

WIN-63-5004

HITACHI



HITACHI INDUSTRIAL COMPUTER

HF-W200E

RAS FEATURES MANUAL

[illegible]

USER'S MANUAL

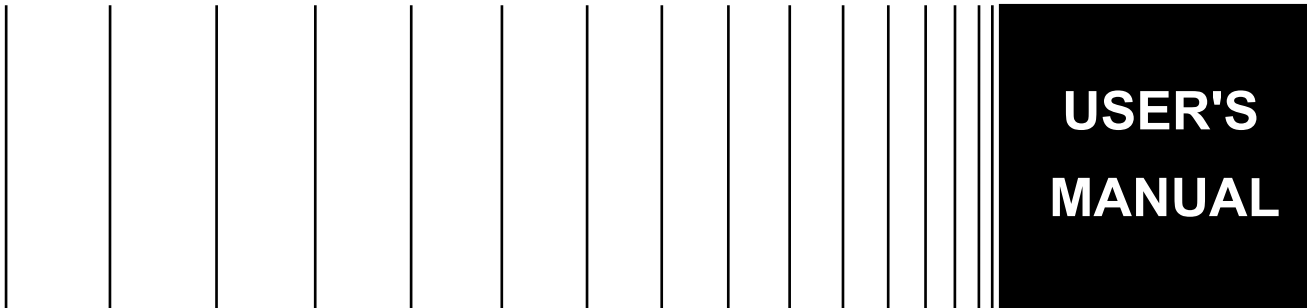
HITACHI INDUSTRIAL COMPUTER

HF-W200E

RAS FEATURES MANUAL

Read and retain this manual.

- Read safety instructions carefully, and ensure that you understand them before beginning operations.
- Keep this manual on hand for reference.



If you plan to export this product, verify the laws and regulations that are relevant to export to foreign countries, including the Foreign Exchange and Foreign Trade Act of Japan and the export controls of the United States, and complete all required procedures. If anything is unclear, contact a sales representative.

First Edition, September 2025, WIN-63-5004

- Do not copy the contents of this manual without permission.
- The contents of this manual are subject to change without notice.

All Rights Reserved, Copyright © 2025, Hitachi Industrial Products, Ltd.



SAFETY INSTRUCTIONS

Carefully read and sufficiently understand the following safety precautions before operating the equipment.

- Operate the equipment by following the instructions and procedures described in this manual.
- In particular, note the safety precautions displayed on the equipment or in this manual and follow them. Failure to do so might result in personal injury and property damage, including damage to the equipment.
- Safety precautions are indicated by one of the following headings. A heading is either a safety alert symbol; a word such as DANGER, WARNING, CAUTION, or NOTICE; or a combination of both.



This is a safety alert symbol. This symbol is used to indicate potential hazards that might result in personal injury or death. Follow the instructions in the safety messages that follow this symbol to avoid possible injury or death.



DANGER: This symbol is used to indicate imminent hazards that are highly likely to result in serious personal injury or death.



WARNING: This symbol is used to indicate potential hazards that might result in serious personal injury or death.



CAUTION: This symbol is used to indicate potential hazards that might result in moderate or minor personal injury.

NOTICE: This symbol is used to indicate hazards that might result in equipment or property damage, but not personal injury.

The term *serious injury* as described here is an injury that has aftereffects and requires hospitalization for medical treatment or long-term follow-up care. Examples of serious injuries are vision loss, burns (caused by dry heat), low-temperature burns, electric-shock injuries, broken bones, and poisoning. A *moderate or minor injury* is an injury that does not require either hospitalization for medical treatment or long-term follow-up care. Examples of such injuries are burns and electric-shock injuries.

Damage not related to personal injury is damage other than personal injury. Examples of such damage are damage or loss of personal property, failure of or damage to the product, and loss of data.

The heading NOTE is used to indicate a cautionary note about the handling or operation of the equipment.



SAFETY INSTRUCTIONS (Continued)

- Do not attempt to perform any operations that are not described in this manual. If you encounter any problems with the equipment, contact your maintenance personnel.
- Read this manual carefully, and sufficiently understand the directions and precautions in this manual before operating the equipment.
- Keep this manual nearby so that you are able to refer to it as needed.
- Although every effort has been made in this manual to specify the most complete and relevant precautions regarding the equipment, unexpected incidents might occur. When using the equipment, use your own judgment on matters related to safety, in addition to following the instructions herein.



SAFETY INSTRUCTIONS (Continued)

1. Safety Warnings in This Manual

1.1 Safety warnings indicated as NOTICE

- When failure of a storage is anticipated, the storage may experience hardware failure in the near future. We recommend you to back up the data and replace this equipment. Please contact the system administrator or maintenance personnel.

(Page 2-2)

- When the OS deadlocks, processes in the OS cannot run as scheduled, and the facility that is using this equipment might be affected due to delays in processing. If the OS deadlocks, resolve the problem immediately.

(Page 2-4)

- When the **Hardware status** window shows an error in the hardware, resolve the problem causing the error immediately.

(Page 4-2)

- When failure of a storage is anticipated, the storage might experience hardware failure in the near future. We recommend you to back up the data and replace this equipment. Please contact the system administrator or maintenance personnel.

(Page 4-8)



SAFETY INSTRUCTIONS (Continued)

- The log function exits asynchronously without waiting for data to actually be written to a log file. This means that this function does not return an error even when writing to a log file fails for some reason. We recommend recording important information in the OS event log.

(Page 6-12)

- The CPU load increases while memory dump files are being collected. While the CPU is under a high load, the operation of user applications can be disturbed. Make sure that you do not collect memory dump files by using the log information collection window while applications for business use are running on this equipment.

(Page 7-4)

- While the equipment is running in simulation mode, monitoring of the actual hardware status is disabled. Errors, including abnormal temperatures, cannot be detected. Never use this equipment in simulation mode for business use. Use the simulation feature only for testing user applications and checking the notification interface of the RAS software.

(Page 8-2)

PREFACE

This manual describes how to use the Reliability, Availability, and Serviceability (RAS) features of the HITACHI INDUSTRIAL COMPUTER HF-W200E (abbreviated hereinafter to *this equipment*).

Structure of this manual

This manual consists of the following:

- CHAPTER 1 CAPABILITIES OF THE RAS FEATURES
- CHAPTER 2 ITEMS MONITORED BY THE RAS FEATURES
- CHAPTER 3 SETTING UP THE RAS FEATURES
- CHAPTER 4 CHECKING THE HARDWARE STATUS
- CHAPTER 5 CONTROLLING THE HARDWARE
- CHAPTER 6 LIBRARY FUNCTIONS
- CHAPTER 7 FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS
- CHAPTER 8 SIMULATING THE HARDWARE STATUS

Precautions for using the RAS features

- Precautions for the RAS external contact interface

This manual describes a RAS external contact interface. Note that this interface is an optional feature.

- Precautions for event log entries at the startup of the SNMP service

When you enable the SNMP service, a standard feature of Windows®, in order to use remote notifications, an error log entry with event ID 1500 might be recorded upon startup of the SNMP service. This event log entry is recorded when SNMP trap notifications have not been set up. Set up the trap notifications as described in *4.4.3 Enabling remote notifications*.

- User Account Control

If User Account Control (UAC) is enabled in Windows®, User Account Control dialog boxes might be displayed when you attempt to run an application or a command. In this case, click **OK** or **Continue**.

Trademarks

- Microsoft®, Windows®, Windows Server®, Windows NT®, and Visual Basic® are registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- All other product names of software and hardware described in this manual that are not from Hitachi are the registered trademarks, trademarks, or products of their respective owners.

Note on storage capacity calculations

- Memory capacities and requirements, file sizes and storage requirements, etc., are calculated according to the formula 2^n :
 - 1 KB (kilobyte) = 1,024 bytes (2^{10} bytes)
 - 1 MB (megabyte) = 1,048,576 bytes (2^{20} bytes)
 - 1 GB (gigabyte) = 1,073,741,824 bytes (2^{30} bytes)
 - 1 TB (terabyte) = 1,099,511,627,776 bytes (2^{40} bytes)
- Disk capacities are calculated according to the formula 10^n :
 - 1 KB (kilobyte) = 1,000 bytes (10^3 bytes)
 - 1 MB (megabyte) = 1,000² bytes (10^6 bytes)
 - 1 GB (gigabyte) = 1,000³ bytes (10^9 bytes)
 - 1 TB (terabyte) = 1,000⁴ bytes (10^{12} bytes)

CONTENTS

 SAFETY INSTRUCTIONS	S-1
CHAPTER 1 CAPABILITIES OF THE RAS FEATURES	1-1
CHAPTER 2 ITEMS MONITORED BY THE RAS FEATURES	2-1
2.1 Monitoring the Temperature Inside the Chassis	2-1
2.2 Storage Failure Prediction (SMART monitoring)	2-2
2.3 Memory Monitoring	2-3
2.4 OS Deadlock Monitoring	2-4
2.5 Watchdog Timer Monitoring	2-5
2.5.1 Automatic retriggering feature for a watchdog timer	2-5
2.5.2 Using a watchdog timer to monitor user programs	2-5
CHAPTER 3 SETTING UP THE RAS FEATURES	3-1
3.1 RAS Setup Window	3-1
3.1.1 Overview	3-1
3.1.2 Opening the RAS Setup window	3-2
3.1.3 Using the RAS Setup window	3-3
3.1.4 Editing pop-up notification messages	3-8
CHAPTER 4 CHECKING THE HARDWARE STATUS	4-1
4.1 Hardware Status Window	4-2
4.1.1 Overview	4-2
4.1.2 Hardware status icon	4-3
4.1.3 Hardware status window	4-7
4.2 RAS Event Notifications	4-10
4.2.1 Overview	4-10
4.2.2 Detecting an event	4-10
4.2.3 Example of using event objects	4-10
4.3 Pop-up Notifications	4-11
4.3.1 Overview	4-11
4.3.2 Messages to be displayed	4-12
4.3.3 Pop-up notification settings	4-12
4.4 Remote Notifications	4-13
4.4.1 Overview	4-13
4.4.2 Hardware conditions that can be acquired by using remote notifications	4-14

4.4.3 Enabling remote notifications	4-15
4.4.4 Objects in the extended MIB for HF-W	4-20
4.4.5 Extended MIB file for HF-W	4-25
4.5 Status Acquisition by Using the RAS Library	4-26
 CHAPTER 5 CONTROLLING THE HARDWARE	 5-1
5.1 Automatic Shutdown of the Equipment	5-2
5.1.1 Automatic shutdown when detecting an abnormally high temperature	5-3
5.2 Memory Dump due to Timeout Detection	5-4
5.3 Controlling the Hardware by Using the RAS Library	5-4
 CHAPTER 6 LIBRARY FUNCTIONS	 6-1
6.1 RAS Library	6-1
6.1.1 Overview	6-1
6.1.2 Shutdown function (BSSysShut)	6-3
6.1.3 Watchdog timer control function (WdtControl)	6-4
6.1.4 Control functions for the general-purpose external contact outputs (GendoControlN)	6-6
6.1.5 Get functions for the general-purpose external contact inputs (GetGendiN)	6-9
6.1.6 Log function (MConWriteMessage)	6-11
6.1.7 Get function for the memory condition (GetMemStatus)	6-13
6.1.8 Get function for the drive condition (hfwDiskStat)	6-15
6.2 Sample Programs	6-17
 CHAPTER 7 FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS	 7-1
7.1 Notifying the Cause of the STOP Error Code	7-1
7.1.1 Overview	7-1
7.1.2 Supported causes of the STOP error code	7-2
7.1.3 Event log	7-3
7.2 Log Information Collection Window	7-4
7.2.1 Overview	7-4
7.2.2 Starting the log information collection window	7-4
7.2.3 Using the log information collection window	7-5
7.2.4 Closing the log information collection window	7-7
7.3 Logging Trends in the Temperature Inside the Chassis	7-8
7.3.1 Overview	7-8
7.3.2 Log files	7-8
7.3.3 Logging frequency setup command	7-10

CHAPTER 8 SIMULATING THE HARDWARE STATUS 8-1

8.1 Hardware Status Simulation 8-1

 8.1.1 Overview 8-1

 8.1.2 Using the simulation feature 8-3

 8.1.3 Precautions when you use the Simulation Tool window 8-13

 8.1.4 Event log entries 8-13

 8.1.5 Remote notifications 8-14

FIGURES

Figure 1-1 RAS Setup window	1-2
Figure 1-2 Hardware status icon	1-3
Figure 2-1 Example flowchart of monitoring the operational state of a user program	2-5
Figure 3-1 RAS Setup window	3-1
Figure 3-2 Shutdown settings items	3-3
Figure 3-3 Items in the watchdog timer settings	3-4
Figure 3-4 Items in the pop-up notification settings	3-5
Figure 3-5 Advanced pop-up notification settings	3-6
Figure 3-6 Format of a message definition file	3-8
Figure 4-1 Hardware status window	4-2
Figure 4-2 Example of displaying the description of an icon (when the hardware status is normal)	4-6
Figure 4-3 Example of displaying the description of an icon (when the hardware status has an error)	4-6
Figure 4-4 Menu of the hardware status icon	4-6
Figure 4-5 Starting the Hardware status window	4-7
Figure 4-6 Hardware status window (example display in the case of an error)	4-9
Figure 4-7 Example pop-up message notification	4-11
Figure 6-1 Operation Example of GendoControlN Function	6-7
Figure 6-2 Operation Example of the External general purpose output	6-8
Figure 6-3 Format of log information	6-11
Figure 7-1 Folder structure for collected data	7-7
Figure 7-2 Format of the log information (1)	7-9
Figure 7-3 Format of the log information (2)	7-9
Figure 8-1 Simulation Tool window	8-1
Figure 8-2 Procedure for using the simulation mode	8-3
Figure 8-3 Description of the Simulation Tool window	8-6

TABLES

Table 1-1 Overview of RAS features	1-1
Table 3-1 Features that can be set up in the RAS Setup window	3-1
Table 3-2 Section names and defined messages	3-9
Table 3-3 Items displayed in the object list for each option selected in the event list	3-10
Table 4-1 Hardware status icon	4-5
Table 4-2 Temperature condition and displayed information	4-8
Table 4-3 Storage condition and displayed information	4-8
Table 4-4 Reported events	4-10
Table 4-5 Displayed messages	4-12
Table 4-6 Objects related to the hardware status	4-20
Table 4-7 Objects related to the RAS feature settings	4-22
Table 4-8 Objects related to operating modes	4-22
Table 4-9 Objects related to the extended MIB for HF-W	4-23
Table 4-10 Objects related to the trap notification (when an error occurs)	4-24
Table 4-11 Objects related to trap notifications (when the equipment has recovered from an error)	4-25
Table 4-12 Objects related to trap notifications (operating modes)	4-25
Table 6-1 RAS library functions	6-1
Table 6-2 Actions for WdtControl that are specified by dwCmd	6-4
Table 6-3 Bit Allocation of ucOutput and ucMask for GendoControlN	6-6
Table 6-4 Bit Allocation of pcInput for GetGendiN	6-9
Table 6-5 List of values stored in Dimm_Status	6-13
Table 6-6 List of values stored in Disk_Status	6-15
Table 6-7 List of provided sample programs	6-17
Table 7-1 Supported STOP Error Causes	7-2
Table 7-2 Event log entries recorded by this feature	7-3
Table 7-3 Log files	7-8
Table 8-1 Event log entries recorded by this feature	8-13

This page is intentionally left blank.

CHAPTER 1 CAPABILITIES OF THE RAS FEATURES

The HF-W series comes with Reliability, Availability, and Serviceability (RAS) features to provide high reliability. The following is an overview of the RAS features.

Table 1-1 Overview of RAS features

Category		Item
Monitoring		Hardware status monitoring
		OS deadlock monitoring
		Watchdog timer monitoring
GUI features settings		RAS Setup window
Status checks	GUI display	Hardware status window
	Notifications	Event notifications
		Pop-up notifications
		Remote notifications
		Status acquisition by using library functions
Control	Shutdown/startup suppression	Automatic shutdown
		Shutdown via library functions
		Controlling general-purpose external contacts
Library functions		RAS library
Maintenance/ failure analysis	Memory dumps	Memory dump collection
		Error cause notification with STOP error code
		Log information collection window
		Maintenance operation support commands
		Logging trends of the temperature inside the chassis
Simulation		Hardware status simulation

1. CAPABILITIES OF THE RAS FEATURES

Monitoring

(1) Hardware status monitoring

Monitors the hardware status of this equipment including the status of the storages as well as the temperature inside the chassis.

(2) OS deadlock monitoring

This feature monitors the operational state of the OS by using the dedicated timer implemented within this equipment.

(3) Watchdog timer monitoring

This feature monitors whether processes are scheduled properly by using the watchdog timer implemented within this equipment. This feature also offers a library for using the watchdog timer.

GUI features settings

(4) RAS Setup window

This window provides a graphical user interface for configuring RAS feature settings, including the conditions on automatic shutdown and the watchdog timer settings.

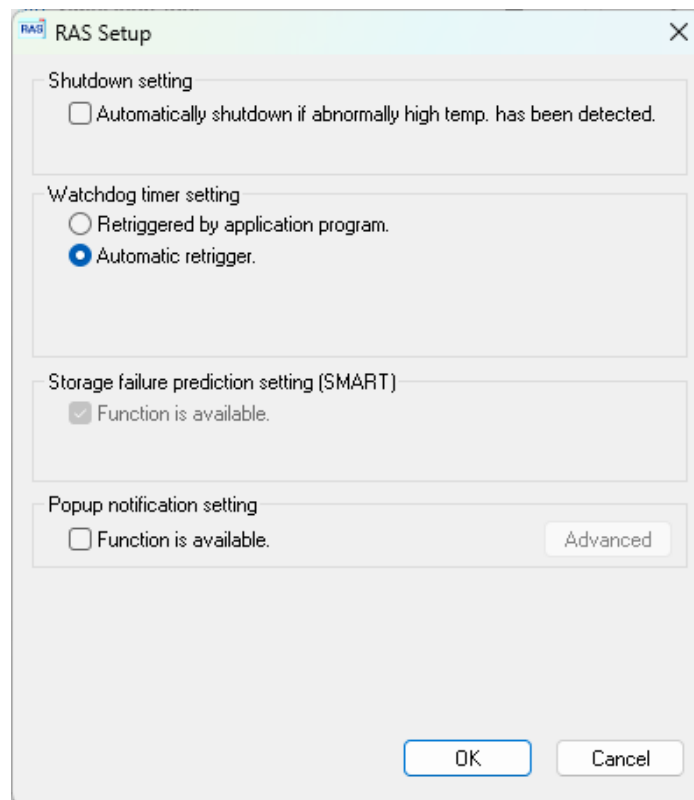


Figure 1-1 RAS Setup window

Status checks

(5) Hardware status window

This window displays the hardware status of this equipment via a graphical interface. An icon in the notification area of the taskbar displays the hardware status at all times.

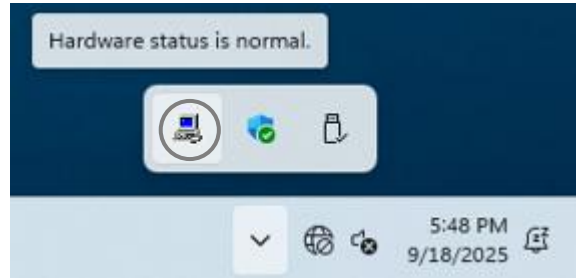


Figure 1-2 Hardware status icon

(6) Event notifications

This feature allows user applications to check the hardware status of this equipment by monitoring the status of event objects.

(7) Pop-up notifications

This feature displays pop-up messages to notify users that errors have occurred in the equipment hardware.

(8) Remote notifications

This feature enables a remote device to check the hardware status of the equipment. It also notifies the remote device whenever the hardware status changes.

(9) Status acquisition by using library functions

This feature allows user applications to obtain the hardware status of this equipment by using the RAS library.

Control

(10) Automatic shutdown

Automatically shuts down the equipment when abnormal temperature inside the chassis is detected.

Configure the settings shown in (4) *RAS Setup window* to enable or disable the automatic shutdown feature.

(11) Shutdown via library functions

You can shut down the equipment from a user application by using the RAS library.

1. CAPABILITIES OF THE RAS FEATURES

(12) Controlling general-purpose external contacts

This feature allows users to control the general-purpose external contacts by using the RAS library.

two input and two output general purpose external contacts are available for a user.

If you use these contacts, signals can be input from an external device to this equipment, and signals can be output from this equipment to an external device.

Library functions

(13) RAS library

This feature offers library functions for recording log information, in addition to the library functions offered by the status check (9) and control (10 and 12) items.

Maintenance/failure analysis

(14) Memory dump collection

This feature records the contents of the system memory in a file (a memory dump file) by The Watchdog timer times out or Keyboard operation (hold down Ctrl key and press Scroll Lock key twice), for example, after the equipment stops unexpectedly. By analyzing the data in this memory dump, you can investigate the cause of the failure.

(15) Error cause notification with STOP error code

This function detects a blue screen caused by an error with the STOP error code 0x80 and records the cause of the blue screen in the event log.

(16) Log information collection window

In this window you can use a graphical user interface to collect log data and memory dump files for this equipment.

(17) Maintenance operation support commands

These commands include a command used for saving failure information, such as memory dump files and event log files, to external media.

(18) Logging trends of the temperature inside the chassis

This feature periodically measures the temperature inside the chassis of this equipment and records the data in a file.

Simulation

(19) Hardware status simulation

This feature simulates the hardware status of this equipment. You can use this feature to test user applications and check the notification interface of the RAS software without an actual hardware failure having occurred.

This manual explains features (1) through (13), (15), (16), (18), and (19). For details about the other features, see the *HF-W200E INSTRUCTION MANUAL* (manual number WIN-62-5004).

This page is intentionally left blank.

CHAPTER 2 ITEMS MONITORED BY THE RAS FEATURES

This chapter describes the items monitored by the RAS features.

2.1 Monitoring Temperature inside the Chassis

The fan monitoring feature monitors the multiple fans located in this equipment, and notifies users when one of the fans malfunctions via the following methods:

- (1) Hardware status window
- (2) Event notifications
- (3) Pop-up notification
- (4) Remote notifications
- (5) Automatic shutdown
- (6) Alarm lamp

For details about (1) through (4), see *Chapter 4. CHECKING THE HARDWARE STATUS*. For details about (5), see *5.1 Automatic Shutdown of the Equipment*.

If the temperature abnormality of the equipment is recovered, the alarm lamp is turn off.

2.2 Storage Failure Prediction (SMART monitoring)

The Storages in this equipment have the Self-Monitoring, Analysis and Reporting Technology (SMART) feature, which continuously monitors the condition of the Storage and anticipates a failure before the failure manifests itself. The storage failure prediction function notifies in the following methods when there is a possibility that failure may occur in a storage in near future.

- (1) Hardware status window
- (2) Event notifications
- (3) Pop-up notifications
- (4) Remote notifications
- (5) hfwDiskStat function in the RAS library

For details about (1) through (4), see *Chapter 4. CHECKING THE HARDWARE STATUS*.

For details about (5), see *6.1.8 Get function for the storage condition (hfwDiskStat)*.

NOTICE

When failure of a storage is anticipated, the storage may experience hardware failure in the near future. We recommend you to back up the data and replace this equipment. Please contact the system administrator or maintenance personnel.

NOTE

- It is not possible for the Storage Failure Prediction Function to anticipate all failures. Therefore, a storage may fail before the Storage Failure Prediction Function anticipates any failures.
 - This function only monitors the internal storages that are recognized during OS startup. If you connect a new storage or replace this equipment, for example, for maintenance, it may take a long time to recognize the new storage at the first startup after the new storage is connected, and the storage may not be recognized as a storage to be monitored. If this situation happens, restart this equipment.
-

2.3 Memory Monitoring

Error Checking and Correcting (ECC) using In-Band technology is installed in this equipment. A single bit error in the memory can be automatically corrected without causing any problems to the operation of the equipment. On the other hand, if the frequency of memory error correction is high, that is considered to be caused by memory failure, and we recommend replacing this equipment from the viewpoint of preventive maintenance. Please contact the system administrator or maintenance personnel.

If the memory might have failed, the memory monitoring feature notifies users of this information via the following methods:

- (1) Event notifications
- (2) Pop-up notifications
- (3) Remote notifications
- (4) GetMemStatus function in the RAS library

For details about (1) through (3), see *Chapter 4. CHECKING THE HARDWARE STATUS*. For details about (4), see *6.1.7 Get function for the memory condition (GetMemStatus)*.

2.4 OS Deadlock Monitoring

The OS deadlock monitoring function uses the watchdog timer implemented in this device to monitor situations where a real-time priority process cannot operate for more than 60 seconds (hereinafter referred to as OS deadlock). If the OS deadlocks up due to a kernel crash or a driver monopolizing the CPU, a memory dump will be taken. For details on taking a memory dump, see *5.2 Memory Dump due to Timeout Detection*.

When using this function, select [Automatic retrigger] in the Watchdog Timer Settings in the RAS Setup window.

NOTICE
When the OS deadlocks, processes in the OS cannot run as scheduled, and the facility that is using this equipment might be affected due to delays in processing. If the OS deadlocks, resolve the problem immediately.

NOTE
For this feature, an OS deadlock is defined as a state where a process with the highest priority (real-time priority class) cannot run

2.5 Watchdog Timer Monitoring

This equipment has a watchdog timer. By retriggering a watchdog timer automatically, this feature can be used, to monitor the operation of the OS and a user program..

2.5.1 Automatic retriggering feature for a watchdog timer

This feature monitors the OS by having the RAS software's watchdog timer monitoring process automatically retrigger the timer.

For details about how to use this feature , see *2.4 OS Deadlock Monitoring*.

2.5.2 Using a watchdog timer to monitor user programs

When you use the watchdog timer for monitoring the operational state of a user program, the user program to be monitored must periodically retrigger the watchdog timer (that is, reset the timeout counter of the watchdog timer to the initial value).

Figure 2-1 shows an example of monitoring the operational state of a user program.

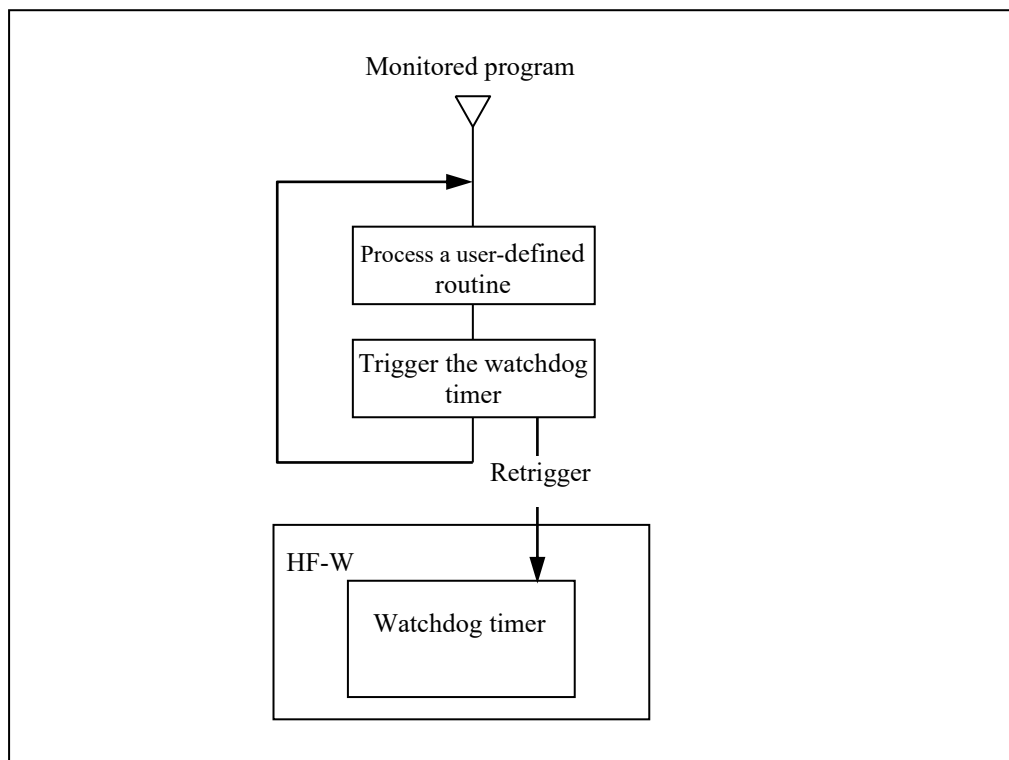


Figure 2-1 Example flowchart of monitoring the operational state of a user program

2. ITEMS MONITORED BY THE RAS FEATURES

The fact that a timeout occurred means that the monitored program was not able to retrigger the watchdog timer for at least the length of the timeout.

A program can use a watchdog timer by calling the library function `WdtControl`. For details about how to use the `WdtControl` function and how the WDTTO contact of the RAS external contact interface works, see 6.1.3 *Watchdog timer control function (WdtControl)*.

NOTE

- Make sure to stop the watchdog timer to prevent a timeout from occurring when terminating monitoring due to the termination of the user program or system shutdown.
 - When you want to use the `WdtControl` function, select Retriggered by application program under Watchdog timer setting in the RAS Setup window.
This will disable the automatic retriggering feature for the watchdog timer.
-

CHAPTER 3 SETTING UP THE RAS FEATURES

3.1 RAS Setup Window

3.1.1 Overview

In the RAS Setup window, you can set up the following features.

Table 3-1 Features that can be set up in the RAS Setup window

Item
Shutdown settings
Watchdog timer settings
Pop-up notification settings

Figure 3-1 shows the RAS Setup window. This figure shows the factory default settings.

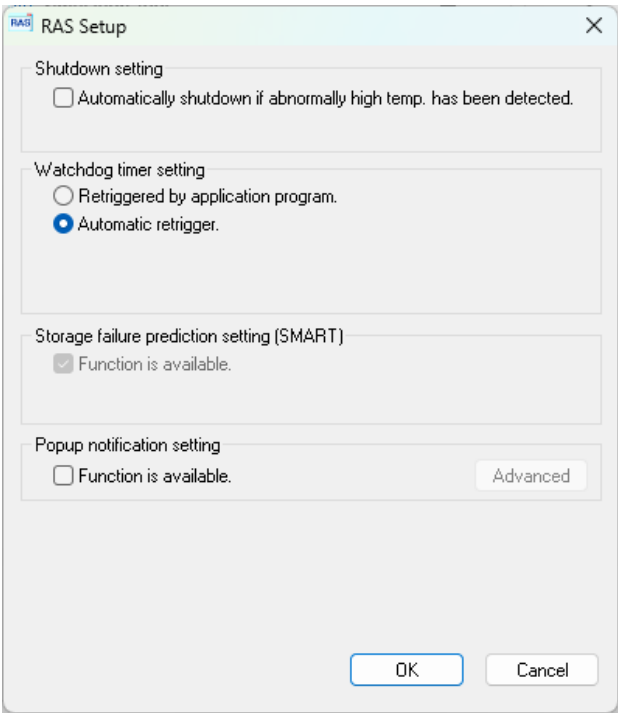


Figure 3-1 RAS Setup window

3. SETTING UP THE RAS FEATURES

3.1.2 Opening the RAS Setup window

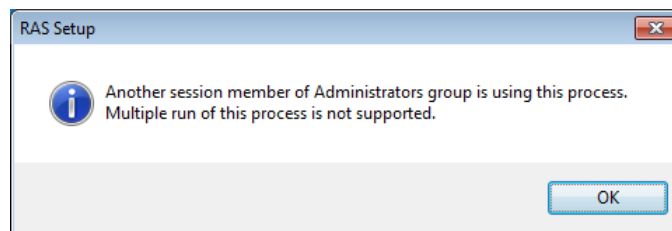
To open the RAS Setup window, perform the following procedure.

Before you open this window, you need to sign in to the computer as an administrator.

1. Click **Start**.
2. From the list of applications, click **RAS Software**.
3. Click **RAS Setup**.

NOTE

The RAS Setup window cannot be used by multiple users at the same time. If you use, for example, user switching to try to start instances of this window from multiple consoles, the following message appears. If you receive this message, close the RAS Setup window from other consoles, and then try opening the RAS Setup window again.



3.1.3 Using the RAS Setup window

(1) Shutdown settings

You can select whether this equipment is automatically shut down in each of the following cases: an abnormally high temperature.

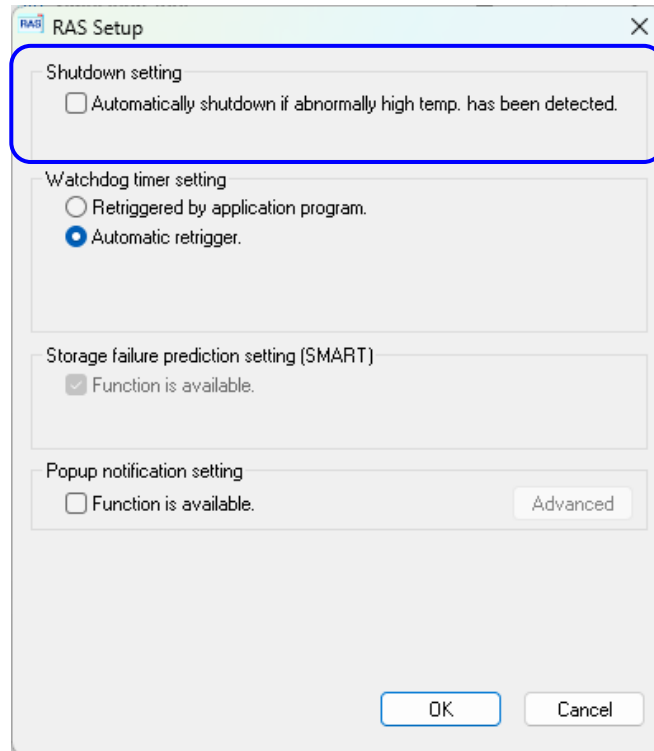


Figure 3-2 Shutdown settings items

- **Automatically shutdown if abnormally high temp. has been detected.** check box
 - Selected: This equipment will automatically shut down.
 - Cleared: This equipment will not automatically shut down (factory default setting).

NOTE

- After an automatic shutdown initiated by this feature is complete, the power turns off.
-

3. SETTING UP THE RAS FEATURES

(2) Watchdog timer settings

You can set up the watchdog timer of this equipment.

You can select one of the following ways of using the watchdog timer by clicking the radio button corresponding to the item

- **Retriggered by an application program.**
- **Automatic retrigger.**

The factory default setting is **Automatic retrigger**.

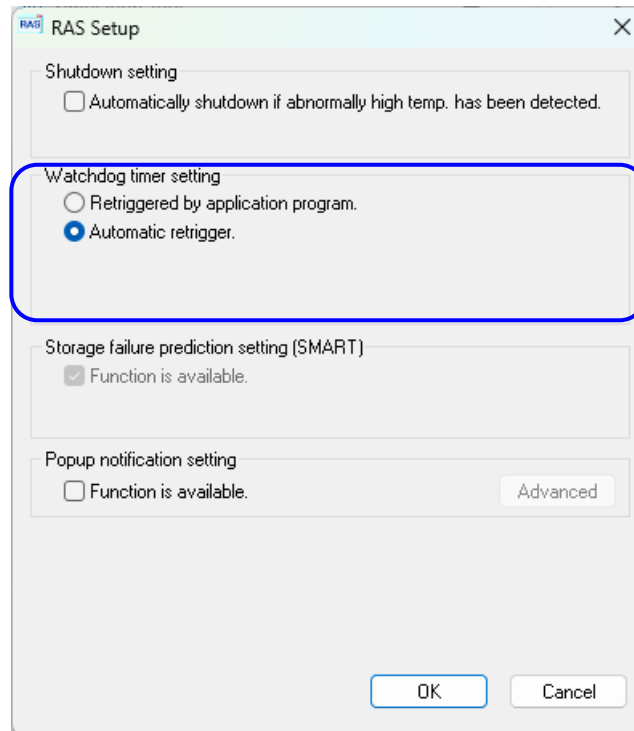


Figure 3-3 Items in the watchdog timer settings

- **Retriggered by application programs**

If you select this item, you can monitor the operational state of a user program by controlling the watchdog timer by using the `WdtControl` function of the RAS library.

NOTE

When you change the setting to **Retriggered by application program**, the watchdog timer stops once. When a user application retriggers the watchdog timer by using the `WdtControl` function of the RAS library, the watchdog timer resumes its countdown.

- **Automatic retrigger**

If you select this item, you can monitor the operation of the OS using the watchdog timer automatic retriggering feature. In addition, you cannot use the watchdog timer by calling the `WdtControl` function of the RAS library.

(3) Pop-up notification settings

You can set up the pop-up notification settings. Clicking **Advanced** allows you to configure the advanced settings of this feature.

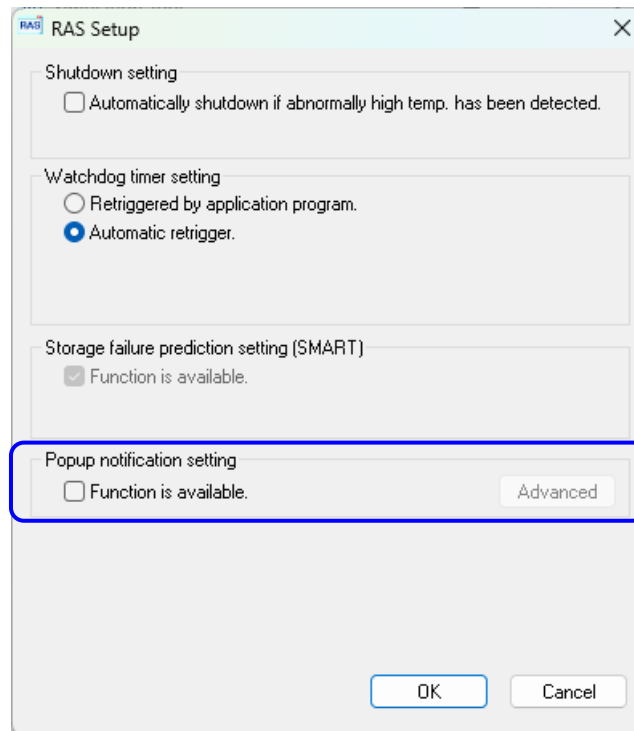


Figure 3-4 Items in the pop-up notification settings

- **Function is available** check box

- Selected: Pop-up notifications are enabled.
- Cleared: Pop-up notifications are disabled (factory default setting).

To change the current setting, click the check box.

If the **Function is available** check box is not selected, **Advanced** is grayed out and unavailable.

3. SETTING UP THE RAS FEATURES

- **Advanced** button

Click **Advanced** to display the following window.

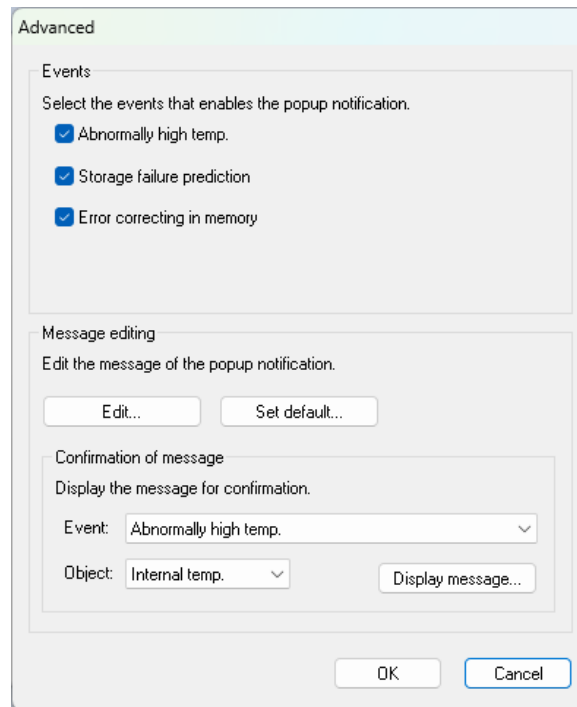


Figure 3-5 Advanced pop-up notification settings

Events

- **Abnormally high temp.**
- **Storage failure prediction**
- **Error correcting in memory**

You can disable or enable pop-up notifications for each of these items.

- Check box for each item
 - Selected: Pop-up notifications are enabled (factory default setting)
 - Cleared: Pop-up notifications are disabled.

To change the current setting, click the check box.

If the monitoring feature is disabled, the system will not send notifications even if pop-up notifications are enabled.

Message editing

You can edit the pop-up notification messages and check the messages after you edit them. For information about how to edit and check these messages, see *3.1.4 Editing pop-up notification messages*.

To apply any advanced settings that you changed, click **OK**. To discard the changes, click **Cancel**. The window then closes and the changes are lost.

(4) **Applying the changes you made in (1) through (3) or discarding the changes**

If you changed the settings in (1) through (3) and want to apply the changes, click **OK**. The RAS Setup window closes and the settings are applied immediately. To discard the changed settings, click **Cancel**. The RAS Setup window then closes and the changes are lost.

3. SETTING UP THE RAS FEATURES

3.1.4 Editing pop-up notification messages

(1) Editing pop-up notification messages

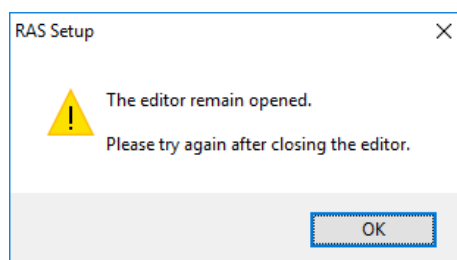
If you want to edit the messages that are used for pop-up notifications, click **Edit**. Notepad launches, and the message definition file for pop-up notifications is opened. Edit the messages in the following format.

NOTE

While you are editing messages, you cannot do the following:

- Click **Edit**.
- Click **Set default**.
- Click **Display message**.
- Close the RAS Setup window (by clicking **OK** or **Cancel**).

If you attempt to perform one of these actions, the following cautionary message appears.



If you click **OK** when this message appears, you return to the RAS Setup window.

■ Format of a message definition file

The format of a message definition file is as follows.

```

;-- Example of the definition of messages --
[TEMP]           ← Section
Line1=""
Line2=""
Line3="Temperature exceeded prescribed value."
                ← Key           ← Value
```

Figure 3-6 Format of a message definition file

A message definition file consists of sections, keys, and key values.

Each section contains keys and their values. A key and its value are separated by an equal sign (=).

Lines that start with a semicolon (;) are comment lines.

■ Descriptions in a message definition file

1. Section

The following table shows a list of section names that you can define for this feature, and an explanation of the message that you define for each section.

Table 3-2 Section names and defined messages

Section name	Defined message
[TEMP]	A message displayed when an abnormal temperature is detected inside the chassis.
[STR1-SMART]	A message displayed when a storage failure prediction (SMART) is detected for Storage.
[DIMM1-ERR]	A message displayed when error corrections are detected frequently in DIMM1.

2. Keys

For a key, specify the line number of the line displayed as a part of the pop-up message.

In this feature, you can use the keys Line1 through Line5 for each section.

If you specify keys other than Line1 through Line5, those keys are ignored.

3. Values

Specify one line of the message displayed as a part of the pop-up message for a value.

For each key, you can assign a maximum of 50 bytes of characters (up to 25 double-byte characters). If you specify more than 50 bytes of characters, the characters from the 51th byte onward are ignored.

If a line includes space characters, enclose the entire value in double quotation marks ("). An empty value is treated as a newline character.

NOTE

- When you save the changes, make sure you use **Save** in the menu. If you do not, the changes you made might not be saved properly.
 - While you are editing a message definition file, do not use another application to edit the same file. If you edit a message definition file from multiple applications at the same time, the changes might not be saved properly.
 - When you edit a pop-up notification message, make sure that the message clearly states that an error occurred. If you continue to use this equipment while an error persists, your system might be significantly affected.
-

3. SETTING UP THE RAS FEATURES

(2) Checking pop-up notification messages

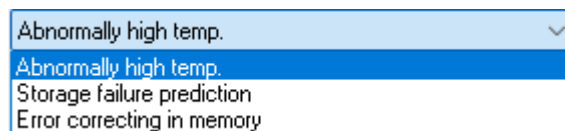
You can check the changes you made in the message for each of the following items:

- **Abnormally high temp.**
- **Storage failure prediction**
- **Error correcting in memory**

The following shows how to check changes you make to the messages.

1. From the **Event** list, select the event you want to check.

In this list, only the items under **Events** selected by using the check boxes appear. If no items under **Events** have been selected by using the check boxes, you cannot select an item from the event list.



Example of Selecting Abnormally high temp.

2. From the **Object** list, select the object that you want to check.

The items that appear in this list depend on the item that you selected in step 1.

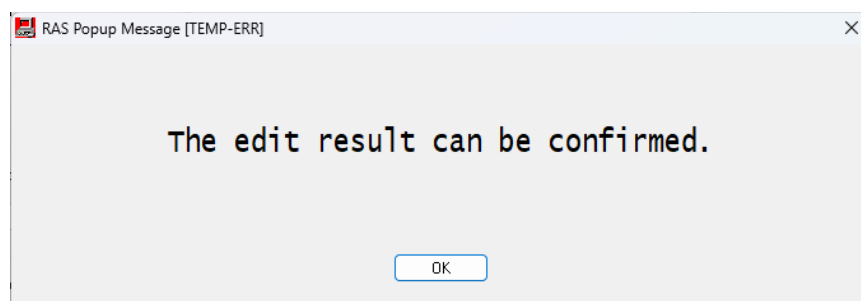
The following table shows items displayed in the **Object** list for each option you select in the **Event** list.

Table 3-3 Items displayed in the object list for each option selected in the event list

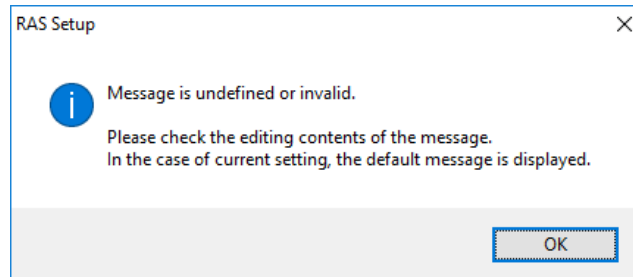
Option in the event list	Items in the object list
Abnormally high temp.	Temperature inside the chassis
Storage failure prediction	M.2 SSD1
Error correction in memory	DIMM1

3. Click **Display message**.

A pop-up notification message appears based on the changes you made. After you confirm the message, click **OK**.

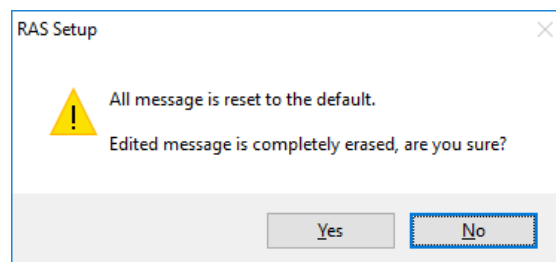


If the message has not been edited or there is something wrong in the message definition file, the following message appears. Click **OK** to go back to the RAS Setup window. Correct the changes that you made in the message.



(3) Restoring the default messages

If you want to return the pop-up notification messages to the default values, click **Set default**. The following message appears. Click **Yes** to discard the changes you made in the message definition file.



If you click **No**, the changes are not deleted, and the messages do not return to the default values.

This page is intentionally left blank.

CHAPTER 4 CHECKING THE HARDWARE STATUS

You can check the hardware status of this equipment by using the following methods:

(1) Checking by using a GUI

You can check the hardware status of this equipment by using a graphical interface. For details, see *4.1 Hardware Status Window*.

(2) Checking by using a user application

A user application can check the hardware status of this equipment by monitoring the status of event objects. For details, see *4.2 RAS Event Notifications*.

A user application can also obtain the hardware status of this equipment by using the RAS library. For details, see *4.5 Status Acquisition by Using the RAS Library*.

(3) Checking on the desktop of this equipment

A pop-up message appears to notify users when errors occur in the hardware of this equipment. For details, see *4.3 Pop-up Notifications*.

(4) Checking from a remote device

A remote device can check the hardware status of this equipment. A remote device can also be notified whenever the hardware status changes. For details, see *4.4 Remote Notifications*.

4.1 Hardware Status Window

4.1.1 Overview

After you sign in to this equipment, there will always be an icon in the notification area of the taskbar to display the hardware status. If you double-click or right-click this icon to display a pop-up menu and then click **Display Hardware status**, detailed information about the hardware status of this equipment is displayed.

This window displays the following information:

- Condition of the temperature inside the chassis
- Storage failure prediction (SMART monitoring) condition

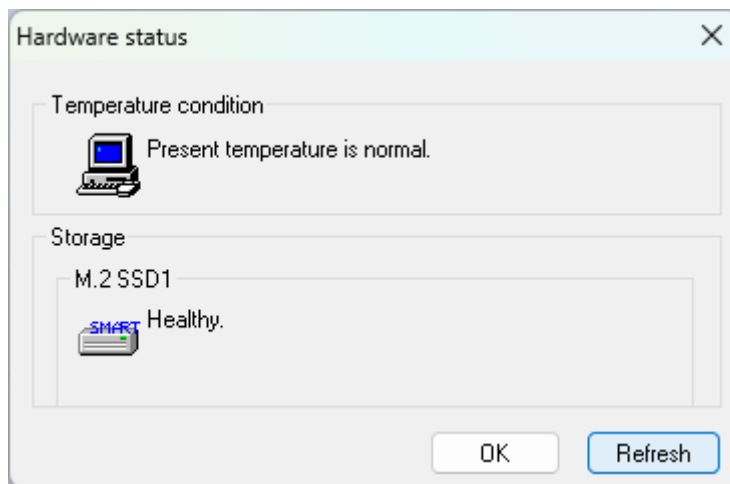


Figure 4-1 Hardware status window

NOTICE

When the **Hardware status** window shows an error in the hardware, resolve the problem causing the error immediately.

NOTE

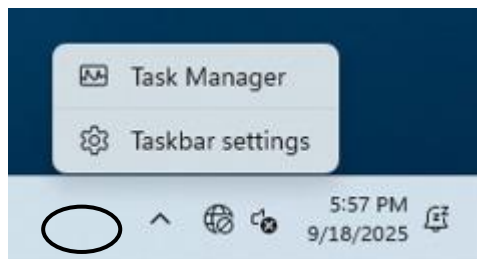
- This window displays only the internal Storages that are recognized at OS startup. If you connect a new Storages or replace a Storage with a new Storage, for example, for maintenance, it might take a long time to recognize the new Storage the first time the equipment starts up after the new Storages is connected, and information about the Storages might not be displayed. If such cases, restart this equipment.

4.1.2 Hardware status icon

There will always be an icon in the notification area of the taskbar to display the hardware status.



Note that, when you are using the factory default settings, the icon does not appear. If you click the arrow at the side of the notification area, the icon appears. If you want to always have the icon appear in the notification area of the taskbar, right-click the task bar, click **Taskbar Settings** in the displayed menu, click **Other System tray icons** in the displayed window, and then turn the icon on for **envdisp MFC Application**.

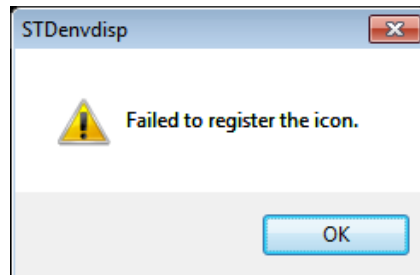


Right-click the taskbar to display the menu.

4. CHECKING THE HARDWARE STATUS

NOTE

In rare cases, registering the hardware status icon to the taskbar fails, and the following message appears. If this happens, perform the following procedure to retry the registration of the hardware status icon.



1. Click **OK** in the dialog box.
 2. Click **Start**.
 3. In the list of applications, click **RAS Software**, and then click **RAS Status**
-

- (1) List of displayed icons and description of each icon

Table 4-1 shows a list of displayed icons and a description of each icon. A description of the displayed icon appears when you point the mouse to an icon.

Table 4-1 Hardware status icon

No.	Hardware status	Icon	Description of the icon
1	Normal		The hardware status is normal.
2	Error		Abnormal TEMP detected.
3			Abnormal TEMP detected. Storage failure possible.
4	Caution		Storage failure possible.

No. 4: If a hardware status error is detected at the same time, the icon for a hardware status error is displayed.

4. CHECKING THE HARDWARE STATUS

Figures 4-2 and 4-3 show examples of displaying the description of an icon when the hardware status of this equipment is normal and when the hardware status has an error.

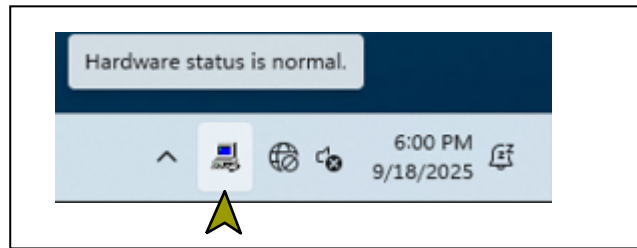


Figure 4-2 Example of displaying the description of an icon (when the hardware status is normal)

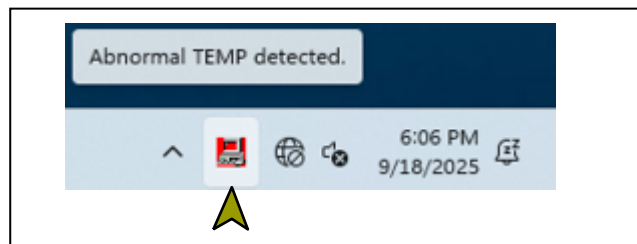


Figure 4-3 Example of displaying the description of an icon (when the hardware status has an error)

(2) Menu of the hardware status icon

If you right-click the icon, a pop-up menu appears.

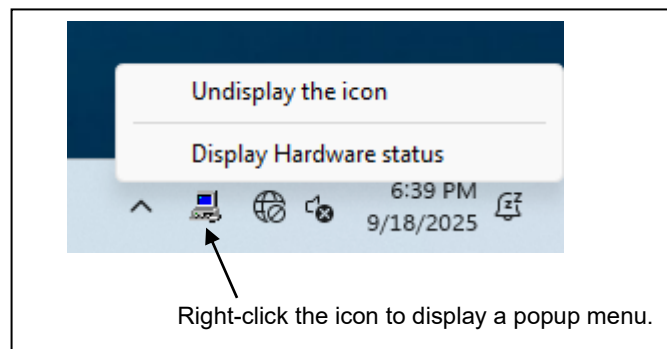


Figure 4-4 Menu of the hardware status icon

- **Display hardware status**

Click this to display the **Hardware status** window.

- **Undisplay the icon**

Click this to delete the icon from the notification area of the taskbar.

4.1.3 Hardware status window

The **Hardware status** window shows the details of the hardware status of this equipment.

Figure 4-5 shows how to open the **Hardware status** window.

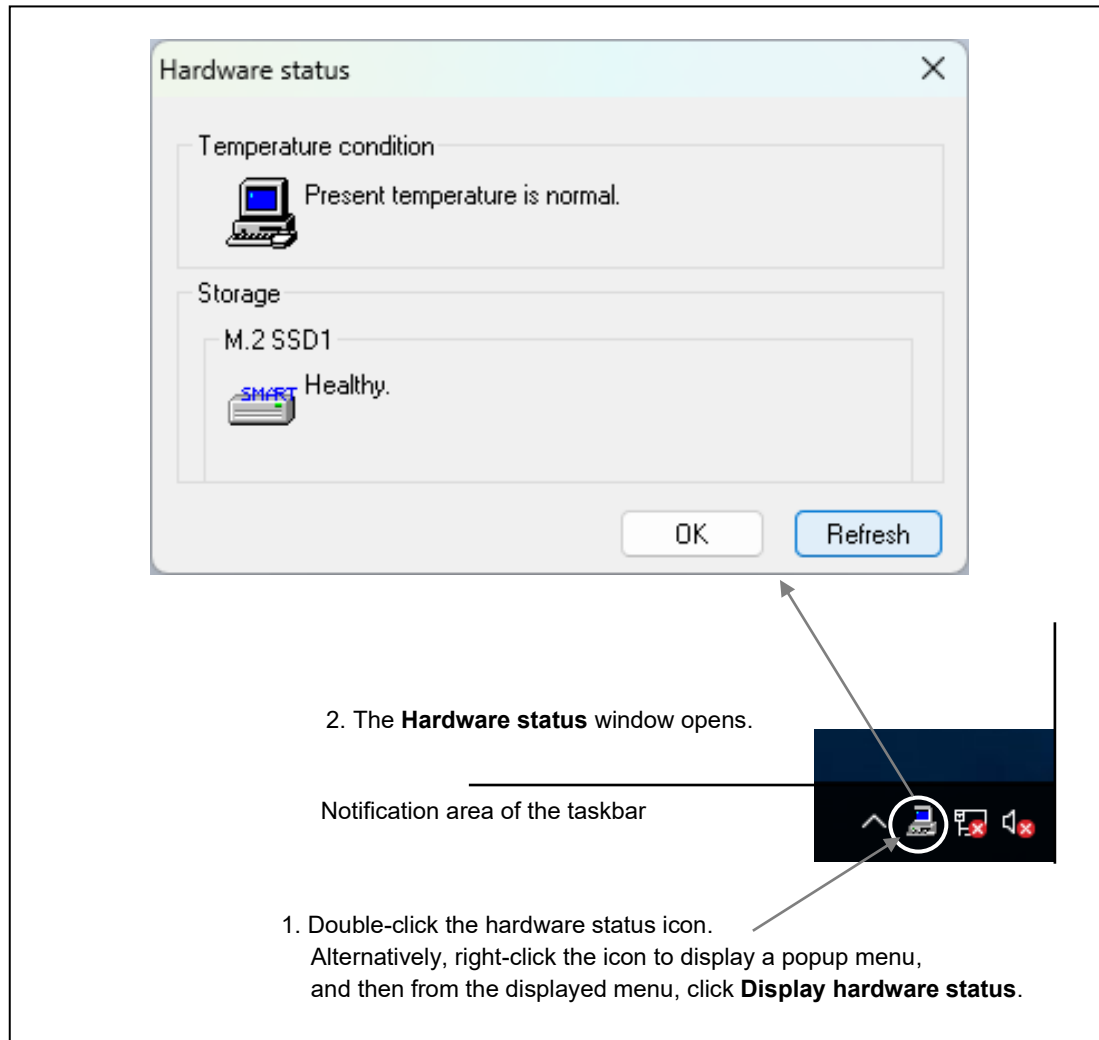


Figure 4-5 Starting the Hardware status window

4. CHECKING THE HARDWARE STATUS

(1) Description of the window

1. Temperature condition

This shows the current status of the temperature inside the chassis.

Table 4-2 Temperature condition and displayed information

Temperature status	Icon	Information
Normal		Present temperature is normal.
Abnormally high temperature		Temperature in the unit has exceeded the limit.

3. Storage condition

This shows the current status of the Storages. In the following areas, the Storage conditions are displayed.

Table 4-3 Storage condition and displayed information

No.	Storage condition	Icon	Information
1	Normal		Healthy.
2	Failure prediction by SMART		A failure might be imminent.

NOTICE

When failure of a storage is anticipated, the storage may experience hardware failure in near future. We recommend you to back up the data and replace this equipment. Please contact the system administrator or maintenance personnel.

8. Refresh button

If you click this button, the latest hardware status is acquired, and the information in the window is refreshed.

9. OK button

Click this button to close the **Hardware status** window

Figure 4-6 shows the **Hardware status** window when there is a hardware status error.

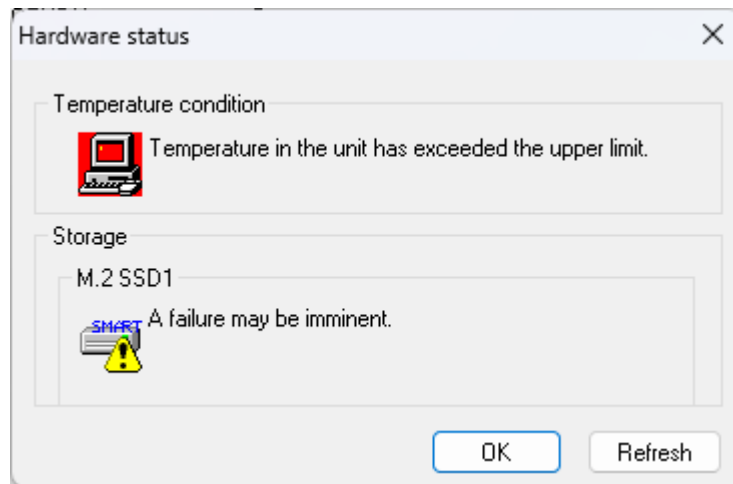


Figure 4-6 Hardware status window (example display in the case of an error)

4.2 RAS Event Notifications

4.2.1 Overview

When an event that must be reported to a user occurs, such as a hardware failure, this feature notifies an application of the event by setting an event object to the signaled state.

The application can detect event such as hardware failures by monitoring when the event objects are set to the signaled state.

Event objects are reset to the nonsignaled state when the cause of the event is cleared.

4.2.2 Detecting an event

To detect an event:

1. Use the `OpenEvent` Windows API function to get the handle to the event object.

Specify `SYNCHRONIZE` for the parameter `dwDesiredAccess` (the access to the event object).

2. Use the `WaitForSingleObject` or `WaitForMultipleObject` Windows API function to monitor when the event object is set to the signaled state.

Table 4-4 lists the events to be reported to a user and their respective event objects.

Table 4-4 Reported events

No.	Event	Event object name
1	The temperature inside the chassis became abnormal.	W2KRAS_TEMP_ERR_EVENT
2	SMART anticipated a failure in one of the Storage.	W2KRAS_HDD_PREDICT_EVENT
3	Frequent error corrections occurred in one of the memory slots.	HFW_MEMORY_ERR_EVENT

NOTE

When you use an event object in a program, you need to add `Global¥` to the beginning of the name of the event object.

4.2.3 Example of using event objects

We provide a sample program in C (`MemErr.c`) to show how to monitor event objects. For the name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

4.3 Pop-up Notifications

4.3.1 Overview

When an event that must be reported to a user occurs, such as a hardware failure, this feature notifies a user of the event by displaying a pop-up message on the desktop. By using this feature, a user can identify that an event such as a hardware failure occurred.

More specifically, a pop-up message is displayed in the following cases:

- Abnormally high temp
- Storage failure prediction
- Error correcting in memory

Figure 4-7 shows an example pop-up message notification when a system fan failure has occurred.

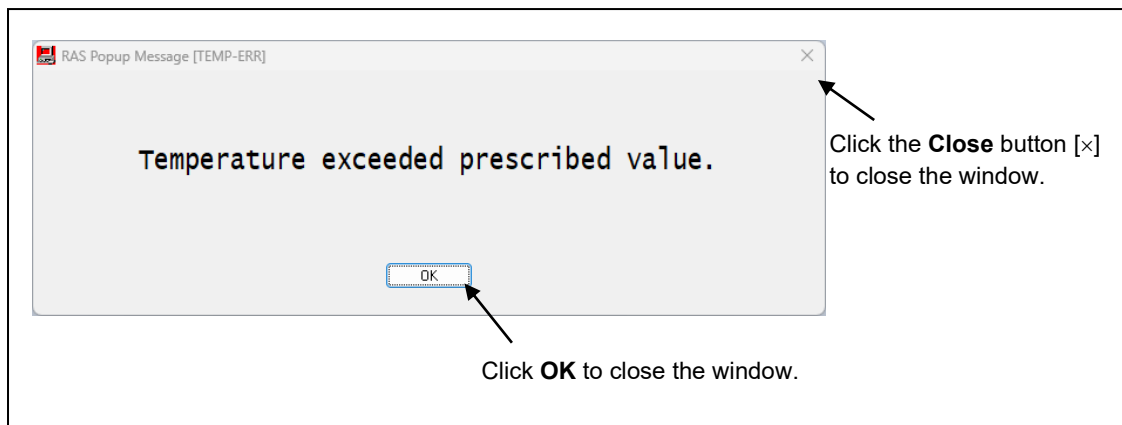


Figure 4-7 Example pop-up message notification

4. CHECKING THE HARDWARE STATUS

4.3.2 Messages to be displayed

Table 4-5 shows a list of the pop-up notification messages that this feature outputs.

You can edit these messages. For details about editing messages, see *3.1.4 Editing pop-up notification messages*.

Table 4-5 Displayed messages

No.	Event	Pop-up notification message
1	The temperature inside the chassis became abnormal.	Temperature exceeded prescribed value.
2	SMART anticipated a storage failure.	A failure may be imminent on the storage of the M.2 SSD1
	Frequent error corrections occur in memory.	In the DIMM 1, error corrections have occurred with high frequency.

4.3.3 Pop-up notification settings

This feature can be enabled or disabled in the RAS Setup window. In the factory default settings, this feature is disabled. If this feature is disabled, pop-up messages are not displayed.

For details, see *3.1.3 Using the RAS Setup window*.

4.4 Remote Notifications

4.4.1 Overview

If you use this feature from a remote device, you can check hardware conditions over the network that, without this feature, can only be checked when you are nearby this equipment. With this feature, even when hardware conditions cannot be checked nearby this equipment because, for example, the system administrator is away from this equipment or this equipment is built into the facility, the hardware conditions can be checked from a remote device.

This feature uses the Simple Network Management Protocol (SNMP) to notify users of hardware conditions. This allows you to use commercially available network management software that supports SNMP and to monitor distributed instances of this equipment and other devices, all from a single location.

NOTE

- Remote notifications use SNMP, a protocol in the application layer of the TCP/IP, and User Datagram Protocol (UDP) in the transport layer. This means that if the network load is high, the hardware conditions might not be received properly.
 - Remote notifications use the SNMP service, a standard feature of Windows®. For details about how to enable the standard Windows® SNMP service, see *4.4.3 Enabling remote notifications*.
-

4. CHECKING THE HARDWARE STATUS

4.4.2 Hardware conditions that can be acquired by using remote notifications

The following hardware conditions and settings can be acquired from a remote device:

- Temperature condition inside the chassis
- Storage condition
- Memory condition
- RAS feature settings
- Operational mode (normal mode)
- Version information of the extended Management Information Base (MIB) for HF-W

The following transitions of the hardware conditions are sent as notifications to users by using a trap.

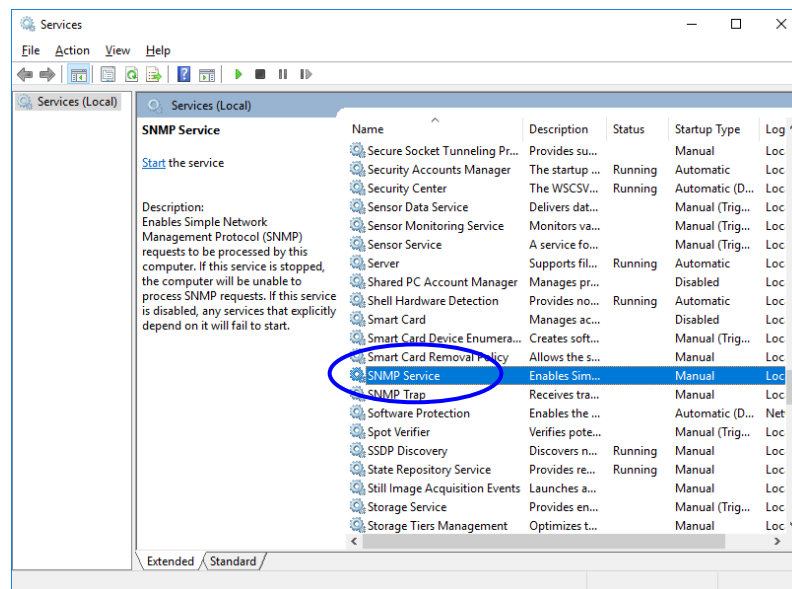
- (1) Temperature condition inside the chassis
 - Normal → Error
 - Error → Normal
- (2) Storage condition
 - Normal → Failure anticipated
- (3) Memory condition
 - Normal → Frequent error corrections
 - Frequent error corrections → Normal
- (4) Operating mode
 - HF-W stopped → Started in normal mode
 - Running in normal mode → Running in simulation mode

4.4.3 Enabling remote notifications

This feature is disabled when you are using the factory default settings. Remote notifications use the standard Windows® SNMP service. If you enable the SNMP service, remote notifications are enabled.

When using remote notifications, perform the following procedure to enable the SNMP service:

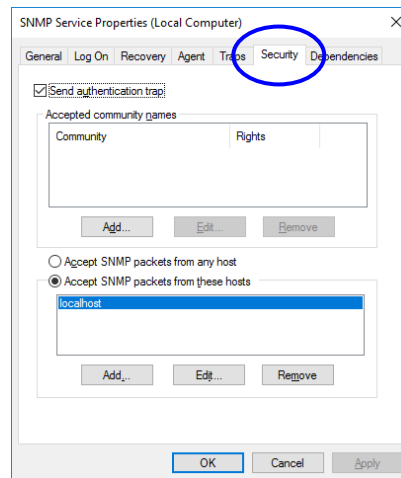
- (1) Opening the SNMP Service Properties window
 1. If you have not signed in to the computer as an administrator, do so.
 2. Open the Services window:
 - Open the **Control Panel**.
 - Double-click **System and Security, Windows Tools, and Services**.
 3. Double-click **SNMP Service** to open the SNMP Service Properties window.



4. CHECKING THE HARDWARE STATUS

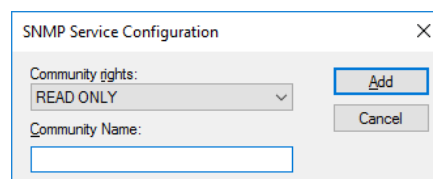
(2) SNMP security configuration

1. In the SNMP Service Properties window, click the **Security** tab.



2. If you want to send a trap message whenever authentication fails, select the **Send authentication trap** check box.

3. Under **Accepted community names**, click **Add**. The SNMP Service Configuration window is displayed. In the **Community rights** list, select **READ ONLY**. In the **Community Name** box, enter the community name that you want to use, and then click **Add**.



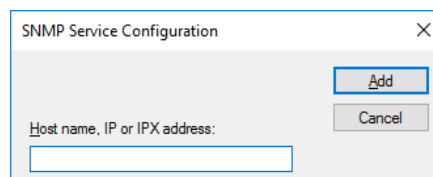
4. Specify whether to accept SNMP packets from hosts.

If you want to accept SNMP packets from any manager on the network:

- Select **Accept SNMP packets from any host**.

If you want to restrict SNMP packets:

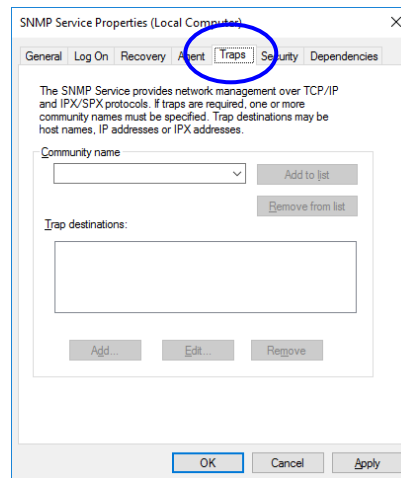
- Select **Accept SNMP packets from these hosts**.
- Click **Add**.
- The SNMP Service Configuration window appears. Enter the host name or the IP or IPX address of the host that you want to accept SNMP packets from, and then click **Add**.



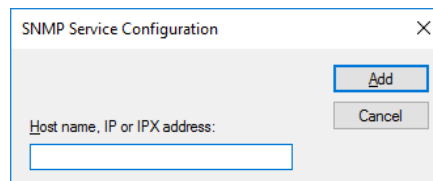
5. In the SNMP Service Properties window, click **Apply**.

(3) SNMP trap configuration

1. In the SNMP Service Properties window, click the **Traps** tab.



2. Under **Community name**, enter the name of the community that trap messages are to be sent to, and then click **Add to list**.
3. Under **Trap destinations**, click **Add**. The SNMP Service Configuration window is displayed. Enter the host name or the IP or IPX address of a destination that you want to send traps to, and then click **Add**.

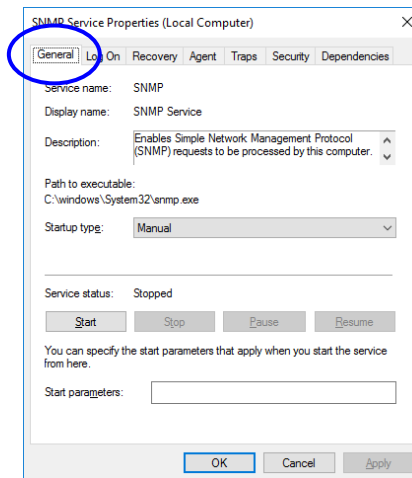


4. In the SNMP Service Properties window, click **Apply**.

4. CHECKING THE HARDWARE STATUS

(4) Starting the SNMP service

1. In the SNMP Services Properties window, click the **General** tab.



2. Click **Start**. The SNMP service starts and remote notifications for hardware statuses are enabled.
3. To start the SNMP service automatically upon the next OS startup, in the **Startup type** list, select **Automatic**.
4. In the SNMP Service Properties window, click **OK**.

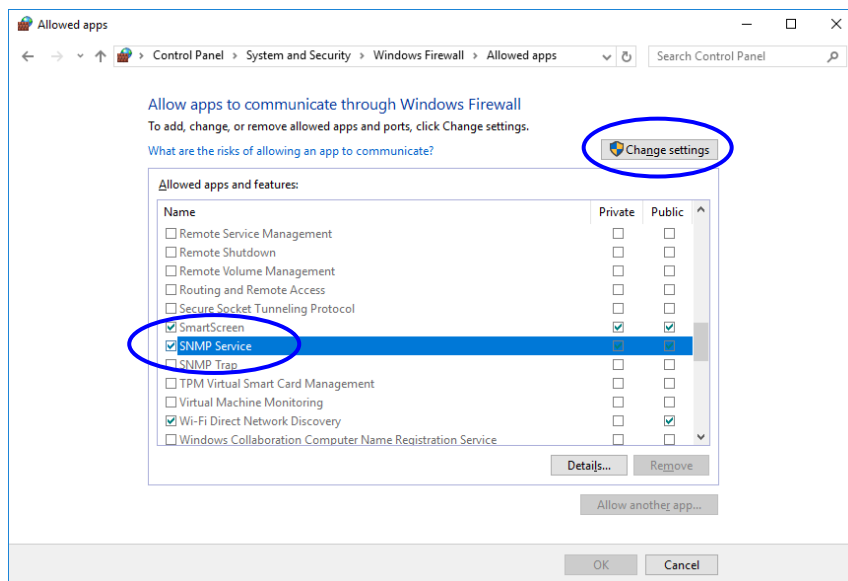
NOTE

- If, when the SNMP service starts, there is an error for which notifications by using a trap is configured, the trap is sent when the SNMP service starts.
- If Windows Firewall is configured to block the SNMP service, you cannot acquire the hardware status of a remote device. If the firewall is set up to block the SNMP service, perform the following procedure to undo the setting.

Note, however, that the SNMP service can pass through the Windows Firewall by default, so in that case you do not have to perform the following procedure:

1. If you have not signed in to the computer as an administrator, do so.
2. Open the **Control Panel**, and then click **System and Security**.
3. Click **Allow an app through Windows Defender Firewall** of **Windows Defender Firewall**.

4. The **Allowed apps** window appears. Click **Change settings**, and under **Allowed apps and features**, select the **SNMP Service** check box.



5. Click **OK**.

4. CHECKING THE HARDWARE STATUS

4.4.4 Objects in the extended MIB for HF-W

To acquire the hardware status of this equipment from a remote device, use the extended MIB for HF-W. This section provides a list of objects defined in the extended MIB for HF-W and a description of these objects.

(1) Objects related to the hardware conditions and settings

Table 4-6 lists the objects related to the hardware status and a description of these objects. The object ID of each object is obtained either by replacing the following *x* with the object in the following table or replacing the following *y* with the object number in the table.

For *N* in the object number, specify a value in the range from 1 to the number of monitoring targets.

For *z*, specify the index number acquired from each entry.

However, for the extended MIB for HF-W, the value of *N* specified for acquisition of an index number is the same as the index number that can be acquired. (If 1 is specified for *N*, the index number is 1.)

Object ID: .iso.org.dod.internet.private.enterprises.Hitachi.systemExMib.
hfwExMib.hfwRasStatus.*x* (where *x* is an object in the following table)

or

.1.3.6.1.4.1.116.5.45.1.*y* (where *y* is an object number in the following table)

Table 4-6 Objects related to the hardware status

(1/3)

No.	Object	Object number	Description	Description of the values
1	hfwTemp.tempNumber	2.1.0	Number of monitored temperatures	--
2	hfwTemp.tempTable.TempEntry.tempIndex	2.2.1.1. <i>N</i>	Index number of tempEntry	--
3	hfwTemp.tempTable.TempEntry.tempName	2.2.1.2. <i>z</i>	Name of the temperature to be monitored	Internal temperature: Temperature inside the chassis
4	hfwTemp.tempTable.TempEntry.tempStatus	2.2.1.3. <i>z</i>	Temperature condition	1: Normal 2: Error
5	hfwDrv.drivNumber	3.1.0	Number of monitored storage	--
6	hfwDrv.drivTable.drivEntry.drivIndex	3.2.1.1. <i>N</i>	Index number of drivEntry	--
7	hfwDrv.drivTable.drivEntry.drivStatus	3.2.1.2. <i>z</i>	Storage condition	1: Healthy 3: Failure anticipated 99: Unknown
8	hfwDrv.drivTable.drivEntry.drivType	3.2.1.4. <i>z</i>	Storage type	2: SSD 99: Unknown
9	hfwDrv.drivTable.drivEntry.drivUsedEndurance	3.2.1.5. <i>z</i>	Percentage of data writes to the storage (unit: %)	0~100
10	hfwMem.memNumber	5.1.0	Number of monitored memory slots	--
11	hfwMem.memTable.memEntry.memIndex	5.2.1.1. <i>N</i>	Index of memEntry	--

No.	Object	Object number	Description	Description of the values
12	hfwMem.memTable.memEntry.memName	5.2.1.2.z	DIMM name	DIMM1: DIMM1 slot
13	hfwMem.memTable.memEntry.memStatus	5.2.1.3.z	Memory condition	1: Normal 2: Error (Frequent error corrections occur or a possible failure is detected).

No. 1: For this equipment, the value is set to 1.

No. 5: The number of internal storage that can be installed in the HF-W is set for the number of monitored storages .
For this equipment, the value is set to 1

No. 10: The number of memory slots is set for the number of monitored memory slots.
For this equipment, the value is set to 1.

4. CHECKING THE HARDWARE STATUS

Table 4-7 shows a list of the objects related to the RAS feature settings and a description of those objects. The object ID of each object is obtained either by replacing the following *x* with the object in the following table or replacing the following *y* with the object number in the table.

Object ID: .iso.org.dod.internet.private.enterprises.Hitachi.systemExMib.
hfwExMib.hfwRasSetting.*x* (where *x* is an object in the following table)
or
.1.3.6.1.4.1.116.5.45.2.*y* (where *y* is an object number in the following table)

Table 4-7 Objects related to the RAS feature settings

No.	Object	Object number	Description	Description of the values
2	hfwTempAutoShutdown	2	Automatic shutdown when the temperature is abnormal	1: Enabled 2: Disabled
3	hfwSmartEnableSetting	4	Storage failure prediction (SMART monitoring) settings	1: Enabled 2: Disabled

Table 4-8 shows a list of the objects related to operating modes and a description of those objects. The object ID of each object is obtained either by replacing the following *x* with the object in the following table or replacing the following *y* with the object number in the table.

Object ID: .iso.org.dod.internet.private.enterprises.Hitachi.systemExMib.
hfwExMib.hfwRasInfo.*x* (where *x* is an object in the following table)
or
.1.3.6.1.4.1.116.5.45.3.*y* (where *y* is an object number in the following table)

Table 4-8 Objects related to operating modes

No.	Object	Object number	Description	Description of the values
1	hfwRasMode	1	Operating mode	1: Normal mode 2: Simulation mode

Table 4-9 shows a list of the objects related to the version information of the extended MIB for HF-W and a description of those objects. The object ID of each object is obtained either by replacing the following *x* with the object in the following table or replacing the following *y* with the object number in the table.

Object ID: .iso.org.dod.internet.private.enterprises.Hitachi.system.
 hfw.hfwExMibInfo.*x* (where *x* is an object in the following table)
or
 .1.3.6.1.4.1.116.3.45.1.*y* (where *y* is an object number in the following table)

Table 4-9 Objects related to the extended MIB for HF-W

No.	Object	Object number	Description	Description of the values
1	Version	1	Version number of the extended MIB for HF-W	--
2	Revision	6	Revision number of the extended MIB for HF-W	--

4. CHECKING THE HARDWARE STATUS

(2) Objects related to trap notifications

Table 4-10 shows a list of the objects related to trap notifications when an error occurs, as well as a description and notification data for those objects. The enterprise ID for the trap notification when an error occurs is as follows:

```
Enterprise ID: .iso.org.dod.internet.private.enterprises.Hitachi.systemAP.  
             hfwMibTrap.hfwRasErrorTrap  
  
             or  
  
             .1.3.6.1.4.1.116.7.45.1
```

Table 4-10 Objects related to the trap notification (when an error occurs)

No.	Object	Trap number	Description	Notification data	
				Object used	Value
1	hfwTempError	2	The temperature inside the chassis becomes abnormal.	tempName	Internal temperature
				tempStatus	2: Error
				hfwTempStMsg	Internal temperature exceeded prescribed value.
2	hfwSmartDetect	3	A failure is anticipated for the Storage.	drvIndex	The Storage number of the Storage for which a failure is anticipated by SMART
				drvStatus	3: Failure anticipated
				hfwSmartStMsg	A failure may be imminent on the storage of the M.2 SSD%1.
3	hfwMemError	6	Frequent error corrections occur.	memName	Name of the memory slot with frequent error corrections
				memStatus	2: Error (frequent error correction)
				hfwMemStMsg	In the %2, error correcting have occurred with high frequency.

No. 2 %1 denotes the Storage number.

No. 3 %2 denotes the name of the memory slot for which error corrections occur frequently or for which a possible memory failure was detected.

Table 4-11 shows a list of the objects related to the trap notification when the equipment has recovered from an error and a description of those objects. The enterprise ID for the trap notification when the equipment has recovered from an error is as follows:

```
Enterprise ID: .iso.org.dod.internet.private.enterprises.Hitachi.systemAP.  
             hfwMibTrap.hfwRasRecoverTrap  
  
             or  
  
             .1.3.6.1.4.1.116.7.45.2
```

Table 4-11 Objects related to trap notifications (when the equipment has recovered from an error)

No.	Object	Trap number	Description	Notification data	
				Object used	Value
1	hfwTempRecover	2	Recovery from abnormal temperature in the chassis	tempName	Internal temperature
				tempStatus	1: Normal
				hfwTempStMsg	Internal temperature returned to prescribed value.
2	hfwMemRecover	6	Recovery from frequent error corrections	memName	Name of the recovered memory slot
				memStatus	1: Normal
				hfwMemStMsg	In the %I, frequency of the error correctings deteriorated.

No. 1: %2 denotes the name of the memory slot that recovered from frequent error corrections.

Table 4-12 shows a list of the objects related to trap notifications when the equipment starts in normal mode and a description of those objects. The enterprise ID for the trap notifications related to operating modes is as follows:

```
Enterprise ID: .iso.org.dod.internet.private.enterprises.Hitachi.systemAP.
               hfwMibTrap.hfwRasInfoTrap
               or
               .1.3.6.1.4.1.116.7.45.3
```

Table 4-12 Objects related to trap notifications (operating modes)

No.	Object	Trap number	Description	Notification data	
				Object used	Value
1	hfwRasServiceStarted	1	Startup in normal mode	hfwRasMode	1: Normal mode
				hfwRasStartMsg	RAS Service is running.
2	hfwSimulationModeStarted	2	Transition to simulation mode	hfwRasMode	2: Simulation mode
				hfwRasStartMsg	RAS Service switched to Simulation Mode.

4.4.5 Extended MIB file for HF-W

The extended MIB file for HF-W is as follows:

Extended MIB file for HF-W: %ProgramFiles%\HFWRAS¥mib¥hfwExMib.mib

4. CHECKING THE HARDWARE STATUS

4.5 Status Acquisition by Using the RAS Library

You can use the RAS library to acquire the following hardware conditions. For details about the RAS library, see *6.1 RAS Library*.

- To acquire the memory condition: Use the `GetMemStatus` function.
- To acquire the storage condition: Use the `hfwDiskStat` function.

CHAPTER 5 CONTROLLING THE HARDWARE

The RAS features can exercise the following controls over this equipment:

(1) Automatic shutdown of the equipment

When a hardware error occurs the equipment can automatically shut down. For details, see *5.1 Automatic Shutdown of the Equipment*.

(2) Memory dump capture due to timeout detection

Memory dump due to timeout detection When a watchdog timer timeout occurs, a memory dump can be generated. For details, see *5.2 Memory Dump due to Timeout Detection*.

(3) Controlling the hardware by using the RAS library

A user application can control the hardware of this equipment by using the RAS library. For details, see *5.3 Controlling the Hardware by Using the RAS Library*.

5.1 Automatic Shutdown of the Equipment

This feature automatically shuts down the equipment when running the equipment would pose a danger when abnormally high temperature is detected. By doing so, you can protect internal parts such as a processor from thermal degradation and prevent thermal runaway of the system due to malfunction of this equipment.

5.1.1 Automatic shutdown when detecting an abnormally high temperature

When the temperature sensor in this equipment detects that the temperature is abnormally high inside the chassis, the equipment automatically shuts down.

- This feature can be enabled or disabled in the RAS Setup window. When using the factory default settings, this feature is disabled. For details, see *3.1.3 Using the RAS Setup window*.
- Alternatively, a user application can detect abnormally high temperatures by using a RAS event and shut down the equipment. For details about RAS events, see *4.2 RAS Event Notifications*.

NOTE

- If the temperature inside the chassis is high, parts can be degraded rapidly due to heat. Then it is not advisable to continue to use this equipment also in terms of the life span of the equipment. On the other hand, if the temperature is abnormally high when the fans are working properly, the abnormally high temperature must be caused by an external factor such as malfunctioning air conditioning at the location of the equipment. You can isolate the cause of the abnormally high temperature while the equipment is running. Because of this, this feature is disabled in the default factory-shipped setting.
 - If you continue to operate this equipment after abnormally high temperature is detected and the temperature further rises to a dangerous level, the equipment is forcibly shut down and the power is turned off regardless of whether this feature is disabled.
-

5.2 Memory Dump due to Timeout Detection

A memory dump is generated when a watchdog timer timeout in the equipment is detected.

If a memory dump is generated by this feature, the equipment is automatically restarted.

The STOP error code when a memory dump is generated is as follow.

- STOP error code: 0x00009222

NOTE

A memory dump may not be generated depending on the situation, for example, when Windows stops responding due to an interrupt with a high interrupt request level (IRQL).

5.3 Controlling the Hardware by Using the RAS Library

By using the RAS library functions, you can shut down the system and control the general-purpose external contacts and the digital LEDs for status indications. For details about the library functions, see *6.1 RAS Library*.

- To shut down the system: Use the `BSSysShut` function.
- To control the watchdog timer: Use the `WdtControl` function.
- To control the general purpose-external contact outputs: Use the `GendoControl` functions.
- To control the general-purpose external contact inputs: Use the `GetGendiN` functions.

This page is intentionally left blank.

CHAPTER 6 LIBRARY FUNCTIONS

A user application can get and control the hardware status of this equipment by using the library functions provided by the RAS features.

For information about the hardware specifications of the RAS external contact interface described in this chapter and how to use each contact, see the *HF-W200E INSTRUCTION MANUAL* (manual number WIN-62-5004).

6.1 RAS Library

6.1.1 Overview

This chapter describes the interface for the library functions provided by the RAS features.

Table 6-1 lists the RAS library functions.

Table 6-1 RAS library functions

No.	Function name	Use	DLL
1	BSSysShut	Shuts down the equipment.	w2kras.dll
2	WdtControl	Retrigger and stops the watchdog timer.	
3	GendoControlN	Controls the external general purpose outputs.	
4	GetGendiN	Gets the status of the external general purpose inputs.	
5	MconWriteMessage	Records a specified message (characters) in the log files of this equipment.	
6	GetMemStatus	Gets the condition of the memory built into this equipment.	hfwras.dll
7	hfwDiskStat	Gets the condition of the storage.	

These functions are offered by the DLLs (w2kras.dll and hfwras.dll).

NOTE

Do not copy or move w2kras.dll and hfwras.dll to another directory. If you do so, the RAS features of this equipment cannot run properly.

6. LIBRARY FUNCTIONS

The functions offered by `w2kras.dll` can be called from Visual Basic®, which requires .NET. When you call the functions numbered 1, 2, and 5 from Visual Basic®, add `_VB` to the end of the name of each function. The function parameters are the same. For example, when you call the `WdtControl` function from Visual Basic®, use the function name `WdtControl_VB`.

The following files are provided as import libraries:

```
%ProgramFiles%\HFWRAS\lib\w2kras.lib
```

```
%ProgramFiles%\HFWRAS\lib\hfwras.lib
```

When you use a library, link the corresponding import library.

The following files are provided as header files for the libraries:

```
%ProgramFiles%\HFWRAS\include\w2kras.h
```

```
%ProgramFiles%\HFWRAS\include\hfwras.h
```

When you use a library in C, include the corresponding header file.

6.1.2 Shutdown function (BSSysShut)

Name

BSSysShut - System shutdown

Syntax

```
#include <w2kras.h>
int BSSysShut(reboot-argument)
int reboot; /*Reboot flag*/
```

Description

BSSysShut shuts down the system.

The reboot argument is used to specify whether to reboot the system after a shutdown.

reboot-argument = 0: The power to this equipment turns off after a shutdown.

reboot-argument $\neq$ 0: The system reboots after a shutdown.

Diagnosis

0: Successful completion (system shutdown processing has started)

1: Shutdown privilege acquisition error

2: Internal error (OS shutdown failed)

Sample program

We provide a sample program that uses this function in C. For the name of the sample program and information about where you can find it, see 6.2 *Sample Programs*.

6.1.3 Watchdog timer control function (WdtControl)

(1) Function interface

Name

WdtControl - Watchdog timer control and status acquisition

Syntax

#include <w2kras.h>

BOOL WdtControl(DWORD *dwCmd*, PDWORD *pdwCount*);

Description

This function performs the action specified by *dwCmd* on the watchdog timer.

To use this function, in the RAS Setup window, select **Retriggered by application program.** under

Watchdog timer setting. If the watchdog timer setting is different, this function terminates with an error. If you call the GetLastError Windows API function, the error code W2KRAS_WDT_NONMANUAL is returned. The following describes the parameters of this function.

dwCmd:

This parameter specifies the action to perform on the watchdog timer. The following options are available for this parameter.

Table 6-2 Actions for WdtControl that are specified by dwCmd

dwCmd	Explanation of the action
WDT_SET (0x00)	Specifies the timeout value in seconds.
WDT_STOP (0x01)	Stops the watchdog timer.

If a value other than these is specified, the function terminates with an error. If you call the GetLastError Windows API function, the error code W2KRAS_INVALID_PARAMETER is returned.

pdwCount:

If *dwCmd* is WDT_SET, you can configure the timeout value of the watchdog timer by using the variable pointed to by *pdwCount* and calling this function.

Specify a value from 1 to 63 in seconds. If a value outside this range is specified, the function terminates with an error. If you call the GetLastError Windows API function, the error code W2KRAS_INVALID_PARAMETER is returned.

When this function returns, the value of the variable pointed to by *pdwCount* is undefined. Do not use the value.

If *dwCmd* is WDT_STOP, the value of *pdwCount* is ignored. When this function returns, the value of the variable pointed to by *pdwCount* is undefined. Do not use the value.

NOTE

If you want to stop monitoring using the watchdog timer when a user program exits or due to shutdown, you must stop the Watchdog timer so that a timeout does not occur.

Diagnosis

If this function ends successfully, the function returns `TRUE`. If the function terminates with an error, the function returns `FALSE`.

If this function terminates with an error, call the `GetLastError` Windows API function to get the error code.

The error codes returned by this function itself are as follows:

Error code (value)	Description
<code>W2KRAS_INVALID_PARAMETER</code> (0x2001)	The specified parameters contain an error.
<code>W2KRAS_WDT_NONMANUAL</code> (0x2002)	This function cannot be used because Retriggered by application program. is not selected under Watchdog timer setting in the RAS Setup window.
<code>W2KRAS_NOT_INITIALIZE</code> (0x2005)	Startup of the RAS software is not yet complete.
<code>W2KRAS_INTERNAL_ERROR</code> (0x2007)	An internal error occurred.

Other error codes come from the Windows API functions used by this function. For details about these error codes, see the Windows API Help.

Sample program

We provide a sample program that uses this function in C. For the name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

For information about how to monitor the operational state of a user program by using the watchdog timer, see *2.5.2 Using a watchdog timer to monitor user programs*.

6.1.4 Control function for the general-purpose external contact outputs (GendoControlN)**Name**

GendoControlN - Control for the external general purpose outputs

Syntax

```
#include <w2kras.h>

BOOL GendoControlN(UCHAR ucOutput, UCHAR ucMask);
```

Description

This function controls external general purpose outputs (GENDO0, GENDO1). The parameters of this function are explained below.

ucOutput:

Sets the output level to the outputs. Table 6-3 shows how bits are allocated to each output. To set a output to the low level, set the bit to “0”. To set the output to the high level, set the bit to “1”.

ucMask:

Specifies a output to be controlled. The bit allocation is the same as *ucOutput* as shown in Table 6-3. If the parameter is to be controlled, set the bit to “1”, Otherwise, set the bit to “0”.

Table 6-3 Bit Allocation of *ucOutput* and *ucMask* for GendoControlN

bit0	GENDO0
bit1	GENDO1
bit2	Unused
bit3	Unused
bit4	Unused
bit5	Unused
bit6	Unused
bit7	Unused

If a value is specified for bit2 to bit7, this function will terminate abnormally, and when the Windows API GetLastError function is called, the error code W2KRAS_INVALID_PARAMETER will be returned.

Diagnosis

If this function ends successfully, the function returns `TRUE`. If the function terminates with an error, the function returns `FALSE`.

If this function terminates with an error, call the `GetLastError` Windows API function to get the error code.

The error codes returned by this function itself are as follows:

Error code (value)	Description
<code>W2KRAS_INVALID_PARAMETER</code> (0x2001)	The specified parameters contain an error.
<code>W2KRAS_INTERNAL_ERROR</code> (0x2007)	An internal error occurred.

Other error codes come from the Windows API functions used by this function. For details about these error codes, see the Windows API Help.

Explanation

The `GendoControlN` function sets the output state of the output with *ucOutput* and specifies a control target with *ucMask*. Figure 6-1 shows an operation example to illustrate the relationship between *ucOutput* and *ucMask*.

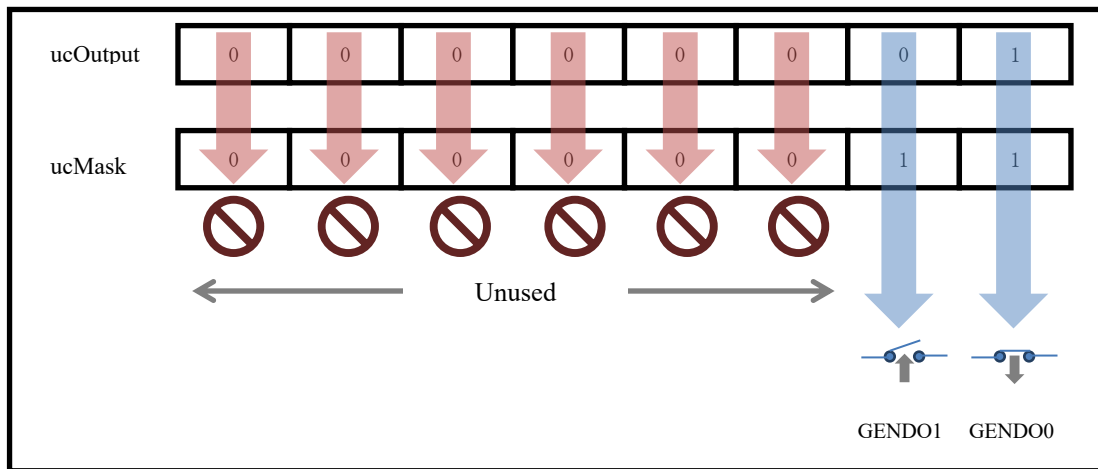


Figure 6-1 Operation Example of `GendoControlN` Function

6. LIBRARY FUNCTIONS

Supplementary information

Figure 6-2 shows the operation of the external general purpose output (GENDO0) when the GendoControlN function is used. The dashed lines show the general purpose output level, and the bold line show the transition of the external general purpose output.

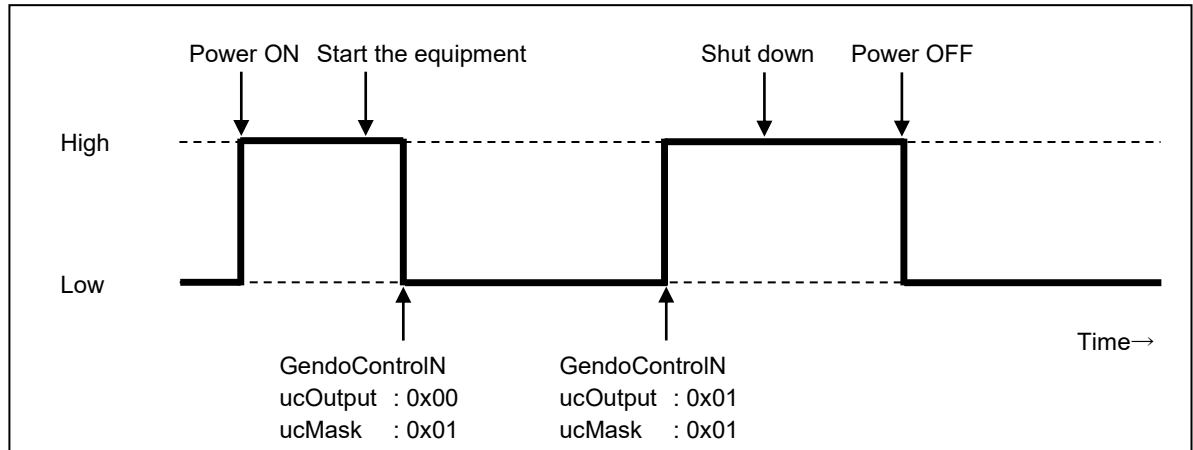


Figure 6-2 Operation Example of the External general purpose output

Sample program

We provide a sample program that uses this function in C. For information about the name of the sample program and where you can find it, see *6.2 Sample Programs*.

6.1.5 Get function for the general-purpose external contact inputs (GetGendiN)

Name

GetGendi - Status acquisition of the general-purpose external contact input

Syntax

```
#include <w2kras.h>

BOOL GetGendiN(PUCHAR pucInput) ;
```

Description

This function acquires the input state of external general purpose inputs (GENDI1, GENDI2). The parameters of this function are explained below.

pucInput:

The input state of external general purpose input is stored. Table 6-4 shows how bits are allocated to each general purpose input. If the input level of a general purpose input is low, “0” is stored to the corresponding bit, and “1” otherwise.

Table 6-4 Bit Allocation of pcInput for GetGendiN

bit0	GENDI1
bit1	GENDI2
bit2	Unused
bit3	Unused
bit4	Unused
bit5	Unused
bit6	Unused
bit7	Unused

6. LIBRARY FUNCTIONS

Diagnosis

If this function completes successfully, the function returns `TRUE`. If this function terminates with an error, the function returns `FALSE`.

When this function terminates with an error, call the `GetLastError` Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
<code>W2KRAS_NOT_INITIALIZE</code> (0x2005)	Startup of the RAS software is not yet complete.
<code>W2KRAS_INTERNAL_ERROR</code> (0x2007)	An internal error occurred.

Other error codes come from the Windows API functions used by this function. For details about these error codes, see the Windows API Help.

Sample program

We provide a sample program that uses this function in C. For information about the name of the sample program and where you can find it, see *6.2 Sample Programs*.

6.1.6 Log function (MConWriteMessage)

Name

MConWriteMessage - Logging

Syntax

```
#include <w2kras.h>

VOID WINAPI MConWriteMessage(LPSTR lpBuffer) ;
```

Description

The MConWriteMessage function writes the specified message (characters) to a log file. The file name is `hfwrasa.log` or `hfwrasb.log`.

The message is written along with the time stamp.

Two log files are available, and the size of each file is 64 KB. When the size of the log recorded in the log file that is currently being used exceeds 64 KB, the log file used for logging switches to the other log file.

The following describes the parameters of this function.

lpBuffer:

This parameter specifies a pointer to a string that contains the message (characters) to be written.

To easily identify the application that records each log entry, specify a message that starts with the name of the application, as shown in the following example.

Checking log information

This function records log information in text format in the following files. When the size of the log recorded in the log file that is currently being used exceeds 64 KB, the log file used for logging switches to another log file.

- %ProgramFiles%\HFWRAS¥log¥hfwrasa.log
- %ProgramFiles%\HFWRAS¥log¥hfwrasb.log

You can check log information by opening the files described here by using an application such as Notepad.

The format of the log information is as follows:

YYYY/MM/DD hh:mm:ss - Specified log information YYYY/MM/DD hh:mm:ss - Specified log information YYYY/MM/DD hh:mm:ss - Specified log information :	YYYY : Year MM : Month DD : Day hh : Hour (24-hour clock) mm : Minute ss : Second
--	--

Figure 6-3 Format of log information

Initially, each of the files contains an EOF character (ASCII code 0x1a).

Sample program

We provide a sample program that uses this function in C. For the name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

NOTICE

The log function exits asynchronously without waiting for data to actually be written to a log file. This means that this function does not return an error even when writing to a log file fails for some reason. We recommend recording important information in the OS event log.

NOTE

- This function has the same name as the message console function provided by the Hitachi software W2K-PLUS, but this function does not output to the message console.
 - To reduce the amount of resources used, this function, for example, opens and closes a pipe every time the function is called. This means that this function has a relatively large overhead. Even when you are recording multiple message lines, use one function call to output the message.
 - This function does not support Unicode strings. Always use ANSI strings. Log entries for messages are stored in text files. In these text files, the combination of the two characters `¥r¥n` is interpreted as a newline.
If you want to include a newline character in a string specified by *lpbuffer*, insert `¥r¥n` in the string.
-

6.1.7 Get function for the memory condition (GetMemStatus)

Name

GetMemStatus - Memory status acquisition

Syntax

```
#include <w2kras.h>

BOOL GetMemStatus(PMEM_DATA pMemData);
```

Description

The GetMemStatus function sets the condition of the memory in this equipment in the structure pointed to by *pMemData*. The following describes the parameters of this function.

pMemData:

This parameter specifies a pointer to a MEM_DATA structure that stores the acquired memory condition.

```
typedef struct MEM_DATA {
    int      Dimm_Number;           //Number of DIMM slots in this equipment
    DWORD    Dimm_Status[4];       //Condition of each DIMM
} MEM_DATA, *PMEM_DATA;
```

When this function ends successfully, the maximum number of DIMMs(1) is stored in Dimm_Number. Each element of Dimm_Status stores a value described in the following table. Note that the number of valid elements is Dimm_Number. The values of elements other than Dimm_Status[0] are undefined for this model. Do not use these values.

Table 6-5 List of values stored in Dimm_Status

Value	Description
MEMORY_NOMAL (0x00)	The memory is working properly.
MEMORY_ERR_DETECT (0x01)	Frequent error correction occurs.
MEMORY_NOT_MOUNTED (0x02)	No DIMMs are installed.

6. LIBRARY FUNCTIONS

For this model, the correspondence between the `Dimm_Status` elements and DIMM names is as follows:

Element	DIMM name
<code>Dimm_Status[0]</code>	DIMM1

Diagnosis

If this function ends successfully, the function returns `TRUE`. If the function terminates with an error, the function returns `FALSE`. When this function terminates with an error, the value stored in `pMemData` is invalid. If this function terminates with an error, call the `GetLastError` Windows API function to get the error code. The error codes returned by this function itself are as follows:

Error code (value)	Description
<code>W2KRAS_INVALID_PARAMETER</code> (0x2001)	The specified parameters contain an error.
<code>W2KRAS_NOT_INITIALIZE</code> (0x2005)	Startup of the RAS software is not yet complete.
<code>W2KRAS_MEMST_INVALID</code> (0x2007)	The memory condition cannot be acquired.

Other error codes come from the Windows API functions used by this function. For details about these error codes, see the Windows API Help.

Sample program

We provide a sample program that uses this function in C. For the name of the sample program and information about where you can find it, see 6.2 *Sample Programs*.

6.1.8 Get function for the storage condition (hfwDiskStat)

Name

hfwDiskStat - Storage status acquisition

Syntax

```
#include <hfwras.h>
BOOL hfwDiskStat(PHFW_DISK_STATUS phfwDiskStatus) ;
```

Description

The hfwDiskStat function sets the storage conditions in a structure pointed to by phfwDiskStatus.

The following describes the parameters of this function.

phfwDiskStatus:

This parameter specifies a pointer to an HFW_DISK_STATUS structure that stores the storage conditions.

```
typedef struct HFW_DISK_STATUS{
    DWORD    Disk_Count;
    DWORD    Disk_Status[16];      //Storage condition
} HFW_DISK_STATUS, *PHFW_DISK_STATUS;
```

When this function is successful, Disk_Count will contain the number of internal storage devices (1) for this device. Each element of Disk_Status will be set to the value shown in the table. At this time, the number of valid elements will be the value returned by Disk_Count. The values of elements other than Disk_Status[0] will be undefined for this device, so do not reference them. Also, the upper 16 bits of Disk_Status are for future use, so their values will be undefined and should not be referenced.

Table 6-6 List of values stored in Disk_Status

Disk_Status	Description
DISKSTAT_HEALTHY (0x00000001)	The storage is working properly.
DISKSTAT_SMART (0x00000008)	Storage failure prediction (SMART) is detected.
DISKSTAT_UNKNOWN (0x00000020)	The storage condition could not be acquired.

6. LIBRARY FUNCTIONS

Diagnosis

If this function ends successfully, the function returns `TRUE`. If the function terminates with an error, the function returns `FALSE`. When this function terminates with an error, the value stored in *phfwDiskStatus* is invalid.

If this function terminates with an error, call the `GetLastError` Windows API function to get the error code.

The error codes returned by this function itself are as follows:

Error code (value)	Description
<code>HFWRAS_INVALID_PARAMETER</code> (0x20000001)	The specified parameters contain an error.
<code>HFWRAS_NOT_INITIALIZE</code> (0x20000002)	Startup of the RAS software is not yet complete.
<code>HFWRAS_INTERNAL_ERROR</code> (0x20000003)	An internal error occurred.

Other error codes come from the Windows API functions used by this function. For details about these error codes, see the Windows API Help.

Sample program

We provide a sample program that uses this function in C. For the name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

6.2 Sample Programs

Sample programs file in C that use the RAS library functions are stored in the %ProgramFiles%\HFWRAS\sample directory. Use these files for reference when you develop a program or to verify the operation of the functions.

Table 6-7 lists the sample programs.

Table 6-7 List of provided sample programs

No.	File name	Description
1	shutd.c	Sample program for the BSSysShut function
2	wdt.c	Sample program for the WdtControl function
3	GendoControlN.c	Sample program for the GendoControlN function
4	GetGendiN.c	Sample program for the GetGendiN function
5	MCon.c	Sample program for the MconWriteMessage function
6	MemErr.c	Sample program for the GetMemStatus function
7	hfwDiskStat.c	Sample program for the hfwDiskStat function

This page is intentionally left blank.

CHAPTER 7 FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

7.1 Notifying the Cause of the STOP Error Code

7.1.1 Overview

This equipment records the memory contents in memory dump files when the system is forcibly recovered from OS lockup, or uncorrectable memory error occurs. At the same time, a blue screen appears and an identifier (NMI_HARDWARE_FAILURE) is displayed.

This function detects a blue screen caused by an error with the STOP error code 0x80 and records the cause of the blue screen in the event log after restart this equipment.

NOTE

These causes may not be recordable in the event log if you perform an action (for example, shutdown) that aborts the process of the STOP error code cause analysis. In the case of a 16GB memory dump file, analysis of the STOP error code takes about two minutes.

7. FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

7.1.2 Supported causes of the STOP error code

This function is activated when a blue screen appears due to a cause described in Table 7-1. Causes other than those in Table 7-1 are not supported.

Table 7-1 Supported STOP Error Causes

Cause	Contents recorded in the Description column of the event log ^{#1}
Watchdog timer time out	WDT timed out. The detailed code is 0x9222.
Uncorrectable memory error occurs	Uncorrectable Error at DIMM1. The detailed code is 0x9218.

#1: The event log is shown in Table 7-2.

For information about the actions to take when a blue screen appears, see Chapter 9 in the *HF-W200E INSTRUCTION MANUAL* (manual number WIN-62-5004).

7.1.3 Event log

Table 7-2 shows a list of the event log entries that this feature records. These event log entries are recorded in the system log.

Table 7-2 Event log entries recorded by this feature

Event ID	Source	Type	Category	Description
800	HFWRAS_SYS	Information	HFWRAS	%1 Detailed code = %2

Note: When an event log entry is recorded, %1 contains a message, and %2 contains the code corresponding to the message in %1.

%1	%2
WDT timed out.	0x9222
Uncorrectable Error at DIMM1.	0x9218

7.2 Log Information Collection Window

7.2.1 Overview

In the log information collection window, you can perform the following actions by using the graphical user interface:

(1) Collecting log data

This feature saves the data used for preventive maintenance and post-failure analysis of a problem. The data is compressed and saved as a single file with the file name `logsave.zip`.

(2) Collecting memory dump files

This feature collects memory dump files saved by the OS. The data is saved as a compressed file with the file name `memory.zip`. At the same time, the minimum memory dump files are also collected.

NOTICE

The CPU load increases while memory dump files are being collected. While the CPU is under a high load, the operation of user applications can be disturbed. Make sure that you do not collect memory dump files by using the log information collection window while applications for business use are running on this equipment.

7.2.2 Starting the log information collection window

To start the log information collection window, perform the following procedure:

Administrator privileges are required to use this window, so sign in to the computer as an administrator and open the window.

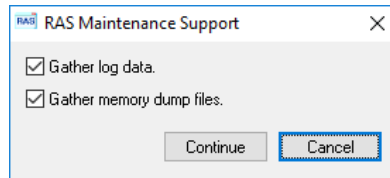
1. Click **Start**.
2. Click **RAS Software**.
3. Click **RAS Maintenance Support**.

NOTE

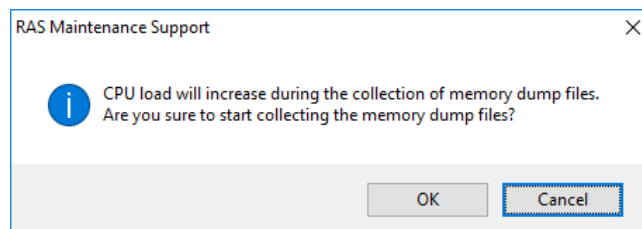
The log information collection window cannot be used by multiple users at the same time. If you use user switching, for example, to try to start instances of the log information collection window from multiple consoles, an error might occur. In such cases, close the log information collection window on the other consoles, and then open the window.

7.2.3 Using the log information collection window

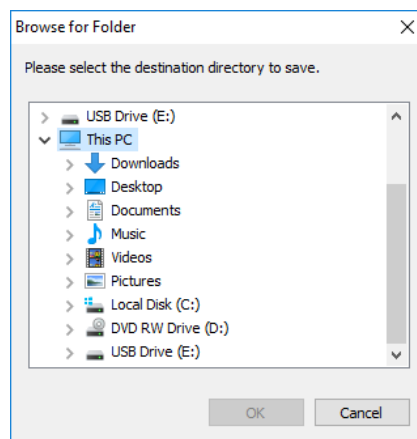
1. The log information collection window appears. By default, both the **Gather log data.** and **Gather memory dump files.** check boxes are selected. If you do not need to use either of one of these two options, clear the check box for the option that you do not need, and then click **Continue**.



2. If the **Gather memory dump files.** check box is selected, the following message appears. Click **OK**. If you click **Cancel**, you return to the log information collection window without executing the maintenance operation.



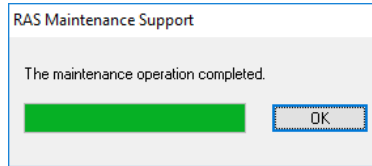
3. The following dialog box appears. Specify the directory in which to save, and then click **OK**. If you want to cancel the maintenance operation, click **Cancel**. If you click **Cancel**, you go back to the log information collection window without executing the maintenance operation.



7. FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

4. The information selected in step 1 is collected. During the process, a window appears to show the progress. If the process finishes successfully, the following window appears.

Do not do anything in the windows that appears during the process until The maintenance operation completed . appears in the following window. After this message appears, click **OK**.



5. A directory is created in the directory specified as the directory in which to save. The name of the directory is based on the date and time of the operation. The collected data is saved in the directory created here. If the following folder structure is not created, collection of log information might have failed. In such cases, collect log information again.

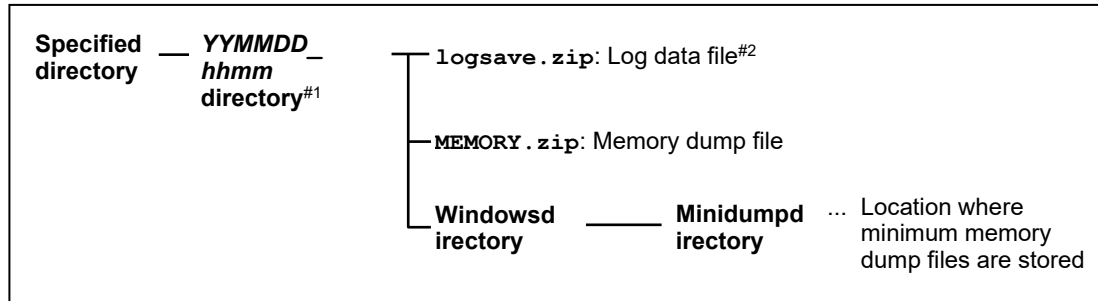


Figure 7-1 Folder structure for collected data

#1: The name of the directory is *YYMMDD_hhmm*.

YY: Lower two digits of the year, *MM*: month, *DD*: day, *hh*: hour, *mm*: minute

Example: If the date is saved at 13:59 on Jan 1, 2025

Directory name: 250101_1359

#2: The following data is saved.

- If **Gather log data.** is selected: `logsave.zip`
- If **Gather memory dump files.** are selected: `MEMORY.zip` and the minimum memory dump files

6. Double-click the saved `logsave.zip` and `MEMORY.zip` files to ensure that you can view the contents of each ZIP file.

If you cannot view the contents of the files, collect log information again, because the ZIP files might have been corrupted.

7.2.4 Closing the log information collection window

To close the log information collection window, click Cancel.

7.3 Logging Trends in the Temperature Inside the Chassis

7.3.1 Overview

This feature periodically measures the temperature inside the chassis of this equipment and records the data in a log file. You can adjust the logging frequency for the temperature inside the chassis by using the logging frequency setup command. The default setting for the logging frequency is 60 minutes. For this setting, you can select from three options: 10, 30, and 60 minutes.

7.3.2 Log files

The temperature inside the chassis is recorded in a log file at the specified logging frequency. When this equipment runs continuously for 8 hours or more, the highest and lowest temperatures across 8 hours are also recorded every 8 hours. For either file, if the log gets full, log entries are overwritten beginning from the oldest entry.

Table 7-8 shows the names of the log files.

Table 7-3 Log files

Parent folder	File name	Description
%ProgramFiles%\HFWRAS\log	temp.csv	The temperature inside the chassis is recorded in this file at the logging frequency. (The maximum number of entries is 51,200.)
	temp_mm.csv	The highest and lowest temperatures in 8 hours are recorded in this file. (The maximum number of entries is 1,100 entries.)

Checking log information

You can check log information by opening the log files by using an application such as Notepad. The files are in CSV format. You can use a spreadsheet or database software to load the log information and draw graphs. You can also use the log information collection window or the `logsave` command to collect the log files. For information about how to use the `logsave` command, see *8.4.1 Log information collection command (logsave)* in the *HF-W200E INSTRUCTION MANUAL* (manual number WIN-62-5004).

Log information format

The format of the log information is as follows:

(1) temp.csv

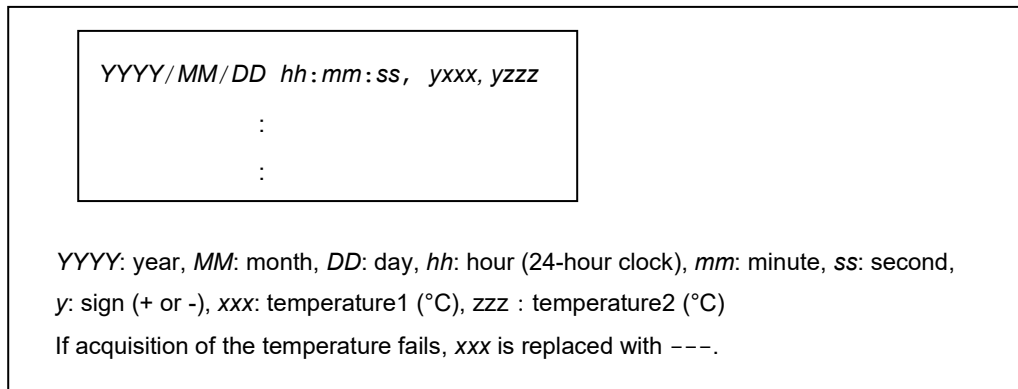


Figure 7-2 Format of the log information (1)

(2) temp_mm.csv

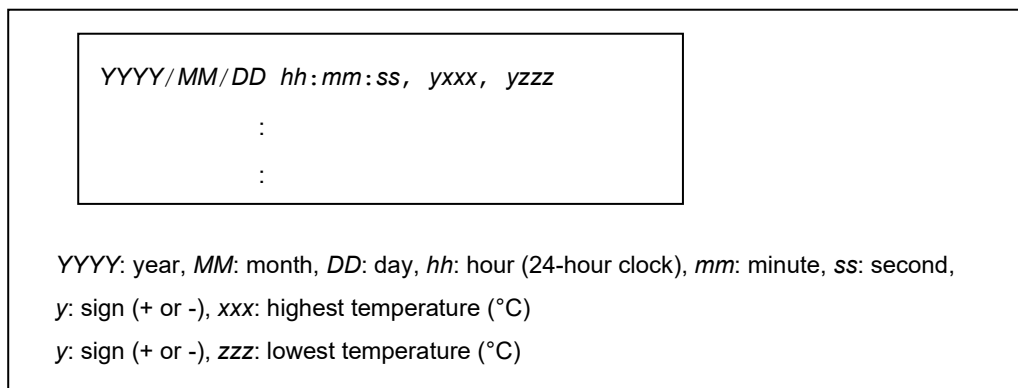


Figure 7-3 Format of the log information (2)

7. FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

7.3.3 Logging frequency setup command

Name

tmplogset - Logging frequency setup

Syntax

tmplogset

Functionality

This command configures the frequency at which to log trends of the temperature inside the chassis.

The following shows how to use the command.

1. Open a command prompt.
Administrator privileges are required to run this command, so sign in to the computer as an administrator, and then open a command prompt.
2. At the command prompt, run the `tmplogset` command. The following initial screen appears along with the current setting. If you enter 2 on the initial screen, the `tmplogset` command exits without changing the settings.

```
>tmplogset
Logging time of the cycle:60 minutes
1. Change at logging cycle [10,30,60 minutes]
2. Exit
:_
```

3. If you enter 1 and then press **Enter**, the following message appears.

```
Please select new time of the cycle.
When the return is input, it becomes like a present setting.
1. 10 minutes
2. 30 minutes
3. 60 minutes
:_
```

7. FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

4. Enter the number corresponding to the new frequency that you want to select, and then press **Enter**. If the number you enter is out of range, the following message appears, prompting you to enter a valid number.

```
The entered setting is invalid.  
Please enter a setting again. [input range: 1-3]
```

5. If you enter a number from 1 to 3, the following message appears. The *x* denotes the time that you selected.

```
New logging time of the cycle is x.  
Is this value set?(y-YES/n-NO)  
:_
```

6. If you enter **y** and then press **Enter**, the logging frequency is updated to the new value, and the command exits. The new setting takes effect when the command exits. If you want to confirm the change, run this command again and check its initial screen.

If you do not want to change the logging frequency, enter **n** and then press **Enter**. The following message appears and the command exits without the setting being changed.

```
The setting takes no effect, because you enter the letter 'n'
```

If you do not have administrator privileges when you run the command, the following message appears and the command exits.

```
>tmplogset  
You do not have the privilege to run this command.  
Please run this command again on "Administrator: Command Prompt".
```

If an internal error occurs when you run the command, the following message appears and the command exits.

```
Error: Systemcall failed. (API-name:error-code)
```

In this message, *API-name* indicates the name of the Windows API that encountered the error. *error-code* shows the error code in hexadecimal. If this message appears, run the command again.

This page is intentionally left blank.

CHAPTER 8 SIMULATING THE HARDWARE STATUS

8.1 Hardware Status Simulation

8.1.1 Overview

This feature simulates the hardware status and the I/O status of the general-purpose external contact of this equipment. By simulating these statuses, you can test a user application and check the notification interface of the RAS software without an actual hardware failure or a connection with the external contact.

To simulate the hardware status and the I/O status of the general-purpose external contact, you need to set the RAS software to a special mode called *simulation mode*. In simulation mode, the monitoring of the actual hardware status is disabled. Never use this equipment in simulation mode for business use.

This feature simulates the following conditions:

- Temperature condition inside the chassis
- Storage condition (Storage failure prediction (SMART monitoring) condition)
- Memory condition
- I/O status of the general-purpose external contact

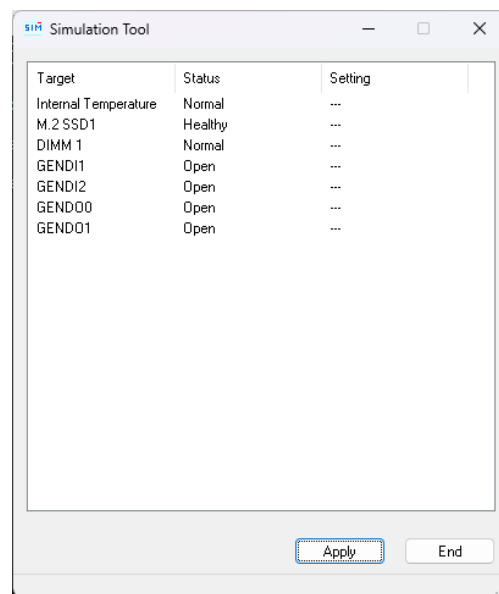


Figure 8-1 Simulation Tool window

8. SIMULATING THE HARDWARE STATUS

The monitoring feature of the RAS software detects change in the simulated hardware status and I/O status of the general-purpose external contact, and then notifies the user of the change through various interfaces.

For information about the interfaces used for notification, see the following sections in this manual:

- Temperature condition inside the chassis: *2.1 Monitoring the Temperature Inside the Chassis*
- Storage condition: *2.2 Storage Failure Prediction (SMART Monitoring)*
- Memory condition: *2.3 Memory Monitoring*
- Input status of the general-purpose external contact: (2) in *6.1.5 Get functions for the general-purpose external contact inputs*
- Output status of the general-purpose external contact: (2) in *6.1.4 Control functions for the general-purpose external contact outputs*

NOTICE

While the equipment is running in simulation mode, monitoring of the actual hardware status is disabled. Errors, including abnormal temperatures, cannot be detected. Never use this equipment in simulation mode for business use. Use the simulation feature only for testing user applications and checking the notification interface of the RAS software.

NOTE

- When checking for temperature abnormalities in the simulation mode, change the shutdown function setting as necessary before performing the check.
 - In simulation mode, the memory-monitoring feature records an event (event ID: 525) in the event log only when a memory error is detected for the first time. Afterward, even if the memory error persists, the memory monitoring feature does not record events in the event log.
 - In simulation mode, when the Storage failure prediction feature records an event (event ID: 265) of SMART detection in the event log, the string *XXXXXXXX* is used for the model name of the Storage. Also note that you cannot simulate a Storage condition of "Unknown" or multiple conditions (for example, SMART detection and power-on (used) hours exceeded) that occur at the same time.
 - Simulation of the input status of general-purpose external contact is used to check the output from a library function, rather than for checking the output of an actual general-purpose external contact.
-

8.1.2 Using the simulation feature

Run the simulation mode start command at the command prompt to set the RAS software to the simulation mode. When the RAS software transitions to the simulation mode, the Simulation Tool window appears on the screen. You can use this window to simulate the condition of hardware devices. To exit the simulation mode, restart this equipment.

This subsection describes how to use the simulation mode.

(1) Overview of the procedure for using the simulation feature

Figure 8-2 shows a rough flowchart of using this feature. The RAS software runs in simulation mode from the time that the simulation mode start command is executed until the OS shuts down.

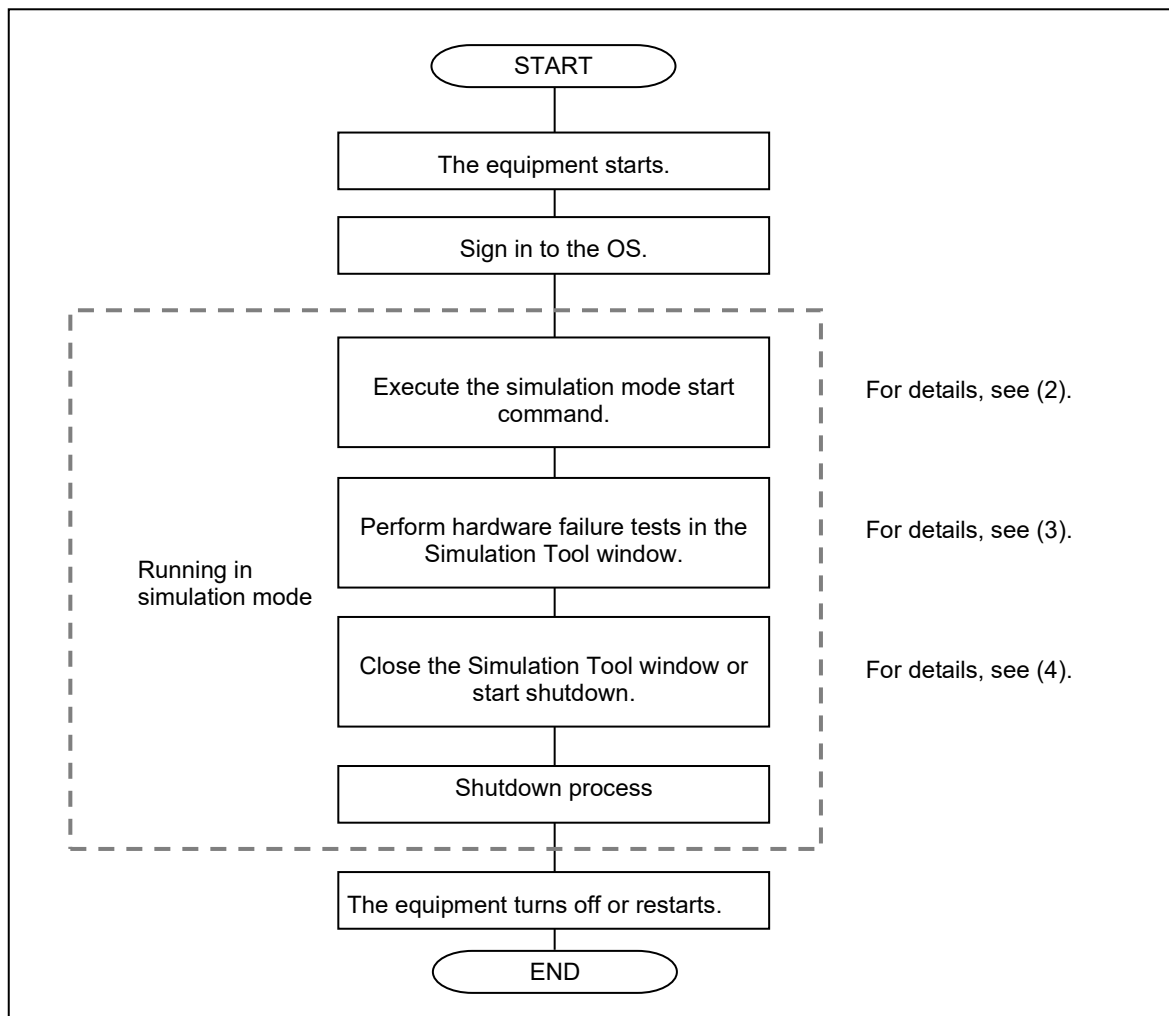


Figure 8-2 Procedure for using the simulation mode

8. SIMULATING THE HARDWARE STATUS

(2) Starting the simulation mode

To start the simulation mode, run the simulation mode start command (the `simrasstart` command) at the command prompt.

NOTE

- The simulation mode cannot be started from a remote desktop. Before you start the simulation mode, other signed-in users must sign out.
 - If the RAS software has already detected a hardware failure, the simulation mode cannot start. Remove the cause of the failure before using the simulation mode.
-

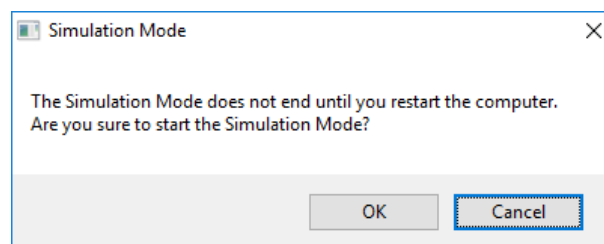
1. Open a command prompt.

You need to have administrator privileges to run the simulation mode start command. Sign in to the computer as an administrator and open a command prompt.

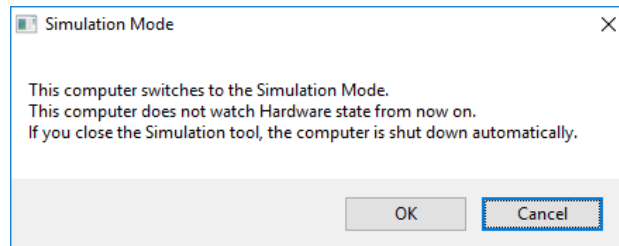
2. At the command prompt, enter the following, and then press **Enter**.

```
>%SystemDrive%\Program Files\HFWRAS\sim\simrasstart
```

3. The following message about exiting the simulation mode appears. Click **OK**. If you click **Cancel**, the simulation mode does not start.



4. The following message appears, indicating that the simulation mode starts. Click **OK**. If you click **Cancel**, the simulation mode does not start.



5. The Simulation Tool window appears. This equipment runs in simulation mode from this point forward. Monitoring of hardware failures is now disabled.

NOTE

The simulation feature performs the following actions while running in simulation mode:

- Every 10 seconds, the Windows® Exclamation sound is played twice. (This sound is audible only when speakers are connected.)
-

8. SIMULATING THE HARDWARE STATUS

(3) Using the Simulation Tool window

When the RAS software transitions to the simulation mode, the Simulation Tool window appears as shown in Figure 8-3.

You can use the Simulation Tool window to change the condition of hardware devices.

When the Simulation Tool window starts, all hardware devices are set to the normal status.

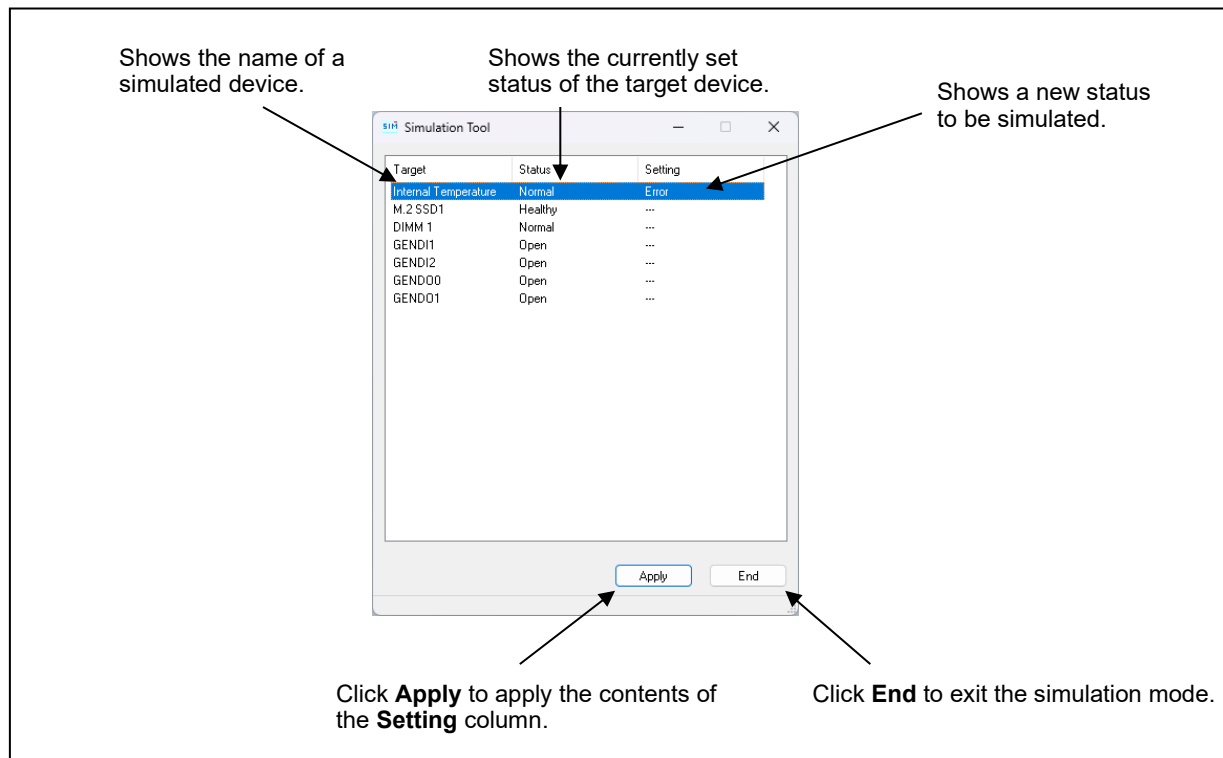


Figure 8-3 Description of the Simulation Tool window

- **Target**

Shows the name of each simulated hardware device.

Category	Target
Temperature condition inside the chassis	Internal temperature
Storage condition	M.2 SSD1
Memory condition	DIMM1
Input status of the general-purpose external contact	GENDI1、 GENDI2
Output status of the general-purpose external contact	GENDO0、 GENDO1

- **Status**

Shows the currently set status of each simulated hardware device. The following shows a list of statuses for each hardware device.

Category	Target	Status
Temperature condition inside the chassis	Internal temperature	Normal, Error
Storage condition	M.2 SSD1	Healthy, SMART Detected,
Memory condition	DIMM1	Normal, Error
Input status of the general-purpose external contact	GENDI1、 GENDI2	Open, Closed ^{#1} ,
Output status of the general-purpose external contact	GENDO0、 GENDO1	Open, Closed ^{#1} ,

#1: When the contact state is high, it is displayed as "Open" and when it is low, it is displayed as "Closed."

After the Simulation Tool window starts, all hardware devices are set to **Normal** or **Healthy**.

- **Setting**

Shows a new status to be simulated for each target hardware device.

If no status to be simulated is set, "---" appears. (After the Simulation Tool window starts, "---" is displayed for all hardware devices.)

- **Apply** button

If you click this button, the contents of the **Setting** column are applied to the status of the hardware devices.

The monitoring feature of the RAS software detects changes in the status of the hardware devices and notifies users of the changes through various interfaces.

NOTE

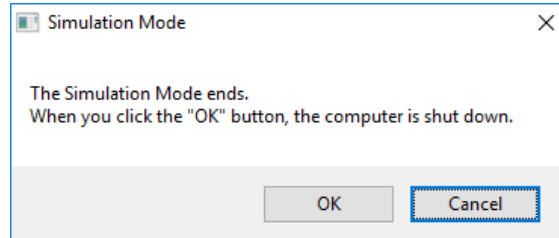
A new hardware status is applied to the notification interface of the RAS software when the following time elapses after you click **Apply** in the Simulation Tool window. Wait for the following times to elapse before you check the results of a simulation.

- Temperature condition inside the chassis: About 15 seconds later
 - Storage condition: About 5 seconds later
 - Memory condition: About 10 seconds later
 - Input status of the general-purpose external contact: Immediately
-

8. SIMULATING THE HARDWARE STATUS

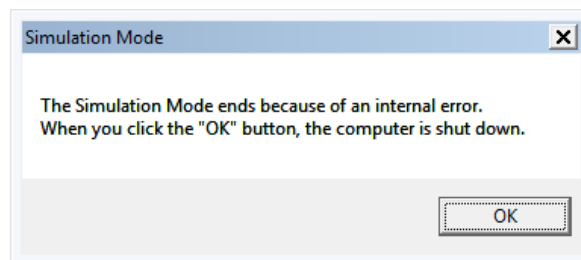
- **End** button

If you click this button, a shutdown is performed to exit the simulation mode. Before a shutdown is performed, the following message appears. Save your data and perform any other necessary tasks, then click **OK**. If you click **Cancel**, the Simulation Tool window is not closed.



NOTE

Even if the Simulation Tool window exits due to an internal error or other reasons, shutdowns are performed automatically to exit the simulation mode. Before a shutdown is executed, the following message appears. Save your data and perform any other necessary actions, and then click **OK**.



- **Minimize** button ([_] button)

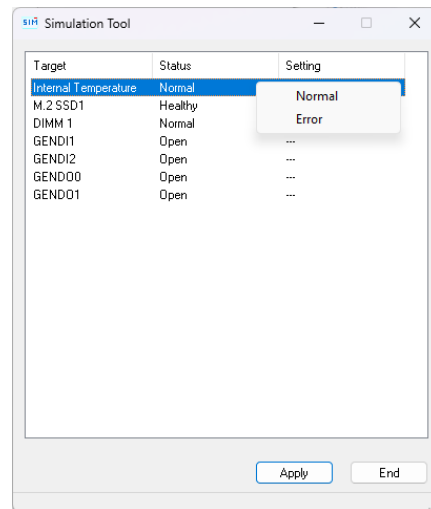
Click the **Minimize** button in the upper-right corner of the Simulation Tool window to minimize the window.

- **Close** button ([X] button)

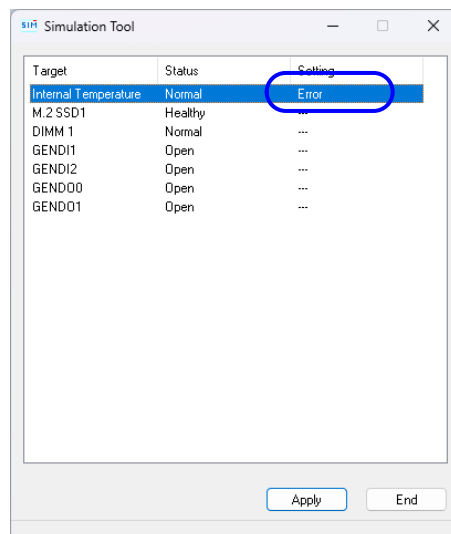
Click the **Close** button in the upper-right corner of the Simulation Tool window to shut down and exit the simulation mode. The behavior after you click this button is the same as when you click **End**.

The following procedure shows how to simulate a hardware status by using the Simulation Tool window.

1. Right-click a hardware item that you want to simulate. A pop-up menu appears. The menu lists the statuses that you can select based on the current hardware status.

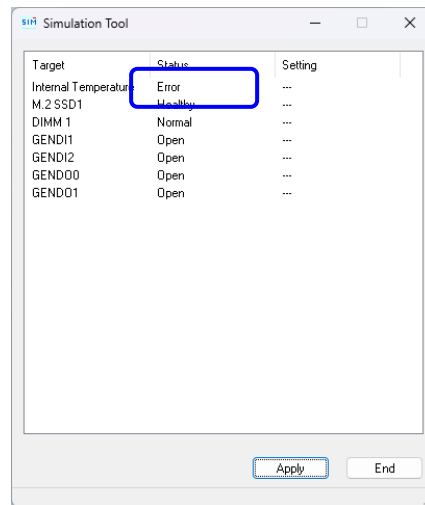


2. If you select a status that you want to simulate from the pop-up menu, the selected status is displayed in the **Setting** column.



8. SIMULATING THE HARDWARE STATUS

3. To apply the status displayed in the **Setting** column to the hardware status, click **Apply**. This updates the **Status** column in the Simulation Tool window.



NOTE

If nothing appears in the **Setting** column, (that is, --- is shown) or the value is the same as the **Status** column after you click **Apply**, the current status does not change.

The following shows a list of items in the pop-up menu that are displayed when you right-click each hardware item.

In the pop-up menu, the current status and the statuses to which the current status can transition are displayed. Note that if the current status is the only status that can be displayed, only a grayed-out **None** is displayed in the menu.

- Internal temperature condition

No.	Current status	Statuses in the pop-up menu	Note
1	Normal	Normal, Error	
2	Error		

- Storage condition

No.	Current status	Statuses in the pop-up menu	Note
1	Healthy	Healthy, SMART Detected	
2	SMART Detected		#1,

#1: Transition from **SMART Detected** to **Healthy** means that the target storage is replaced with a new one and that the new storage is connected.

- Memory condition

No.	Current status	Statuses in the popup menu	Note
1	Normal	Normal, Error, Failure, Not Mounted	
2	Error		

- Input status of general-purpose external contacts

No.	Current status	Statuses in the pop-up menu	Note
1	Open	Open, Closed	
2	Closed		

- Output status of general-purpose external contacts

No.	Current status	Statuses in the popup menu	Note
1	Open	None	#
2	Closed		#

#: The pop-up menu contains only **None** because only the control result of GendoControlN (the control function for general-purpose external contact outputs) is displayed.

8. SIMULATING THE HARDWARE STATUS

(4) Exiting the simulation mode

To exit the simulation mode, shut down or restart this equipment. There are no restrictions on how to shut down or restart the equipment. In the same way as in its normal operating mode, this equipment is shut down or restarted in the following cases (for the following reasons):

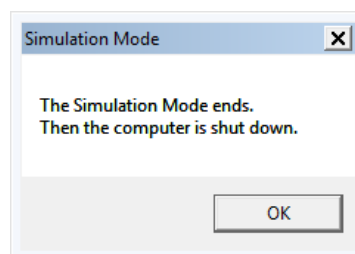
- A shutdown is executed from the **Start** menu.
- A system shutdown API function, such as the `BSSysshut` and `ExitWindowsEx` functions, is executed.
- The equipment is automatically shut down because abnormal temperature inside the chassis.
- **Ctrl+Alt+Delete** is pressed and the power button in the lower-right corner of the screen is used to shut down the equipment.
- The power is forcibly turned off by using the power switch.

The hardware simulation feature automatically shuts down the equipment in the following cases:

- The **End** button or the **Close** button [×] in the Simulation Tool window is clicked.
- The Simulation Tool window terminates with an error.

NOTE

- As explained here, shutting down or restarting the equipment exits the simulation mode. To start the simulation mode again, execute the simulation mode start command after the equipment restarts.
- If you perform a shutdown or sign-out during the transition to the simulation mode, the following message appears, indicating that the simulation mode will end and that the system will shut down.



- When a remote connection is used, the shutdown process can be delayed when the shutdown is executed for one of the preceding reasons.
-

8.1.3 Precautions when you use the Simulation Tool window

- (1) When the new status to be simulated is finalized

From the time that you select a new status to be simulated from the pop-up menu in the Simulation Tool window until when you click **Apply**, the new status to be simulated is not finalized and can be changed.

The actual status used for simulation is the one shown in the **Setting** column in the Simulation Tool window when you click **Apply**. If nothing is set for **Setting** (that is, --- appears), the status displayed for **Status** is maintained.

8.1.4 Event log entries

To clearly show which log entries for hardware failure originate from the simulation feature, the feature records the event log entries listed in Table 8-1.

Note that a log entry with event ID 252 is recorded when you click **Apply** in the Simulation Tool window. This log entry is recorded even when none of the items in the **Setting** column is specified.

Table 8-1 Event log entries recorded by this feature

Event ID	Source	Type	Category	Description
250	HFWSIM_SYS	Information	HFWSIM	The simulation mode started.
251	HFWSIM_SYS	Information	HFWSIM	The simulation mode ended.
252	HFWSIM_SYS	Information	HFWSIM	The following conditions were set in the simulation mode. Internal Temperature: %1 M.2 SSD1 : %2 DIMM1 : %3 GENDI1 : %4 GENDI2 : %5

%x in this table denotes the status that was simulated by clicking **Apply** in the Simulation Tool window.

8. SIMULATING THE HARDWARE STATUS

8.1.5 Remote notifications

This feature notifies users of the transition of the RAS software to simulation mode by using trap notifications, so that an SNMP manager that monitors this equipment from a remote location can identify that the hardware statuses that it acquires and the trap notifications for hardware failure (and hardware recovery) that it receives are generated in simulation mode. In addition, the object value for the operating mode of the RAS software is changed to the one for the simulation mode.

NOTE

The contents of the hardware statuses that can be acquired and the trap notifications that can be sent are the same as in the normal mode. For details about the objects of the extended MIB for HF-W, see *4.4 Remote Notifications*.
