read latency Internal Name: totalReadLatency VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Read latency Internal Name: totalReadLatency
Total Write Latency (TOTAL_WRITE_LATENCY)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Write latency Internal Name: totalWriteLatency VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Write latency Internal Name: totalWriteLatency
Outstanding Requests (OUTSTANDING_REQUESTS)	OUTSTANDING_READ_REQUESTS + OUTSTANDING_WRITE_REQUESTS
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Average number of outstanding read requests Internal Name: readOIO VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Average number of outstanding read requests Internal Name: readOIO
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Virtual disc Counters: Description: Average number of outstanding write requests Internal Name: writeOIO

PFM - View name (PFM - Manager name)	Data Sources
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Virtual disk Counters: Average number of outstanding write requests Internal Name: writeOIO

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(14) VM Status Detail(PD_VM)

The following table lists the data sources in each fields of the VM Status Detail(PD_VM) record.

Table L–14: The data sources in each fields of the VM Status Detail(PD_VM) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary General: DNS Name VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary DNS Name
VM Name (VM_NAME)	VMware vSphere Client Select machine: Virtual Machine Main Window: Upper side of tab view VMware vSphere Web Client Select machine: Virtual Machine Main Window: Upper side of tab view
Status (STATUS)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary General: State VMware vSphere Web Client Select machine: Host TAB: VMs State

PFM - View name (PFM - Manager name)	Data Sources
Information (INFORMATION)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary General: Guest OS VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary Guest OS
Snapshot (SNAPSHOT)	VMware vSphere Client Select machine: Virtual Machine Right-click: Snapshots / Snapshot Manager VMware vSphere Web Client Select machine: Virtual Machine Actions: Snapshots / Snapshot Manager

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(15) VM Status(PI_VI)

The following table lists the data sources in each fields of the VM Status(PI_VI) record.

Table L–15: The data sources in each fields of the VM Status(PI_VI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Clocks (CLOCKS)	[In case of the UseVcpuMax value set N as instance information] VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Processors General: Processor Speed VMware vSphere Web Client Select machine: Host TAB: Summary Hardware: CPU CPU Cores [In case of the UseVcpuMax value set Y as instance information]

PFM - View name (PFM - Manager name)	Data Sources
Clocks (CLOCKS)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation CPU: Resource Settings: Limit If Resource Settings is Unlimited, same as [In case of the UseVcpuMax value set N as instance information]. VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware CPU: Limit If Resource Settings is Unlimited, same as [In case of the UseVcpuMax value set N as instance information].
Count (COUNT)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary General: CPU VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware CPU
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	$\sum$ PI_VCI.USED
Insufficient (INSUFFICIENT)	$\sum$ PI_VCI.INSUFFICIENT
Request (REQUEST)	USED + INSUFFICIENT
Host Used % (HOST_USED_PERCENT)	USED / PI.CLOCKS * 100
Used % (USED_PERCENT)	USED / CLOCKS * 100
Insufficient % (INSUFFICIENT_PERCENT)	INSUFFICIENT / CLOCKS * 100
Request % (REQUEST_PERCENT)	REQUEST / CLOCKS * 100
Used Per Request (USED_PER_REQUEST)	USED / REQUEST * 100
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	INSUFFICIENT / REQUEST * 100
Affinity (AFFINITY)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary Commands: Edit Settings Virtual Machine Properties:

PFM - View name (PFM - Manager name)	Data Sources
Affinity (AFFINITY)	TAB: Resources Advanced CPU: Scheduling Affinity VMware vSphere Web Client (ESXi Host) Select machine: Virtual Machine Edit settings: Virtual Hardware CPU: Scheduling Affinity
Share (SHARE)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation CPU: Resource Settings: Shares VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware: CPU Shares
Max (MAX)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation CPU: Resource Settings: Limit VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware: CPU Limit
Min (MIN)	VMware vSphere Client Select machine: Virtual Machine TAB: Resource Allocation CPU: Resource Settings: Reservation VMware vSphere Web Client Select machine: Virtual Machine TAB: Summary VM Hardware: CPU Reservation
Expectation (EXPECTATION)	PI.CLOCKS * EXPECTATION_PERCENT / 100
Max % (MAX_PERCENT)	MAX / CLOCKS * 100
Min % (MIN_PERCENT)	MIN / CLOCKS * 100
Expectation % (EXPECTATION_PERCENT)	SHARE / ∑ SHARE * 100
Snapshot (SNAPSHOT)	VMware vSphere Client Select machine: Virtual Machine Right-click: Snapshots / Snapshot Manager

PFM - View name (PFM - Manager name)	Data Sources
Snapshot (SNAPSHOT)	VMware vSphere Web Client Select machine: Virtual Machine Actions: Snapshots / Snapshot Manager
Co-Stop (CO_STOP)	USED / USED_PERCENT * CO_STOP_PERCENT
Co-Stop % (CO_STOP_PERCENT)	VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: CPU Counters: Description: Co-stop Internal Name: costop VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: CPU Counters: Co-stop Internal Name: costop

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(16) Host Generic Data Detail(PD_HGDD)

The following table lists the data sources in each fields of the Host Generic Data Detail(PD_HGDD) record.

Table L–16: The data sources in each fields of the Host Generic Data Detail(PD_HGDD) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	The section name specified in the user-defined record definition file
Data Name (DATA_NAME)	The counter name specified in the user-defined record definition file VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Any Counters: Internal Name VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Any

PFM - View name (PFM - Manager name)	Data Sources
Data Name (DATA_NAME)	Internal Name
Object Name (OBJECT_NAME)	The object name of the counter specified in the user-defined record definition file VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Any Objects: Description VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Any Target Objects
String Data (STRING_DATA)	The counter value specified in the user-defined record definition file
Double Data (DOUBLE_DATA)	The counter value specified in the user-defined record definition file

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(17) Host Generic Data Interval(PI_HGDI)

The following table lists the data sources in each fields of the Host Generic Data Interval(PI_HGDI) record.

Table L–17: The data sources in each fields of the Host Generic Data Interval(PI_HGDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	The section name specified in the user-defined record definition file
Data Name (DATA_NAME)	The counter name specified in the user-defined record definition file VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Any Counters: Internal Name VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Any Internal Name
Object Name (OBJECT_NAME)	The object name of the counter specified in the user-defined record definition file

PFM - View name (PFM - Manager name)	Data Sources
Object Name (OBJECT_NAME)	VMware vSphere Client Select machine: Host TAB: Performance Chart Options: Any Objects: Description VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Chart Options: Any Target Objects
Sampling Time (SAMPLING_TIME)	VMware vSphere Client Select machine: Host TAB: Performance Graph: Time VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Graph: Time
String Data (STRING_DATA)	The counter value specified in the user-defined record definition file
Double Data (DOUBLE_DATA)	The counter value specified in the user-defined record definition file

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(18) VM Generic Data Detail(PD_VGDD)

The following table lists the data sources in each fields of the VM Generic Data Detail(PD_VGDD) record.

Table L–18: The data sources in each fields of the VM Generic Data Detail(PD_VGDD) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Section Name (SECTION_NAME)	The section name specified in the user-defined record definition file
Data Name (DATA_NAME)	The counter name specified in the user-defined record definition file VMware vSphere Client Select machine: Virtual Machine

PFM - View name (PFM - Manager name)	Data Sources
Data Name (DATA_NAME)	TAB: Performance Chart Options: Any Counters: Internal Name VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Any Internal Name
Object Name (OBJECT_NAME)	The object name of the counter specified in the user-defined record definition file VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Any Objects: Description VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Any Target Objects
String Data (STRING_DATA)	The counter value specified in the user-defined record definition file
Double Data (DOUBLE_DATA)	The counter value specified in the user-defined record definition file

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(19) VM Generic Data Interval(PI_VGDI)

The following table lists the data sources in each fields of the VM Generic Data Interval(PI_VGDI) record.

Table L–19: The data sources in each fields of the Host VM Generic Data Interval(PI_VGDI) record (VMware)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	VMware vSphere Client Select machine: Host TAB: Performance Graph: Time

PFM - View name (PFM - Manager name)	Data Sources
Sampling Time (SAMPLING_TIME)	VMware vSphere Web Client Select machine: Host TAB: Monitor / Performance / Advanced Graph: Time
Section Name (SECTION_NAME)	The section name specified in the user-defined record definition file
Data Name (DATA_NAME)	The counter name specified in the user-defined record definition file VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Any Counters: Internal Name VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Any Internal Name
Object Name (OBJECT_NAME)	The object name of the counter specified in the user-defined record definition file VMware vSphere Client Select machine: Virtual Machine TAB: Performance Chart Options: Any Objects: Description VMware vSphere Web Client Select machine: Virtual Machine TAB: Monitor / Performance / Advanced Chart Options: Any Target Objects
String Data (STRING_DATA)	The counter value specified in the user-defined record definition file
Double Data (DOUBLE_DATA)	The counter value specified in the user-defined record definition file

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(20) VM Virtual Disk Detail(PD_VDKD)

The following table lists the data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record.

Table L–20: The data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record (VMware)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--

PFM-View name (PFM-Manager name)	Data Sources
Datastore ID (DATASTORE_ID)	VMware vSphere Client None VMware vSphere Web Client (ESXi Host) Select machine: Storage / Datastore UUID
Datastore Name (DATASTORE_NAME)	VMware vSphere Client Select machine: Host TAB: Configuration Hardware: Storage Datastores: Identification VMware vSphere Web Client Select machine: Host TAB: Datastores Name
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Controller Name (CONTROLLER_NAME)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary Commands: Edit Settings Virtual Machine Properties: Hardware SCSI Controller IDE SATA Controller VMware vSphere Web Client Select machine: Virtual Machine Edit Settings: Virtual Hardware SCSI Controller Other / Controllers / IDE Other / Controllers / SATA Controller
Bus Number (BUS_NUMBER)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary Commands: Edit Settings Virtual Machine Properties: Hardware SCSI Controller IDE SATA Controller VMware vSphere Web Client Select machine: Virtual Machine Edit Settings: Virtual Hardware SCSI Controller Other / Controllers / IDE Other / Controllers / SATA Controller
Unit Number (UNIT_NUMBER)	VMware vSphere Client Select machine: Virtual Machine

PFM-View name (PFM-Manager name)	Data Sources
Unit Number (UNIT_NUMBER)	TAB: Summary Commands: Edit Settings Virtual Machine Properties: Hardware Hard disk: Virtual Device Node VMware vSphere Web Client Select machine: Virtual Machine Edit Settings: Virtual Hardware Hard disk: Virtual Device Node
Disk UUID (DISK_UUID)	--
Capacity (CAPACITY)	VMware vSphere Client Select machine: Virtual Machine TAB: Summary Commands: Edit Settings Virtual Machine Properties: Hardware Hard disk: Provisioned Size VMware vSphere Web Client Select Machine: Virtual Machine TAB: Summary VM Hardware Hard disk: Capacity

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(21) POD Status Detail(PD_PODD)

The following table lists the data sources in each fields of the POD Status Detail (PD_PODD) record.

Table L–21: The data sources in each fields of the POD Status Detail(PD_PODD) record (VMware)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Status (STATUS)	--
Container Count (CONTAINER_COUNT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(22) POD Status Interval(PI_PODI)

The following table lists the data sources in each fields of the POD Status Interval (PI_PODI) record.

Table L–22: The data sources in each fields of the POD Status Interval(PI_PODI) record (VMware)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(23) POD Container Status Interval(PI_POCI)

The following table lists the data sources in each fields of the POD Container Status Interval (PI_POCI) record.

Table L–23: The data sources in each fields of the POD Container Status Interval(PI_POCI) record (VMware)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Container ID (CONTAINER_ID)	--
Container Name (CONTAINER_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

L.2 In Hyper-V

This section describes the field values of the data sources in Hyper-V.

(1) Host CPU Status(PI_HCI)

The following table lists the data sources in each fields of the Host CPU Status(PI_HCI) record.

Table L–24: The data sources in each fields of the Host CPU Status(PI_HCI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
CPU ID (CPU_ID)	C	C:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Name
CPU Name (CPU_NAME)	C	C:Win32_Processor.Name
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Clocks (CLOCKS)	C	C:Win32_Processor.MaxClockSpeed
Used (USED)	CLOCKS * USED_PERCENT / 100	--
Unused (UNUSED)	CLOCKS * (100 - USED_PERCENT) / 100	--
Used % (USED_PERCENT)	TB * Δ C / Δ T / 100000	C:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.PercentTotalRunTime T: Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Timestamp_PerfTime TB:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Frequency_PerfTime
Unused % (UNUSED_PERCENT)	100 -USED_PERCENT	--

Legend:

C: Counter value.

T: Time value.

TB: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(2) Host Logical Disk Status(PI_HLDI)

The following table lists the data sources in each fields of the Host Logical Disk Status(PI_HLDI) record.

Table L–25: The data sources in each fields of the Host Logical Disk Status(PI_HLDI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Disk ID (DISK_ID)	C	C:Win32_Volume.Name (Only in case of Win32_Volume.DriveType = 3 (Hard disk))
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Size (SIZE)	C / 1024 / 1024	C:Win32_Volume.Capacity
Used (USED)	SIZE - FREE	--
Free (FREE)	C / 1024 / 1024	C:Win32_Volume.FreeSpace
Used % (USED_PERCENT)	(USED / SIZE) * 100	--
Last Update (LAST UPDATE)	--	--
Free % (FREE_PERCENT)	(FREE / SIZE) * 100	--

Legend:

C: Counter value.

--: The field stores the raw (not altered) value of performance data acquired.

(3) Host Memory Status(PI_HMI)

The following table lists the data sources in each fields of the Host Memory Status(PI_HMI) record.

Table L–26: The data sources in each fields of the Host Memory Status(PI_HMI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Size (SIZE)	C / 1024 / 1024	C:Win32_ComputerSystem.TotalPhysicalMemory
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Used (USED)	SIZE - UNUSED	--
VMM Used (VMM_USED)	--	--
VM Used (VM_USED)	--	--
Unused (UNUSED)	C	C:Win32_PerfFormattedData_PerfOS_Memory.AvailableMbytes
VM Swap Used (VM_SWAP_USED)	--	--
Host Swap Used (HOST_SWAP_USED)	--	--
Total Used (TOTAL_USED)	USED	--
Used % (USED_PERCENT)	(USED / SIZE) * 100	--
VMM Used % (VMM_USED_PERCENT)	--	--
VM Used % (VM_USED_PERCENT)	--	--
VM Swap Used % (VM_SWAP_USED_PERCENT)	--	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--	--
Total Used % (TOTAL_USED_PERCENT)	(TOTAL_USED / SIZE) * 100	--
Swap IO (SWAP_IO)	--	--
Swap In IO (SWAP_IN_IO)	--	--
Swap Out IO (SWAP_OUT_IO)	--	--

Legend:

C: Counter value.

--: The field stores the raw (not altered) value of performance data acquired.

(4) Host Network Status(PI_HNI)

The following table lists the data sources in each fields of the Host Network Status(PI_HNI) record.

Table L–27: The data sources in each fields of the Host Network Status(PI_HNI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Net ID (NET_ID)	C	C:Win32_PerfRawData_Tcpip_NetworkInterface.Name
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Rate (RATE)	Δ C / (Δ T / TB) / 1024	C:Win32_PerfRawData_Tcpip_NetworkInterface.BytesTotalPersec T:Win32_PerfRawData_Tcpip_NetworkInterface.Timestamp_PerfTime TB:Win32_PerfRawData_Tcpip_NetworkInterface.Frequency_PerfTime
Send Rate (SEND_RATE)	Δ C / (Δ T / TB) / 1024	C:Win32_PerfRawData_Tcpip_NetworkInterface.BytesSentPersec T:Win32_PerfRawData_Tcpip_NetworkInterface.Timestamp_PerfTime TB:Win32_PerfRawData_Tcpip_NetworkInterface.Frequency_PerfTime
Recv Rate (RECV_RATE)	Δ C / (Δ T / TB) / 1024	Win32_PerfRawData_Tcpip_NetworkInterface.BytesReceivedPersec T:Win32_PerfRawData_Tcpip_NetworkInterface.Timestamp_PerfTime TB:Win32_PerfRawData_Tcpip_NetworkInterface.Frequency_PerfTime

Legend:

C: Counter value.

T: Time value.

TB: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(5) Host Physical Disk Status(PI_HPDI)

The following table lists the data sources in each fields of the Host Physical Disk Status(PI_HPDI) record.

Table L–28: The data sources in each fields of the Host Physical Disk Status(PI_HPDI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Disk ID (DISK_ID)	C	C:Win32_PerfRawData_PerfDisk_PhysicalDisk.Name
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Speed (SPEED)	Δ C / (Δ T / TB) / 1024	C:Win32_PerfRawData_PerfDisk_PhysicalDisk.DiskBytesPersec T:Win32_PerfRawData_PerfDisk_PhysicalDisk.Timestamp_PerfTime TB:Win32_PerfRawData_PerfDisk_PhysicalDisk.Frequency_PerfTime
Read Speed (READ_SPEED)	Δ C / (Δ T / TB) / 1024	Win32_PerfRawData_PerfDisk_PhysicalDisk.DiskReadBytesPersec T:Win32_PerfRawData_PerfDisk_PhysicalDisk.Timestamp_PerfTime TB:Win32_PerfRawData_PerfDisk_PhysicalDisk.Frequency_PerfTime
Write Speed (WRITE_SPEED)	Δ C / (Δ T / TB) / 1024	Win32_PerfRawData_PerfDisk_PhysicalDisk.DiskWriteBytesPersec T:Win32_PerfRawData_PerfDisk_PhysicalDisk.Timestamp_PerfTime TB:Win32_PerfRawData_PerfDisk_PhysicalDisk.Frequency_PerfTime
Requests (REQUESTS)	Δ C / (Δ T / TB)	Win32_PerfRawData_PerfDisk_PhysicalDisk.DiskTransfersPersec T:Win32_PerfRawData_PerfDisk_PhysicalDisk.Timestamp_PerfTime TB:Win32_PerfRawData_PerfDisk_PhysicalDisk.Frequency_PerfTime
Read Requests (READ_REQUESTS)	Δ C / (Δ T / TB)	Win32_PerfRawData_PerfDisk_PhysicalDisk.DiskReadsPersec T:Win32_PerfRawData_PerfDisk_PhysicalDisk.Timestamp_PerfTime TB:Win32_PerfRawData_PerfDisk_PhysicalDisk.Frequency_PerfTime
Write Requests (WRITE_REQUESTS)	Δ C / (Δ T / TB)	Win32_PerfRawData_PerfDisk_PhysicalDisk.DiskWritesPersec T:Win32_PerfRawData_PerfDisk_PhysicalDisk.Timestamp_PerfTime TB:Win32_PerfRawData_PerfDisk_PhysicalDisk.Frequency_PerfTime
Commands (COMMANDS)	--	--
Abort Commands (ABORT_COMMANDS)	--	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--	--
Bus Resets (BUS_RESETS)	--	--
Device Latency (DEVICE_LATENCY)	--	--
Device Read Latency (DEVICE_READ_LATENCY)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Device Write Latency (DEVICE_READ_LATENCY)	--	--
Kernel Latency (KERNEL_LATENCY)	--	--
Kernel Read Latency (KERNEL_READ_LATENCY)	--	--
Kernel Write Latency (KERNEL_WRITE_LATENCY)	--	--
Queue Latency (QUEUE_LATENCY)	--	--
Queue Read Latency (QUEUE_READ_LATENCY)	--	--
Queue Write Latency (QUEUE_WRITE_LATENCY)	--	--

Legend:

C: Counter value.

T: Time value.

TB: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(6) Host Status Detail(PD)

The following table lists the data sources in each fields of the Host Status Detail(PD) record.

Table L–29: The data sources in each fields of the Host Status Detail(PD) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Status (STATUS)	--	--
Host Name (HOST_NAME)	Setting for monitored: VM_Host	--
Reason (REASON)	--	--
Product (PRODUCT)	C1, C2, C3, C4	C1:Win32_OperatingSystem.Caption

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Product (PRODUCT)	C1, C2, C3, C4	C2:Win32_OperatingSystem.OtherTypeDescription C3:Win32_OperatingSystem.CSDVersion C4:Win32_OperatingSystem.Version
VM Count (VM_COUNT)	The number of the instance of Msvm_ComputerSystem	--
VM Active (VM_ACTIVE)	Msvm_ComputerSystem.EnabledState is the number of 2(Enabled) in the number of the instance of Msvm_ComputerSystem	--

Legend:

Cn: Counter value.

--: The field stores the raw (not altered) value of performance data acquired.

(7) Host Status(PI)

The following table lists the data sources in each fields of the Host Status(PI) record.

Table L–30: The data sources in each fields of the Host Status(PI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Clocks (CLOCKS)	C * COUNT	C:Win32_Processor.MaxClockSpeed
Count (COUNT)	The number of C except "_Total"	C:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Name
Sampling Time (SAMPLING_TIME)	C	C:Win32_OperatingSystem.LocalDateTime
Used (USED)	CLOCKS * USED_PERCENT / 100	--
VMM Used (VMM_USED)	CLOCKS * VMM_USED_PERCENT / 100	--
VM Used (VM_USED)	CLOCKS * VM_USED_PERCENT / 100	--
VMM Console Used (VMM_CONSOLE_USED)	--	--
VMM Kernel Used (VMM_KERNEL_USED)	--	--
VMM Others Used (VMM_OTHERS_USED)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Unused (UNUSED)	CLOCKS * UNUSED_PERCENT / 100	--
Used % (USED_PERCENT)	(TB which C1 is equal to "_Total") * Δ C2 / Δ T / 100000	C1:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Name C2:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.PercentTotalRunTime T:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Timestamp_PerfTime TB:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Frequency_PerfTime
VMM Used % (VMM_USED_PERCENT)	(TB which C1 is equal to "_Total") * Δ C2 / Δ T / 100000	C1:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Name C2:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.PercentHypervisorRunTime T:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Timestamp_PerfTime TB:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Frequency_PerfTime
VM Used % (VM_USED_PERCENT)	(TB which C1 is equal to "_Total") * Δ C2 / Δ T / 100000	C1:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Name C2:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.PercentGuestRunTime T:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Timestamp_PerfTime TB:Win32_PerfRawData_HvStats_HyperVHypervisorLogicalProcessor.Frequency_PerfTime
VMM Console Used % (VMM_CONSOLE_USED_PERCENT)	--	--
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)	--	--
VMM Others Used % (VMM_OTHERS_USED_PERCENT)	--	--
Unused % (UNUSED_PERCENT)	100 - USED_PERCENT	--
Insufficient (INSUFFICIENT)	--	--
Insufficient % (INSUFFICIENT_PERCENT)	--	--
Co-Stop (CO_STOP)	--	--
Co-Stop % (CO_STOP_PERCENT)	--	--

Legend:

C, Cn: Counter value.

T: Time value.

TB: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(8) VM CPU Status(PI_VCI)

The following table lists the data sources in each fields of the VM CPU Status(PI_VCI) record.

Table L–31: The data sources in each fields of the VM CPU Status(PI_VCI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	PD_VM.VM_ID	--
CPU ID (CPU_ID)	[Root-Partition] <Id> which is part of "Root VP<Id>" in C1 [Child-Partition] <Id> which is part of "<Virtual Machine Name>:Hv VP <Id>:" in C2	C1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.Name C2:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Name
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	PD_VM.VM_NAME	--
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Used (USED)	C * USED_PERCENT / 100	C:Win32_Processor.MaxClockSpeed
Insufficient (INSUFFICIENT)	--	--
Request (REQUEST)	--	--
Used % (USED_PERCENT)	[Root-Partition] TB1 * Δ C1 / Δ T1 / 100000 [Child-Partition] TB2 * Δ C2 / Δ T2 / 100000	C1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.PercentTotalRunTime T1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.Timestamp_PerfTime TB1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.Frequency_PerfTime C2:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.PercentTotalRunTime T2:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Timestamp_PerfTime TB2:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Frequency_PerfTime
Insufficient % (INSUFFICIENT_PERCENT)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Request % (REQUEST_PERCENT)	--	--
Used Per Request (USED_PER_REQUEST)	--	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--	--
Co-Stop (CO_STOP)	--	--
Co-Stop % (CO_STOP_PERCENT)	--	--

Legend:

C, Cn: Counter value.

T: Time value.

TBn: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(9) VM Logical Disk Status(PI_VLDI)

The following table lists the data sources in each fields of the VM Logical Disk Status(PI_VLDI) record.

Table L–32: The data sources in each fields of the VM Logical Disk Status(PI_VLDI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	--	--
Disk ID (DISK_ID)	--	--
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	--	--
Sampling Time (SAMPLING_TIME)	--	--
Size (SIZE)	--	--
Used (USED)	--	--
Free (FREE)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Used % (USED_PERCENT)	--	--
Free % (FREE_PERCENT)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(10) VM Memory Status(PI_VMI)

The following table lists the data sources in each fields of the VM Memory Status(PI_VMI) record.

Table L–33: The data sources in each fields of the VM Memory Status(PI_VMI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	PD_VM.VM_ID	--
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	PD_VM.VM_NAME	--
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Size (SIZE)	C1 * C2 / 1024 / 1024	C1:Msvm_Memory.BlockSize C2:Msvm_Memory.NumberOfBlocks
Used (USED)	SIZE	--
Resource Used (RESOURCE_USED)	SIZE	--
VM Swap Used (VM_SWAP_USED)	--	--
Host Swap Used (HOST_SWAP_USED)	--	--
Unused (UNUSED)	0 (Fixed)	--
Used % (USED_PERCENT)	100 (Fixed)	--
Resource Used % (RESOURCE_USED_PERCENT)	100 (Fixed)	--
VM Swap Used % (VM_SWAP_USED_PERCENT)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--	--
VM Swap IO (VM_SWAP_IO)	--	--
VM Swap In (VM_SWAP_IN)	--	--
VM Swap Out (VM_SWAP_OUT)	--	--
Working Size (WORKING_SIZE)	--	--
Working Size % (WORKING_SIZE_PERCENT)	--	--
Share (SHARE)	--	--
Max (MAX)	--	--
Min (MIN)	--	--
Expectation (EXPECTATION)	--	--
Max % (MAX_PERCENT)	--	--
Min % (MIN_PERCENT)	--	--
Expectation % (EXPECTATION_PERCENT)	--	--

Legend:

C, Cn: Counter value.

--: The field stores the raw (not altered) value of performance data acquired.

(11) VM Network Status(PI_VNI)

The following table lists the data sources in each fields of the VM Network Status(PI_VNI) record.

Table L–34: The data sources in each fields of the VM Network Status(PI_VNI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	PD_VM.VM_ID	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Net ID (NET_ID)	<In 2012 Hyper-V> [Root-Partition] C1 + ":" + C2 (concatenated with colon) [Child-Partition] C3 + ":" + C4 (concatenated with colon) <In 2012 R2 Hyper-V, 2016 Hyper-V, and 2019 Hyper-V> [Root-Partition] C5 + ":" + C6 (concatenated with colon) [Child-Partition] C5 + ":" + C7 (concatenated with colon)	C1:Msvm_SwitchLANEndPoint.MacAddress C2:Msvm_SwitchLANEndPoint.ElementName C3:Msvm_VmLANEndPoint.MacAddress C4:Msvm_VirtualSwitch.ElementName C5:Msvm_LANEndpoint.MacAddress C6:Msvm_LANEndpoint.ElementName C7:Msvm_VirtualEthernetSwitch.ElementName
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	PD_VM.VM_NAME	--
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Rate (RATE)	Δ C / (Δ T / TB) / 1024	C:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.BytesPersec T:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.Timestamp_PerfTime TB:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.Frequency_PerfTime
Send Rate (SEND_RATE)	Δ C / (Δ T / TB) / 1024	C:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.BytesReceivedPersec T:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.Timestamp_PerfTime TB:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.Frequency_PerfTime
Recv Rate (RECV_RATE)	Δ C / (Δ T / TB) / 1024	C:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.BytesSentPersec T:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.Timestamp_PerfTime TB:Win32_PerfRawData_NvspPortStats_HyperVVirtualSwitchPort.Frequency_PerfTime

Legend:

C, Cn: Counter value.

T: Time value.

TB: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(12) VM Physical Disk Status(PI_VPDI)

The following table lists the data sources in each fields of the VM Physical Disk Status(PI_VPDI) record.

Table L–35: The data sources in each fields of the VM Physical Disk Status(PI_VPDI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	PD_VM.VM_ID	--
Disk ID (DISK_ID)	<In 2012 Hyper-V> [Virtual hard disk] C1 [Physical hard disk] "Disk " + C2 <In 2012 R2 Hyper-V, 2016 Hyper-V, and 2019 Hyper-V> [Virtual hard disk] C3 [Physical hard disk] "Disk " + C2	C1:Msvm_ResourceAllocationSettingData.Connection C2:Win32_DiskDrive.Index C3:Msvm_StorageAllocationSettingData.HostResource
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	PD_VM.VM_NAME	--
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Speed (SPEED)	READ_SPEED + WRITE_SPEED	--
Read Speed (READ_SPEED)	<In 2012 Hyper-V> Δ C1 / (Δ T1 / TB1) / 1024 <In 2012 R2 Hyper-V, 2016 Hyper-V, and 2019 Hyper-V> Δ C2 / (Δ T2 / TB2) / 1024	C1:Win32_PerfRawData_StorageStats_HyperVVirtualStorageDevice.ReadBytesPersec T1:Win32_PerfRawData_StorageStats_HyperVVirtualStorageDevice.Timestamp_PerfTime TB1:Win32_PerfRawData_StorageStats_HyperVVirtualStorageDevice.Frequency_PerfTime C2:Win32_PerfRawData_Counters_HyperVVirtualStorageDevice.ReadBytesPersec T2:Win32_PerfRawData_Counters_HyperVVirtualStorageDevice.Timestamp_PerfTime TB2:Win32_PerfRawData_Counters_HyperVVirtualStorageDevice.Frequency_PerfTime
Write Speed (WRITE_SPEED)	<In 2012 Hyper-V> Δ C1 / (Δ T1 / TB1) / 1024 <In 2012 R2 Hyper-V, 2016 Hyper-V, and 2019 Hyper-V> Δ C2 / (Δ T2 / TB2) / 1024	C1:Win32_PerfRawData_StorageStats_HyperVVirtualStorageDevice.WriteBytesPersec T1:Win32_PerfRawData_StorageStats_HyperVVirtualStorageDevice.Timestamp_PerfTime TB1:Win32_PerfRawData_StorageStats_HyperVVirtualStorageDevice.Frequency_PerfTime C2:Win32_PerfRawData_Counters_HyperVVirtualStorageDevice.WriteBytesPersec

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Write Speed (WRITE_SPEED)	<In 2012 Hyper-V> Δ C1 / (Δ T1 / TB1) / 1024 <In 2012 R2 Hyper-V, 2016 Hyper-V, and 2019 Hyper-V> Δ C2 / (Δ T2 / TB2) / 1024	T2:Win32_PerfRawData_Counters_HyperVVirtualStorageDevice.Timestamp_PerfTime TB2:Win32_PerfRawData_Counters_HyperVVirtualStorageDevice.Frequency_PerfTime
Requests (REQUESTS)	--	--
Read Requests (READ_REQUESTS)	--	--
Write Requests (WRITE_REQUESTS)	--	--
Commands (COMMANDS)	--	--
Abort Commands (ABORT_COMMANDS)	--	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--	--
Bus Resets (BUS_RESETS)	--	--

Legend:

C, Cn: Counter value.

Tn: Time value.

TBn: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(13) VM Virtual Disk Status(PI_VVDI)

The following table lists the data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record.

Table L–36: The data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record (Hyper-V)

PFM-View name (PFM-Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	--	--
Disk ID (DISK_ID)	--	--
VM Host Name (VM_HOST_NAME)	--	--

PFM-View name (PFM-Manager name)	Data Sources	
	Calculating Formula	WMI Classes
VM Name (VM_NAME)	--	--
Sampling Time (SAMPLING_TIME)	--	--
Speed (SPEED)	--	--
Read Speed (READ_SPEED)	--	--
Write Speed (WRITE_SPEED)	--	--
Requests Per Sec (REQUESTS_PER_SEC)	--	--
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	--	--
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	--	--
Total Latency (TOTAL_LATENCY)	--	--
Total Read Latency (TOTAL_READ_LATENCY)	--	--
Total Write Latency (TOTAL_WRITE_LATENCY)	--	--
Outstanding Requests (OUTSTANDING_REQUESTS)	--	--
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	--	--
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(14) VM Status Detail(PD_VM)

The following table lists the data sources in each fields of the VM Status Detail(PD_VM) record.

Table L–37: The data sources in each fields of the VM Status Detail(PD_VM) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	[Root-Partition] "Root"(Fixed) [Child-Partition] C	C:Msvm_ComputerSystem.Name
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	C	C:Msvm_ComputerSystem.ElementName
Status (STATUS)	The following character string corresponding to C 1:"OTHER", 2:"ON", 3:"OFF", 4:"STOPPING", 5:"NA", 6:"OFFLINE", 7:"TEST", 8:"DEFERRED", 9:"QUIESCE", 32768:"PAUSED", 32769:"SUSPENDED", 32770:"STARTING", 32771:"SNAPSHOTTING", 32773:"SAVING", 32774:"STOPPING", 32776:"PAUSINIG", 32777:"RESUMING", others:"UNKNOWN"	C:Msvm_ComputerSystem.EnabledState
Information (INFORMATION)	--	--
Snapshot (SNAPSHOT)	--	--

Legend:

C: Counter value.

--: The field stores the raw (not altered) value of performance data acquired.

(15) VM Status(PI_VI)

The following table lists the data sources in each fields of the VM Status(PI_VI) record.

Table L–38: The data sources in each fields of the VM Status(PI_VI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	PD_VM.VM_ID	--
Clocks (CLOCKS)	C * COUNT	C:Win32_Processor.MaxClockSpeed
Count (COUNT)	[Root-Partition] The number of C1 except "_Total" [Child-Partition] The number of according vm_name with <Name> which is part of "<Name>:Hv VP" in C2	C1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.Name C2:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Name
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	PD_VM.VM_NAME	--
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME	--
Used (USED)	CLOCKS * USED_PERCENT / 100	--
Insufficient (INSUFFICIENT)	--	--
Request (REQUEST)	--	--
Host Used % (HOST_USED_PERCENT)	--	--
Used % (USED_PERCENT)	[Root-Partition] TB1 * Δ C1 / Δ T1 / 100000 [Child-Partition] (The total of TB3# * Δ C3 / Δ T3 / 100000) / COUNT	C1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.PercentTotalRunTime T1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.Timestamp_PerfTime TB1:Win32_PerfRawData_HvStats_HyperVHypervisorRootVirtualProcessor.Frequency_PerfTime C2:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Name C3:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.PercentTotalRunTime T3:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Timestamp_PerfTime TB3:Win32_PerfRawData_HvStats_HyperVHypervisorVirtualProcessor.Frequency_PerfTime
Insufficient % (INSUFFICIENT_PERCENT)	--	--
Request % (REQUEST_PERCENT)	--	--
Used Per Request (USED_PER_REQUEST)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--	--
Affinity (AFFINITY)	--	--
Share (SHARE)	[Root-Partition] -- [Child-Partition] C	C:Msvm_ProcessorSettingData.Weight
Max (MAX)	[Root-Partition] -- [Child-Partition] MAX_PERCENT * CLOCKS / 100	--
Min (MIN)	[Root-Partition] -- [Child-Partition] MIN_PERCENT * CLOCKS / 100	--
Expectation (EXPECTATION)	--	--
Max % (MAX_PERCENT)	[Root-Partition] -- [Child-Partition] C / 1000	C:Msvm_ProcessorSettingData.Limit
Min % (MIN_PERCENT)	[Root-Partition] -- [Child-Partition] C / 1000	C:Msvm_ProcessorSettingData.Reservation
Expectation % (EXPECTATION_PERCENT)	--	--
Snapshot (SNAPSHOT)	--	--
Co-Stop (CO_STOP)	--	--
Co-Stop % (CO_STOP_PERCENT)	--	--

Legend:

C, Cn: Counter value.

Tn: Time value.

TBn: Time base.

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

#

TB3 which is same as <Virtual Host Name> of "<Virtual Host Name>:Hv VP <Id>".

(16) Host Generic Data Detail(PD_HGDD)

The following table lists the data sources in each fields of the Host Generic Data Detail(PD_HGDD) record.

30,35,35

Table L–39: The data sources in each fields of the Host Generic Data Detail(PD_HGDD) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Section Name (SECTION_NAME)	--	--
Data Name (DATA_NAME)	--	--
Object Name (OBJECT_NAME)	--	--
String Data (STRING_DATA)	--	--
Double Data (DOUBLE_DATA)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(17) Host Generic Data Interval(PI_HGDI)

The following table lists the data sources in each fields of the Host Generic Data Interval(PI_HGDI) record.

30,35,35

Table L–40: The data sources in each fields of the Host Generic Data Interval(PI_HGDI) record (Hyper-V)

PFM-View name (PFM-Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Section Name (SECTION_NAME)	--	--
Data Name (DATA_NAME)	--	--

PFM-View name (PFM-Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Object Name (OBJECT_NAME)	--	--
Sampling Time (SAMPLING_TIME)	--	--
String Data (STRING_DATA)	--	--
Double Data (DOUBLE_DATA)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(18) VM Generic Data Detail(PD_VGDD)

The following table lists the data sources in each fields of the VM Generic Data Detail(PD_VGDD) record.

30,35,35

Table L–41: The data sources in each fields of the VM Generic Data Detail(PD_VGDD) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	--	--
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	--	--
Section Name (SECTION_NAME)	--	--
Data Name (DATA_NAME)	--	--
Object Name (OBJECT_NAME)	--	--
String Data (STRING_DATA)	--	--
Double Data (DOUBLE_DATA)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(19) VM Generic Data Interval(PI_VGDI)

The following table lists the data sources in each fields of the VM Generic Data Interval(PI_VGDI) record.

30,35,35

Table L–42: The data sources in each fields of the Host VM Generic Data Interval(PI_VGDI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
VM ID (VM_ID)	--	--
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	--	--
Sampling Time (SAMPLING_TIME)	--	--
Section Name (SECTION_NAME)	--	--
Data Name (DATA_NAME)	--	--
Object Name (OBJECT_NAME)	--	--
String Data (STRING_DATA)	--	--
Double Data (DOUBLE_DATA)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(20) VM Virtual Disk Detail(PD_VDKD)

The following table lists the data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record.

Table L–43: The data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
Datastore ID (DATASTORE_ID)	--	--
Datastore Name (DATASTORE_NAME)	--	--
VM ID (VM_ID)	--	--
VM Host Name (VM_HOST_NAME)	--	--
VM Name (VM_NAME)	--	--
Controller Name (CONTROLLER_NAME)	--	--
Bus Number (BUS_NUMBER)	--	--
Unit Number (UNIT_NUMBER)	--	--
Disk UUID (DISK_UUID)	--	--
Capacity (CAPACITY)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(21) POD Status Detail(PD_PODD)

The following table lists the data sources in each fields of the POD Status Detail(PD_PODD) record.

Table L–44: The data sources in each fields of the POD Status Detail(PD_PODD) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
POD ID (POD_ID)	--	--
POD Name (POD_NAME)	--	--
Status (STATUS)	--	--
Container Count (CONTAINER_COUNT)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(22) POD Status Interval(PI_PODI)

The following table lists the data sources in each fields of the POD Status Interval(PI_PODI) record.

Table L–45: The data sources in each fields of the POD Status Interval(PI_PODI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
POD ID (POD_ID)	--	--
POD Name (POD_NAME)	--	--
Sampling Time (SAMPLING_TIME)	--	--
CPU Used (CPU_USED)	--	--
CPU Used % (CPU_USED_PERCENT)	--	--
Memory Used (MEMORY_USED)	--	--
Memory Used % (MEMORY_USED_PERCENT)	--	--
Disk Speed (DISK_SPEED)	--	--
Disk Read Speed (DISK_READ_SPEED)	--	--
Disk Write Speed (DISK_WRITE_SPEED)	--	--
Disk Requests (DISK_REQUESTS)	--	--
Disk Read Requests (DISK_READ_REQUESTS)	--	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--	--
Network Rate (NETWORK_RATE)	--	--
Network Send Rate (NETWORK_SEND_RATE)	--	--
Network Recv Rate (NETWORK_RECV_RATE)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(23) POD Container Status Interval(PI_POCI)

The following table lists the data sources in each fields of the POD Container Status Interval(PI_POCI) record.

Table L–46: The data sources in each fields of the POD Container Status Interval(PI_POCI) record (Hyper-V)

PFM - View name (PFM - Manager name)	Data Sources	
	Calculating Formula	WMI Classes
Record Type (INPUT_RECORD_TYPE)	--	--
Record Time (RECORD_TIME)	--	--
Interval (INTERVAL)	--	--
VA DeviceID (VADEVICEID)	--	--
POD ID (POD_ID)	--	--
POD Name (POD_NAME)	--	--
Sampling Time (SAMPLING_TIME)	--	--
CPU Used (CPU_USED)	--	--
CPU Used % (CPU_USED_PERCENT)	--	--
Memory Used (MEMORY_USED)	--	--
Memory Used % (MEMORY_USED_PERCENT)	--	--
Disk Speed (DISK_SPEED)	--	--
Disk Read Speed (DISK_READ_SPEED)	--	--
Disk Write Speed (DISK_WRITE_SPEED)	--	--
Disk Requests (DISK_REQUESTS)	--	--
Disk Read Requests (DISK_READ_REQUESTS)	--	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--	--
Network Rate (NETWORK_RATE)	--	--
Network Send Rate (NETWORK_SEND_RATE)	--	--
Network Recv Rate (NETWORK_RECV_RATE)	--	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

L.3 In KVM

This section describes the field values of the data sources in KVM.

(1) Host CPU Status(PI_HCI)

The following table lists the data sources in each fields of the Host CPU Status(PI_HCI) record.

Table L–47: The data sources in each fields of the Host CPU Status(PI_HCI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
CPU ID (CPU_ID)	The "processor" line in the "/proc/cpuinfo" file
CPU Name (CPU_NAME)	The "model name" line in the "/proc/cpuinfo" file
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Clocks (CLOCKS)	The value of "/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq" file / 1000 In case of not exist the file "/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq", the "cpu MHz" value in "/proc/cpuinfo" file
Used (USED)	CLOCKS * (USED_PERCENT / 100)
Unused (UNUSED)	CLOCKS - USED
Used % (USED_PERCENT)	Δ The "cpu0~" line and the "usr+nice+sys"column in the "/proc/stat" file / Δ Collection time
Unused % (UNUSED_PERCENT)	100 - USED_PERCENT

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(2) Host Logical Disk Status(PI_HLDI)

The following table lists the data sources in each fields of the Host Logical Disk Status(PI_HLDI) record.

Table L–48: The data sources in each fields of the Host Logical Disk Status(PI_HLDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	The "FileSystem" column in the result of "df -lkP"
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	The "1024-blocks" column in the result of "df -lkP" / 1024
Used (USED)	The "Used" column in the result of "df -lkP" / 1024

PFM - View name (PFM - Manager name)	Data Sources
Free (FREE)	The "Available" column in the result of "df -lkP" / 1024
Used % (USED_PERCENT)	(USED / SIZE) * 100
Last Update (LAST_UPDATE)	--
Free % (FREE_PERCENT)	(FREE / SIZE) * 100

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(3) Host Memory Status(PI_HMI)

The following table lists the data sources in each fields of the Host Memory Status(PI_HMI) record.

Table L–49: The data sources in each fields of the Host Memory Status(PI_HMI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Size (SIZE)	"Mem:" line and "total" column in the result of "free -m"
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	For Linux 6 (x64) "-/+ buffers/cache" line and "used" column in the result of "free -m" For Linux 7 "Mem:" line and "total" column - ("free" column + "buffers" column + "cached" column) in the result of "free -m -w"
VMM Used (VMM_USED)	"RSS" column in the result of "ps aux" / 1024 The total of ksmd, virt-manager, libvirtd processes
VM Used (VM_USED)	The total of PI_VMI.RESOURCE_USED
Unused (UNUSED)	For Linux 6 (x64) "-/+ buffers/cache" line and "free" column in the result of "free -m" For Linux 7 "Mem:" line and "free" column + "buffers" column + "cached" column in the result of "free -m -w"
VM Swap Used (VM_SWAP_USED)	The total of PI_VMI.VM_SWAP_USED
Host Swap Used (HOST_SWAP_USED)	The total of PI_VMI.HOST_SWAP_USED
Total Used (TOTAL_USED)	USED + VM_SWAP_USED + HOST_SWAP_USED
Used % (USED_PERCENT)	(USED / SIZE) * 100
VMM Used % (VMM_USED_PERCENT)	(VMM_USED / SIZE) * 100

PFM - View name (PFM - Manager name)	Data Sources
VM Used % (VM_USED_PERCENT)	(VM_USED / SIZE) * 100
VM Swap Used % (VM_SWAP_USED_PERCENT)	(VM_SWAP_USED / SIZE) * 100
Host Swap Used % (HOST_SWAP_USED_PERCENT)	(HOST_SWAP_USED / SIZE) * 100
Total Used % (TOTAL_USED_PERCENT)	(TOTAL_USED / SIZE) * 100
Swap IO (SWAP_IO)	SWAP_IN_IO + SWAP_OUT_IO
Swap In IO (SWAP_IN_IO)	"pages swapped in" line in the result of "vmstat -s" * "getconf PAGE_SIZE" / 1024 / 1024
Swap Out IO (SWAP_OUT_IO)	"pages swapped out" line in the result of "vmstat -s" * "getconf PAGE_SIZE" / 1024 / 1024

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(4) Host Network Status(PI_HNI)

The following table lists the data sources in each fields of the Host Network Status(PI_HNI) record.

Table L–50: The data sources in each fields of the Host Network Status(PI_HNI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Net ID (NET_ID)	For Linux 6 (x64) Interface that is not found in the results of "virsh dumpxml {domain}" within the interfaces shown by the "ifconfig" For Linux 7 Interface that is not found in the results of "virsh dumpxml {domain}" within the "Interface" field of the `/proc/net/dev` file
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE
Send Rate (SEND_RATE)	For Linux 6 (x64) Δ TX bytes in the "ifconfig" / Δ collection time / 1024 For Linux 7 Δ Transmit Bytes in the `/proc/net/dev` file / Δ collection time / 1024
Recv Rate (RECV_RATE)	For Linux 6 (x64) Δ RX bytes in the "ifconfig" / Δ collection time / 1024 For Linux 7 Δ Receive Bytes in the `/proc/net/dev` file / Δ collection time / 1024

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(5) Host Physical Disk Status(PI_HPDI)

The following table lists the data sources in each fields of the Host Physical Disk Status(PI_HPDI) record.

Table L–51: The data sources in each fields of the Host Physical Disk Status(PI_HPDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	The "Device:" column in the result of "iostat -x -k -d 1 1"
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	The "rkB/s" column in the result of "iostat -x -k -d 1 1"
Write Speed (WRITE_SPEED)	The "wkB/s" column in the result of "iostat -x -k -d 1 1"
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	The "r/s" column in the result of "iostat -x -k -d 1 1"
Write Requests (WRITE_REQUESTS)	The "w/s" column in the result of "iostat -x -k -d 1 1"
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--
Device Latency (DEVICE_LATENCY)	--
Device Read Latency (DEVICE_READ_LATENCY)	--
Device Write Latency (DEVICE_READ_LATENCY)	--
Kernel Latency (KERNEL_LATENCY)	--
Kernel Read Latency (KERNEL_READ_LATENCY)	--
Kernel Write Latency (KERNEL_WRITE_LATENCY)	--
Queue Latency (QUEUE_LATENCY)	--

PFM - View name (PFM - Manager name)	Data Sources
Queue Read Latency (QUEUE_READ_LATENCY)	--
Queue Write Latency (QUEUE_WRITE_LATENCY)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(6) Host Status Detail(PD)

The following table lists the data sources in each fields of the Host Status Detail(PD) record.

Table L–52: The data sources in each fields of the Host Status Detail(PD) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Status (STATUS)	--
Host Name (HOST_NAME)	Setting for monitored VM_Host parameter
Reason (REASON)	--
Product (PRODUCT)	Display each values of "virsh version" in comma separated value
VM Count (VM_COUNT)	The number of lines in "virsh list -all"
VM Active (VM_ACTIVE)	The number of lines which "status" in the result of "virsh list -all" is linage of "running, idle, paused"

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(7) Host Status(PI)

The following table lists the data sources in each fields of the Host Status(PI) record.

Table L–53: The data sources in each fields of the Host Status(PI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Clocks (CLOCKS)	PI_HCI.CLOCKS * COUNT

PFM - View name (PFM - Manager name)	Data Sources
Count (COUNT)	Count the number of CPUs in "/proc/cpuinfo"
Sampling Time (SAMPLING_TIME)	date -Iseconds
Used (USED)	CLOCKS * USED_PERCENT / 100
VMM Used (VMM_USED)	CLOCKS * VMM_USED_PRECENT / 100
VM Used (VM_USED)	CLOCKS * VM_USED_PERCENT / 100
VMM Console Used (VMM_CONSOLE_USED)	CLOCKS * VMM_CONSOLE_USED_PERCENT / 100
VMM Kernel Used (VMM_KERNEL_USED)	CLOCKS * VMM_KERNEL_USED_PERCENT / 100
VMM Others Used (VMM_OTHERS_USED)	CLOCKS * VMM_OTHERS_USED_PERCENT / 100
Unused (UNUSED)	CLOCKS - USED
Used % (USED_PERCENT)	Δ (The "CPU" line and "usr+nice+sys" column in the "/proc/stat" file) / (Δ Collection time * 100 * COUNT) * 100
VMM Used % (VMM_USED_PERCENT)	USED_PERCENT - VM_USED_PERCENT
VM Used % (VM_USED_PERCENT)	Δ (The "CPU" line and "guest" column in the "/proc/stat" file) / (Δ Collection time * 100 * COUNT) * 100
VMM Console Used % (VMM_CONSOLE_USED_PERCEN T)	The "virt-manager" line and "PID" column in the result of "ps aux" Δ (The "PID" line and "TIME+" column of the "virt-manager" section in the result of "top -b -n 1") / (Δ Collection time * COUNT) * 100
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)	The "libvirtd, ksmd" line and "PID" column in the result of "ps aux" Δ (The "libvirtd,ksmd" line and "PID" column in the result of "top -b -n 1") / (Δ Collection time * COUNT) * 100
VMM Others Used % (VMM_OTHERS_USED_PERCENT)	VMM_USED_PERCENT - VMM_CONSOLE_USED_PERCENT - VMM_KERNEL_USED_PERCENT
Unused % (UNUSED_PERCENT)	UNUSED / CLOCKS * 100
Insufficient (INSUFFICIENT)	--
Insufficient % (INSUFFICIENT_PERCENT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(8) VM CPU Status(PI_VCI)

The following table lists the data sources in each fields of the VM CPU Status(PI_VCI) record.

Table L–54: The data sources in each fields of the VM CPU Status(PI_VCI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
CPU ID (CPU_ID)	The "VCPU" line in the result of "virsh vcpuinfo {domain}"
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	PI_HCI.CLOCKS * USED_PERCENT / 100
Insufficient (INSUFFICIENT)	--
Request (REQUEST)	--
Used % (USED_PERCENT)	Δ The "CPU time" line in the result of "virsh vcpuinfo {domain}" / Δ Collection time * 100
Insufficient % (INSUFFICIENT_PERCENT)	--
Request % (REQUEST_PERCENT)	--
Used Per Request (USED_PER_REQUEST)	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(9) VM Logical Disk Status(PI_VLDI)

The following table lists the data sources in each fields of the VM Logical Disk Status(PI_VLDI) record.

Table L–55: The data sources in each fields of the VM Logical Disk Status(PI_VLDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Free (FREE)	--
Used % (USED_PERCENT)	--
Free % (FREE_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(10) VM Memory Status(PI_VMI)

The following table lists the data sources in each fields of the VM Memory Status(PI_VMI) record.

Table L–56: The data sources in each fields of the VM Memory Status(PI_VMI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	The "Kbytes" value in "total" line and in the result of "pmap -x {pid}" / 1024
Used (USED)	SIZE
Resource Used (RESOURCE_USED)	The "Dirty" value in "total" line and in the result of "pmap -x {pid}" / 1024

PFM - View name (PFM - Manager name)	Data Sources
VM Swap Used (VM_SWAP_USED)	The "RSS" value in "total" line and in the result of "pmap -x {pid}" / 1024 - Resource Used
Host Swap Used (HOST_SWAP_USED)	SIZE - The "RSS" value in "total" line and in the result of "pmap -x {pid}" / 1024
Unused (UNUSED)	0(Fixed)
Used % (USED_PERCENT)	(USED / SIZE) * 100
Resource Used % (RESOURCE_USED_PERCENT)	(RESOURCE_USED / SIZE) * 100
VM Swap Used % (VM_SWAP_USED_PERCENT)	(VM_SWAP_USED / SIZE) * 100
Host Swap Used % (HOST_SWAP_USED_PERCENT)	(HOST_SWAP_USED / SIZE) * 100
VM Swap IO (VM_SWAP_IO)	--
VM Swap In (VM_SWAP_IN)	--
VM Swap Out (VM_SWAP_OUT)	--
Working Size (WORKING_SIZE)	The "Usedmemory" line in the result of "virsh dominfo {domain}" / 1024
Working Size % (WORKING_SIZE_PERCENT)	(WORKING_SIZE / SIZE) * 100
Share (SHARE)	--
Max (MAX)	The "Max memory" line in the result of "virsh dominfo {domain}"
Min (MIN)	--
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	MAX / PI_HMI.SIZE * 100
Min % (MIN_PERCENT)	--
Expectation % (EXPECTATION_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(11) VM Network Status(PI_VNI)

The following table lists the data sources in each fields of the VM Network Status(PI_VNI) record.

Table L–57: The data sources in each fields of the VM Network Status(PI_VNI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--

PFM - View name (PFM - Manager name)	Data Sources
VM ID (VM_ID)	PD_VM.VM_ID
Net ID (NET_ID)	The "interface" value in the result of "virsh dumpxml {domain}" / mac address
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE
Send Rate (SEND_RATE)	For Linux 6 (x64) (Δ "RX bytes" value in the result of "ifconfig" / Δ Collection time) / 1024 For Linux 7 Δ Receive Bytes in the `/proc/net/dev` file / Δ Collection time / 1024
Recv Rate (RECV_RATE)	For Linux 6 (x64) (Δ "TX bytes" value in the result of "ifconfig" / Δ Collection time) / 1024 For Linux 7 Δ Transmit Bytes in the `/proc/net/dev` file / Δ Collection time / 1024

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(12) VM Physical Disk Status(PI_VPDI)

The following table lists the data sources in each fields of the VM Physical Disk Status(PI_VPDI) record.

Table L–58: The data sources in each fields of the VM Physical Disk Status(PI_VPDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Disk ID (DISK_ID)	The character string in the "`source file=''`" which is inside of the "`<disk type='file' device='disk'>`" in the result of "dumpxml {domain}"
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED

PFM - View name (PFM - Manager name)	Data Sources
Read Speed (READ_SPEED)	(Δ The "rd_bytes" line in the result of "virsh domblkstat {domain} {device}" / Δ Collection time) / 1024
Write Speed (WRITE_SPEED)	(Δ The "wr_bytes" line in the result of "virsh domblkstat {domain} {device}" / Δ Collection time) / 1024
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	Δ The "rd_req" line in the result of "virsh domblkstat {domain} {device}" / Δ Collection time
Write Requests (WRITE_REQUESTS)	Δ The "wr_req" line in the result of "virsh domblkstat {domain} {device}" / Δ Collection time
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(13) VM Virtual Disk Status(PI_VVDI)

The following table lists the data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record.

Table L–59: The data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests Per Sec (REQUESTS_PER_SEC)	--

PFM - View name (PFM - Manager name)	Data Sources
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	--
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	--
Total Latency (TOTAL_LATENCY)	--
Total Read Latency (TOTAL_READ_LATENCY)	--
Total Write Latency (TOTAL_WRITE_LATENCY)	--
Outstanding Requests (OUTSTANDING_REQUESTS)	--
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	--
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(14) VM Status Detail(PD_VM)

The following table lists the data sources in each fields of the VM Status Detail(PD_VM) record.

Table L–60: The data sources in each fields of the VM Status Detail(PD_VM) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	The "UUID" line in the result of "virsh dominfo {domain}"
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	The "Name" column in the result of "virsh list -all"
Status (STATUS)	The "state" line in the result of "virsh dominfo {domain}"
Information (INFORMATION)	The "OS Type" line in the result of "virsh dominfo {domain}"
Snapshot (SNAPSHOT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(15) VM Status(PI_VI)

The following table lists the data sources in each fields of the VM Status(PI_VI) record.

Table L–61: The data sources in each fields of the VM Status(PI_VI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Clocks (CLOCKS)	PI_HCI.CLOCKS * The number of usable CPUs
Count (COUNT)	The "CPU(s)" line in the result of "virsh dominfo {domain}"
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	CLOCKS * USED_PERCENT / 100
Insufficient (INSUFFICIENT)	--
Request (REQUEST)	--
Host Used % (HOST_USED_PERCENT)	USED / PI.CLOCKS * 100
Used % (USED_PERCENT)	The "QEMU" line and "PID" column in the result of "ps aux" Δ The "PID" line and "TIME+" column in the result of "top -b -n 1" / (Δ Collection time * COUNT)
Insufficient % (INSUFFICIENT_PERCENT)	--
Request % (REQUEST_PERCENT)	--
Used Per Request (USED_PER_REQUEST)	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--
Affinity (AFFINITY)	The "CPU Affinity" column in the result of "virsh vcpuinfo {domain}"
Share (SHARE)	--
Max (MAX)	CLOCKS
Min (MIN)	--
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	100%(Fixed)
Min % (MIN_PERCENT)	--
Expectation % (EXPECTATION_PERCENT)	--
Snapshot (SNAPSHOT)	--
Co-Stop (CO_STOP)	--

PFM - View name (PFM - Manager name)	Data Sources
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(16) Host Generic Data Detail(PD_HGDD)

The following table lists the data sources in each fields of the Host Generic Data Detail(PD_HGDD) record.

Table L–62: The data sources in each fields of the Host Generic Data Detail(PD_HGDD) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(17) Host Generic Data Interval(PI_HGDI)

The following table lists the data sources in each fields of the Host Generic Data Interval(PI_HGDI) record.

Table L–63: The data sources in each fields of the Host Generic Data Interval(PI_HGDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--

PFM - View name (PFM - Manager name)	Data Sources
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
Sampling Time (SAMPLING_TIME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(18) VM Generic Data Detail(PD_VGDD)

The following table lists the data sources in each fields of the VM Generic Data Detail(PD_VGDD) record.

Table L–64: The data sources in each fields of the VM Generic Data Detail(PD_VGDD) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(19) VM Generic Data Interval(PI_VGDI)

The following table lists the data sources in each fields of the VM Generic Data Interval(PI_VGDI) record.

Table L–65: The data sources in each fields of the VM Generic Data Interval(PI_VGDI) record (KVM)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(20) VM Virtual Disk Detail(PD_VDKD)

The following table lists the data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record.

Table L–66: The data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record (KVM)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Datastore ID (DATASTORE_ID)	--
Datastore Name (DATASTORE_NAME)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Controller Name (CONTROLLER_NAME)	--

PFM-View name (PFM-Manager name)	Data Sources
Bus Number (BUS_NUMBER)	--
Unit Number (UNIT_NUMBER)	--
Disk UUID (DISK_UUID)	--
Capacity (CAPACITY)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(21) POD Status Detail(PD_PODD)

The following table lists the data sources in each fields of the POD Status Detail(PD_PODD) record.

Table L–67: The data sources in each fields of the POD Status Detail(PD_PODD) record (KVM)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Status (STATUS)	--
Container Count (CONTAINER_COUNT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(22) POD Status Interval(PI_PODI)

The following table lists the data sources in each fields of the POD Status Interval(PI_PODI) record.

Table L–68: The data sources in each fields of the POD Status Interval(PI_PODI) record (KVM)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--

PFM-View name (PFM-Manager name)	Data Sources
Status (STATUS)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(23) POD Container Status Interval(PI_POCI)

The following table lists the data sources in each fields of the POD Container Status Interval(PI_POCI) record.

Table L–69: The data sources in each fields of the POD Container Status Interval(PI_POCI) record (KVM)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--

PFM-View name (PFM-Manager name)	Data Sources
Container ID (CONTAINER_ID)	--
Container Name (CONTAINER_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

L.4 In Docker environment

This section describes the field values of the data sources in Docker environment.

(1) Host CPU Status(PI_HCI)

The following table lists the data sources in each fields of the Host CPU Status(PI_HCI) record.

Table L–70: The data sources in each fields of the Host CPU Status(PI_HCI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--

PFM - View name (PFM - Manager name)	Data Sources
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
CPU ID (CPU_ID)	--
CPU Name (CPU_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Clocks (CLOCKS)	--
Used (USED)	--
Unused (UNUSED)	--
Used % (USED_PERCENT)	--
Unused % (UNUSED_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(2) Host Logical Disk Status(PI_HLDI)

The following table lists the data sources in each fields of the Host Logical Disk Status(PI_HLDI) record.

Table L–71: The data sources in each fields of the Host Logical Disk Status(PI_HLDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Free (FREE)	--
Used % (USED_PERCENT)	--
Last Update (LAST_UPDATE)	--
Free % (FREE_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(3) Host Memory Status(PI_HMI)

The following table lists the data sources in each fields of the Host Memory Status(PI_HMI) record.

Table L–72: The data sources in each fields of the Host Memory Status(PI_HMI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Size (SIZE)	--
Sampling Time (SAMPLING_TIME)	--
Used (USED)	--
VMM Used (VMM_USED)	--
VM Used (VM_USED)	--
Unused (UNUSED)	--
VM Swap Used (VM_SWAP_USED)	--
Host Swap Used (HOST_SWAP_USED)	--
Total Used (TOTAL_USED)	--
Used % (USED_PERCENT)	--
VMM Used % (VMM_USED_PERCENT)	--
VM Used % (VM_USED_PERCENT)	--
VM Swap Used % (VM_SWAP_USED_PERCENT)	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--
Total Used % (TOTAL_USED_PERCENT)	--
Swap IO (SWAP_IO)	--
Swap In IO (SWAP_IN_IO)	--
Swap Out IO (SWAP_OUT_IO)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(4) Host Network Status(PI_HNI)

The following table lists the data sources in each fields of the Host Network Status(PI_HNI) record.

Table L–73: The data sources in each fields of the Host Network Status(PI_HNI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Net ID (NET_ID)	--
Sampling Time (SAMPLING_TIME)	--
Rate (RATE)	--
Send Rate (SEND_RATE)	--
Recv Rate (RECV_RATE)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(5) Host Physical Disk Status(PI_HPDI)

The following table lists the data sources in each fields of the Host Physical Disk Status(PI_HPDI) record.

Table L–74: The data sources in each fields of the Host Physical Disk Status(PI_HPDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	--
Sampling Time (SAMPLING_TIME)	--
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests (REQUESTS)	--
Read Requests (READ_REQUESTS)	--
Write Requests (WRITE_REQUESTS)	--
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--

PFM - View name (PFM - Manager name)	Data Sources
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--
Device Latency (DEVICE_LATENCY)	--
Device Read Latency (DEVICE_READ_LATENCY)	--
Device Write Latency (DEVICE_READ_LATENCY)	--
Kernel Latency (KERNEL_LATENCY)	--
Kernel Read Latency (KERNEL_READ_LATENCY)	--
Kernel Write Latency (KERNEL_WRITE_LATENCY)	--
Queue Latency (QUEUE_LATENCY)	--
Queue Read Latency (QUEUE_READ_LATENCY)	--
Queue Write Latency (QUEUE_WRITE_LATENCY)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(6) Host Status Detail(PD)

The following table lists the data sources in each fields of the Host Status Detail(PD) record.

Table L–75: The data sources in each fields of the Host Status Detail(PD) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Status (STATUS)	--
Host Name (HOST_NAME)	Setting for monitored VM_Host parameter
Reason (REASON)	--
Product (PRODUCT)	• Command `Version`, `ApiVersion`, `GitCommit`, `GoVersion`, `Os`, and `Arch` of the server information displayed by the `docker version` in comma separated values • API

PFM - View name (PFM - Manager name)	Data Sources
Product (PRODUCT)	`Version`, `ApiVersion`, `GitCommit`, `GoVersion`, `Os`, and `Arch` of `/version` in comma separated values
VM Count (VM_COUNT)	• Command The number of lines in "docker ps - a" • API `Containers` of `/info`
VM Active (VM_ACTIVE)	For Docker environment (Linux): • Command The number of lines in " docker ps" • API(Docker environment 1.7.1-1.9.1) In "`/containers/`*container-ID*`/json`", the number of the containers with non-zero `/State/Pid` • API(Docker environment 1.10.3 or later) ContainersRunning + ContainersPaused of `/info` For Docker environment (Windows): • Command The number of lines in " docker ps" • API ContainersRunning + ContainersPaused of `/info`

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(7) Host Status(PI)

The following table lists the data sources in each fields of the Host Status(PI) record.

Table L–76: The data sources in each fields of the Host Status(PI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Clocks (CLOCKS)	--
Count (COUNT)	--
Sampling Time (SAMPLING_TIME)	--
Used (USED)	--
VMM Used (VMM_USED)	--
VM Used (VM_USED)	--
VMM Console Used (VMM_CONSOLE_USED)	--

PFM - View name (PFM - Manager name)	Data Sources
VMM Kernel Used (VMM_KERNEL_USED)	--
VMM Others Used (VMM_OTHERS_USED)	--
Unused (UNUSED)	--
Used % (USED_PERCENT)	--
VMM Used % (VMM_USED_PERCENT)	--
VM Used % (VM_USED_PERCENT)	--
VMM Console Used % (VMM_CONSOLE_USED_PERCENT)	--
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)	--
VMM Others Used % (VMM_OTHERS_USED_PERCENT)	--
Unused % (UNUSED_PERCENT)	--
Insufficient (INSUFFICIENT)	--
Insufficient % (INSUFFICIENT_PERCENT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(8) VM CPU Status(PI_VCI)

The following table lists the data sources in each fields of the VM CPU Status(PI_VCI) record.

Table L–77: The data sources in each fields of the VM CPU Status(PI_VCI) record (Docker environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	For Docker environment (Linux): PD_VM.VM_ID For Docker environment (Windows):

PFM-View name (PFM-Manager name)	Data Sources
VM ID (VM_ID)	--
CPU ID (CPU_ID)	For Docker environment (Linux): • Command None • API In "/containers/*container-ID*/stats?stream=0", the index of "cpu_stats/cpu_usage/percpu_usage" For Docker environment (Windows): --
VM Host Name (VM_HOST_NAME)	For Docker environment (Linux): PD_VM.VM_HOST_NAME For Docker environment (Windows): --
VM Name (VM_NAME)	For Docker environment (Linux): PD_VM.VM_NAME For Docker environment (Windows): --
Sampling Time (SAMPLING_TIME)	For Docker environment (Linux): PI_VI.SAMPLING_TIME For Docker environment (Windows): --
Used (USED)	--
Insufficient (INSUFFICIENT)	--
Request (REQUEST)	--
Used % (USED_PERCENT)	For Docker environment (Linux): • Command None • API In "/containers/*container-ID*/stats?stream=0", "Δ cpu_stats/cpu_usage/percpu_usage"/"Δ cpu_stats/system_cpu_usage" * number of "cpu_stats/cpu_usage/percpu_usage" * 100 For Docker environment (Windows): --
Insufficient % (INSUFFICIENT_PERCENT)	--
Request % (REQUEST_PERCENT)	--
Used Per Request (USED_PER_REQUEST)	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(9) VM Logical Disk Status(PI_VLDI)

The following table lists the data sources in each fields of the VM Logical Disk Status(PI_VLDI) record.

Table L–78: The data sources in each fields of the VM Logical Disk Status(PI_VLDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Free (FREE)	--
Used % (USED_PERCENT)	--
Free % (FREE_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(10) VM Memory Status(PI_VMI)

The following table lists the data sources in each fields of the VM Memory Status(PI_VMI) record.

Table L–79: The data sources in each fields of the VM Memory Status(PI_VMI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--

PFM - View name (PFM - Manager name)	Data Sources
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI_VI.SAMPLING_TIME
Size (SIZE)	For Docker environment (Linux): • Command None • API In "/containers/*container-ID*/stats?stream=0", "memory_stats/limit" / 1024 / 1024 For Docker environment (Windows): • Command Total Memory of / docker info • API MemTotal / 1024 / 1024 of / info
Used (USED)	For Docker environment (Linux): • Command None • API In "/containers/*container-ID*/stats?stream=0", "memory_stats/usage" / 1024 / 1024 For Docker environment (Windows): • Command None • API In "/containers/*container-ID*/stats?stream=0", "memory_stats/privateworkingset" / 1024 / 1024
Resource Used (RESOURCE_USED)	--
VM Swap Used (VM_SWAP_USED)	--
Host Swap Used (HOST_SWAP_USED)	--
Unused (UNUSED)	--
Used % (USED_PERCENT)	(USED / SIZE) * 100
Resource Used % (RESOURCE_USED_PERCENT)	--
VM Swap Used % (VM_SWAP_USED_PERCENT)	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--
VM Swap IO (VM_SWAP_IO)	--
VM Swap In (VM_SWAP_IN)	--
VM Swap Out (VM_SWAP_OUT)	--
Working Size (WORKING_SIZE)	--

PFM - View name (PFM - Manager name)	Data Sources
Working Size % (WORKING_SIZE_PERCENT)	--
Share (SHARE)	--
Max (MAX)	• Command In "`docker inspect` *container-ID*", "`HostConfig/Memory`" / 1024 / 1024 • API In "`/containers/`*container-ID*`/json`", "`HostConfig/Memory`" / 1024 / 1024
Min (MIN)	• Command In "`docker inspect` *container-ID*", "`HostConfig/MemoryReservation`" / 1024 / 1024 • API "`/containers/`*container-ID*`/json`" of "`HostConfig/MemoryReservation`" / 1024 / 1024
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	• Command MAX / Total Memory of "`docker info`" * 100 • API MAX / (MemTotal of `/info` / 1024 / 1024) * 100
Min % (MIN_PERCENT)	• Command MIN / Total Memory of "`docker info`" * 100 • API MIN / (MemTotal of "`/info`" / 1024 / 1024) * 100
Expectation % (EXPECTATION_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(11) VM Network Status(PI_VNI)

The following table lists the data sources in each fields of the VM Network Status(PI_VNI) record.

Table L–80: The data sources in each fields of the VM Network Status(PI_VNI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Net ID (NET_ID)	• Command None • API

PFM - View name (PFM - Manager name)	Data Sources
Net ID (NET_ID)	In "/containers/*container-ID*/stats?stream=0", "networks/*key name*"
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI_VI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE
Send Rate (SEND_RATE)	• Command None • API In "/containers/*container-ID*/stats?stream=0"," Δ networks/*key name*/tx_bytes" / Δ read / 1024
Recv Rate (RECV_RATE)	• Command None • API In "/containers/*container-ID*/stats?stream=0"," Δ networks/*key name*/rd_bytes" / Δ read / 1024

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(12) VM Physical Disk Status(PI_VPDI)

The following table lists the data sources in each fields of the VM Physical Disk Status(PI_VPDI) record.

Table L–81: The data sources in each fields of the VM Physical Disk Status(PI_VPDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Disk ID (DISK_ID)	For Docker environment (Linux): • Command None • API In "/containers/*container-ID*/stats?stream=0", "blkio_stats/io_service_bytes_recursive/*index*/major" + ":" + "blkio_stats/io_service_bytes_recursive/*index*/minor" For Docker environment (Windows):

PFM - View name (PFM - Manager name)	Data Sources
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI_VI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	For Docker environment (Linux): • Command None • API In "`/containers/`*container-ID*`/stats?stream=0`", "`blkio_stats/io_service_bytes_recursive/`*index*`/op`" is `Read`, " Δ `blkio_stats/io_service_bytes_recursive/`*index*`/value`" / Δ read / 1024 For Docker environment (Windows): • Command None • API In "`/containers/`*container-ID*`/stats?stream=0`", Δ `storage_stats/read_size_bytes` / Δ read / 1024
Write Speed (WRITE_SPEED)	For Docker environment (Linux): • Command None • API In "`/containers/`*container-ID*`/stats?stream=0`", "`blkio_stats/io_service_bytes_recursive/`*index*`/op`" is `Write`, " Δ `blkio_stats/io_service_bytes_recursive/`*index*`/value`" / Δ read / 1024 For Docker environment (Windows): • Command None • API "`/containers/`*container-ID*`/stats?stream=0`", Δ `storage_stats/write_size_bytes`" / Δ read / 1024
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	For Docker environment (Linux): • Command None • API In "`/containers/`*container-ID*`/stats?stream=0`", "`blkio_stats/io_serviced_recursive/`*index*`/op`" is `Read`, " Δ `blkio_stats/io_serviced_recursive/`*index*`/value`" / Δ read For Docker environment (Windows): • Command None

PFM - View name (PFM - Manager name)	Data Sources
Read Requests (READ_REQUESTS)	• API In "`/containers/`*container-ID*`/stats?stream=0`", Δ `storage_stats/read_count_normalized`"/ Δ read
Write Requests (WRITE_REQUESTS)	For Docker environment (Linux): • Command None • API In "`/containers/`*container-ID*`/stats?stream=0`", "`blkio_stats/io_serviced_recursive/`*index*`/op`" is `Write`, " Δ `blkio_stats/io_serviced_recursive/`*index*`/value`"/ Δ read For Docker environment (Windows): • Command None • API In "`/containers/`*container-ID*`/stats?stream=0`", Δ `storage_stats/write_count_normalized`"/ Δ read
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(13) VM Virtual Disk Status(PI_VVDI)

The following table lists the data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record.

Table L–82: The data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--

PFM - View name (PFM - Manager name)	Data Sources
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests Per Sec (REQUESTS_PER_SEC)	--
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	--
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	--
Total Latency (TOTAL_LATENCY)	--
Total Read Latency (TOTAL_READ_LATENCY)	--
Total Write Latency (TOTAL_WRITE_LATENCY)	--
Outstanding Requests (OUTSTANDING_REQUESTS)	--
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	--
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(14) VM Status Detail(PD_VM)

The following table lists the data sources in each fields of the VM Status Detail(PD_VM) record.

Table L–83: The data sources in each fields of the VM Status Detail(PD_VM) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	• Command

PFM - View name (PFM - Manager name)	Data Sources
VM ID (VM_ID)	Id of "`docker inspect` *container-ID*" • API Id of "`/containers/json?all=1`"
VM Host Name (VM_HOST_NAME)	• Command "`Config/Hostname`" of "`docker inspect` *container-ID*" • API "`Config/Hostname`" of "`/containers/`*container-ID*`/json`"
VM Name (VM_NAME)	• Command Name of "`docker inspect` *container-ID*" • API Name of "`/containers/`*container-ID*`/json`"
Status (STATUS)	• Command None • API In "`/containers/`*container-ID*`/json` ", `State/Dead` is true →DEAD `State/OOMKilled` is true →OOMKILLED `State/Paused` is true →PAUSED `State/Restarting` is true →STARTING `State/Running` is true →ON All is false →OFF
Information (INFORMATION)	• Command COMMAND of "`docker ps -a`" • API Path of "`/containers/`*container-ID*`/json`"
Snapshot (SNAPSHOT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(15) VM Status(PI_VI)

The following table lists the data sources in each fields of the VM Status(PI_VI) record.

Table L–84: The data sources in each fields of the VM Status(PI_VI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Clocks (CLOCKS)	--

PFM - View name (PFM - Manager name)	Data Sources
Count (COUNT)	• Command CPUs of "`docker info`" • API NCPU of "`/info`"
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	• Command None • API SystemTime of "`/info`"
Used (USED)	--
Insufficient (INSUFFICIENT)	--
Request (REQUEST)	--
Host Used % (HOST_USED_PERCENT)	--
Used % (USED_PERCENT)	For Docker environment (Linux): • Command CPU % of "`docker stats`" • API "`/containers/`*container-ID*`/stats?stream=0`", " Δ `cpu_stats/cpu_usage/total_usage`" / " Δ `cpu_stats/system_cpu_usage`" * number of "`cpu_stats/cpu_usage/percpu_usage`" * 100 For Docker environment (Windows): • Command CPU % of "`docker stats`" • API Δ `cpu_stats/cpu_usage/total_usage` of "`/containers/`*container-ID*`/stats?stream=0`" / (Δ read of "`/containers/`*container-ID*`/stats?stream=0`" * NCPU of "`/info`") * 100
Insufficient % (INSUFFICIENT_PERCENT)	--
Request % (REQUEST_PERCENT)	--
Used Per Request (USED_PER_REQUEST)	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--
Affinity (AFFINITY)	• Command "`HostConfig/CpusetCpus`" of "`docker inspect` *container-ID*" • API "`HostConfig/CpusetCpus`" of "`/containers/`*container-ID*`/json`"
Share (SHARE)	• Command "`HostConfig/CpuShares`" of "`docker inspect` *container-ID*"

PFM - View name (PFM - Manager name)	Data Sources
Share (SHARE)	• API "HostConfig/CpuShares" of "/containers/*container-ID*/json"
Max (MAX)	--
Min (MIN)	--
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	--
Min % (MIN_PERCENT)	--
Expectation % (EXPECTATION_PERCENT)	SHARE#1 / ∑ SHARE#2 * 100 #1 When SHARE is 0, calculate using the default value 1024. #2 ∑ SHARE is a total value of SHARE for containers that are running.
Snapshot (SNAPSHOT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(16) Host Generic Data Detail(PD_HGDD)

The following table lists the data sources in each fields of the Host Generic Data Detail(PD_HGDD) record.

Table L–85: The data sources in each fields of the Host Generic Data Detail(PD_HGDD) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(17) Host Generic Data Interval(PI_HGDI)

The following table lists the data sources in each fields of the Host Generic Data Interval(PI_HGDI) record.

Table L–86: The data sources in each fields of the Host Generic Data Interval(PI_HGDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
Sampling Time (SAMPLING_TIME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(18) VM Generic Data Detail(PD_VGDD)

The following table lists the data sources in each fields of the VM Generic Data Detail(PD_VGDD) record.

Table L–87: The data sources in each fields of the VM Generic Data Detail(PD_VGDD) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--

PFM - View name (PFM - Manager name)	Data Sources
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(19) VM Generic Data Interval(PI_VGDI)

The following table lists the data sources in each fields of the VM Generic Data Interval(PI_VGDI) record.

Table L–88: The data sources in each fields of the VM Generic Data Interval(PI_VGDI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(20) VM Virtual Disk Detail(PD_VDKD)

The following table lists the data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record.

Table L–89: The data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record (Docker environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--

PFM-View name (PFM-Manager name)	Data Sources
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Datastore ID (DATASTORE_ID)	--
Datastore Name (DATASTORE_NAME)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Controller Name (CONTROLLER_NAME)	--
Bus Number (BUS_NUMBER)	--
Unit Number (UNIT_NUMBER)	--
Disk UUID (DISK_UUID)	--
Capacity (CAPACITY)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(21) POD Status Detail(PD_PODD)

The following table lists the data sources in each fields of the POD Status Detail(PD_PODD) record.

Table L–90: The data sources in each fields of the POD Status Detail(PD_PODD) record (Docker environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Status (STATUS)	--
Container Count (CONTAINER_COUNT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(22) POD Status Interval(PI_PODI)

The following table lists the data sources in each fields of the POD Status Interval(PI_PODI) record.

Table L–91: The data sources in each fields of the POD Status Interval(PI_PODI) record (Docker environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(23) POD Container Status Interval(PI_POCI)

The following table lists the data sources in each fields of the POD Container Status Interval(PI_POCI) record.

Table L–92: The data sources in each fields of the POD Container Status Interval(PI_POCI) record (Docker environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Container ID (CONTAINER_ID)	--
Container Name (CONTAINER_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

L.5 In Podman environment

This section describes the field values of the data sources in Podman environment.

(1) Host CPU Status(PI_HCI)

The following table lists the data sources in each fields of the Host CPU Status(PI_HCI) record.

Table L–93: The data sources in each fields of the Host CPU Status(PI_HCI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
CPU ID (CPU_ID)	`processor` line in the `/proc/cpuinfo` file
CPU Name (CPU_NAME)	`model name` line in the `/proc/cpuinfo` file
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Clocks (CLOCKS)	Value of the `/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq` file / 1000 Value of `cpu MHz` of the `/proc/cpuinfo` file if the `/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq` file does not exist
Used (USED)	CLOCKS * (USED_PERCENT / 100)
Unused (UNUSED)	CLOCKS - USED
Used % (USED_PERCENT)	Δ (`cpu0~` line and the (`usr` + `nice` + `sys` column) in the `/proc/stat` file / 100#) / Δ (Collection time * 100) #: The CPU utilization time of `/proc/stat` is indicated in hundredths of a second.
Unused % (UNUSED_PERCENT)	(UNUSED / CLOCKS) * 100

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(2) Host Logical Disk Status(PI_HLDI)

The following table lists the data sources in each fields of the Host Logical Disk Status(PI_HLDI) record.

Table L–94: The data sources in each fields of the Host Logical Disk Status(PI_HLDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--

PFM - View name (PFM - Manager name)	Data Sources
Disk ID (DISK_ID)	ID with `tmpfs` in the `FileSystem` column and `/dev/shm` in the `Mounted on` column of `df-lkP`. Alternatively, an ID starting with `/dev` in the `FileSystem` column of `df-lkP`, except for all indicated by `mount -t iso9660`.
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	`1024-blocks` column in `df -lkP` / 1024
Used (USED)	`Used` column in `df -lkP` / 1024
Free (FREE)	`Available` column in `df -lkP` / 1024
Used % (USED_PERCENT)	(USED / SIZE) * 100
Last Update (LAST_UPDATE)	--
Free % (FREE_PERCENT)	(FREE / SIZE) * 100

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(3) Host Memory Status(PI_HMI)

The following table lists the data sources in each fields of the Host Memory Status(PI_HMI) record.

Table L–95: The data sources in each fields of the Host Memory Status(PI_HMI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Size (SIZE)	`host.MemTotal` in `/usr/bin/podman info --format=json` / 1024 / 1024
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	SIZE - UNUSED
VMM Used (VMM_USED)	--
VM Used (VM_USED)	The total of PI_VMI.USED
Unused (UNUSED)	`host.MemFree` in `/usr/bin/podman info --format=json` / 1024 / 1024
VM Swap Used (VM_SWAP_USED)	--
Host Swap Used (HOST_SWAP_USED)	--
Total Used (TOTAL_USED)	USED
Used % (USED_PERCENT)	(USED / SIZE) * 100

PFM - View name (PFM - Manager name)	Data Sources
VMM Used % (VMM_USED_PERCENT)	--
VM Used % (VM_USED_PERCENT)	(VM_USED / SIZE) * 100
VM Swap Used % (VM_SWAP_USED_PERCENT)	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--
Total Used % (TOTAL_USED_PERCENT)	(TOTAL_USED / SIZE) * 100
Swap IO (SWAP_IO)	SWAP_IN_IO + SWAP_OUT_IO
Swap In IO (SWAP_IN_IO)	`pages swapped in` line in `vmstat -s` * `getconf PAGE_SIZE` / 1024 / 1024
Swap Out IO (SWAP_OUT_IO)	`pages swapped out` line in `vmstat -s` * `getconf PAGE_SIZE` / 1024 / 1024

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(4) Host Network Status(PI_HNI)

The following table lists the data sources in each fields of the Host Network Status(PI_HNI) record.

Table L–96: The data sources in each fields of the Host Network Status(PI_HNI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Net ID (NET_ID)	Interfaces not starting with `veth` in the `Inter-face` column of the `/proc/net/dev` file
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE
Send Rate (SEND_RATE)	Δ Value of `Transmit Bytes` in the `/proc/net/dev` file / Δ Collection time / 1024
Recv Rate (RECV_RATE)	Δ Value of `Receive Bytes` in the `/proc/net/dev` file / Δ Collection time / 1024

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(5) Host Physical Disk Status(PI_HPDI)

The following table lists the data sources in each fields of the Host Physical Disk Status(PI_HPDI) record.

Table L–97: The data sources in each fields of the Host Physical Disk Status(PI_HPDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	Directory name in `ls /sys/block`
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	((Δ Value of the `read sectors` column in the `/sys/block/`*directory-name*`/stat` file) * 512 / 1024 / 1024) / Δ Collection time
Write Speed (WRITE_SPEED)	((Δ Value of the `write sectors` column in the`/sys/block/`*directory-name*`/stat` file) * 512 / 1024 / 1024) / Δ Collection time
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	Δ Value of the `read I/Os` column in the`/sys/block/`*directory-name*`/stat` file / Δ Collection time
Write Requests (WRITE_REQUESTS)	Δ Value of the `write I/Os` column in the`/sys/block/`*directory-name*`/stat` file / Δ Collection time
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--
Device Latency (DEVICE_LATENCY)	--
Device Read Latency (DEVICE_READ_LATENCY)	--
Device Write Latency (DEVICE_READ_LATENCY)	--
Kernel Latency (KERNEL_LATENCY)	--
Kernel Read Latency (KERNEL_READ_LATENCY)	--
Kernel Write Latency (KERNEL_WRITE_LATENCY)	--
Queue Latency (QUEUE_LATENCY)	--

PFM - View name (PFM - Manager name)	Data Sources
Queue Read Latency (QUEUE_READ_LATENCY)	--
Queue Write Latency (QUEUE_WRITE_LATENCY)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(6) Host Status Detail(PD)

The following table lists the data sources in each fields of the Host Status Detail(PD) record.

Table L–98: The data sources in each fields of the Host Status Detail(PD) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Status (STATUS)	--
Host Name (HOST_NAME)	Setting for monitored VM_Host parameter
Reason (REASON)	--
Product (PRODUCT)	`Version`, `GoVersion`, and `OsArch` values in `/usr/bin/podman version --format=json` in comma separated values
VM Count (VM_COUNT)	Number of lines in `{{.IsInfra}}` with the false value in `/usr/bin/podman container list --no-trunc --all --format '{{.ID}},{{.IsInfra}}'`
VM Active (VM_ACTIVE)	Number of lines in `{{.IsInfra}}` with the false value and `{{.Status}}` with `Up` *elapsed-time* or `Paused` in `/usr/bin/podman container list --no-trunc --all --format '{{.ID}},{{.IsInfra}},{{.Status}}'`

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(7) Host Status(PI)

The following table lists the data sources in each fields of the Host Status(PI) record.

Table L–99: The data sources in each fields of the Host Status(PI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Clocks (CLOCKS)	PI_HCI.CLOCKS * COUNT
Count (COUNT)	Counts the number of CPUs in the `/proc/cpuinfo` file
Sampling Time (SAMPLING_TIME)	date -Iseconds
Used (USED)	CLOCKS * USED_PERCENT / 100
VMM Used (VMM_USED)	--
VM Used (VM_USED)	The total of PI_VI.USED
VMM Console Used (VMM_CONSOLE_USED)	--
VMM Kernel Used (VMM_KERNEL_USED)	--
VMM Others Used (VMM_OTHERS_USED)	--
Unused (UNUSED)	CLOCKS - USED
Used % (USED_PERCENT)	Δ (`CPU` line and the `usr` + `nice` + `sys` column in the `/proc/stat` file) / (Δ Collection time * 100 * `COUNT`) * 100
VMM Used % (VMM_USED_PERCENT)	--
VM Used % (VM_USED_PERCENT)	VM_USED / CLOCKS * 100
VMM Console Used % (VMM_CONSOLE_USED_PERCENT)	--
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)	--
VMM Others Used % (VMM_OTHERS_USED_PERCENT)	--
Unused % (UNUSED_PERCENT)	UNUSED / CLOCKS * 100
Insufficient (INSUFFICIENT)	--
Insufficient % (INSUFFICIENT_PERCENT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(8) VM CPU Status(PI_VCI)

The following table lists the data sources in each fields of the VM CPU Status(PI_VCI) record.

Table L–100: The data sources in each fields of the VM CPU Status(PI_VCI) record (Podman environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
CPU ID (CPU_ID)	`cpu` column in the `/sys/fs/cgroup/cpu,cpuacct/`*path-to-the-container*`/cpuacct.usage_all` file
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	PI_HCI.CLOCKS * USED_PERCENT / 100
Insufficient (INSUFFICIENT)	--
Request (REQUEST)	--
Used % (USED_PERCENT)	Δ (`user` column + `system` column in `/sys/fs/cgroup/cpu,cpuacct/`*path-to-the-container*`/cpuacct.usage_all`) / Δ Collection time * 100
Insufficient % (INSUFFICIENT_PERCENT)	--
Request % (REQUEST_PERCENT)	--
Used Per Request (USED_PER_REQUEST)	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(9) VM Logical Disk Status(PI_VLDI)

The following table lists the data sources in each fields of the VM Logical Disk Status(PI_VLDI) record.

Table L–101: The data sources in each fields of the VM Logical Disk Status(PI_VLDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Free (FREE)	--
Used % (USED_PERCENT)	--
Free % (FREE_PERCENT)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(10) VM Memory Status(PI_VMI)

The following table lists the data sources in each fields of the VM Memory Status(PI_VMI) record.

Table L–102: The data sources in each fields of the VM Memory Status(PI_VMI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Size (SIZE)	PI_HMI.SIZE

PFM - View name (PFM - Manager name)	Data Sources
Used (USED)	Value in the `/sys/fs/cgroup/memory/`*path-name-to-the-container*`/memory.usage_in_bytes` file / 1024 / 1024
Resource Used (RESOURCE_USED)	--
VM Swap Used (VM_SWAP_USED)	--
Host Swap Used (HOST_SWAP_USED)	--
Unused (UNUSED)	SIZE - Used
Used % (USED_PERCENT)	(USED / SIZE) * 100
Resource Used % (RESOURCE_USED_PERCENT)	--
VM Swap Used % (VM_SWAP_USED_PERCENT)	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--
VM Swap IO (VM_SWAP_IO)	--
VM Swap In (VM_SWAP_IN)	--
VM Swap Out (VM_SWAP_OUT)	--
Working Size (WORKING_SIZE)	--
Working Size % (WORKING_SIZE_PERCENT)	--
Share (SHARE)	--
Max (MAX)	Value in the `/usr/bin/podman container inspect --format '{{.HostConfig.Memory}}'` / 1024 / 1024
Min (MIN)	Value in the `/usr/bin/podman container inspect --format '{{.HostConfig.MemoryReservation }}'` / 1024 / 1024
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	MAX / SIZE * 100
Min % (MIN_PERCENT)	MIN / SIZE * 100
Expectation % (EXPECTATION_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(11) VM Network Status(PI_VNI)

The following table lists the data sources in each fields of the VM Network Status(PI_VNI) record.

Table L–103: The data sources in each fields of the VM Network Status(PI_VNI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Net ID (NET_ID)	`inter-face` column in the `/proc/`*process-ID-of-the-container*`/net/dev` file
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PD_VCI.SAMPLING_TIME
Rate (RATE)	SEND_RATE + RECV_RATE
Send Rate (SEND_RATE)	Δ Value of `Transmit Bytes` in the `/proc/`*process-ID-of-the-container*`/net/dev` file / Δ Collection time / 1024
Recv Rate (RECV_RATE)	Δ Value of `Receive Bytes` in the `/proc/`*process-ID-of-the-container*`/net/dev` file / Δ Collection time / 1024

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(12) VM Physical Disk Status(PI_VPDI)

The following table lists the data sources in each fields of the VM Physical Disk Status(PI_VPDI) record.

Table L–104: The data sources in each fields of the VM Physical Disk Status(PI_VPDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Disk ID (DISK_ID)	`Device ID` column in the `/sys/fs/cgroup/blkio/`*path-name-to-the-container*`/blkio.throttle.io_service_bytes` file
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME

PFM - View name (PFM - Manager name)	Data Sources
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Speed (SPEED)	READ_SPEED + WRITE_SPEED
Read Speed (READ_SPEED)	(Δ Value of operation is "Read" in the `/sys/fs/cgroup/blkio/`*path-name-to-the-container*`/blkio.throttle.io_service_bytes` file / Δ Collection time) / 1024
Write Speed (WRITE_SPEED)	(Δ Value of operation is "Write" in the `/sys/fs/cgroup/blkio/`*path-name-to-the-container*`/blkio.throttle.io_service_bytes` file / Δ Collection time) / 1024
Requests (REQUESTS)	READ_REQUESTS + WRITE_REQUESTS
Read Requests (READ_REQUESTS)	(Δ Value of operation is "Read" in the `/sys/fs/cgroup/blkio/`*path-name-to-the-container*`/blkio.throttle.io_serviced` file / Δ Collection time)
Write Requests (WRITE_REQUESTS)	(Δ Value of operation is "Write" in the `/sys/fs/cgroup/blkio/`*path-name-to-the-container*`/blkio.throttle.io_serviced` file / Δ Collection time)
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(13) VM Virtual Disk Status(PI_VVDI)

The following table lists the data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record.

Table L–105: The data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--

PFM - View name (PFM - Manager name)	Data Sources
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests Per Sec (REQUESTS_PER_SEC)	--
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	--
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	--
Total Latency (TOTAL_LATENCY)	--
Total Read Latency (TOTAL_READ_LATENCY)	--
Total Write Latency (TOTAL_WRITE_LATENCY)	--
Outstanding Requests (OUTSTANDING_REQUESTS)	--
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	--
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(14) VM Status Detail(PD_VM)

The following table lists the data sources in each fields of the VM Status Detail(PD_VM) record.

Table L–106: The data sources in each fields of the VM Status Detail(PD_VM) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	Value of `{{.ID}}` when `{{.IsInfra}}` has the false value in `/usr/bin/podman container list --no-trunc --all --format '{{.ID}},{{.IsInfra}}'`

PFM - View name (PFM - Manager name)	Data Sources
VM Host Name (VM_HOST_NAME)	Value of `{{.Config.Hostname}}` in `/usr/bin/podman container inspect --format '{{.Config.Hostname}}'`
VM Name (VM_NAME)	Value of `{{.Name}}` in `/usr/bin/podman container inspect --format '{{.Name}}'`

Status (STATUS)	The following values in `/usr/bin/podman container inspect --format '{{.State.Running}},{{.State.Paused}},{{.State.Restarting}},{{.State.OOMKilled}},{{.State.Dead}},{{.State.Pid}}'`:
	`{{.State.Dead}}` is true → `DEAD`
	`{{.State.OOMKilled}}` is true → `OOMKilled`
	`{{.State.Restarting}}` is true → `STARTING`
	`{{.State.Paused}}` is true → `PAUSED`
	`{{.State.Running}}` is true and `{{State.Pid}}` is not ""0"" → `ON`
	All are false → `OFF`
	None of the above → `UNKNOWN`

Information (INFORMATION)	Value of `{{.Path}}` in `/usr/bin/podman container inspect --format '{{.Path}}'`

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(15) VM Status(PI_VI)

The following table lists the data sources in each fields of the VM Status(PI_VI) record.

Table L–107: The data sources in each fields of the VM Status(PI_VI) record (Docker environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	PD_VM.VM_ID
Clocks (CLOCKS)	--
Count (COUNT)	--
VM Host Name (VM_HOST_NAME)	PD_VM.VM_HOST_NAME
VM Name (VM_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
Used (USED)	PI.CLOCKS * USED_PERCENT / 100
Insufficient (INSUFFICIENT)	--
Request (REQUEST)	--
Host Used % (HOST_USED_PERCENT)	USED / PI.CLOCKS * 100

PFM - View name (PFM - Manager name)	Data Sources
Used % (USED_PERCENT)	Values of the `user` column and `system` column in the `/sys/fs/cgroup/cpu,cpuacct/`*path-to-the-container*`/cpuacct.usage_all` file Σ ((Δ `user` column + Δ `system` column) / Δ Collection time)
Insufficient % (INSUFFICIENT_PERCENT)	--
Request % (REQUEST_PERCENT)	--
Used Per Request (USED_PER_REQUEST)	--
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	--
Affinity (AFFINITY)	--
Share (SHARE)	--
Max (MAX)	--
Min (MIN)	--
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	--
Min % (MIN_PERCENT)	--
Expectation % (EXPECTATION_PERCENT)	--
Snapshot (SNAPSHOT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

Δ : This time collected value - Last time collected value.

--: The field stores the raw (not altered) value of performance data acquired.

(16) Host Generic Data Detail(PD_HGDD)

The following table lists the data sources in each fields of the Host Generic Data Detail(PD_HGDD) record.

Table L–108: The data sources in each fields of the Host Generic Data Detail(PD_HGDD) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--

PFM - View name (PFM - Manager name)	Data Sources
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(17) Host Generic Data Interval(PI_HGDI)

The following table lists the data sources in each fields of the Host Generic Data Interval(PI_HGDI) record.

Table L–109: The data sources in each fields of the Host Generic Data Interval(PI_HGDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
Sampling Time (SAMPLING_TIME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(18) VM Generic Data Detail(PD_VGDD)

The following table lists the data sources in each fields of the VM Generic Data Detail(PD_VGDD) record.

Table L–110: The data sources in each fields of the VM Generic Data Detail(PD_VGDD) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--

PFM - View name (PFM - Manager name)	Data Sources
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(19) VM Generic Data Interval(PI_VGDI)

The following table lists the data sources in each fields of the VM Generic Data Interval(PI_VGDI) record.

Table L–111: The data sources in each fields of the VM Generic Data Interval(PI_VGDI) record (Podman environment)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(20) VM Virtual Disk Detail(PD_VDKD)

The following table lists the data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record.

Table L–112: The data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record (Podman environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Datastore ID (DATASTORE_ID)	--
Datastore Name (DATASTORE_NAME)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Controller Name (CONTROLLER_NAME)	--
Bus Number (BUS_NUMBER)	--
Unit Number (UNIT_NUMBER)	--
Disk UUID (DISK_UUID)	--
Capacity (CAPACITY)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(21) POD Status Detail(PD_PODD)

The following table lists the data sources in each fields of the POD Status Detail(PD_PODD) record.

Table L–113: The data sources in each fields of the POD Status Detail(PD_PODD) record (Podman environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	Value of `{{.ID}}` in `/usr/bin/podman pod list --no-trunc --format '{{.ID}},{{.Status}}'`

PFM-View name (PFM-Manager name)	Data Sources
POD Name (POD_NAME)	Value of `Config/name` in `/usr/bin/podman pod inspect`
Status (STATUS)	Value of `{{.Status}}` in `/usr/bin/podman pod list --no-trunc --format '{{.ID}},{{.Status}}'`
Container Count (CONTAINER_COUNT)	Number of `Containers` in `/usr/bin/podman pod inspect`

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(22) POD Status Interval(PI_PODI)

The following table lists the data sources in each fields of the POD Status Interval(PI_PODI) record.

Table L–114: The data sources in each fields of the POD Status Interval(PI_PODI) record (Podman environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	PD_PODD.POD_ID
POD Name (POD_NAME)	PD_PODD.POD_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
CPU Used (CPU_USED)	Σ `PI_POCI.CPU_USED` of the containers with the same `POD_ID` value
CPU Used % (CPU_USED_PERCENT)	(CPU_USED / PI.CLOCKS) * 100
Memory Used (MEMORY_USED)	Σ `PI_POCI.MEMORY_USED` of the containers with the same `POD_ID` value
Memory Used % (MEMORY_USED_PERCENT)	(MEMORY_USED / PI_HMI.SIZE) * 100
Disk Speed (DISK_SPEED)	DISK_READ_SPEED + DISK_WRITE_SPEED
Disk Read Speed (DISK_READ_SPEED)	Σ `PI_POCI.DISK_READ_SPEED` of the containers with the same `POD_ID` value
Disk Write Speed (DISK_WRITE_SPEED)	Σ `PI_POCI.DISK_WRITE_SPEED` of the containers with the same `POD_ID` value
Disk Requests (DISK_REQUESTS)	DISK_READ_REQUESTS + DISK_WRITE_REQUESTS
Disk Read Requests (DISK_READ_REQUESTS)	Σ `PI_POCI.DISK_READ_REQIESTS` of the containers with the same `POD_ID` value
Disk Write Requests (DISK_WRITE_REQUESTS)	Σ `PI_POCI.DISK_WRITE_REQUESTS` of the containers with the same `POD_ID` value

PFM-View name (PFM-Manager name)	Data Sources
Network Rate (NETWORK_RATE)	NETWORK_SEND_RATE + NETWORK_RECV_RATE
Network Send Rate (NETWORK_SEND_RATE)	Σ `PI_POCI.NETWORK_SEND_RATE` of the containers with the same `POD_ID` value
Network Recv Rate (NETWORK_RECV_RATE)	Σ `PI_POCI.NETWORK_RECV_RATE` of the containers with the same `POD_ID` value

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(23) POD Container Status Interval(PI_POCI)

The following table lists the data sources in each fields of the POD Container Status Interval(PI_POCI) record.

Table L–115: The data sources in each fields of the POD Container Status Interval(PI_POCI) record (Podman environment)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	PD_PODD.POD_ID
POD Name (POD_NAME)	PD_PODD.POD_NAME
Container ID (CONTAINER_ID)	PD_VM.VM_ID
Container Name (CONTAINER_NAME)	PD_VM.VM_NAME
Sampling Time (SAMPLING_TIME)	PI.SAMPLING_TIME
CPU Used (CPU_USED)	PI_VI.USED
CPU Used % (CPU_USED_PERCENT)	PI_VI.USED_PERCENT
Memory Used (MEMORY_USED)	PI_VMI.USED
Memory Used % (MEMORY_USED_PERCENT)	PI_VMI.USED_PERCENT
Disk Speed (DISK_SPEED)	PI_VPDI.SPEED
Disk Read Speed (DISK_READ_SPEED)	PI_VPDI.READ_SPEED
Disk Write Speed (DISK_WRITE_SPEED)	PI_VPDI.WRITE_SPEED
Disk Requests (DISK_REQUESTS)	PI_VPDI.REQUESTS
Disk Read Requests (DISK_READ_REQUESTS)	PI_VPDI.READ_REQUESTS

PFM-View name (PFM-Manager name)	Data Sources
Disk Write Requests (DISK_WRITE_REQUESTS)	PI_VPDI.WRITE_REQUESTS
Network Rate (NETWORK_RATE)	PI_VNI.RATE
Network Send Rate (NETWORK_SEND_RATE)	PI_VNI.SEND_RATE
Network Recv Rate (NETWORK_RECV_RATE)	PI_VNI.RECV_RATE

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

L.6 In logical partitioning feature

This section describes the field values of the data sources in logical partitioning feature

(1) Host CPU Status(PI_HCI)

The following table lists the data sources in each fields of the Host CPU Status(PI_HCI) record.

Table L–116: The data sources in each fields of the Host CPU Status(PI_HCI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
CPU ID (CPU_ID)	CORE# field of the PHYSICAL_CPU_USAGE record
CPU Name (CPU_NAME)	NAME field of the PHYSICAL_CPU_USAGE record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Clocks (CLOCKS)	CAPACITY field of the PHYSICAL_CPU_USAGE record
Used (USED)	USED field of the PHYSICAL_CPU_USAGE record
Unused (UNUSED)	UNUSED field of the PHYSICAL_CPU_USAGE record
Used % (USED_PERCENT)	USED% field of the PHYSICAL_CPU_USAGE record
Unused % (UNUSED_PERCENT)	UNUSED% field of the PHYSICAL_CPU_USAGE record

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(2) Host Logical Disk Status(PI_HLDI)

The following table lists the data sources in each fields of the Host Logical Disk Status(PI_HLDI) record.

Table L–117: The data sources in each fields of the Host Logical Disk Status(PI_HLDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Free (FREE)	--
Used % (USED_PERCENT)	--
Last Update (LAST_UPDATE)	--
Free % (FREE_PERCENT)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(3) Host Memory Status(PI_HMI)

The following table lists the data sources in each fields of the Host Memory Status(PI_HMI) record.

Table L–118: The data sources in each fields of the Host Memory Status(PI_HMI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Size (SIZE)	MEM field of the SYSTEM_CONFIGURATION record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Used (USED)	USED field when the NAME field SYSTEM_USAGE_SUMMARY record contains MEM
VMM Used (VMM_USED)	USED field when the NAME field SYSTEM_MEM_USAGE record contains SYS
VM Used (VM_USED)	USED field when the NAME field SYSTEM_MEM_USAGE record contains LPAR
Unused (UNUSED)	UNUSED field when the NAME field SYSTEM_USAGE_SUMMARY record contains MEM
VM Swap Used (VM_SWAP_USED)	--

PFM - View name (PFM - Manager name)	Data Sources
Host Swap Used (HOST_SWAP_USED)	--
Total Used (TOTAL_USED)	USED field when the NAME field SYSTEM_USAGE_SUMMARY record contains MEM
Used % (USED_PERCENT)	USED% field when the NAME field SYSTEM_USAGE_SUMMARY record contains MEM
VMM Used % (VMM_USED_PERCENT)	USED% field when the NAME field SYSTEM_MEM_USAGE record contains SYS
VM Used % (VM_USED_PERCENT)	USED% field when the NAME field SYSTEM_MEM_USAGE record contains LPAR
VM Swap Used % (VM_SWAP_USED_PERCENT)	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--
Total Used % (TOTAL_USED_PERCENT)	USED% field when the NAME field SYSTEM_USAGE_SUMMARY record contains MEM
Swap IO (SWAP_IO)	--
Swap In IO (SWAP_IN_IO)	--
Swap Out IO (SWAP_OUT_IO)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(4) Host Network Status(PI_HNI)

The following table lists the data sources in each fields of the Host Network Status(PI_HNI) record.

Table L–119: The data sources in each fields of the Host Network Status(PI_HNI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Net ID (NET_ID)	SID field + P# field of the PHYSICAL_NIC_USAGE record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Rate (RATE)	T_BYTE field of the PHYSICAL_NIC_USAGE record
Send Rate (SEND_RATE)	S_BYTE field of the PHYSICAL_NIC_USAGE record
Recv Rate (RECV_RATE)	R_BYTE field of the PHYSICAL_NIC_USAGE record

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(5) Host Physical Disk Status(PI_HPDI)

The following table lists the data sources in each fields of the Host Physical Disk Status(PI_HPDI) record.

Table L–120: The data sources in each fields of the Host Physical Disk Status(PI_HPDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Disk ID (DISK_ID)	SID field + P# field of the PHYSICAL_HBA_USAGE record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests (REQUESTS)	INT field of the PHYSICAL_HBA_USAGE record
Read Requests (READ_REQUESTS)	--
Write Requests (WRITE_REQUESTS)	--
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--
Device Latency (DEVICE_LATENCY)	--
Device Read Latency (DEVICE_READ_LATENCY)	--
Device Write Latency (DEVICE_READ_LATENCY)	--
Kernel Latency (KERNEL_LATENCY)	--
Kernel Read Latency (KERNEL_READ_LATENCY)	--
Kernel Write Latency (KERNEL_WRITE_LATENCY)	--
Queue Latency (QUEUE_LATENCY)	--
Queue Read Latency (QUEUE_READ_LATENCY)	--
Queue Write Latency (QUEUE_WRITE_LATENCY)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(6) Host Status Detail(PD)

The following table lists the data sources in each fields of the Host Status Detail(PD) record.

Table L–121: The data sources in each fields of the Host Status Detail(PD) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Status (STATUS)	--
Host Name (HOST_NAME)	Input parameter to the plug-in (VM_Host)
Reason (REASON)	--
Product (PRODUCT)	PRODUCT field of the MONITORING_INFORMATION record
VM Count (VM_COUNT)	DEF_LPARs field of the SYSTEM_CONFIGURATION record
VM Active (VM_ACTIVE)	ACT_LPARs field of the SYSTEM_CONFIGURATION record

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(7) Host Status(PI)

The following table lists the data sources in each fields of the Host Status(PI) record.

Table L–122: The data sources in each fields of the Host Status(PI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Clocks (CLOCKS)	CPU_CAP field of the SYSTEM_CONFIGURATION record
Count (COUNT)	COREs field of the SYSTEM_CONFIGURATION record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Used (USED)	USED field when the NAME field of the SYSTEM_USAGE_SUMMARY record contains CPU
VMM Used (VMM_USED)	Total of the USED fields when the NAME field of the SYSTEM_CPU_USAGE record contains SYS1 and when the NAME field contains SYS2
VM Used (VM_USED)	Total of the USED fields when the NAME field of the SYSTEM_CPU_USAGE record contains SHR_LPAR and when the NAME field contains DED_LPAR
VMM Console Used (VMM_CONSOLE_USED)	--

PFM - View name (PFM - Manager name)	Data Sources
VMM Kernel Used (VMM_KERNEL_USED)	USED field when the NAME field of the SYSTEM_CPU_USAGE record contains SYS1
VMM Others Used (VMM_OTHERS_USED)	USED field when the NAME field of the SYSTEM_CPU_USAGE record contains SYS2
Unused (UNUSED)	UNUSED field when the NAME field of the SYSTEM_USAGE_SUMMARY record contains CPU
Used % (USED_PERCENT)	USED% field when the NAME field of the SYSTEM_USAGE_SUMMARY record contains CPU
VMM Used % (VMM_USED_PERCENT)	Total of the USED% fields when the NAME field of the SYSTEM_CPU_USAGE record contains SYS1 and when the NAME field contains SYS2
VM Used % (VM_USED_PERCENT)	Total of the USED% fields when the NAME field of the SYSTEM_CPU_USAGE record contains SHR_LPAR and when the NAME field contains DED_LPAR
VMM Console Used % (VMM_CONSOLE_USED_PERCENT)	--
VMM Kernel Used % (VMM_KERNEL_USED_PERCENT)	USED% field when the NAME field of the SYSTEM_CPU_USAGE record contains SYS1
VMM Others Used % (VMM_OTHERS_USED_PERCENT)	USED% field when the NAME field of the SYSTEM_CPU_USAGE record contains SYS2
Unused % (UNUSED_PERCENT)	UNUSED% field when the NAME field of the SYSTEM_USAGE_SUMMARY record contains CPU
Insufficient (INSUFFICIENT)	--
Insufficient % (INSUFFICIENT_PERCENT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(8) VM CPU Status(PI_VCI)

The following table lists the data sources in each fields of the VM CPU Status(PI_VCI) record.

Table L–123: The data sources in each fields of the VM CPU Status(PI_VCI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	L# field of the LOGICAL_CPU_USAGE record
CPU ID (CPU_ID)	CPU# field of the LOGICAL_CPU_USAGE record

PFM - View name (PFM - Manager name)	Data Sources
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	NAME field of the LOGICAL_CPU_USAGE record
Sampling Time (SAMPLING_TIME)	CRR_DATE_TIME field of the MONITORING_INFORMATION record
Used (USED)	USED field of the LOGICAL_CPU_USAGE record
Insufficient (INSUFFICIENT)	Total of the ROB field and DELAY fields of the LOGICAL_CPU_USAGE record
Request (REQUEST)	Total of the USED, ROB, and DELAY fields of the LOGICAL_CPU_USAGE record
Used % (USED_PERCENT)	USED% field of the LOGICAL_CPU_USAGE record
Insufficient % (INSUFFICIENT_PERCENT)	ROB% field and DELAY% field of the LOGICAL_CPU_USAGE record
Request % (REQUEST_PERCENT)	Total of the USED%, ROB%, and DELAY% fields of the LOGICAL_CPU_USAGE record
Used Per Request (USED_PER_REQUEST)	USED / REQUEST
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	INSUFFICIENT / REQUEST
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(9) VM Logical Disk Status(PI_VLDI)

The following table lists the data sources in each fields of the VM Logical Disk Status(PI_VLDI) record.

Table L–124: The data sources in each fields of the VM Logical Disk Status(PI_VLDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Free (FREE)	--

PFM - View name (PFM - Manager name)	Data Sources
Used % (USED_PERCENT)	--
Free % (FREE_PERCENT)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(10) VM Memory Status(PI_VMI)

The following table lists the data sources in each fields of the VM Memory Status(PI_VMI) record.

Table L–125: The data sources in each fields of the VM Memory Status(PI_VMI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Size (SIZE)	--
Used (USED)	--
Resource Used (RESOURCE_USED)	--
VM Swap Used (VM_SWAP_USED)	--
Host Swap Used (HOST_SWAP_USED)	--
Unused (UNUSED)	--
Used % (USED_PERCENT)	--
Resource Used % (RESOURCE_USED_PERCENT)	--
VM Swap Used % (VM_SWAP_USED_PERCENT)	--
Host Swap Used % (HOST_SWAP_USED_PERCENT)	--
VM Swap IO (VM_SWAP_IO)	--
VM Swap In (VM_SWAP_IN)	--
VM Swap Out (VM_SWAP_OUT)	--
Working Size (WORKING_SIZE)	--
Working Size % (WORKING_SIZE_PERCENT)	--
Share (SHARE)	--

PFM - View name (PFM - Manager name)	Data Sources
Max (MAX)	--
Min (MIN)	--
Expectation (EXPECTATION)	--
Max % (MAX_PERCENT)	--
Min % (MIN_PERCENT)	--
Expectation % (EXPECTATION_PERCENT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(11) VM Network Status(PI_VNI)

The following table lists the data sources in each fields of the VM Network Status(PI_VNI) record.

Table L–126: The data sources in each fields of the VM Network Status(PI_VNI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	L# field of the LOGICAL_NIC_USAGE record
Net ID (NET_ID)	SID field + P# field of the LOGICAL_NIC_USAGE record
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	NAME field of the LOGICAL_NIC_USAGE record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Rate (RATE)	T_BYTE field of the LOGICAL_NIC_USAGE record
Send Rate (SEND_RATE)	S_BYTE field of the LOGICAL_NIC_USAGE record
Recv Rate (RECV_RATE)	R_BYTE field of the LOGICAL_NIC_USAGE record

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(12) VM Physical Disk Status(PI_VPDI)

The following table lists the data sources in each fields of the VM Physical Disk Status(PI_VPDI) record.

Table L–127: The data sources in each fields of the VM Physical Disk Status(PI_VPDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	L# field of the LOGICAL_HBA_USAGE record
Disk ID (DISK_ID)	SID field + P# field of the LOGICAL_HBA_USAGE record
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	NAME field of the LOGICAL_HBA_USAGE record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests (REQUESTS)	INT field of the LOGICAL_HBA_USAGE record
Read Requests (READ_REQUESTS)	--
Write Requests (WRITE_REQUESTS)	--
Commands (COMMANDS)	--
Abort Commands (ABORT_COMMANDS)	--
Abort Commands % (ABORT_COMMANDS_PERCENT)	--
Bus Resets (BUS_RESETS)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(13) VM Virtual Disk Status(PI_VVDI)

The following table lists the data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record.

Table L–128: The data sources in each fields of the VM Virtual Disk Status(PI_VVDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--

PFM - View name (PFM - Manager name)	Data Sources
Disk ID (DISK_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Speed (SPEED)	--
Read Speed (READ_SPEED)	--
Write Speed (WRITE_SPEED)	--
Requests Per Sec (REQUESTS_PER_SEC)	--
Read Requests Per Sec (READ_REQUESTS_PER_SEC)	--
Write Requests Per Sec (WRITE_REQUESTS_PER_SEC)	--
Total Latency (TOTAL_LATENCY)	--
Total Read Latency (TOTAL_READ_LATENCY)	--
Total Write Latency (TOTAL_WRITE_LATENCY)	--
Outstanding Requests (OUTSTANDING_REQUESTS)	--
Outstanding Read Requests (OUTSTANDING_READ_REQUESTS)	--
Outstanding Write Requests (OUTSTANDING_WRITE_REQUESTS)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(14) VM Status Detail(PD_VM)

The following table lists the data sources in each fields of the VM Status Detail(PD_VM) record.

Table L–129: The data sources in each fields of the VM Status Detail(PD_VM) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	L# field of the LPAR_CONFIGURATION record
VM Host Name (VM_HOST_NAME)	--

PFM - View name (PFM - Manager name)	Data Sources
VM Name (VM_NAME)	NAME field of the LPAR_CONFIGURATION record
Status (STATUS)	STATE field of the LPAR_CONFIGURATION record
Information (INFORMATION)	INFORMATION field of the LPAR_CONFIGURATION record
Snapshot (SNAPSHOT)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(15) VM Status(PI_VI)

The following table lists the data sources in each fields of the VM Status(PI_VI) record.

Table L–130: The data sources in each fields of the VM Status(PI_VI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	L# field of the LPAR_CPU_USAGE record
Clocks (CLOCKS)	CPU_CAP field of the LPAR_CONFIGURATION record
Count (COUNT)	COREs field of the LPAR_CONFIGURATION record
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	NAME field of the LPAR_CPU_USAGE record
Sampling Time (SAMPLING_TIME)	CURR_DATE_TIME field of the MONITORING_INFORMATION record
Used (USED)	USED field of the LPAR_CPU_USAGE record
Insufficient (INSUFFICIENT)	Total of the ROB and DELAY fields of the LPAR_CPU_USAGE record
Request (REQUEST)	Total of the USED, ROB, and DELAY fields of the LPAR_CPU_USAGE record
Host Used % (HOST_USED_PERCENT)	HST_USED% field of the LPAR_CPU_USAGE record
Used % (USED_PERCENT)	USED% field of the LPAR_CPU_USAGE record
Insufficient % (INSUFFICIENT_PERCENT)	Total of the ROB% and DEALY% fields of the LPAR_CPU_USAGE record
Request % (REQUEST_PERCENT)	Total of the USED%, ROB%, and DELAY% fields of the LPAR_CPU_USAGE record
Used Per Request (USED_PER_REQUEST)	Used / Request
Insufficient Per Request (INSUFFICIENT_PER_REQUEST)	Insufficient / Request
Affinity (AFFINITY)	--
Share (SHARE)	CPU_WIGHT field of the LPAR_CONFIGURATION record

PFM - View name (PFM - Manager name)	Data Sources
Max (MAX)	CPU_MAX field of the LPAR_CONFIGURATION record
Min (MIN)	--
Expectation (EXPECTATION)	CPU_SRV field of the LPAR_CONFIGURATION record
Max % (MAX_PERCENT)	CPU_MAX% field of the LPAR_CONFIGURATION record
Min % (MIN_PERCENT)	--
Expectation % (EXPECTATION_PERCENT)	CPU_SRV% field of the LPAR_CONFIGURATION record
Snapshot (SNAPSHOT)	--
Co-Stop (CO_STOP)	--
Co-Stop % (CO_STOP_PERCENT)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(16) Host Generic Data Detail(PD_HGDD)

The following table lists the data sources in each fields of the Host Generic Data Detail(PD_HGDD) record.

Table L–131: The data sources in each fields of the Host Generic Data Detail(PD_HGDD) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(17) Host Generic Data Interval(PI_HGDI)

The following table lists the data sources in each fields of the Host Generic Data Interval(PI_HGDI) record.

Table L–132: The data sources in each fields of the Host Generic Data Interval(PI_HGDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
Sampling Time (SAMPLING_TIME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(18) VM Generic Data Detail(PD_VGDD)

The following table lists the data sources in each fields of the VM Generic Data Detail(PD_VGDD) record.

Table L–133: The data sources in each fields of the VM Generic Data Detail(PD_VGDD) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:
--: The field stores the raw (not altered) value of performance data acquired.

(19) VM Generic Data Interval(PI_VGDI)

The following table lists the data sources in each fields of the VM Generic Data Interval(PI_VGDI) record.

Table L–134: The data sources in each fields of the VM Generic Data Interval(PI_VGDI) record (logical partitioning feature)

PFM - View name (PFM - Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
VM ID (VM_ID)	--
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Sampling Time (SAMPLING_TIME)	--
Section Name (SECTION_NAME)	--
Data Name (DATA_NAME)	--
Object Name (OBJECT_NAME)	--
String Data (STRING_DATA)	--
Double Data (DOUBLE_DATA)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(20) VM Virtual Disk Detail(PD_VDKD)

The following table lists the data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record.

Table L–135: The data sources in each fields of the VM Virtual Disk Detail(PD_VDKD) record (logical partitioning feature)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
Datastore ID (DATASTORE_ID)	--
Datastore Name (DATASTORE_NAME)	--
VM ID (VM_ID)	--

PFM-View name (PFM-Manager name)	Data Sources
VM Host Name (VM_HOST_NAME)	--
VM Name (VM_NAME)	--
Controller Name (CONTROLLER_NAME)	--
Bus Number (BUS_NUMBER)	--
Unit Number (UNIT_NUMBER)	--
Disk UUID (DISK_UUID)	--
Capacity (CAPACITY)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(21) POD Status Detail(PD_PODD)

The following table lists the data sources in each fields of the POD Status Detail(PD_PODD) record.

Table L–136: The data sources in each fields of the POD Status Detail(PD_PODD) record (logical partitioning feature)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Status (STATUS)	--
Container Count (CONTAINER_COUNT)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(22) POD Status Interval(PI_PODI)

The following table lists the data sources in each fields of the POD Status Interval(PI_PODI) record.

Table L–137: The data sources in each fields of the POD Status Interval(PI_PODI) record (logical partitioning feature)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--

PFM-View name (PFM-Manager name)	Data Sources
Record Time (RECORD_TIME)	--
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

(23) POD Container Status Interval(PI_POCI)

The following table lists the data sources in each fields of the POD Container Status Interval(PI_POCI) record.

Table L–138: The data sources in each fields of the POD Container Status Interval(PI_POCI) record (logical partitioning feature)

PFM-View name (PFM-Manager name)	Data Sources
Record Type (INPUT_RECORD_TYPE)	--
Record Time (RECORD_TIME)	--

PFM-View name (PFM-Manager name)	Data Sources
Interval (INTERVAL)	--
VA DeviceID (VADEVICEID)	--
POD ID (POD_ID)	--
POD Name (POD_NAME)	--
Container ID (CONTAINER_ID)	--
Container Name (CONTAINER_NAME)	--
Sampling Time (SAMPLING_TIME)	--
CPU Used (CPU_USED)	--
CPU Used % (CPU_USED_PERCENT)	--
Memory Used (MEMORY_USED)	--
Memory Used % (MEMORY_USED_PERCENT)	--
Disk Speed (DISK_SPEED)	--
Disk Read Speed (DISK_READ_SPEED)	--
Disk Write Speed (DISK_WRITE_SPEED)	--
Disk Requests (DISK_REQUESTS)	--
Disk Read Requests (DISK_READ_REQUESTS)	--
Disk Write Requests (DISK_WRITE_REQUESTS)	--
Network Rate (NETWORK_RATE)	--
Network Send Rate (NETWORK_SEND_RATE)	--
Network Recv Rate (NETWORK_RECV_RATE)	--

Legend:

--: The field stores the raw (not altered) value of performance data acquired.

M. Fields Affected by Setting the PFM - RM for Virtual Machine

The following describes the fields affected by setting described at *2.1.4(2) Setting PFM - RM for Virtual Machine*. An example is shown below and the table shows the different values for each setting.

The example monitoring target is as follows:

VMware: Hyper-Threading is enabled

CPU: Physical processor for 1 core: 2GHz.

Number of physical processor cores: 2

Number of logical processor cores: 4

CPU resources used: 3GHz

Table M–1: Fields affected by PFM - RM for Virtual Machine setup

No.	Record	Field	Affected	Setting UseHTPhysicalClocks	
				If N is specified	If Y is specified
1	Host Status (PI)	Clocks	Y	2GHz * 4 = 8GHz	2GHz * 2 = 4GHz
2		Count	N	2	2
3		Used	N	3GHz	3GHz
4		VMM Used	N	0.5GHz	0.5GHz
5		VM Used	N	3GHz	3GHz
6		VMM Console Used	N	0GHz	0GHz
7		VMM Kernel Used	N	0.3GHz	0.3GHz
8		VMM Others Used	N	VMM Used(0.5GHz) - VMM Console Used(0GHz) - VMM Kernel Used(0.3GHz) = 0.2GHz	VMM Used(0.5GHz) - VMM Console Used(0GHz) - VMM Kernel Used(0.3GHz) = 0.2GHz
9		UnUsed	Y	Clocks(8Ghz) - Used(3GHz) = 5GHz	Clocks(4Ghz) - Used(3GHz) = 1GHz
10		Used %	Y	Used(3GHz) / Clocks(8GHz) = 37.5%	Used(3GHz) / Clocks(4GHz) = 75%
11		VMM Used %	Y	VMM Used(0.5GHz) / Clocks(8GHz) = 6.25%	VMM Used(0.5GHz) / Clocks(4GHz) = 12.5%
12		VM Used %	Y	VM Used(3GHz) / Clocks(8GHz) = 37.5%	VM Used(3GHz) / Clocks(4GHz) = 75%
13		VMM Console Used %	Y	VMM Console Used(0GHz) / Clocks(8GHz) = 0%	VMM Console Used(0GHz) / Clocks(4GHz) = 0%
14		VMM Kernel Used %	Y	VMM Kernel Used(0.3GHz) / Clocks(8GHz) = 3.75%	VMM Kernel Used(0.3GHz) / Clocks(4GHz) = 7.5%
15		VMM Others Used	Y	VMM Others Used(0.2GHz) / Clocks(8GHz) = 2.5%	VMM Others Used(0.2GHz) / Clocks(4GHz) = 5%
16		Unused %	Y	Unused(5GHz) / Clocks(8GHz) = 62.5%	Unused(1GHz) / Clocks(4GHz) = 25%

No.	Record	Field	Affected	Setting UseHTPhysicalClocks	
				If N is specified	If Y is specified
17	VM Status (PI_VI)	Host Used %	Y	Used(3GHz) / PI.Clocks(8GHz) = 37.5%	Used(3GHz) / PI.Clocks(4GHz) = 75%
18		Expectation	Y	PI.Clocks(8GHz) * (Expectation %(30%) / 100) = 2.4GHz	PI.Clocks(4GHz) * (Expectation %(30%) / 100) = 1.2GHz

Legend:

Y: The field is affected by the setting.

N: The field is not affected by the setting.

N. What to Do When the Message KAVL20516-W Is Output Repeatedly

When you restart the system or when a system in a cluster configuration is switched, the message KAVL20516-W can be output repeatedly depending on the time when the PFM - RM for Virtual Machine service starts. This can occur even though a problem has not occurred (for example, the system is temporarily under high load or takes a long time to communicate with the monitoring target as shown in the *Figure 1-10* in *1.3.1 (2) General procedure for collecting performance data*).

If you restart the PFM - RM for Virtual Machine service, the message KAVL20516-W is no longer output.

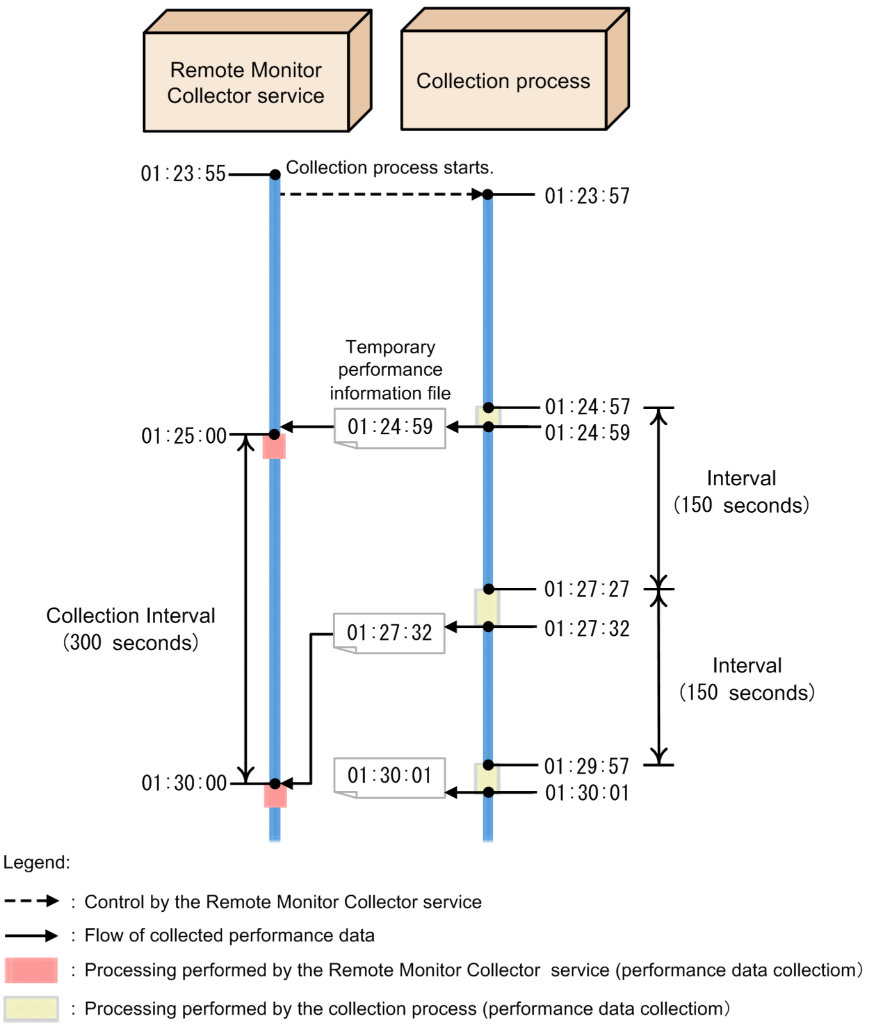
In addition, you can configure PFM - RM for Virtual Machine so that the KAVL20516-W message is not output when the system is restarted or switched in a cluster configuration. To do this, use either of the following settings:

(a) Changing the collection interval settings (`Collection Interval` and `Interval`) with which performance data is collected (if `UseOffset` is set to disable (`N`))

When you set up the instance environment, set the `Interval` value to half of the `Collection Interval` value, which enables the collection process of the Remote Monitor Collector service to read a new temporary performance information file.

The following is an example when you set the `Collection Interval` value of the record to 300 seconds, and the `Interval` value in the instance environment to 150 seconds.

Figure N–1: Example procedure for when changing the collection interval value of performance data

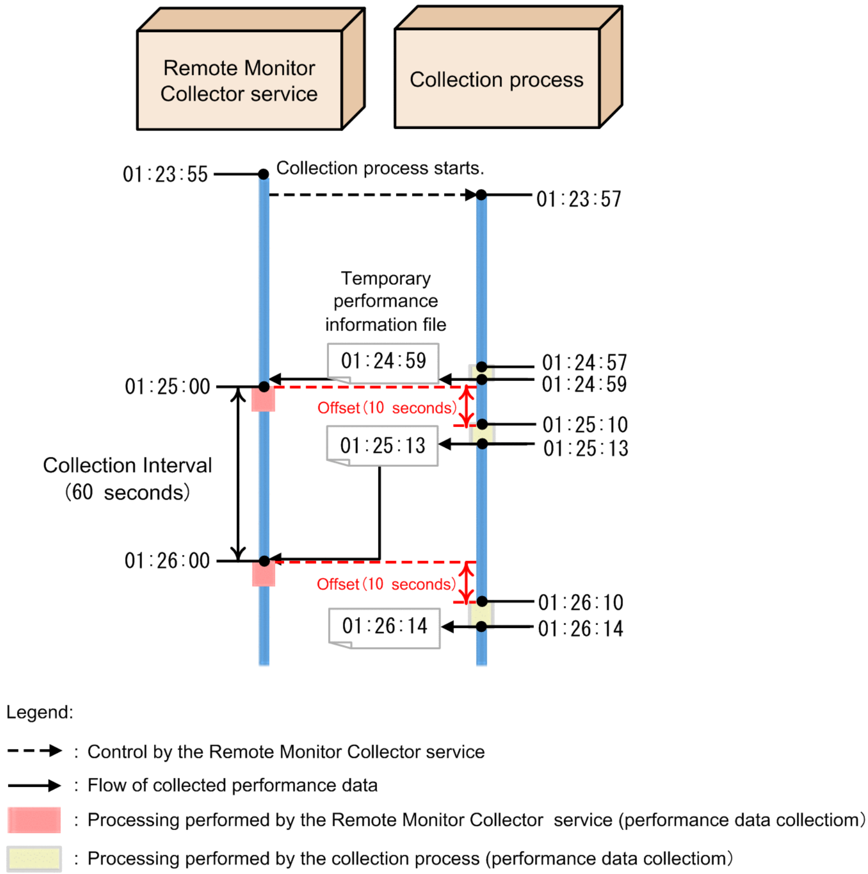


(b) Using a zero reference point based on which the collection process starts (if `UseOffset` is set to enable (`Y`))

If you set a short time (such as the minimum value of 60 seconds) to the `Collection Interval` value for each record and the `Interval` value of the instance environment, you cannot set half of the `Collection Interval` value to the `Interval` value. In that case, set `UseOffset` to enable (`Y`) in the PFM - RM for Virtual Machine setup file so that the collection process starts based on a zero reference point. For details about the settings, see *2.1.4(2) Setting PFM - RM for Virtual Machine*.

The following is an example when you set the `UseOffset` value to enable (`Y`), the `Collection Interval` value of the record to 60 seconds, and the `Offset` value of the instance environment to 10 seconds.

Figure N–2: Example procedure for when using a zero reference point



> **Tip**
>
> If you set the `UseOffset` value to enable (`Y`), the first collection immediately after the service starts is performed one minute after the collection process starts. Subsequent collections start in the period (in seconds, specified in the `Offset` value during startup of the instance environment) from the time the next collection process is to be started (which is the zero reference point).

If multiple instances are monitored and you choose to use a zero reference point for the collection process, you might need the settings described below.

Load balancing for the collection process

If you set up more than one instance while using the default `Offset` value (10 seconds) in the instance environment, collection processes start at the same time. In an environment where a large number of instances exist, the load on the PFM - RM for Virtual Machine host increases significantly, which might cause an operational problem.

To prevent this, set a different `Offset` value for each instance.

The Offset value can be 1 through 3600. However, taking into account the time it takes for the Remote Monitor Collector service to read a temporary performance information file, specify a time about 10 seconds longer than the Collection Offset value of each record.

Figure N–3: Example for when the same Offset value is used for all instances

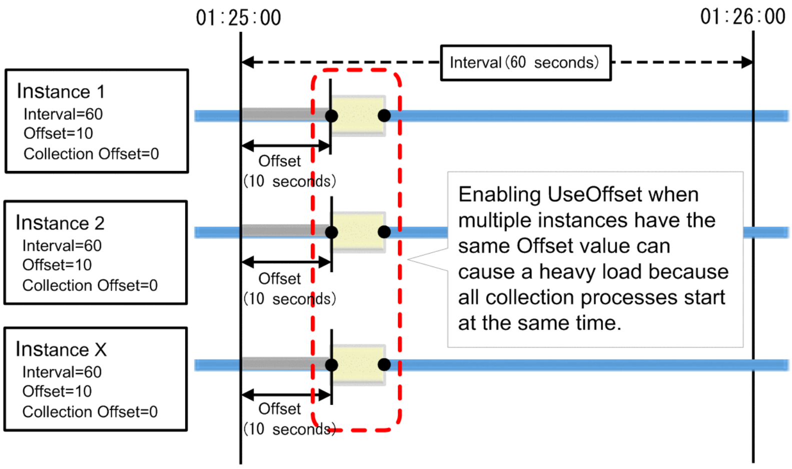
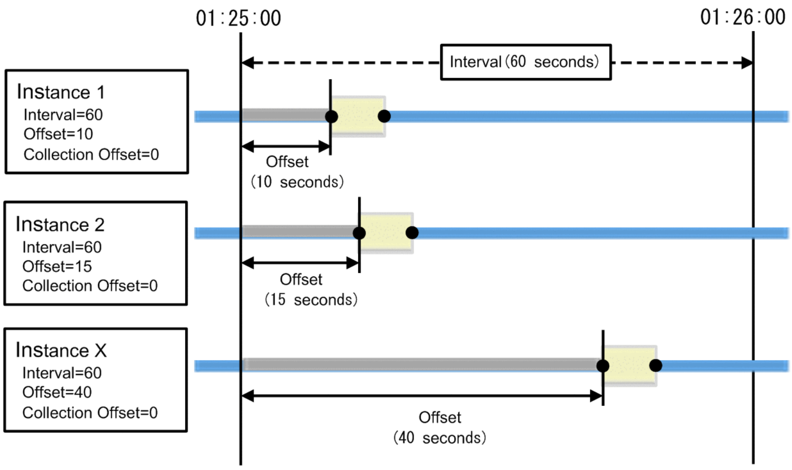


Figure N–4: Example for when a different Offset value is used for each instance

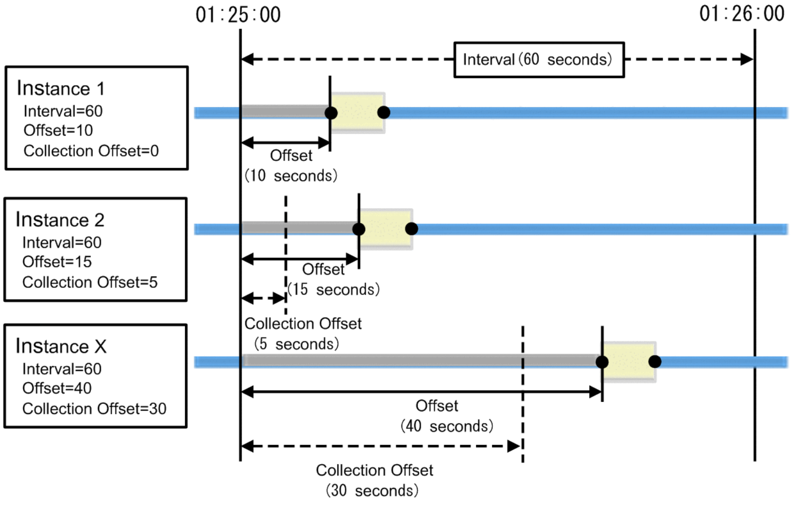


Changing the Collection Offset value

Setting a larger Offset value close the Collection Interval value (which determines when the Remote Monitor Collector service reads a temporary performance information file) can lead a repeated KAVL20516-W message output. To prevent this, use the Collection Offset setting for each record to change the time for which the service reads the temporary file.

If you do so, use the same Collection Offset value for all records in the instance.

Figure N–5: Example for when a different Collection Offset value is used for each instance



O. Version Revisions

This appendix shows the changes that have been made to each version of the manual.

O.1 Revisions in 12-50

- Added the Podman environment of Red Hat Enterprise Linux(R) Server 8 to the targets whose resources are to be monitored.
- Red Hat Enterprise Linux(R) Server 8 is now supported when KVM is to be monitored.
- Added `SSH_Type` to the instance information.
- Added Windows Server 2019 standard OpenSSH as an SSH client when KVM is to be monitored.
- Changed the version of the data model to 9.0.
- Changed the version of the alarm table for monitoring templates to 12.50.
- With the change of data model, the versions of the following reports that use fields for which the type of their data model was changed in version 9.0 or that refer to such reports were changed:
    - Host CPU Used Status
    - Host Disk Used
    - Host Disk Used Status
    - VM CPU Allocation Value
    - VM CPU Insufficient
    - VM CPU Trend
    - VM CPU Used
    - VM CPU Used Status
    - VM Disk Used
- Added the next item to records:
    - POD Status Detail
    - POD Status Interval
    - POD Container Status Interval
- Added the fields for the following records:
    - Host Logical Disk Status (PI_HLDI)
    - Host Status (PI)
    - VM CPU Status (PI_VCI)
    - VM Logical Disk Status (PI_VLDI)
    - VM Status Detail (PD_VM)
    - VM Status (PI_VI)
- Added the descriptions or actions for when the following messages are output:
    - KAVL20014-W
    - KAVL20201-W

- KAVL20517-W
- KAVL20524-W
- KAVL20526-W

O.2 Revisions in 12-10

- Added PI_VVDI and PD_VDKD records to monitor VMware disk resources.
- Changes were made to the procedure for registering PFM - RM for Virtual Machine.
- Changed the version of the data model to 8.0.
- Added the VM Virtual Disk Allocation Value (8.0) report.
- Changed the version of the alarm table for monitoring templates to 12.10.
- Added the VM Virtual Disk Detail (PD_VDKD) record.
- VMware vSphere Web Client was added as a data source of field values when VMware is monitored.

O.3 Revisions in 12-00

- The following OSs are now supported:
    - Microsoft(R) Windows Server(R) 2019 Datacenter
    - Microsoft(R) Windows Server(R) 2019 Standard
- Deleted a series of descriptions of the following OSs:
    - Windows Server 2008 R2
- The following has been deleted from the monitoring target:
    - 2008 Hyper-V
    - 2008 R2 Hyper-V
    - VMware ESX 3
    - VMware ESX 4
- When the monitoring target is a Docker environment, Docker environment for Windows is now supported.
- User defined records that can be collected only when the monitoring target is a VMware environment are now supported.
- Added Offset to the instance information.
- Changed the version of the data model to 7.0.
- Changed the version of the alarm table for monitoring templates to 12.00.
- Added the Last Update field to the following reports:
    - Host Disk Used (7.0)
    - Host Disk Used Status (7.0)
- Added Last Update fields of the Host Logical Disk Status (PI_HLDI) record, and the sizes of the following records were changed.
- Added the message with the following message ID:

- KAVL20527-W
- KAVL20528-I
- KAVL20529-W
- Added descriptions of what to do when the message KAVL20516-W is output.
- Deleted the message with the following message ID:
  KAVL20400-I to KAVL20488-E

O.4 Revisions in 11-50

- Added Docker environment to the target for resource monitoring.
- For using a cluster system for operation, SSL/TLS connection is added to prerequisite conditions of installation and setup.
- Changed the version of the data model to 6.0.
- Changed the version of the alarm table for monitoring templates to 11.50.
- Added the next report to a list of reports.
  - VM CPU Trend
  - VM Memory Trend
- Added available virtual environment to a list of alarms.
- Added the next item to records.
  - Over 10 Sec Collection Time

O.5 Revisions in 11-00

- In the setting items of the monitoring target, one of the values `1`, `2`, or `3` can now be specified for `Security` when SSL is used.
- Hyper-V on a PFM - RM host can now be monitored.
- Red Hat Enterprise Linux(R) Server 7 is now supported if the monitoring target is a KVM environment.
- Added the `hostname` and `cat` commands as commands required for collecting records if the monitoring target is a KVM environment.
- Changed the version of the alarm table for monitoring templates to 11.00.
- Added fields of the Host Physical Disk Status (`PI_HPDI`) record.
- Added the VM Virtual Disk Status (`PI_VVDI`) record.
- Added the message with the following message ID:
  KAVL20206-W
- Changed the version of the data model to 5.0.
- Deleted a series of descriptions of the following OSs:
  - Windows Server 2003
  - Windows Server 2003 R2
  - Windows Server 2008

- The following languages can now be used with Performance Management:
    - Korean
    - Spanish
    - German
    - French
    - Russian
- The product name was changed from JP1/ITSLM to JP1/SLM.

P. Reference Material for This Manual

This appendix provides reference information, including various conventions, for this manual.

P.1 Related publications

This manual is part of a related set of manuals. The manuals in the set are listed below (with the manual numbers):

Manuals associated with JP1/Performance Management:

- *JP1 Version 12 Performance Management: Getting Started* (3021-3-D75(E))
- *JP1 Version 12 JP1/Performance Management Planning and Configuration Guide* (3021-3-D76 (E))
- *JP1 Version 12 JP1/Performance Management User's Guide* (3021-3-D77(E))
- *JP1 Version 12 JP1/Performance Management Reference* (3021-3-D78(E))

Manuals associated with JP1:

- *Job Management Partner 1/Software Distribution Administrator's Guide Volume 1* (3020-3-S81(E)), for Windows systems

P.2 Conventions: Abbreviations for product names

This manual uses abbreviated names for products of Hitachi and other companies. The following table lists the abbreviated names and their full names.

Abbreviation			Full name or meaning
JP1/IM	JP1/IM - Manager		JP1/Integrated Management - Manager
			JP1/Integrated Management 2 - Manager
	JP1/IM - View		JP1/Integrated Management - View
			JP1/Integrated Management 2 - View
JP1/ITSLM (10-50 or earlier)	JP1/ITSLM - Manager		Job Management Partner 1/IT Service Level Management - Manager
	JP1/ITSLM - UR		Job Management Partner 1/IT Service Level Management - User Response
JP1/SLM	JP1/SLM - Manager		JP1/Service Level Management - Manager
	JP1/SLM - UR		JP1/Service Level Management - User Response
JP1/SD	JP1/Software Distribution		Job Management Partner 1/Software Distribution Client
			Job Management Partner 1/Software Distribution Manager
			Job Management Partner 1/Software Distribution SubManager
Linux	CentOS	CentOS 6 (x64)	CentOS 6.1 (x64) or later
		CentOS 7	CentOS 7.1 or later

Abbreviation			Full name or meaning
Linux	CentOS	CentOS 8	CentOS 8.1 or later
	Linux 6 (x64)		Red Hat Enterprise Linux(R) Server 6.1 (64-bit x86_64) or later
	Linux 7		Red Hat Enterprise Linux(R) Server 7.1 or later
	Linux 8		Red Hat Enterprise Linux(R) Server 8.1 or later
	Oracle Linux	Oracle Linux 6 (x64)	Oracle Linux(R) Operating System 6.1 (x64) or later
		Oracle Linux 7	Oracle Linux(R) Operating System 7.1 or later
		Oracle Linux 8	Oracle Linux(R) Operating System 8.1 or later
	SUSE Linux	SUSE Linux 12	SUSE Linux(R) Enterprise Server 12
		SUSE Linux 15	SUSE Linux(R) Enterprise Server 15
Performance Management			JP1/Performance Management
PFM - Agent	PFM - Agent for JP1/ AJS#	PFM - Agent for JP1/ AJS2	JP1/Performance Management - Agent Option for JP1/AJS2
		PFM - Agent for JP1/ AJS3	JP1/Performance Management - Agent Option for JP1/AJS3
	PFM - Agent for Enterprise Applications		JP1/Performance Management - Agent Option for Enterprise Applications
	PFM - Agent for Service Response		JP1/Performance Management - Agent Option for Service Response
PFM - Base			JP1/Performance Management - Base
PFM - Manager			JP1/Performance Management - Manager
PFM - RM	PFM - RM for Microsoft SQL Server		JP1/Performance Management - Remote Monitor for Microsoft(R) SQL Server
	PFM - RM for Oracle		JP1/Performance Management - Remote Monitor for Oracle
	PFM - RM for Platform	PFM - RM for Platform	JP1/Performance Management - Remote Monitor for Platform
	PFM - RM for Virtual Machine		JP1/Performance Management - Remote Monitor for Virtual Machine
PFM - Web Console			JP1/Performance Management - Web Console
VMware			VMware vSphere ESXi 5
			VMware vSphere ESXi 6

- PFM - Manager, PFM - Agent, PFM - Base, PFM - Web Console, and PFM - RM may be referred to collectively as *Performance Management*.
- VMware system physical servers, Hyper-V system physical servers, KVM system physical servers, host machines of a system with logical partitioning feature, and Docker environment system physical servers are sometimes generically referred to as *physical servers*. Also, VMware system virtual machines, Hyper-V system virtual machines, and logical partitions of a system with logical partitioning feature are sometimes generically referred to as *virtual machines*.
- Also, VMware system virtual machines, Hyper-V system virtual machines, logical partitions of a system with logical partitioning feature, and Docker containers are sometimes generically referred to as *virtual machines*.

- A Docker environment has Docker containers and the host that manages Docker containers (Docker host).

  The Docker environment for Windows is sometimes referred to as the *Docker environment (Windows)* or *Docker container (Windows)*. Likewise, the Docker environment for Linux is sometimes referred to as the *Docker environment (Linux)* or *Docker container (Linux)*.

  Also, the Docker environments for Windows and Linux are sometimes collectively referred to as the *Docker environment* or *Docker container*.

#

This product only runs in a Japanese environment.

P.3 Conventions: Acronyms

This manual also uses the following acronyms:

Acronym	Full name or meaning
CPU	Central Processing Unit
DCOM	Distributed Component Object Model
DHCP	Dynamic Host Configuration Protocol
DMZ	DeMilitarized Zone
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
GMT	Greenwich Mean Time
HA	High Availability
HBA	Host Bus Adapter
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Security
IP	Internet Protocol
IPF	Itanium(R) Processor Family
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
JST	Japan Standard Time
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LPAR	Logical Partition
MSDTC	Microsoft Distributed Transaction Coordinator
NAPT	Network Address Port Translation
NAT	Network Address Translation
NIC	Network Interface Card

Acronym	Full name or meaning
ODBC	Open Database Connectivity
OS	Operating System
RPM	Redhat Package Manager
SAN	Storage Area Network
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer
SSL/TLS	Secure Socket Layer/Transport Layer Security
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UAC	User Access Control
UTC	Universal Time, Coordinated
Web	World Wide Web
WMI	Windows Management Instrumentation

P.4 Conventions: Product names, service IDs, and service keys

Performance Management version 09-00 or later can display the product name as the service ID and service key by enabling the product name display functionality.

Identifiers	Product name display functionality	
	Disabled	Enabled
Service ID	8S*instance-number instance-name*[*hostname*]	*instance-name*[*hostname*]<RM VirtualMachine>(Store)
	8A*instance-number instance-name*[*hostname*]	*instance-name*[*hostname*]<RM VirtualMachine>
Service Key	agt8	RMVM

Hereafter in this manual, service IDs and service keys are shown in the format when the product name display functionality is enabled.

Note that you can enable the product name display functionality only when you satisfy the two conditions listed below:

- The version number of the prerequisite programs (PFM - Manager or PFM - Base) is 09-00 or later.
- The version number of PFM - Web Console and connection-target PFM - Manager is 09-00 or later.

P.5 Conventions: Installation folder

The default installation folder for the Windows version of Performance Management is as follows:

The part indicated as *system-drive*\Program Files might appear differently in a different environment because this part varies depending on the OS environment variable used upon installation.

Installation-folder for PFM - Base:

system-drive`\Program Files(x86)\Hitachi\jp1pc`

This manual uses the term *installation folder* for the PFM - Base installation folder.

Installation-folder for PFM - Management:

system-drive`\Program Files(x86)\Hitachi\jp1pc`

Installation-folder for PFM - Web Console:

system-drive`\Program Files(x86)\Hitachi\jp1pcWebCon`

P.6 Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

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

Q. Glossary

A

action

Performance Management automatically executes an action when monitored data reaches a threshold. The following types of actions are available:

- Email transmission
- Command execution
- SNMP trap generation
- JP1 event generation

Action Handler

This is one of the PFM - Manager or PFM - Base services. This service executes an action.

agent

Refers to PFM - RM, which collects performance data.

alarm

Information that defines the action that is to be taken or the event message that is issued when monitored data reaches a threshold.

alarm table

A table summarizing one or more alarms that define the following types of information:

- Monitored object (process, TCP, Web service, for example)
- Monitored information (CPU usage, the number of bytes received in 1 second, for example)
- Monitored condition (threshold)

B

binding

Binding means associating an alarm with an agent. When an alarm is bound, the user can be notified when the performance data collected by the agent reaches the threshold defined by the alarm.

C

cluster system

A cluster system links together multiple server systems and operates them as a single system.

In this manual, *cluster system* refers to an HA cluster system unless otherwise specified.

See *HA cluster system*.

Correlator

This is one of the PFM - Manager services. This service controls event delivery between services. It evaluates an alarm status, and when the threshold is exceeded, sends an alarm event and an agent event to both the Trap Generator service and PFM - Web Console.

D

data model

Generally refers to the records and fields that each PFM - RM has. Data models are managed by version.

database ID

An ID assigned to each record of PFM - RM to indicate the database in which each record is stored. A database ID indicates the type of records that are stored in the database.

- PI: Indicates a database for storing records of the PI record type.
- PD: Indicates a database for storing records of the PD record type.

drilldown report

This is a report associated with a report or report field, and is used for displaying detailed information or related information about a report.

E

executing node

In server systems that comprise a cluster system, the node that is executing a job (the node on which the logical host is active).

F

failover

A process of switching over the processing of the server that is executing jobs from the executing node to the standby node when an error occurs in a cluster system.

field

Individual operation information entries in a record. Each field serves as a monitoring item for Performance Management.

function ID

This is a 1-byte identifier indicating a function type of a Performance Management program, and is part of a service ID.

H

HA cluster system

A cluster system for achieving high availability, designed to continue running even when an error occurs. When an error occurs on a server that is processing jobs, another server that has been on standby takes over job processing. In this way, jobs are not interrupted by an error and system availability is improved.

In this manual, *cluster system* refers to an HA cluster system unless otherwise specified.

historical report

A report that shows the status history of the monitored target.

I

instance

In this manual, the term *instance* is used in the following ways:

- To indicate a recording format for records

  A record that is recorded on one line is referred to as a *single-instance record*, a record that is recorded on multiple lines is referred to as a *multi-instance record*, and each line in a record is referred to as an *instance*.

- To indicate the PFM - RM startup method

  When a single agent monitors targets on the same host, it is referred to as a *single-instance agent*. When multiple agents monitor targets on the same host, they are referred to as *multi-instance agents*. Each of these multi-instance agents is referred to as an *instance*.

instance number

This is an identifier that indicates a 1-byte control number used in internal processing, and that is part of a service ID.

J

JP1/SLM

A product that performs monitoring from the viewpoint of performance as experienced by the service users of a business system, and that supports service-level maintenance. Linkage with JP1/SLM can enhance monitoring of the operating status.

L

lifetime

A period during which the consistency of the performance data collected into records can be guaranteed.

logical host

This is the logical server that becomes the executing environment for JP1 when it operates in a cluster system. When an error occurs, systems are switched on a logical host basis.

A logical host has its own IP address, and this IP address is inherited during a failover. Therefore, even when physical servers are switched because of an error, clients can use the same IP address to access the server, making it appear as though a single server is always running.

M

management tools

Management tools are various types of commands and GUI functions that are used to check service status and processing performance data. They can be used to perform the following types of operations:

- Displaying service configuration and status
- Saving and restoring performance data
- Exporting performance data to a text file
- Erasing performance data

Master Manager

This is one of the PFM - Manager services, and refers to PFM - Manager's main service.

Master Store

This is one of the PFM - Manager services, and refers to the service that manages alarm events issued by each PFM - RM. The Master Store service uses a database for retaining event data.

monitoring template

Refers to a pre-defined alarm or report provided in PFM - RM. By using a monitoring template, you can easily prepare to monitor the operation status of PFM - RM without having to create complex definitions.

multi-instance record

A record that is recorded on multiple lines. This record has a specific ODBC key field.

See *Instance*.

N

Name Server

A PFM - Manager service that manages the service configuration information in the system.

Non-interactive (command)

Command execution method in which operator input required for command execution are provided by values in option specifications or in definition files.

Executing a command non-interactively saves work when configuring an operation monitoring system and can reduce user workload.

O

ODBC key field

These fields display the primary keys that are necessary to use the data retrieved from records stored in the Store database on either PFM - Manager or PFM - Base. Some ODBC key fields are common to all records, while others are specific to individual records.

P

PD record type

See *Product Detail record type*.

performance data

Operation status data of a resource collected from the monitored system.

Performance Management

A general term that refers to a group of software program products necessary for monitoring and analyzing performance-related problems of a system. Performance Management consists of the following five program products:

- PFM - Manager
- PFM - Web Console
- PFM - Base
- PFM - Agent
- PFM - RM

PFM - Agent

One of the program products comprising Performance Management. PFM - Agent is equivalent to a system monitoring function, and various types of PFM - Agents are available according to the application to be monitored, the database, and the OS. PFM - Agent has the following functions:

- Monitoring the monitored target
- Collecting and recording data from a monitored target

PFM - Base

One of the program products comprising Performance Management. PFM - Base provides Performance Management with the core functions necessary for monitoring operations. PFM - Base is a prerequisite program product for running PFM - RM. PFM - Base has the following functions:

- Management tools such as various types of commands
- Common functions necessary for linking Performance Management to other systems

PFM - Manager

One of the program products comprising Performance Management. PFM - Manager is equivalent to the manager function and has the following functions:

- Management of the Performance Management program products

- Event management

PFM - Manager name

Name that identifies a field stored in the Store database, and that is used to specify a field for a command, for example.

PFM - RM

One of the program products in the Performance Management family. PFM - RM is responsible for system monitoring. Several types of PFM - RM are available, depending on the applications, database, and operating system to be monitored. PFM - RM provides the following features:

- Performance monitoring of target objects
- Collection and recording of data from monitored objects

PFM - View name

Alias for the PFM - Manager name. The PFM - View name is a more intuitive name than the PFM - Manager name. For example, the PFM - Manager name `INPUT_RECORD_TYPE` becomes Record Type as a PFM - View name. The PFM - View name is used to specify a field in the GUI of PFM - Web Console, for example.

PFM - Web Console

One of the program products comprising Performance Management. It provides Web application server functions for centrally monitoring the Performance Management system. PFM - Web Console has the following functions:

- GUI display
- Integrated monitoring and management functions
- Report definition and alarm definition

physical host

An environment specific to each server that makes up a cluster system. A physical host's environment is not inherited by another server during a failover.

physical server

This is a physical server on which a virtual environment operates, and it has various types of resources such as a CPU. Virtual machines on the same physical server share the physical server's resources.

PI record type

See *Product Interval record type*.

Product Detail record type

This record type stores performance data that indicates the system status at a given point in time, such as detailed information about the process that is currently active. Use the PD record type when you want to know the system status at a given point in time, such as the following:

- The system's operation status
- The capacity of the file system currently being used

product ID

This is a one-byte identifier that indicates the program product of Performance Management to which the service of the applicable Performance Management program belongs, and that is part of a service ID.

Product Interval record type

This record type stores performance data for a given period of time (interval), such as the number of processes for 1 minute. Use the PI record type when you want to analyze changes or trends in the system status over time, such as the following:

- Trends in the number of system calls that occurred within a specified period of time
- Trends in the capacity of the file system being used

R

real-time report

A report that indicates the current status of the monitored target.

record

A group of operation information entries categorized by their purpose. A monitoring agent collects operation information from each record. The types of records that can be collected vary depending on the agent program.

Remote Monitor Collector

A PFM - RM service that collects performance data and evaluates the data according to the threshold values set in alarms.

Remote Monitor Store

A PFM - RM service that stores performance data in a database. A separate Remote Monitor Store service is provided with each PFM - Remote Monitor platform.

report

A report defines how performance data collected by PFM - RM is graphically displayed. It primarily defines the following types of information:

- Records to be displayed in a report
- Performance data display items
- Performance data display format (table, graph, and so on)

S

service ID

A unique ID assigned to a Performance Management program. When you use a command to check the system configuration of Performance Management, or when you back up the performance data of each agent, you need to execute the command by specifying the service ID of the Performance Management program. The service ID format varies depending on the settings of the product name display function. For details about the service ID format, see the chapter that explains the functions of Performance Management in the *JP1/Performance Management Planning and Configuration Guide*.

single-instance record

A record that is recorded on a single line. This record does not have a specific ODBC key field.

See *Instance*.

stand-alone mode

This is a status in which PFM - RM only has started. Even when the Master Manager service or Name Server service of PFM - Manager cannot be started because of an error, you can start PFM - RM by itself and collect performance data.

standby node

In server systems that comprise a cluster system, a standby node can assume the jobs that were being executed by the executing node when an error occurred on it.

status management function

This function manages the status of all services running on PFM - Manager and PFM - RM. Using the status management function, the system administrator can accurately assess the service startup or stoppage status on each host, and can therefore take the appropriate corrective actions quickly.

Store database

This database stores the performance data collected by Remote Monitor Collector service.

SYS1

Name of the kernel component in the hypervisor that manages a system with logical partitioning feature.

SYS2

Name of the communication and service component in the hypervisor that manages a system with logical partitioning feature.

V

virtual machine

This refers to a virtual machine that is provided by software and that uses virtual resources provided in the resources of a physical server.

virtual machine monitor

See *VMM*.

VMM

Stands for Virtual Machine Manager, which is the kernel for controlling virtual machines.

VMM console

This is the console for operating VMM.

VMM kernel

This is the kernel portion of VMM.

Index

M

N

O

P

R

S

T

U

Hitachi, Ltd.

6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 100–8280 Japan