WIN-63-5000

HITACHI

HITACHI EMBEDDED CONTROLLER

HF-W400E

RAS FEATURES MANUAL

USER'S
MANUAL

WIN-63-5000

HITACHI

HITACHI EMBEDDED CONTROLLER

HF-W400E

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.**

USER'S MANUAL

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 Administration Regulations of the United States, and complete all required procedures.
If anything is unclear, contact a sales representative.

First Edition, July 2025, WIN-63-5000

⚠ 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 heading is used to indicate imminent hazards that are highly likely to result in serious personal injury or death.

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

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

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

The term *serious personal 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 personal injuries are vision loss, burns (caused by dry heat), low-temperature burns, electric-shock injuries, broken bones, and poisoning. A *moderate or minor personal 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 the failure of a drive is anticipated, the drive might experience hardware failure in the near future. We recommend backing up the data and replacing the drive. For information about how to replace a drive, see the *HF-W400E INSTRUCTION MANUAL (manual number WIN-62-5001).*

  (Page 2-3)

- 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 2-6)

- If this equipment continues to operate after a fan failure is detected, internal parts such as the processor will not cool sufficiently, which might cause the thermal runaway of the system due to a malfunction in the equipment, or result in damage to parts.

  (Page 3-3)

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

  (Page 4-2)

- When the failure of a drive is anticipated, the drive might experience hardware failure in the near future. We recommend backing up the data and replacing the drive. For information about how to replace a drive, see the *HF-W400E INSTRUCTION MANUAL (manual number WIN-62-5001)*.

  (Page 4-9)

- If this equipment continues to operate after a fan failure is detected, internal parts such as the processor will not cool sufficiently, which might cause the thermal runaway of the system due to a malfunction in the equipment, or result in damage to parts. If possible, enable the automatic shutdown feature.

- If the automatic shutdown feature is not used, have a user application detect fan failures by using a RAS event, and shut down the equipment when such failures are detected.

  (Page 5-2)

⚠ 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 6-13)

- 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 fan failures and 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 EMBEDDED CONTROLLER HF-W400E (abbreviated hereinafter to *this equipment*).

Structure of this manual

This manual consists of the following:

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 *4.5.3 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

- Windows® 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) = $1{,}000^2$ bytes (10^6 bytes)

  1 GB (gigabyte) = $1{,}000^3$ bytes (10^9 bytes)

  1 TB (terabyte) = $1{,}000^4$ bytes (10^{12} bytes)

CONTENTS

FIGURES

TABLES

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
		Digital LEDs for status indications
		Remote notifications
		Status acquisition by using library functions
Control	Shutdown/startup suppression	Automatic shutdown
		Shutdown via library functions
		Startup suppression when severe failures occur
		Controlling general-purpose external contacts
		Controlling digital LEDs for status indications
Library functions		RAS library
Maintenance/failure analysis	Memory dumps	Memory dump collection
		Bug check (blue screen) cause notifications
		Log information collection window
		Maintenance operation support commands
		Logging trends of the temperature inside the chassis
Simulation		Hardware status simulation

Monitoring

(1) Hardware status monitoring

This feature monitors the hardware status of this equipment, including fan status, drive status, and 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. As long as the process with the highest priority (real-time priority class) can run properly, the status lamp on the front of this equipment is lit in green.

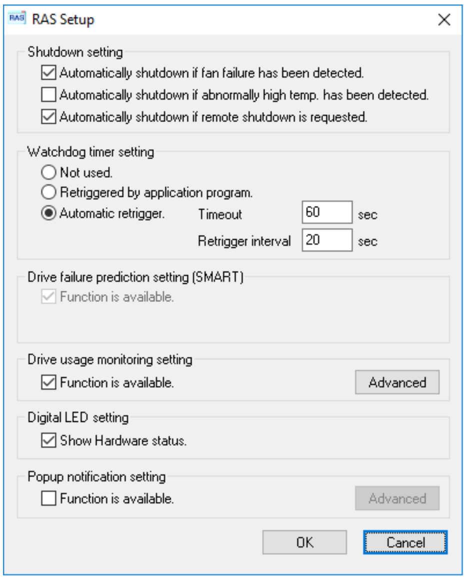
(3) Watchdog timer monitoring

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



RAS Setup

Shutdown setting
☑ Automatically shutdown if fan failure has been detected.
□ Automatically shutdown if abnormally high temp. has been detected.
☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting
○ Not used.
○ Retriggered by application program.
● Automatic retrigger. Timeout 60 sec
Retrigger interval 20 sec

Drive failure prediction setting (SMART)
☑ Function is available.

Drive usage monitoring setting
☑ Function is available.

Digital LED setting
☑ Show Hardware status.

Popup notification setting
□ Function is available.

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.

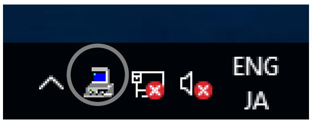


Figure 1-2 Hardware status icon

(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) Digital LEDs for status indications

The digital LEDs for status indications are on the front of this equipment, and notify users when errors occur in the equipment hardware. These LEDs can be used by a user application to, for example, notify the user of application failures.

(9) 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.

(10) 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

(11) Automatic shutdown

This feature automatically shuts down the equipment when it detects a fan failure, an abnormal temperature inside the chassis, or a remote shutdown signal input. Configure the settings shown in *(4) RAS Setup window* to enable or disable the automatic shutdown feature.

(12) Shutdown via library functions

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

(13) Startup suppression when severe failures occur

This feature suppresses startup of this equipment to protect the hardware when a failure, such as a fan failure, is detected during OS startup.

(14) Controlling general-purpose external contacts (HJ-F4070-11, HJ-F4070-12, and HJ-F4070-13) and digital LEDs for status indications

This feature allows users to control the general-purpose external contacts and the digital LEDs for status indications by using the RAS library.

Four input and three output general-purpose external contacts (two inputs and two outputs for HJ-F4070-13) are available. 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

(15) RAS library

This feature offers library functions for recording log information, in addition to the library functions offered by the status check (10) and control (12 and 14) items.

Maintenance/failure analysis

(16) Memory dump collection

This feature records the contents of the system memory in a file (a memory dump file) when the NMI switch is pressed after a failure occurs, for example, after the equipment stops unexpectedly. By analyzing the data in this memory dump, you can investigate the cause of the failure.

(17) Bug check (blue screen) cause notifications

This feature detects a bug check (blue screen), and records the cause in the event log.

(18) 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.

(19) 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.

(20) 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

(21) 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) to (12), (14), (15), (17), (18), (20), and (21). For details about the other features, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

CHAPTER 2 ITEMS MONITORED BY THE RAS FEATURES

This chapter describes the items monitored by the RAS features.

For information about the hardware specifications of the RAS external contact interface (optional) described in this chapter and how to use each contact, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

2.1 Fan Monitoring

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 notifications

(4) Digital LEDs for status indications

(5) Remote notifications

(6) Automatic shutdown

(7) Alarm lamp

For details about (1) to (5), see *Chapter 4. CHECKING THE HARDWARE STATUS*. For details about (6), see *5.1 Automatic Shutdown of the Equipment*.

If you use the MCALL contact of the RAS external contact interface, the external hardware connected to the RAS external contact interface can detect malfunctions in this equipment. The following table shows how the state of the MCALL contact changes.

Table 2-1 State of the equipment in terms of fan monitoring and state of the MCALL contact

Status of this equipment	MCALL contact
During startup of the equipment or when the power is off	Open
When the fans are operating normally or have recovered from a malfunction	Open
During a fan malfunction	Closed

The alarm lamp (as well as the MCALL contact of the RAS external contact interface) is not dedicated to fan monitoring and, therefore, is not turned off (opened) when the system recovers from a fan malfunction as long as a malfunction is detected for other monitored items.

2.2 Monitoring the Temperature Inside the Chassis

This feature monitors the temperature inside the chassis by using the temperature sensor in this equipment, and notifies users when the temperature inside the chassis becomes abnormally high via the following methods:

(1) Hardware status window

(2) Event notifications

(3) Pop-up notifications

(4) Digital LEDs for status indications

(5) Remote notifications

(6) Automatic shutdown

(7) Alarm lamp

For details about (1) to (5), see *Chapter 4. CHECKING THE HARDWARE STATUS*. For details about (6), see *5.1 Automatic Shutdown of the Equipment*.

If you use the MCALL contact of the RAS external contact interface, the external hardware connected to the RAS external contact interface can detect malfunctions in this equipment. The following table shows how the state of the MCALL contact changes.

Table 2-2 State of the equipment in terms of monitoring the temperature inside the chassis and state of the MCALL contact

Status of this equipment	MCALL contact
During startup of the equipment or when the power is off	Open
When the temperature is normal or has recovered from being abnormally high	Open
When the temperature is abnormally high	Closed

The alarm lamp (as well as the MCALL contact of the RAS external contact interface) is not dedicated to monitoring the temperature inside the chassis and, therefore, is not turned off (opened) when the system recovers from an abnormally high temperature as long as a malfunction is detected for other monitored items.

2.3 Drive Failure Prediction (SMART monitoring)

The hard drives in this equipment have the Self-Monitoring, Analysis and Reporting Technology (SMART) feature, which continuously monitors the condition of the drives and anticipates failures before they manifest. The drive failure prediction feature notifies users when a drive might fail in the near future via the following methods:

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

For details about (1) to (5), see *Chapter 4. CHECKING THE HARDWARE STATUS*.
For details about (6), see *6.1.8 Get function for the drive condition (hfwDiskStat)*.

NOTICE

When the failure of a drive is anticipated, the drive might experience hardware failure in the near future. We recommend backing up the data and replacing the drive. For information about how to replace a drive, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

NOTE

- It is not possible for the drive failure prediction feature to anticipate all failures. Therefore, a drive might fail before the drive failure prediction feature anticipates any failures.
- This feature only monitors internal drives that are recognized during OS startup. If you connect a new drive or replace a drive with a new drive for maintenance or other reasons, it might take a long time to recognize the new drive upon the first startup after the new drive is connected, and the drive might not be recognized as a drive to be monitored. In this case, restart this equipment.

2.4 Drive Usage Monitoring

The drive usage monitoring feature adds up the power-on hours of a drive in this equipment. If the total power-on hours exceed a set value, the feature notifies users via the following methods. By using this feature, you can keep track of when to replace drives and prevent drive failures that are caused by using drives for too long.

(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) to (4), see *Chapter 4. CHECKING THE HARDWARE STATUS*.

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

This feature can be enabled or disabled in the RAS Setup window. If you disable this feature, notifications via the methods described above are disabled. For details, see *3.1.3 Using the RAS Setup window*.

NOTE

- This monitoring feature is active while the OS is running, that is, from OS startup to OS shutdown. When the OS is not running, drive power-on hours are not monitored.
- This feature records the serial number (unique ID) of a drive in the equipment in a drive management information file. If you install a drive that does not match the serial numbers recorded in the drive management information file, the cumulative power-on hours for the drive are automatically reset.
- This feature is not designed to anticipate drive failure, but we recommend replacing drives when their power-on hours exceed the set value for preventive maintenance of components with a limited life. For information about how to replace a drive, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).
- This feature only monitors internal drives that are recognized during OS startup. If you connect a new drive or replace a drive with a new drive for maintenance or other reasons, it might take a long time to recognize the new drive upon the first startup after the new drive is connected, and the drive might not be recognized as a drive to be monitored. In this case, restart this equipment.

2.5 Memory Monitoring

Memory with error checking and correcting (ECC) is installed in this equipment. A single-bit error in the memory can be automatically corrected without interfering with the operation of the equipment. On the other hand, memory might have failed in the following cases of single-bit errors, and we recommend that you replace the memory modules for preventive maintenance:

1. A single-bit error frequently occurs while the OS is running.
2. A single-bit error continuously occurs each time the OS starts.

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

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

(2) and (3) report *High frequency of memory error correction* for error case 1, and *Possible memory failure* for error case 2. In the case of (1) and (4), both error cases are handled as memory errors, and the same value is returned.

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

NOTE

For information about how to replace a memory module, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

2.6 OS Deadlock Monitoring

The OS deadlock monitoring feature uses a timer implemented in this equipment to monitor the operational state of the OS to detect situations where a process with real-time priority cannot run. Such situations (hereinafter referred to as an *OS deadlock*) occurs if, for example, a runaway kernel exists or all CPU load is used up by a driver. When an OS deadlock is detected, this feature notifies users via the status lamp on the front of this equipment.

Table 2-3 State of the equipment in terms of OS deadlock monitoring and state of the status lamp

State of this equipment	Status lamp
During startup of the equipment or when the power is off	Red
When an OS deadlock has not occurred or the system has recovered from an OS deadlock	Green
When an OS deadlock has occurred	Red

If you use the CPUSTOP contact of the RAS external contact interface, the external hardware connected to the RAS external contact interface can detect an OS deadlock. The following table shows how the state of the CPUSTOP contact changes.

Table 2-4 State of the equipment in terms of OS deadlock monitoring and state of the CPUSTOP contact

State of this equipment	CPUSTOP contact
During startup of the equipment or when the power is off	Closed
When an OS deadlock has not occurred or the system has recovered from an OS deadlock	Open
When an OS deadlock has occurred	Closed

NOTICE

When an OS deadlock occurs, 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 an OS deadlock occurs, 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.
- When the OS starts, the status lamp is lit green when this feature starts.
- When the OS shuts down, the status lamp becomes red when this feature stops. Note that when the status lamp turns red, the shutdown process is not complete. Do not turn off the power at this point.

Figure 2-1 shows how the state of the status lamp and the state of the CPUSTOP contact change.

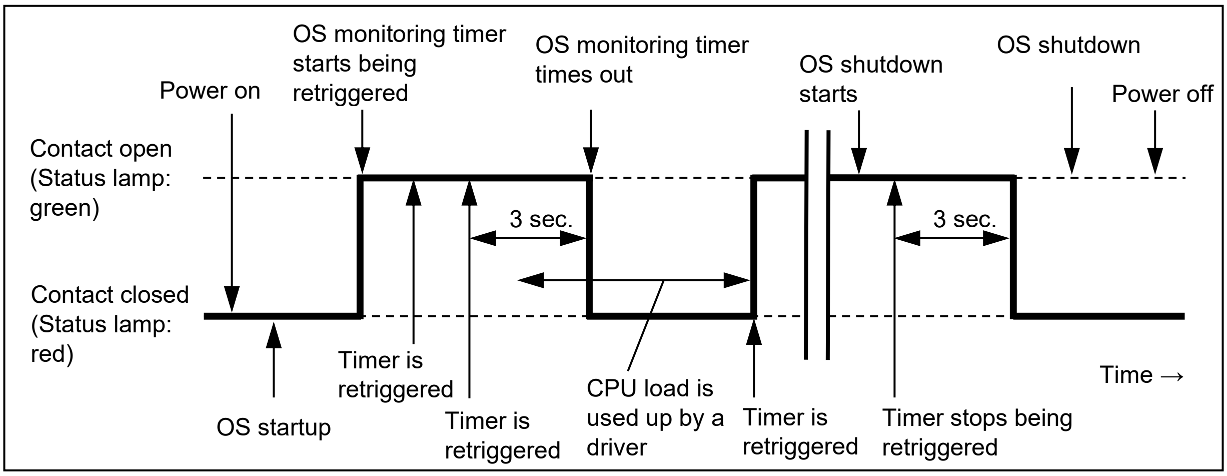

OS monitoring timer starts being retriggered
OS monitoring timer times out
OS shutdown starts
OS shutdown
Power on
Power off
Contact open (Status lamp: green)
Contact closed (Status lamp: red)
3 sec.
3 sec.
Timer is retriggered
Timer is retriggered
OS startup
CPU load is used up by a driver
Timer is retriggered
Timer stops being retriggered
Time →

Figure 2-1 State of the status lamp and the CPUSTOP contact

The dotted lines in Figure 2-1 denote the state of the contact, and the thick line denotes the transition of the state of the contact.

2.7 Watchdog Timer Monitoring

This equipment has a watchdog timer. By retriggering a watchdog timer automatically, this feature monitors whether processes are scheduled properly. In addition, this feature can be used, for example, to monitor the operational state of user programs by using dedicated library functions.

2.7.1 Automatic retriggering feature for a watchdog timer

This feature is implemented via a process that retriggers a watchdog timer periodically. This process runs with the idle priority. If the amount of CPU time used by processes with a priority higher than the idle priority class exceeds a set value, the watchdog timer times out. You can use this feature to, for example, detect a runaway application process and so on.

You can set up the watchdog timer timeout and the retrigger interval in the RAS Setup window. For information about how to use the RAS Setup window, see *3.1.3 Using the RAS Setup window*. The factory default settings are as follows:

Timeout	60 (seconds)
Retrigger interval	20 (seconds)

2.7.2 Using a watchdog timer to monitor user programs

To use a watchdog timer to monitor the operational state of a user program, you can use a configuration where, for example, the user program periodically retriggers the watchdog timer (that is, it resets the watchdog timer to the initial value) and another program checks whether a timeout occurs for the watchdog timer. The flowchart for this configuration is shown below.

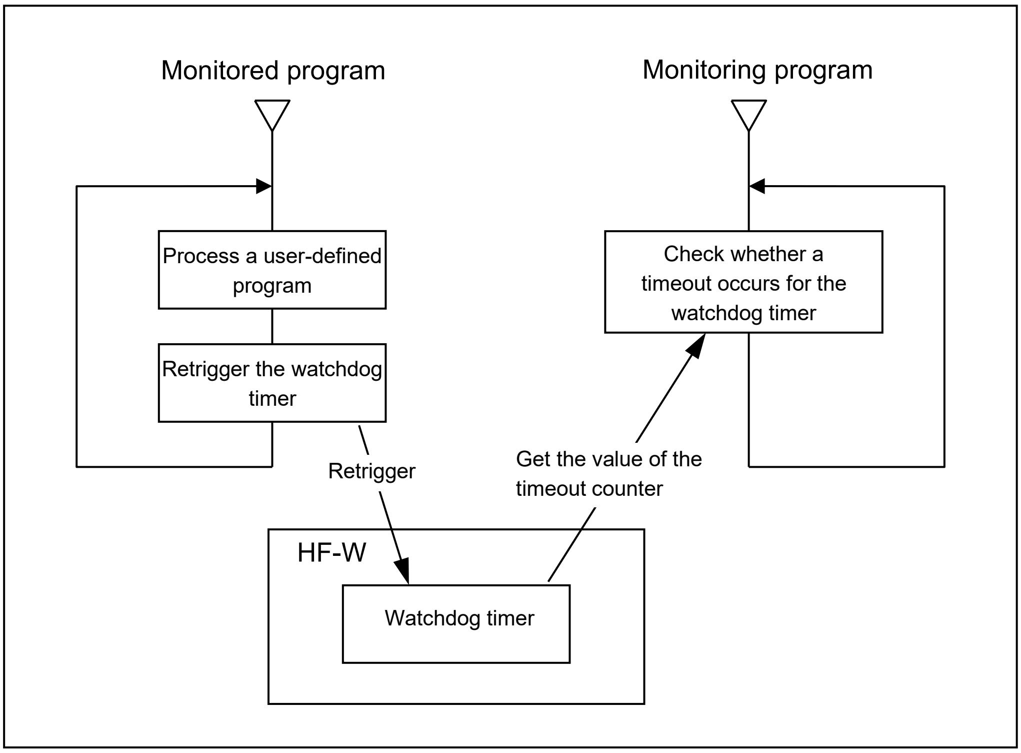

Monitored program
Monitoring program
Process a user-defined program
Retrigger the watchdog timer
Check whether a timeout occurs for the watchdog timer
Retrigger
Get the value of the timeout counter
HF-W
Watchdog timer

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

In Figure 2-2, the monitored program is configured to periodically retrigger the watchdog timer.

The monitoring program periodically checks the timeout counter of the watchdog timer and, if the value of the timeout counter (that is, the remaining time until timeout expires) is zero, a timeout is determined to have occurred.

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. In this example, another program detects the timeout of the watchdog timer, but if you use the RAS external contact interface, the WDTTO contact is closed when a timeout occurs for the watchdog timer. In this case, the external hardware connected to the RAS external contact interface can monitor the operational state of a user program.

A program can use a watchdog timer by calling the library function `WdtControl`. For information 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

If you are using the `WdtControl` function, you cannot use the automatic retriggering feature for the watchdog timer. To use the `WdtControl` function, select **Retriggered by application program** under **Watchdog timer setting** in the RAS Setup window. For information about how to use the RAS Setup window, see *3.1.3 Using the RAS Setup window*.

CHAPTER 3　SETTING UP THE RAS FEATURES

3.1 RAS Setup Window

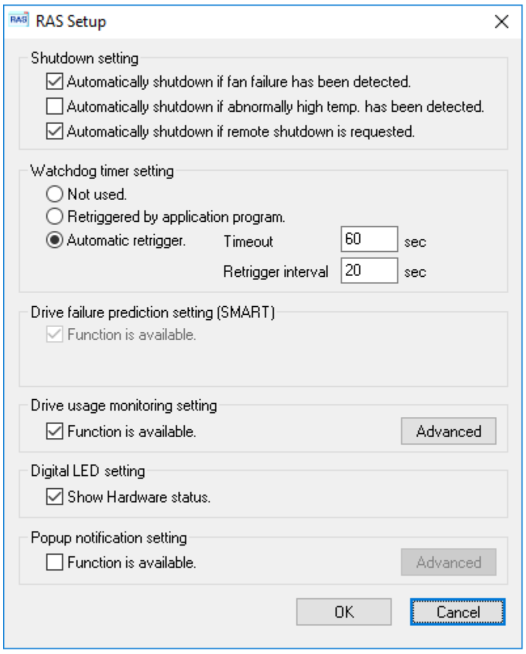
3.1.1 Overview

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	Automatically shut down if a fan failure is detected
	Automatically shut down if an abnormally high temperature is detected
	Automatically shut down if a remote shutdown is requested
Watchdog timer settings	
Drive usage monitoring settings	
Digital LEDs for status indications settings	
Pop-up notification settings	

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



RAS Setup

Shutdown setting

☑ Automatically shutdown if fan failure has been detected.

□ Automatically shutdown if abnormally high temp. has been detected.

☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting

○ Not used.

○ Retriggered by application program.

● Automatic retrigger.　Timeout 60 sec

Retrigger interval 20 sec

Drive failure prediction setting (SMART)

☑ Function is available.

Drive usage monitoring setting

☑ Function is available.

Digital LED setting

☑ Show Hardware status.

Popup notification setting

□ Function is available.

Figure 3-1 RAS Setup window

3.1.2 Opening the RAS Setup window

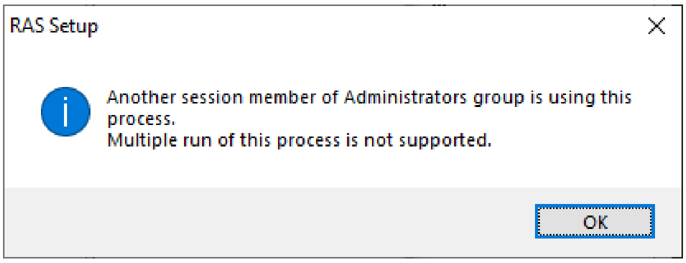
To open the RAS Setup window, perform the following procedure.

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.



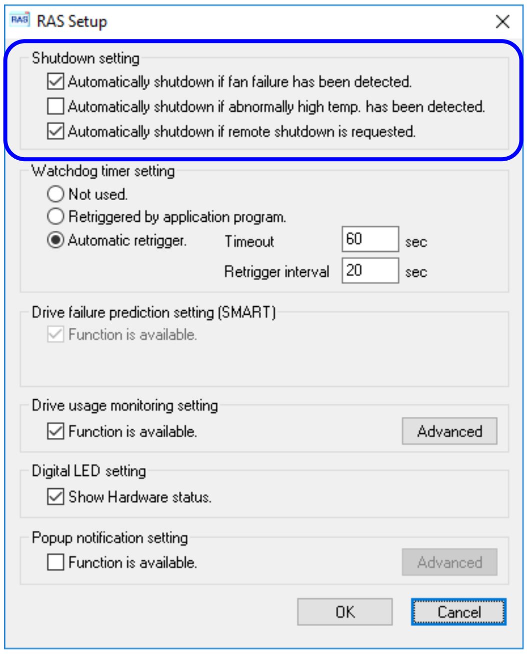
RAS Setup

Another session member of Administrators group is using this process.
Multiple run of this process is not supported.

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: a fan failure, an abnormally high temperature, and a remote shutdown request.



RAS Setup

Shutdown setting

☑ Automatically shutdown if fan failure has been detected.

☐ Automatically shutdown if abnormally high temp. has been detected.

☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting

○ Not used.

○ Retriggered by application program.

● Automatic retrigger. Timeout 60 sec

Retrigger interval 20 sec

Drive failure prediction setting (SMART)

☑ Function is available.

Drive usage monitoring setting

☑ Function is available.

Digital LED setting

☑ Show Hardware status.

Popup notification setting

☐ Function is available.

Figure 3-2 Shutdown settings items

- **"Automatically shutdown if fan failure has been detected"** check box
  - Selected: This equipment will automatically shut down (factory default setting).
  - Cleared: This equipment will not automatically shut down.
- **"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).
- **"Automatically shutdown if remote shutdown is requested"** check box
  - Selected: This equipment will automatically shut down (factory default setting).
  - Cleared: This equipment will not automatically shut down.

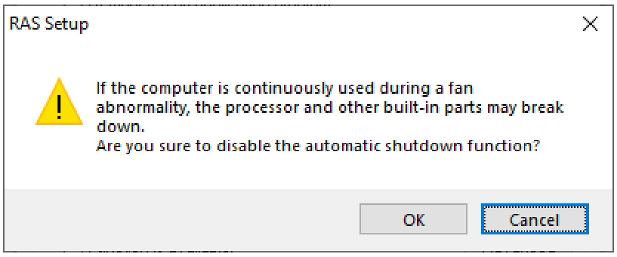
To change the current setting, click the check box that you want to change.

NOTICE

If this equipment continues to operate after a fan failure is detected, internal parts such as the processor will not cool sufficiently, which might cause the thermal runaway of the system due to a malfunction in the equipment, or result in damage to parts.

NOTE

- If you clear the **"Automatically shutdown if fan failure has been detected"** check box, the following cautionary message appears. Enable this feature (select the check box) whenever possible.



RAS Setup

If the computer is continuously used during a fan abnormality, the processor and other built-in parts may break down.
Are you sure to disable the automatic shutdown function?

If you click **Cancel** when this message appears, the check box returns to the selected state. If you click **OK**, the check box is cleared.

- After an automatic shutdown initiated by this feature is complete, the power turns off.
- The remote shutdown input is polled every 5 seconds. This means that it might take up to 5 seconds from when a remote shutdown is requested to when the actual shutdown process starts.

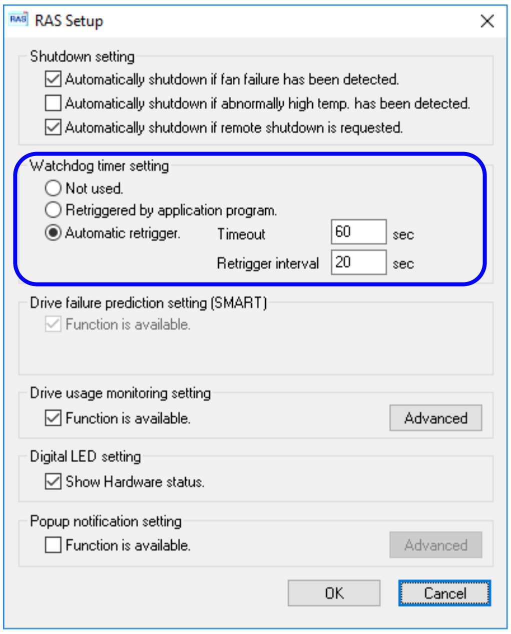
(2) Watchdog timer settings

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:

- Not used
- Retriggered by an application program
- Automatic retrigger

The factory default setting is "Automatic retrigger"



RAS Setup

Shutdown setting
☑ Automatically shutdown if fan failure has been detected.
□ Automatically shutdown if abnormally high temp. has been detected.
☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting
○ Not used.
○ Retriggered by application program.
● Automatic retrigger. Timeout 60 sec
Retrigger interval 20 sec

Drive failure prediction setting (SMART)
☑ Function is available.

Drive usage monitoring setting
☑ Function is available.

Digital LED setting
☑ Show Hardware status.

Popup notification setting
□ Function is available.

Figure 3-3 Items in the watchdog timer settings

- **Not used:**

  If you select this item, the watchdog timer stops, and the watchdog timer will not time out. In addition, you cannot use the watchdog timer by calling the `WdtControl` function of the RAS library.

- **Retriggered by application program:**

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

NOTE

When you change the setting to "**Retriggered by application program**," the watchdog timer stops once. While the watchdog timer is stopped, it will never time out. When a user application retriggers the watchdog timer by using the `WdtControl` function of the RAS library, the watchdog timer resumes its countdown. You can check the state of the watchdog timer (whether it is counting down or stopped) by using the `WdtControl` function.

● **Automatic retrigger:**

If you select this item, the watchdog timer is automatically retriggered by the automatic retriggering feature of the watchdog timer. In addition, you cannot use the watchdog timer by calling the `WdtControl` function of the RAS library.

If this item is selected, you can configure the retrigger interval of the automatic retriggering feature and the timeout of the watchdog timer.

- Timeout

  This sets the timeout of the watchdog timer by using single-byte numeric characters. Enter an integer value from 5 to 60 (in seconds). The factory default setting is 60 seconds.

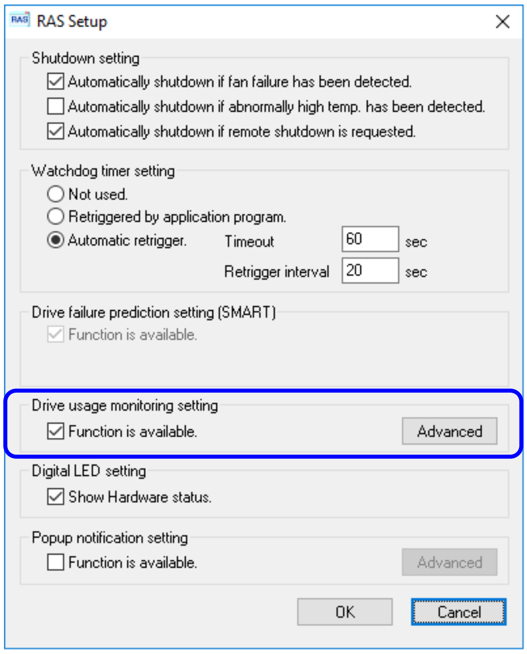
- Retrigger interval

  This sets the retrigger interval of the automatic retriggering feature of the watchdog timer by using single-byte numeric characters. Enter an integer value from 1 to (timeout - 4) (in seconds). The factory default setting is 20 seconds.

You can set a timeout and retrigger interval only if you select **“Automatic retrigger”**. If **“Automatic retrigger”** is not selected, you cannot enter these values.

(3) Drive usage monitoring settings

You can set up the drive usage monitoring settings. By clicking **Advanced**, you can configure the advanced settings of this feature.



RAS Setup

Shutdown setting

☑ Automatically shutdown if fan failure has been detected.

□ Automatically shutdown if abnormally high temp. has been detected.

☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting

○ Not used.

○ Retriggered by application program.

● Automatic retrigger. Timeout 60 sec

Retrigger interval 20 sec

Drive failure prediction setting (SMART)

☑ Function is available.

Drive usage monitoring setting

☑ Function is available. Advanced

Digital LED setting

☑ Show Hardware status.

Popup notification setting

□ Function is available. Advanced

OK Cancel

Figure 3-4 Items in the drive usage monitoring settings

● **"Function is available"** check box

• Selected: Drive usage monitoring is enabled (factory default setting).

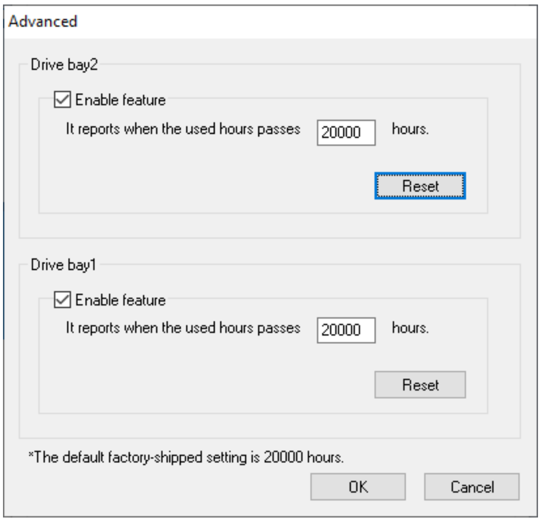
• Cleared: Drive usage monitoring is disabled.

To change the current setting, click the check box.

If drive usage monitoring is disabled, you cannot use the notification features described in *2.4 Drive Usage Monitoring* such as indicating whether the value of the drive power-on (used) hours exceeds the threshold.

● **Advanced** button

If drive usage monitoring is enabled, click **Advanced** to display the window shown in Figure 3-5. If drive usage monitoring is disabled, this button is grayed out and unavailable.



Advanced

Drive bay2

☑ Enable feature

It reports when the used hours passes 20000 hours.

Drive bay1

☑ Enable feature

It reports when the used hours passes 20000 hours.

*The default factory-shipped setting is 20000 hours.

Figure 3-5 Advanced settings for drive usage monitoring

● **“Enable feature”** check box

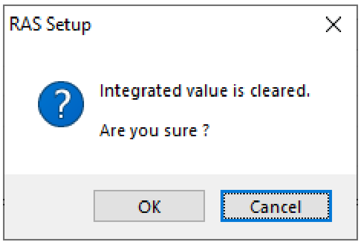
- Selected: Drive usage monitoring is enabled (factory default setting).
- Cleared: Drive usage monitoring is disabled.

● **“It reports when the used hours passes** □ **hours ” box**

Specify a threshold (a single-byte number) for the drive power-on (used) hours so that you can be notified when the value of the drive power-on (used) hours exceeds the threshold. Enter an integer value in hours ranging from 100 to the maximum value at 100-hour increments. The factory default setting is 20000 hours.

The maximum value that can be set is 99900 hours.

If you want to clear the current cumulative power-on (used) hours, click **Reset** to clear the value. The following message then appears:



RAS Setup

Integrated value is cleared.

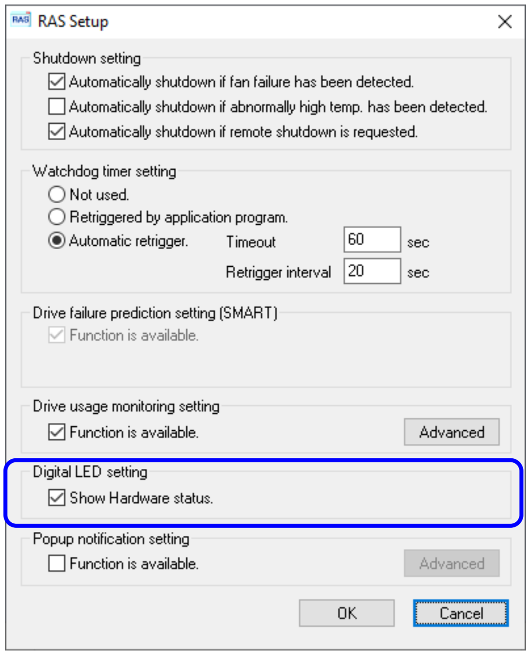
Are you sure ?

If you click **OK** when the message appears, the value of the current cumulative power-on (used) hours is cleared. Click **Cancel** to leave the value as is.

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

(4) Digital LEDs for status indications settings

You can set up the display mode of the digital LEDs for status indications located on the front of the equipment.



RAS Setup

Shutdown setting

☑ Automatically shutdown if fan failure has been detected.

□ Automatically shutdown if abnormally high temp. has been detected.

☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting

○ Not used.

○ Retriggered by application program.

● Automatic retrigger. Timeout 60 sec

Retrigger interval 20 sec

Drive failure prediction setting (SMART)

☑ Function is available.

Drive usage monitoring setting

☑ Function is available. Advanced

Digital LED setting

☑ Show Hardware status.

Popup notification setting

□ Function is available. Advanced

OK Cancel

Figure 3-6 Items in the digital LEDs for status indications settings

- **“Show Hardware status”** check box

  • Selected: Hardware status display mode (factory default setting)

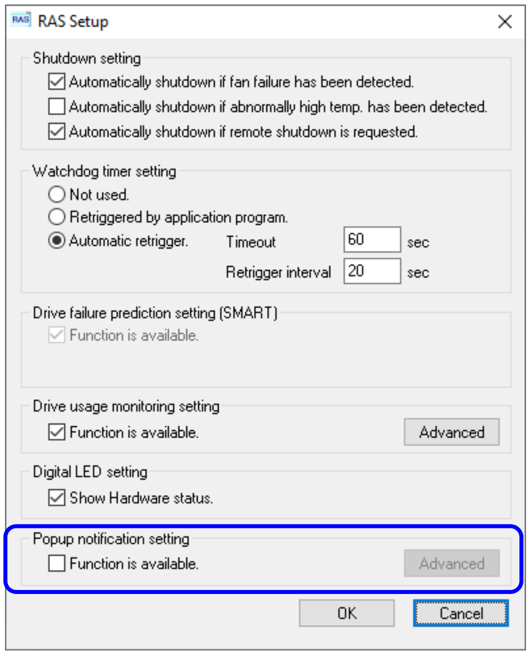
  • Cleared: Application status display mode

  To change the current setting, click the check box.

  For details about the functionality of the digital LEDs for status indications, including its display modes, see *4.4 Functionality of the Digital LEDs for Status Indications*.

(5) Pop-up notification settings

You can set up the pop-up notification settings. By clicking **Advanced**, you can configure the advanced settings of this feature.



RAS Setup

Shutdown setting

☑ Automatically shutdown if fan failure has been detected.

□ Automatically shutdown if abnormally high temp. has been detected.

☑ Automatically shutdown if remote shutdown is requested.

Watchdog timer setting

○ Not used.

○ Retriggered by application program.

● Automatic retrigger. Timeout 60 sec

Retrigger interval 20 sec

Drive failure prediction setting (SMART)

☑ Function is available.

Drive usage monitoring setting

☑ Function is available.

Digital LED setting

☑ Show Hardware status.

Popup notification setting

□ Function is available.

Figure 3-7 Items in the pop-up notification settings

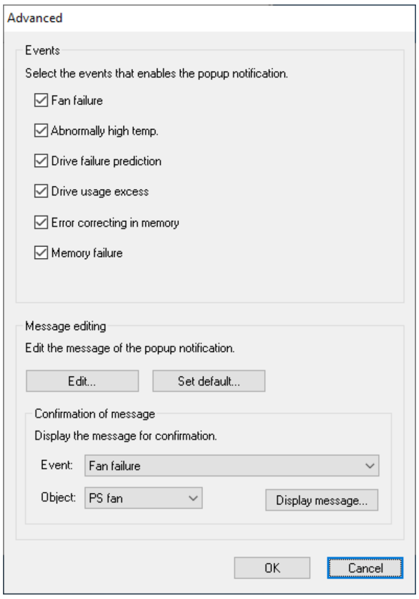
- **"Function is available"** check box
  - Selected: Pop-up notifications are enabled.
  - Cleared: Pop-up notifications are disabled (factory default setting).

  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.

● **Advanced** button

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



Advanced

Events

Select the events that enables the popup notification.

☑ Fan failure

☑ Abnormally high temp.

☑ Drive failure prediction

☑ Drive usage excess

☑ Error correcting in memory

☑ Memory failure

Message editing

Edit the message of the popup notification.

Confirmation of message

Display the message for confirmation.

Event: Fan failure

Object: PS fan

Figure 3-8 Advanced pop-up notification settings

[Events]

- **Fan failure**
- **Abnormally high temp.**
- **Drive failure prediction**
- **Drive usage excess**
- **Error correcting in memory**
- **Memory failure**

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 closes, and then the changes are lost.

(6) Applying the changes you made in (1) to (5) or discarding the changes

If you changed the settings in (1) to (5) and want to apply the changes, click **OK**. The RAS Setup window closes, and then the settings are applied immediately. To discard the changes, click **Cancel**. The RAS Setup window closes, and then the changes are lost.

3.1.4 Editing pop-up notification messages

(1) Editing pop-up notification messages

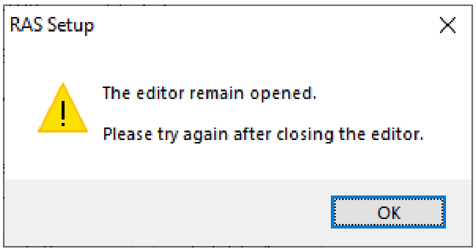
If you want to edit the messages that are used for pop-up notifications, click **Edit.** Notepad launches, and the message definition file for pop-up notifications is opened. Edit the messages in the following format.

NOTE

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

- Clicking **Edit.**
- Clicking **Set default.**
- Clicking **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.



RAS Setup

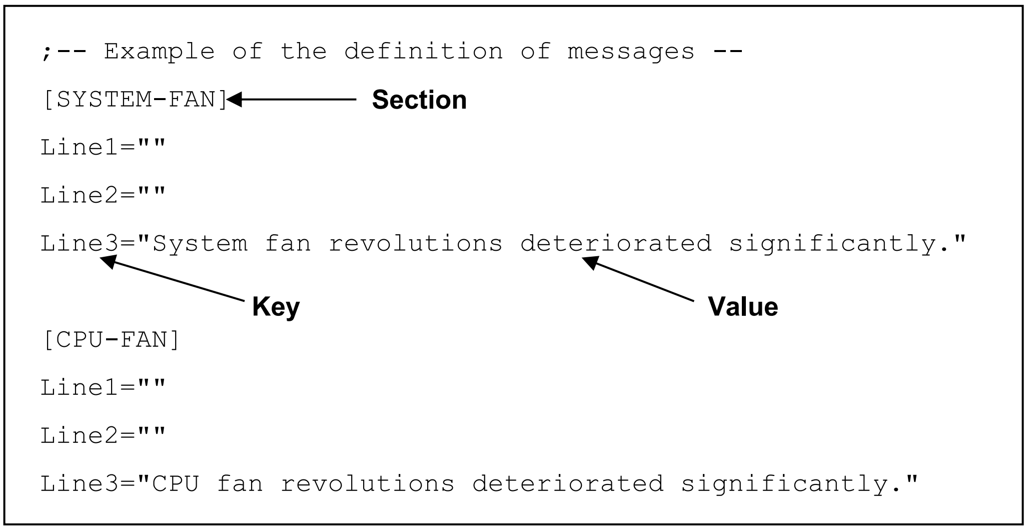
The editor remain opened.

Please try again after closing the editor.

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 --
[SYSTEM-FAN]
Line1=""
Line2=""
Line3="System fan revolutions deteriorated significantly."

[CPU-FAN]
Line1=""
Line2=""
Line3="CPU fan revolutions deteriorated significantly."
```

Figure 3-9 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
`[SYSTEM-FAN]`	A message displayed when a system fan failure is detected
`[CPU-FAN]`	A message displayed when a CPU fan failure is detected.
`[TEMP]`	A message displayed when an abnormal temperature is detected inside the chassis
`[DRV1-SMART]`#1	A message displayed when a drive failure prediction (SMART) is detected for drive bay 1
`[DRV2-SMART]`#1	A message displayed when a drive failure prediction (SMART) is detected for drive bay 2
`[DRV1-OVERRUN]`#1	A message displayed when the value of the drive power-on (used) hours exceeds the defined value for drive bay 1
`[DRV2-OVERRUN]`#1	A message displayed when the value of the drive power-on (used) hours exceeds the defined value for drive bay 2
`[DIMM1-ERR]`	A message displayed when error corrections are detected frequently in DIMM1
`[DIMM2-ERR]`	A message displayed when error corrections are detected frequently in DIMM2
`[DIMM1- FAILURE]`	A message displayed when a memory failure might have occurred in DIMM1
`[DIMM2- FAILURE]`	A message displayed when a memory failure might have occurred in DIMM2

#1: Regardless of the installed drive, this section name is fixed.

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 to Line5 for each section.

If you specify keys other than Line1 to 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 51st 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 this equipment continues to be used while an error persists, the system might be significantly affected.

(2) Checking pop-up notification messages

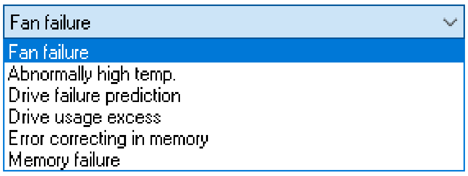
You can check the changes you made in the message for each of the following items:

- Fan failure
- Abnormally high temp.
- Drive failure prediction
- Drive usage excess (When the drive power-on (=used) hours exceeds the threshold)
- Error correcting in memory (When memory correction is detected frequently)
- Memory failure

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

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

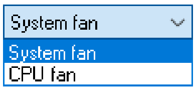
   In this list, only the items under **Events** selected by using the check boxes appear. If no items under **Events** have been selected by using the check boxes, you cannot select an item from the event list.



Example of selecting fan failure

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.



Example of selecting the system fan

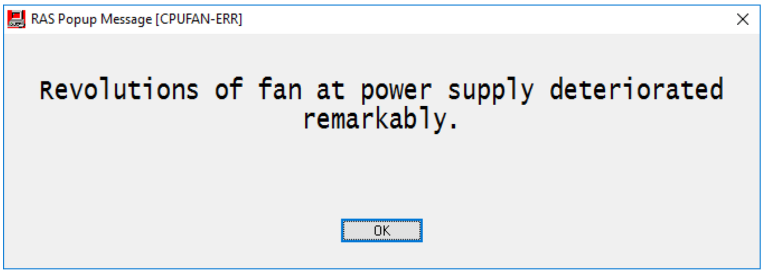
The following table shows items displayed in the object list for each option you select in the event list.

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
Fan failure	System fan,CPU fan
Abnormally high temp.	Temperature inside the chassis
Drive failure prediction	Drive bay1, Drive bay2
Drive usage excess	
Error correction in memory	DIMM1, DIMM2
Memory failure	

3. Click **Display message**

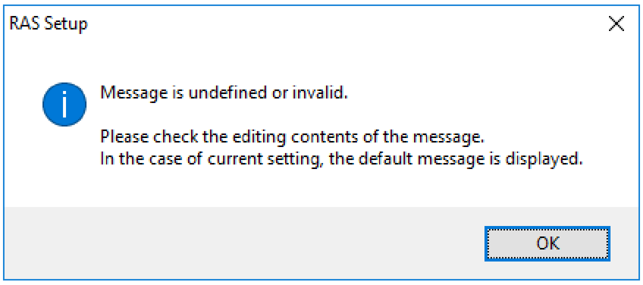
A pop-up notification message appears based on the changes you made. After you confirm the message, click **OK**.



RAS Popup Message [CPUFAN-ERR]

Revolutions of fan at power supply deteriorated remarkably.

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.



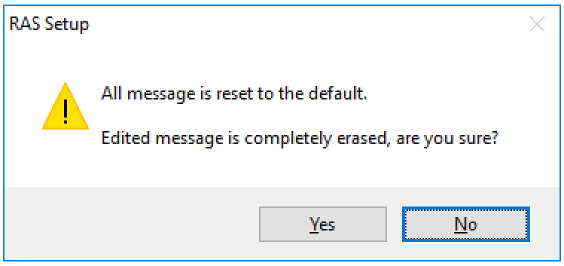
RAS Setup

Message is undefined or invalid.

Please check the editing contents of the message.
In the case of current setting, the default message is displayed.

(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.



RAS Setup

All message is reset to the default.

Edited message is completely erased, are you sure?

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

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 *4.6 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 by using the digital LEDs for status indications on the front of this equipment

The digital LEDs for status indications are on the front of this equipment, and notify users when errors occur in the hardware of this equipment. These LEDs can also be used by user applications to, for example, notify maintenance personnel of failures by displaying a code. For details, see *4.4 Functionality of the Digital LEDs for Status Indications*.

(5) 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 *4.5 Remote Notifications*.

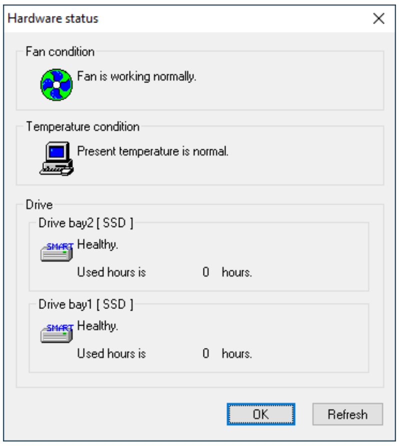
4.1 Hardware Status Window

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 fan
- Condition of the temperature inside the chassis
- Drive failure prediction (SMART monitoring) condition
- Drive power-on (used) hours



Hardware status

Fan condition

Fan is working normally.

Temperature condition

Present temperature is normal.

Drive

Drive bay2 [SSD]

Healthy.

Used hours is 0 hours.

Drive bay1 [SSD]

Healthy.

Used hours is 0 hours.

Figure 4-1 Hardware status window

NOTICE

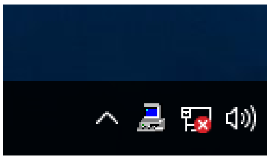
When the Hardware status window shows an error in the hardware, resolve the problem causing the error immediately.

NOTE

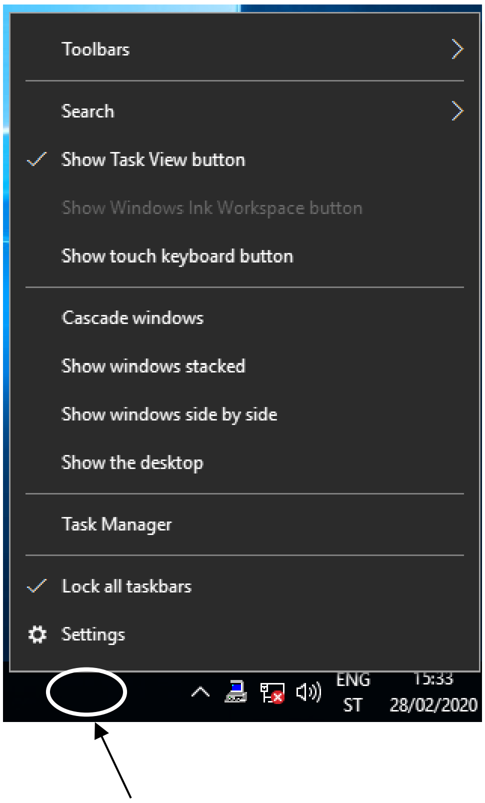
- For information about how to replace a replaceable component, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).
- This window displays only the internal drives that are recognized at OS startup. If you connect a new drive or replace a drive with a new drive, for example, for maintenance, it might take a long time to recognize the new drive the first time the equipment starts up after the new drive is connected, and information about the drive might not be displayed. In this case, 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.



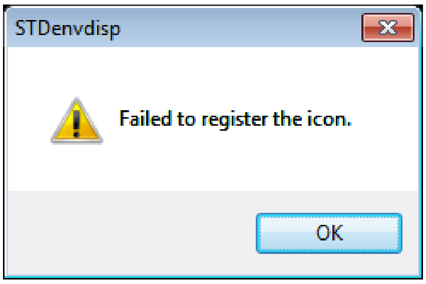
Note that when you are using the factory default settings, the icon does not appear. If you click the arrow at the side of the notification area, the icon appears. To have the icon always appear in the notification area of the taskbar, right-click the taskbar, and then from the menu that appears, click **Settings**. Then, in the window that appears, click **Select which icons appear on the taskbar**, and then turn on the icon for **envdisp MFC Application**.



Right-click the taskbar to display the menu.

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.



Failed to register the 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.

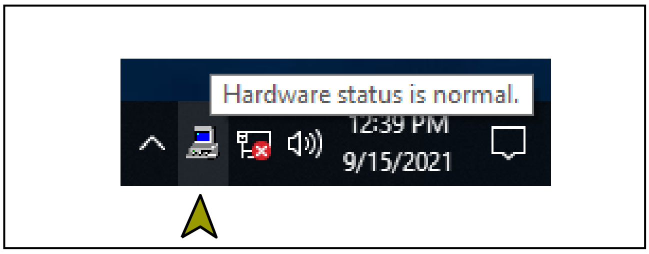
Table 4-1 Hardware status icon

No.	Hardware status	Icon	Description of the icon
1	Normal	🖥	The hardware status is normal.
2	Error	🟥	A fan failure was detected.
3			An abnormal temperature was detected.
4			A fan failure and an abnormal temperature were detected.
5			A fan failure was detected. A drive failure is possible.
6			An abnormal temperature was detected. A drive failure is possible.
7			A fan failure and an abnormal temperature were detected. A drive failure is possible.
8			A fan failure was detected. The power-on (used) hours exceeded the threshold.
9			An abnormal temperature was detected. The power-on (used) hours exceeded the threshold.
10			A fan failure and an abnormal temperature were detected. The power-on (used) hours exceeded the threshold.
11	Caution	⚠	A drive failure is possible.
12			The power-on (used) hours exceeded the threshold.

No. 5 to 7 and 11: When the detections of both power-on (used) hours exceeded the threshold and drive failure prediction (SMART) occur at the same time, the icon does not indicate that the power-on (used) hours exceeded the threshold.

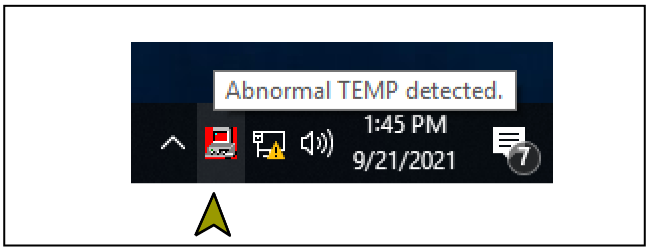
No.11 and 12: If a hardware status error is detected at the same time, the icon for a hardware status error is displayed.

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.



Hardware status is normal.

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

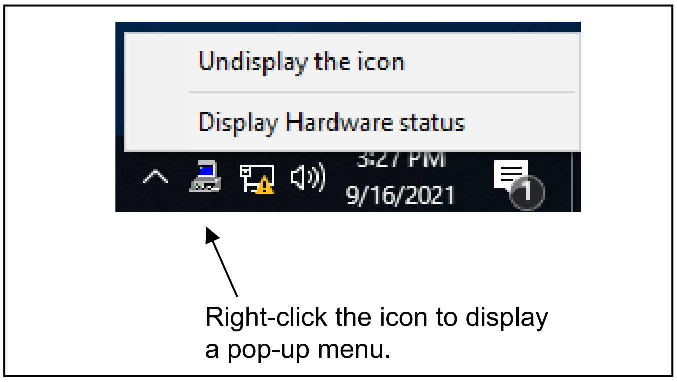


Abnormal TEMP detected.

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.



Right-click the icon to display a pop-up menu.

Figure 4-4 Menu of the hardware status icon

- **Display hardware status**

  Click this to display the Hardware status window.

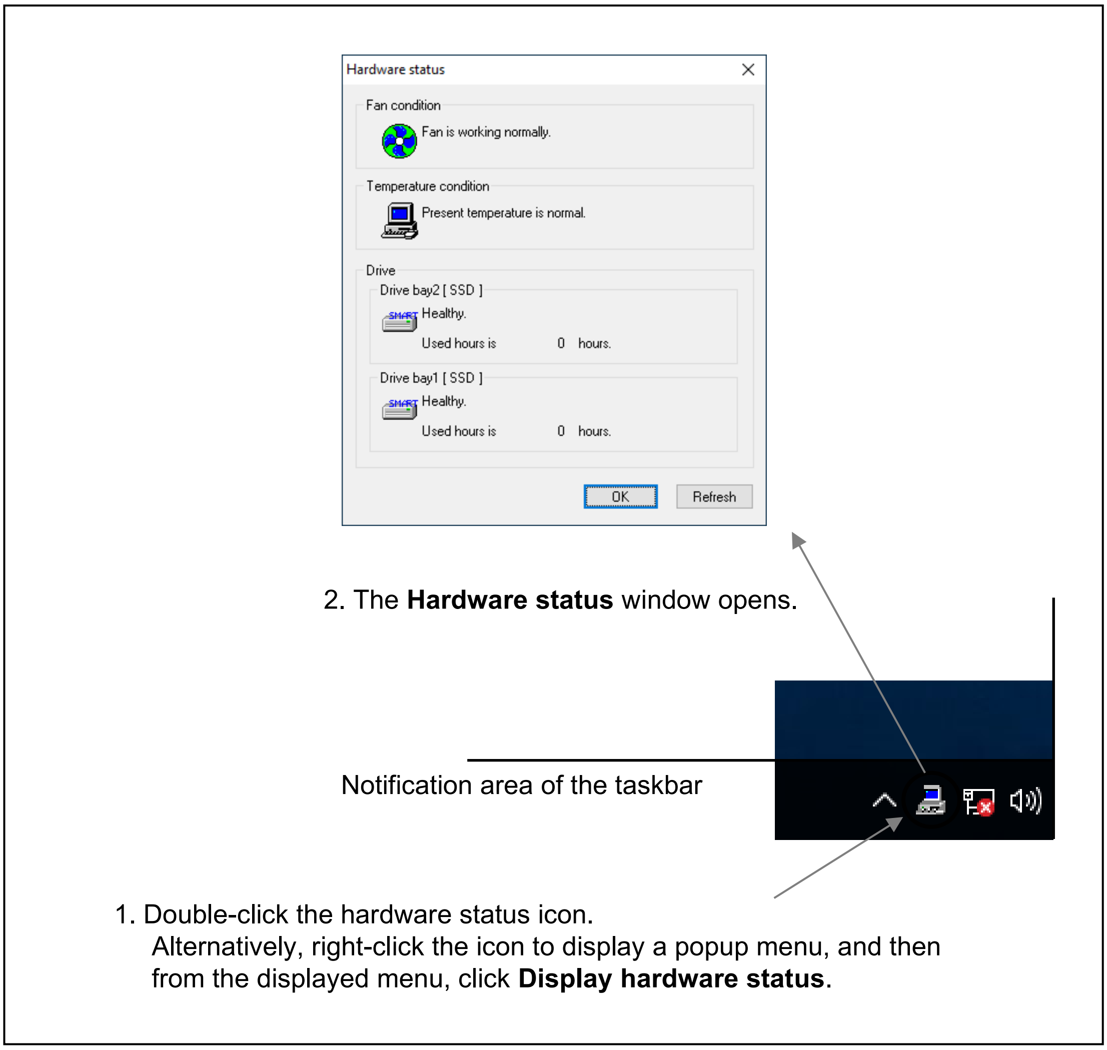
- **Undisplay the icon**

  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.



Hardware status

Fan condition

Fan is working normally.

Temperature condition

Present temperature is normal.

Drive

Drive bay2 [SSD]

Healthy.

Used hours is 0 hours.

Drive bay1 [SSD]

Healthy.

Used hours is 0 hours.

2. The **Hardware status** window opens.

Notification area of the taskbar

1. Double-click the hardware status icon.
   Alternatively, right-click the icon to display a popup menu, and then from the displayed menu, click **Display hardware status**.

Figure 4-5 Starting the Hardware status window

(1) Description of the window

1. Fan condition

This shows the current status of the fans.

Table 4-2 Fan condition and provided information

Fan condition	Icon	Information
Normal	🟩	The fan is working normally.
Excessively low rotation speed	🟥	An excessively low rotation speed was detected. For details, see the event log.

2. Temperature condition

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

Table 4-3 Temperature condition and provided information

Temperature condition	Icon	Information
Normal	⬜	The current temperature is normal.
Abnormally high temperature	🟥	The temperature inside the chassis has exceeded the maximum.

3. Drive condition

This shows the current status of the drives. In the following areas, the drive conditions of drive bay 1 and drive bay 2 are indicated.

Table 4-4 Drive condition and provided information

No.	Drive condition	Icon	Information
1	Normal	SMART	This condition is healthy.
2	Failure prediction by SMART	SMART ⚠	A failure might be imminent.
3	Excessive drive usage	SMART ℹ	The hours that the drive has been used exceed the threshold.
4	Error	?	The condition of the drive is unknown.
5	Not installed		No drive is installed.

NOTICE

When the failure of a drive is anticipated, the drive might experience hardware failure in the near future. We recommend backing up the data and replacing the drive. For information about how to replace a drive, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

4. Drive type

This shows the type of the drive.

Table 4-5 Drive type and provided information

No.	Drive type	Provided information
1	SSD	`[SSD]`
2	Unknown	`[---]`

5. Drive power-on (used) hours

The number of hours that the drive has been used (the cumulative hours to date) is displayed. The value of the cumulative hours is updated every hour between 0 and 100 hours, and every 10 hours after that. The range of the cumulative hours that can be displayed is from 0 to 99990 (hours). If drive usage monitoring is disabled, the number of hours that the drive has been used cannot be displayed.

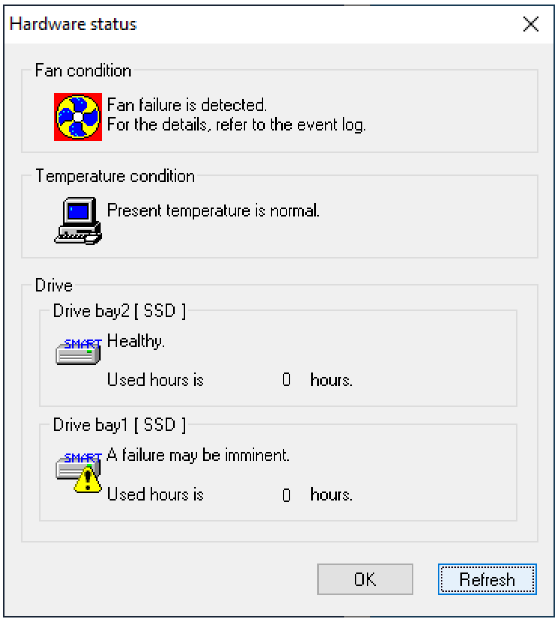
6. **Refresh** button

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

7. **OK** button

Click this button to close the Hardware status window.

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



Hardware status

Fan condition

Fan failure is detected.
For the details, refer to the event log.

Temperature condition

Present temperature is normal.

Drive

Drive bay2 [SSD]

Healthy.

Used hours is 0 hours.

Drive bay1 [SSD]

A failure may be imminent.

Used hours is 0 hours.

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 events such as hardware failures by monitoring when the event objects are set to the signaled state.

Event objects are reset to the non-signaled 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 4-6 shows a list of the events to be reported to a user and their respective event objects.

Table 4-6 Reported events

No.	Event	Event object name
1	A system fan failure occurred.	`W2KRAS_SYSFAN_ERR_EVENT`
2	A CPU fan failure occurred.	`W2KRAS_CPUFAN_ERR_EVENT`
3	The temperature inside the chassis became abnormal.	`W2KRAS_TEMP_ERR_EVENT`
4	A remote shutdown request was generated.	`W2KRAS_RMTSTDN_EVENT`
5	SMART anticipated a failure in one of the drives.	`W2KRAS_HDD_PREDICT_EVENT`
6	The power-on (used) hours exceeded the threshold for one of the drives.	`HFW_HDD_OVERRUN_EVENT`
7	Frequent error corrections occurred in one of the memory slots.	`HFW_MEMORY_ERR_EVENT`
8	A possible failure was detected in one of the memory slots.	

No. 7 and 8: The same event object name is assigned because both events indicate a possible memory failure.

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

A sample program in C (`FanErr.c`) is available to show how to monitor event objects. For the file name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

4.3 Pop-up Notifications

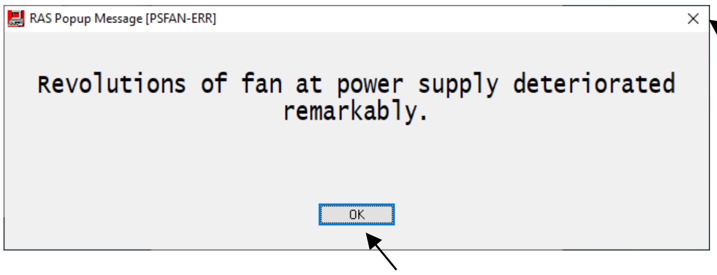
4.3.1 Overview

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:

- ● A fan failure is detected.
- ● An abnormally high temperature is detected inside the chassis.
- ● A drive failure is anticipated (by SMART).
- ● The drive power-on (used) hours exceeded the threshold.
- ● An error correction in memory is frequently detected.
- ● A possible memory failure is detected.

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



RAS Popup Message [PSFAN-ERR]

Revolutions of fan at power supply deteriorated remarkably.

Click **Close** ([×] button) to close the window.

Click **OK** to close the window.

Figure 4-7 Example pop-up message notification

4.3.2 Messages to be displayed

Table 4-7 shows a list of the pop-up notification messages generated by this feature.

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

Table 4-7 Messages to be displayed

No.	Event	Pop-up notification message
1	A system fan failure occurred.	The rotation speed of the system fan became excessively lower.
2	A CPU fan failure occurred.	The rotation speed of the CPU fan became excessively lower.
3	The temperature inside the chassis became abnormal.	The temperature exceeded the prescribed value.
4	SMART anticipated a drive failure.	A failure might be imminent on the drive in drive bay *%1*.
5	The drive power-on (used) hours exceeded the threshold.	The number of hours that the drive has been used in drive bay *%1* exceeded the prescribed value.
6	Frequent memory error corrections occurred.	Error corrections occurred with high frequency in DIMM *%1*.
7	A possible memory failure was detected.	DIMM *%1* might have failed.

No. 4 and 5: *%1* denotes the drive bay number.

No. 6 and 7: *%1* denotes the DIMM slot number.

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 Functionality of the Digital LEDs for Status Indications

4.4.1 Overview

When an event that must be reported to a user occurs, such as a hardware failure, this feature notifies the user of the event by displaying a code via the digital LEDs for status indications located on the front of this equipment. By using this feature, a user can identify the occurrence of an event such as a hardware failure.

In addition, a user application can output any code on the digital LEDs for status indications by using library functions. A code is displayed as a two-digit hexadecimal number.

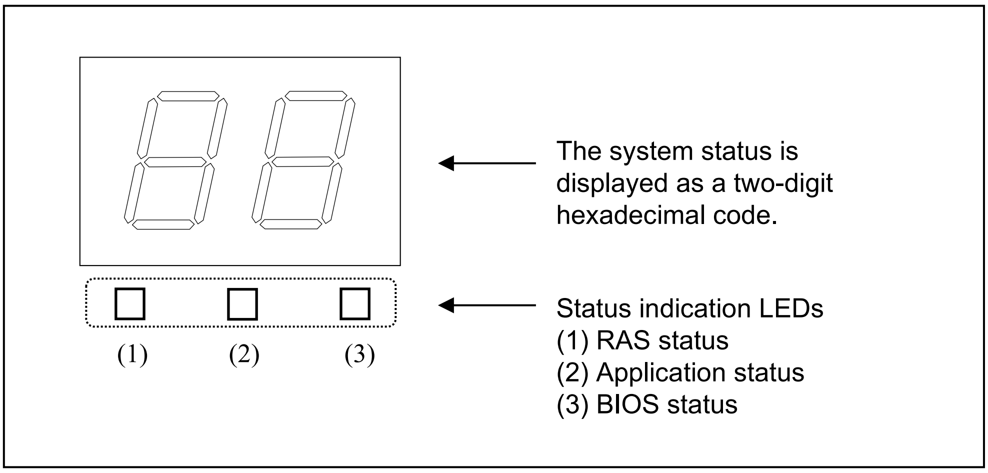


Figure 4-8 Digital LEDs for status indications

4.4.2 Status codes to be displayed

(1) Hardware status codes

A hardware status code is displayed when an error has occurred in the hardware status of this equipment. If the hardware status is normal, no hardware status code is displayed.

When a hardware status code is displayed, the leftmost LED in the status indication LEDs is lit.

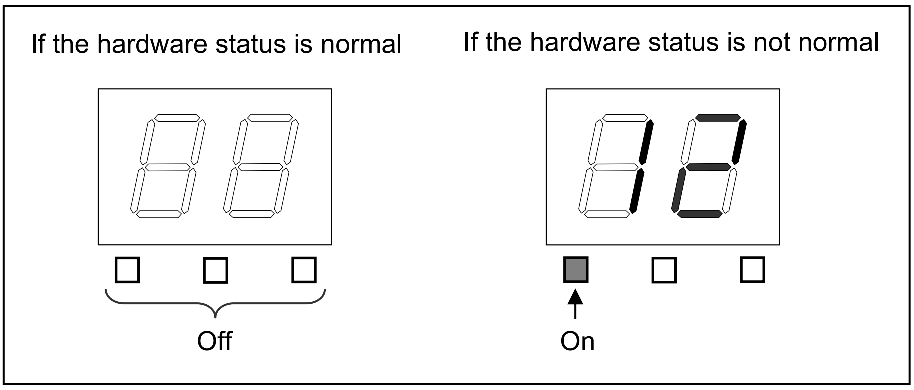

If the hardware status is normal
If the hardware status is not normal
Off
On

Figure 4-9 Hardware status codes

Table 4-8 shows a list of hardware status codes.

Table 4-8 Hardware status codes

No.	Code	Cause	Priority
1	12	A system fan failure occurred.	1
2	13	A CPU fan failure occurred.	
3	21	The temperature inside the chassis became abnormal.	2
4	31	SMART anticipated a drive failure in drive bay 1.	3
5	32	SMART anticipated a drive failure in drive bay 2.	

If multiple failures occur at the same time, the status code with the highest priority (the status code with the smallest value in the *Priority* column of Table 4-8) is displayed. If multiple failures with the same priority occur at the same time, the status code of the failure that was detected last is displayed.

(2) Application status codes

An application status code is displayed by a user application by using the library functions provided by this feature.

When an application status code is displayed, the center LED of the status indication LEDs is lit.

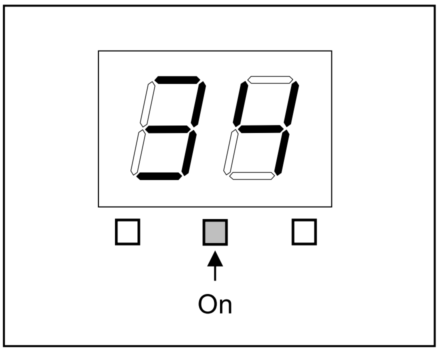


Figure 4-10 Application status code

(3) Codes displayed for a bug check (blue screen)

These error codes are displayed when a bug check (blue screen) occurs during system operation. Regardless of the status display mode, this error code is displayed with the highest priority. For information about the status display modes, see *4.4.3 Status display modes*.

When a bug check (blue screen) occurs, the status indication LEDs are lit as shown in Figure 4-11.

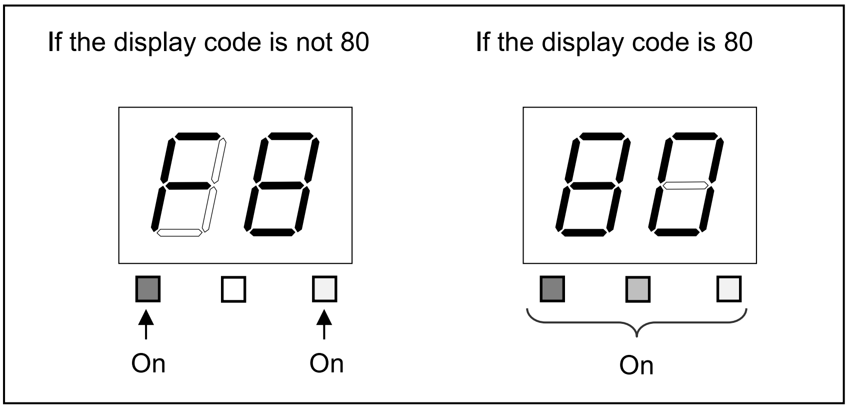


Figure 4-11 Status indication LEDs when a bug check (blue screen) occurs

Table 4-9 shows a list of codes displayed when a bug check (blue screen) occurs.

Table 4-9 Codes displayed when a bug check (blue screen) occurs

No.	Code	Cause
1	F8	A parity error occurred in the PCI bus.
2	F9	
3	FA	An uncorrectable memory error occurred.
4	Fb	The **NMI** button is pressed.
5	80	An error other than the preceding occurred.

For information about the priority used to display each code, see *4.4.4 Priorities of codes to be displayed*.

4.4.3 Status display modes

This feature has two display modes: hardware status display mode and application status display mode.

Table 4-10 Status display modes

Status display mode	Description
Hardware status display mode	When the hardware status is normal, an application status code is displayed. When the hardware status has an error, a hardware status code has higher display priority.
Application status display mode	Only an application status code is displayed. Even when the hardware status has an error, the hardware status code is not displayed.

The status display mode can be configured in the RAS Setup window. The factory default setting is the hardware status display mode.

For information about how to use the RAS Setup window, see *3.1.3 Using the RAS Setup window*.

Note that if the hardware status is normal and there is no application status code to display, the LEDs are all off.

Figure 4-12 shows an example of what is displayed in the RAS status display mode. The dotted lines in Figure 4-12 denote the respective display modes, and the thick line denotes the transition of the state of the LEDs.

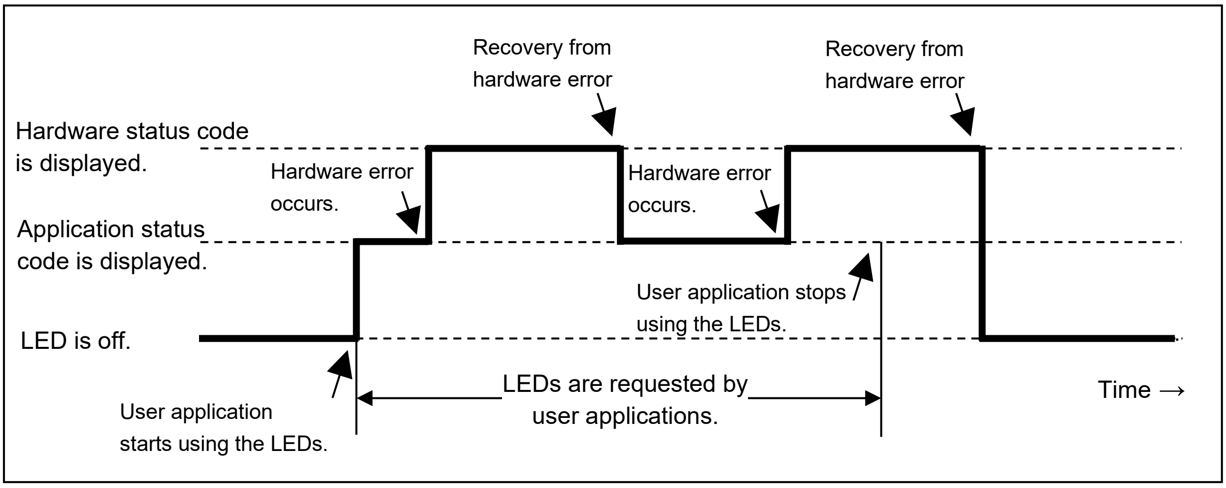

Recovery from hardware error
Recovery from hardware error
Hardware status code is displayed.
Hardware error occurs.
Hardware error occurs.
Application status code is displayed.
User application stops using the LEDs.
LED is off.
LEDs are requested by user applications.
Time →
User application starts using the LEDs.

Figure 4-12 Example of the status display mode

4.4.4 Priorities of codes to be displayed

The following shows the priorities of the codes displayed by this feature.

(1) In the hardware status display mode

Table 4-11 Priority of codes in the hardware status display mode

Code class	Priority
Codes displayed for a bug check (blue screen)	1
Hardware status codes	2
Application status codes	3

- The code displayed when a bug check (blue screen) occurs has the highest priority.
- The priority of a hardware status code differs depending on its cause. For details, see *4.4.2 Status codes to be displayed*.

(2) In the application status display mode

Table 4-12 Priority of codes in the application status display mode

Code class	Priority
Codes displayed for a bug check (blue screen)	1
Application status codes	2

- The code displayed when a bug check (blue screen) occurs has the highest priority.

4.4.5 Control functions for the digital LEDs for status indications

The following functions are offered as library functions for controlling the digital LEDs for status indications. For details about the library functions, see *6.1 RAS library*.

● To display an application status code: Use the `SetStCode7seg` function.

● To turn off the currently displayed application status code: Use the `TurnOff7seg` function.

● To set the status display mode: Use the `SetMode7seg` function.

4.5 Remote Notifications

4.5.1 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 *4.5.3 Enabling remote notifications*.

4.5.2 Hardware conditions that can be acquired by using remote notifications

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

- Fan condition
- Condition of the temperature inside the chassis
- Drive condition
- Memory condition
- I/O status of the general-purpose external contacts
- RAS features settings
- Operating 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) Fan condition

- Normal → Error
- Error → Normal

(2) Condition of the temperature inside the chassis

- Normal → Error
- Error → Normal

(3) Drive condition

- Normal → Failure anticipated
- Normal → Used hours exceeded the threshold

(4) Memory condition

- Normal → Frequent error corrections
- Normal → Possible failure detected
- Frequent error corrections → Normal

(5) Operating mode

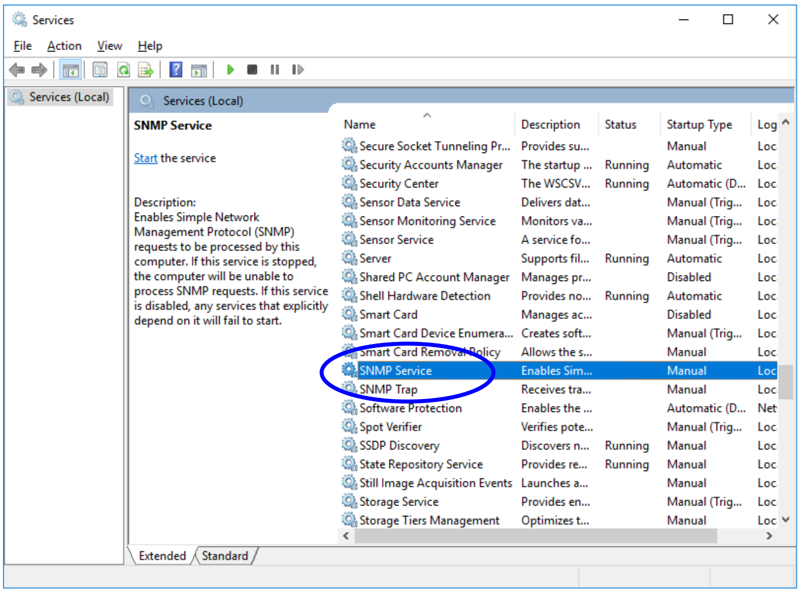
- HF-W stopped → Started in normal mode
- Running in normal mode → Running in simulation mode

4.5.3 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 done so already, sign in to the computer as an administrator.
2. Open the Services window:
   - Open the **Control Panel**.
   - Double-click **System and Security**, **Administrative Tools**, and **Services**.
3. Double-click **SNMP Service** to open the SNMP Service Properties window.



SNMP Service

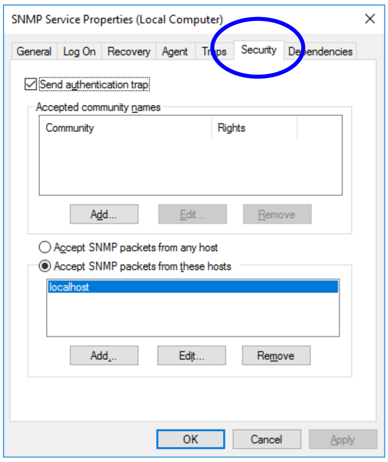
Start the service

Description:
Enables Simple Network Management Protocol (SNMP) requests to be processed by this computer. If this service is stopped, the computer will be unable to process SNMP requests. If this service is disabled, any services that explicitly depend on it will fail to start.

Name	Description	Status	Startup Type	Log
Secure Socket Tunneling Pr...	Provides su...		Manual	Loc
Security Accounts Manager	The startup ...	Running	Automatic	Loc
Security Center	The WSCSV...	Running	Automatic (D...	Loc
Sensor Data Service	Delivers dat...		Manual (Trig...	Loc
Sensor Monitoring Service	Monitors va...		Manual (Trig...	Loc
Sensor Service	A service fo...		Manual (Trig...	Loc
Server	Supports fil...	Running	Automatic	Loc
Shared PC Account Manager	Manages pr...		Disabled	Loc
Shell Hardware Detection	Provides no...	Running	Automatic	Loc
Smart Card	Manages ac...		Disabled	Loc
Smart Card Device Enumera...	Creates soft...		Manual (Trig...	Loc
Smart Card Removal Policy	Allows the s...		Manual	Loc
SNMP Service	Enables Sim...		Manual	Loc
SNMP Trap	Receives tra...		Manual	Loc
Software Protection	Enables the ...		Automatic (D...	Net
Spot Verifier	Verifies pote...		Manual (Trig...	Loc
SSDP Discovery	Discovers n...	Running	Manual	Loc
State Repository Service	Provides re...	Running	Manual	Loc
Still Image Acquisition Events	Launches a...		Manual	Loc
Storage Service	Provides en...		Manual (Trig...	Loc
Storage Tiers Management	Optimizes t...		Manual	Loc

(2) SNMP security configuration

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



☑ Send authentication trap

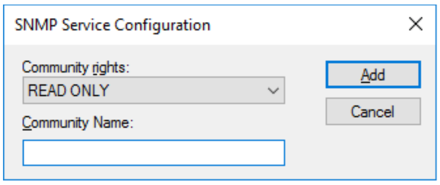
○ Accept SNMP packets from any host

● Accept SNMP packets from these hosts

localhost

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**.



Community rights: READ ONLY

Community Name:

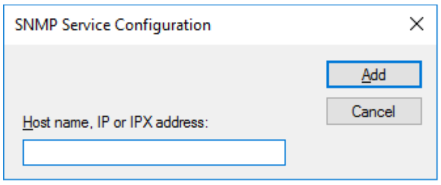
4. Specify whether to accept SNMP packets from hosts.

   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**.

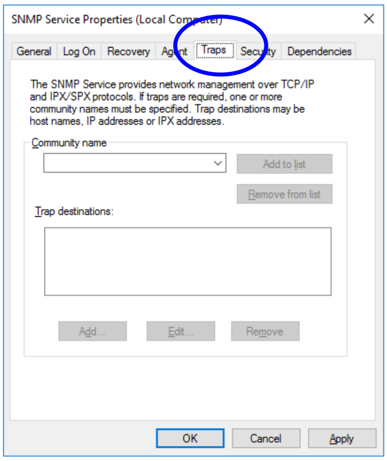


Host name, IP or IPX address:

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

(3) SNMP trap configuration

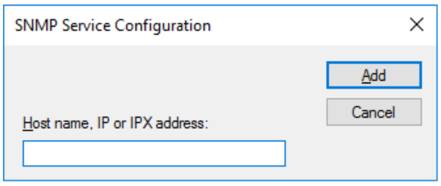
1. In the SNMP Service Properties window, click the **Traps** tab.



The SNMP Service provides network management over TCP/IP and IPX/SPX protocols. If traps are required, one or more community names must be specified. Trap destinations may be host names, IP addresses or IPX addresses.

2. Under **Community name**, enter the name of the community that trap messages are to be sent to, and then click **Add to list**.

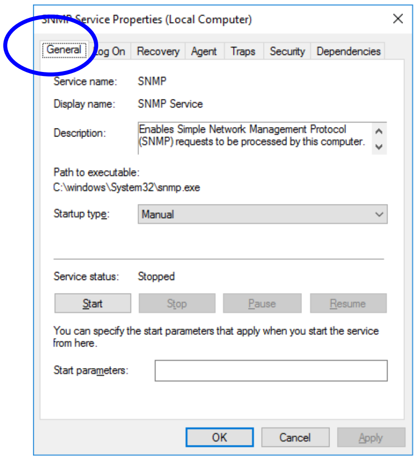
3. Under **Trap destinations**, click **Add**. The SNMP Service Configuration window appears. Enter the host name or the IP or IPX address of a destination that you want to send traps to, and then click **Add**.



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

(4) Starting the SNMP service

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



SNMP Service Properties (Local Computer)

Service name: SNMP

Display name: SNMP Service

Description: Enables Simple Network Management Protocol (SNMP) requests to be processed by this computer.

Path to executable:
C:\windows\System32\snmp.exe

Startup type: Manual

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

2. Click **Start**. The SNMP service starts and remote notifications for hardware statuses are enabled.

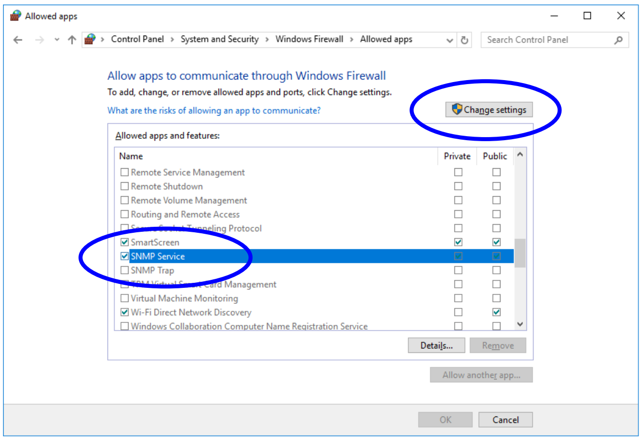
3. To start the SNMP service automatically upon the next OS startup, in the **Startup type** list, select **Automatic**.

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

NOTE

- If there is an error for which notifications by using a trap is configured when the SNMP service starts, the trap is sent at this timing.
- If Windows Firewall is configured to block the SNMP service, you cannot acquire the hardware status via a remote device. If the firewall is set up to block the SNMP service, perform the following procedure to undo the setting.
  Note that the SNMP service can pass through the Windows Firewall by default. In that case, you do not have to perform the following procedure:
  1. If you have not done so already, sign in to the computer as an administrator.
  2. Open the **Control Panel**, and then click **System and Security**.
  3. Under **Windows Firewall**, click **Allow an app or feature through Windows Firewall**.

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



Allow apps to communicate through Windows Firewall

To add, change, or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

Allowed apps and features:

Name	Private	Public
□ Remote Service Management	□	□
□ Remote Shutdown	□	□
□ Remote Volume Management	□	□
□ Routing and Remote Access	□	□
□ Secure Socket Tunneling Protocol	□	□
☑ SmartScreen	☑	☑
☑ SNMP Service	□	☑
□ SNMP Trap	□	□
□ TPM Virtual Smart Card Management	□	□
□ Virtual Machine Monitoring	□	□
☑ Wi-Fi Direct Network Discovery	□	☑
□ Windows Collaboration Computer Name Registration Service	□	□

5. Click **OK**.

4.5.4 Objects in the extended MIB for HF-W

To acquire the hardware status of this equipment via 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 4-13 shows a list of 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.

Note that 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 4-13 Objects related to the hardware status

(1/3)

No.	Object	Object number	Description	Description of the values
1	`hfwFan.fanNumber`	1.1.0	Number of monitored fans	--
2	`hfwFan.fanTable.fanEntry.fanIndex`	1.2.1.1.*N*	Index number of `fanEntry`	--
3	`hfwFan.fanTable.fanEntry.fanName`	1.2.1.2.z	Fan name	`System fan`: System fan `CPU fan`: CPU fan
4	`hfwFan.fanTable.fanEntry.fanStatus`	1.2.1.3.z	Fan condition	1: Normal 2: Error
5	`hfwTemp.tempNumber`	2.1.0	Number of monitored temperatures	--
6	`hfwTemp.tempTable.TempEntry.tempIndex`	2.2.1.1.*N*	Index number of `tempEntry`	--
7	`hfwTemp.tempTable.TempEntry.tempName`	2.2.1.2.z	Name of the temperature to be monitored	`Internal temperature`: Temperature inside the chassis
8	`hfwTemp.tempTable.TempEntry.tempStatus`	2.2.1.3.z	Temperature condition	1: Normal 2: Error
9	`hfwHdd.hddNumber`	3.1.0	Number of monitored drive bays	--
10	`hfwHdd.hddTable.hddEntry.hddIndex`	3.2.1.1.*N*	Index number of `hddEntry`	--

(2/3)

No.	Object	Object number	Description	Description of the values
11	hfwHdd.hddTable.hddEntry.hddStatus	3.2.1.2.z	Drive condition	1: Normal 2: Not installed 3: Failure anticipated 5: Used hours exceeded the threshold 99: Unknown
12	hfwHdd.hddTable.hddEntry.hddUseTime	3.2.1.3.z	Drive power-on (used) hours (in hours)	--
13	hfwHdd.hddTable.hddEntry.hddType	3.2.1.4.z	Drive type	1: HDD 2: SSD 99: Unknown
14	hfwMem.memNumber	5.1.0	Number of monitored memory slots	--
15	hfwMem.memTable.memEntry.memIndex	5.2.1.1.*N*	Index number of memEntry	--
16	hfwMem.memTable.memEntry.memName	5.2.1.2.z	DIMM name	DIMM1: DIMM1 slot DIMM2: DIMM2 slot
17	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.) 3: Not installed
18	hfwGenDI.gendiNumber	6.1.0	Number of general-purpose external contact inputs	--
19	hfwGenDI.gendiTable.gendiEntry.gendiIndex	6.2.1.1.*N*	Index number of gendiEntry	--

(3/3)

No.	Object	Object number	Description	Description of the values
20	hfwGenDI.gendiTable.gendiEntry.gendiName	6.2.1.2.z	Name of the general-purpose external contact input	GENDI: GENDI contact GENDI0: GENDI0 contact GENDI1: GENDI1 contact GENDI2: GENDI2 contact
21	hfwGenDI.gendiTable.gendiEntry.gendoStatus	6.2.1.3.z	Input status of the general-purpose external contact	0: Open 1: Closed
22	hfwGenDO.gendoNumber	7.1.0	Number of general-purpose external contact outputs	--
23	hfwGenDO.gendoTable.gendoEntry.gendoIndex	7.2.1.1.*N*	Index number of gendoEntry	--
24	hfwGenDO.gendoTable.gendoEntry.gendoName	7.2.1.2.z	Name of the general-purpose external contact output	GENDO0: GENDO0 contact GENDO1: GENDO1 contact GENDO2: GENDO2 contact
25	hfwGenDO.gendoTable.gendoEntry.gendoStatus	7.2.1.3.z	Output status of the general-purpose external contact	0: Open 1: Closed

No. 1: For this equipment, the value is set to 2.

No. 5: For this equipment, the value is set to 1.

No. 9: The number of internal drives that can be installed in the HF-W is set for the number of monitored drive bays.
For this equipment, the value is set to 2.

No. 14: The number of memory slots is set for the number of monitored memory slots.
For this equipment, the value is set to 2.

No. 18: For this equipment, the value is set to 4.

No. 22: For this equipment, the value is set to 3.

Table 4-14 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 4-14 Objects related to the RAS feature settings

No.	Object	Object number	Description	Description of the values
1	hfwFanAutoShutdown	1	Automatic shutdown when a fan failure is detected	1: Enabled 2: Disabled
2	hfwTempAutoShutdown	2	Automatic shutdown when an abnormally high temperature is detected	1: Enabled 2: Disabled
3	hfwRemoteShutdown	3	Automatic shutdown when a remote shutdown is requested	1: Enabled 2: Disabled

Table 4-15 shows a list of the objects related to the 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 4-15 Objects related to the operating modes

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

Table 4-16 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 4-16 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`	5	Revision number of the extended MIB for HF-W	--

(2) Objects related to trap notifications

Table 4-17 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 4-17 Objects related to trap notifications (when an error occurs)

(1/2)

No.	Object	Trap number	Description	Notification data	
				Object used	Value
1	hfwFanError	1	A fan failure occurs.	fanName	Name of the failed fan
				fanStatus	2: Error
				hfwFanStMsg	Revolution of *%1* deteriorated remarkably.
2	hfwTempError	2	The temperature inside the chassis became abnormal.	tempName	Internal temperature
				tempStatus	2: Error
				hfwTempStMsg	Internal temperature exceeded prescribed value.
3	hfwSmartDetect	3	A drive failure is anticipated.	hddIndex	The drive bay number of the drive for which a failure is anticipated by SMART
				hddStatus	3: Failure anticipated
				hfwSmartStMsg	A failure may be imminent on the drive of the drive bay*%2*.
4	hfwHddOverRun	4	The drive power-on (used) hours exceeded the threshold.	hddIndex	The drive bay number of the drive for which the power-on hours exceeded the threshold
				hddStatus	5: Used hours exceeded the threshold
				hfwHddUseTimeStMsg	Used hours on the drive of the drive bay*%2* exceeded prescribed value.
5	hfwMemError	6	Frequent error corrections occur.	memName	Name of the memory slot with frequent error corrections
				memStatus	2: Error (Frequent error corrections occur)
				hfwMemStMsg	In the *%3*, error correcting have occurred with high frequency.

(2/2)

No.	Object	Trap number	Description	Notification data	
				Object used	Value
6	hfwMemFailure	8	A possible memory failure was detected.	memName	Name of the memory slot for which a possible memory failure was detected
				memStatus	2: Error (A possible memory failure is detected.)
				hfwMemFailMsg	*%3* failure might occur.

No. 1: *%1* denotes the name of the failed fan.

No. 3 and 4: *%2* denotes the drive bay number.

No .5 and 6: %3 denotes the name of the memory slot for which error corrections occur frequently or for which a possible memory failure was detected.

Table 4-18 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 4-18 Objects related to trap notifications (when the equipment has recovered from an error)

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

No. 1: *%1* denotes the name of the recovered fan.

No. 3: *%2* denotes the name of the memory slot that recovered from error corrections.

Table 4-19 shows a list of the objects related to trap notifications when the equipment starts in the 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 4-19 Objects related to trap notifications (operating modes)

No.	Object	Trap number	Description	Notification data	
				Object used	Value
1	hfwRasServic e Started	1	Started in normal mode	hfwRasMode	1: Normal mode
				hfwRasStartMsg	RAS Service is running.
2	hfwSimulatio n ModeStarted	2	Transition to simulation mode	hfwRasMode	2: Simulation mode
				hfwRasStartMsg	RAS Service switched to Simulation Mode.

4.5.5 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`

4.6 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 drive condition: Use the `hfwDiskStat` function.

CHAPTER 5 CONTROLLING THE HARDWARE

The RAS features can enable the following controls of this equipment:

(1) Automatic shutdown of the equipment

When a hardware error occurs or a remote shutdown request through the contact input is detected, the equipment can automatically shut down. For details, see *5.1 Automatic Shutdown of the Equipment*.

(2) 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 *5.2 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 because a fan failure or an abnormally high temperature is detected. Automatically shutting down the equipment can protect the internal parts, such as the processor, from thermal degradation and can prevent thermal runaway of the system due to a malfunction of this equipment. The equipment can also be automatically shut down by a remote shutdown signal input from an external device.

5.1.1 Automatic shutdown when a fan failure is detected

When the equipment detects an error in one of the fans, the equipment automatically shuts down.

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

NOTICE

- If this equipment continues to operate after a fan failure is detected, internal parts such as the processor will not cool sufficiently, which might cause the thermal runaway of the system due to a malfunction in the equipment, or result in damage to parts. If possible, enable the automatic shutdown feature.
- If the automatic shutdown feature is not used, have a user application detect fan failures by using a RAS event, and shut down the equipment when such failures are detected.

NOTE

For information about how to replace a fan, see the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

5.1.2 Automatic shutdown when an abnormally high temperature is detected

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. In 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 an abnormally high temperature by using a RAS event and shut down the equipment. For information about RAS events, see *4.2 RAS Event Notifications*.

NOTE

- If the temperature inside the chassis is high, parts can degrade rapidly due to heat. In this case, continuing to use this equipment is not advisable in terms of the equipment life. On the other hand, if the temperature is abnormally high when the fans are working properly, the abnormally high temperature is likely to 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 when you are using the factory default settings.
- If you continue to operate this equipment after an abnormally high temperature is detected and the temperature inside the chassis further rises to a dangerous level, the equipment forcibly shuts down and the power turns off regardless of whether this feature is disabled. This is intended to prevent the thermal runaway of the system and damage to parts.

5.1.3 Automatic shutdown when a remote shutdown request is detected through the contact input

If this feature is enabled, this equipment automatically shuts down when the remote shutdown contact (RMTSTDN contact) in the RAS external contact interface is closed. Using this feature allows you to shut down this equipment from a remote location.

- This feature can be enabled or disabled in the RAS Setup window. In the factory default settings, this feature is enabled. For details, see *3.1.3 Using the RAS Setup window*.
- Alternatively, a user application can detect a remote shutdown request through the contact input by using a RAS event and shut down the equipment. For information about RAS events, see *4.2 RAS Event Notifications*.

5.2 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 `GendoControlEx` function.
- To control the general purpose-external contact inputs: Use the `GetGendiEx` function.
- To control the digital LEDs for status indications: Use the `SetStCode7seg`, `TurnOff7seg`, and `SetMode7seg` functions.

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-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

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 shows a list of 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`	Retriggers, gets the status of, and stops the watchdog timer	
3	`GendoControlEx`	Opens and closes the general-purpose external contact outputs (`GENDO0`, `GENDO1`, and `GENDO2`)	
4	`GetGendiEx`	Acquires the status of the general-purpose external contact inputs (`GENDI`, `GENDI0`, `GENDI1`, and `GENDI2`)	
5	`MconWriteMessage`	Records a specified message (characters) in the log files of this equipment	
6	`GetMemStatus`	Acquires the status of the memory installed in this equipment	
7	`hfwDiskStat`	Acquires the status of a drive	`hfwras.dll`
8	`SetStCode7seg`	Outputs an application status code by using the digital LEDs for status indications	`ctrl7seg.dll`
9	`TurnOff7seg`	Clears an application status code from the digital LEDs for status indications	
10	`SetMode7seg`	Configures the status display mode of the digital LEDs for status indications	

These functions are offered by the DLLs (`w2kras.dll`, `hfwras.dll`, and `ctrl7seg.dll`).

NOTE

Do not copy or move `w2kras.dll`, `hfwras.dll`, or `ctrl7seg.dll` to another directory. If you do so, the RAS features of this equipment cannot run properly.

The functions offered by `w2kras.dll` and `ctrl7seg.dll` can be called from Visual Basic®, which requires .NET. When you call the functions numbered 1 to 7 and 10 to 12 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
%ProgramFiles%¥HFWRAS¥lib¥ctrl7seg.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
%ProgramFiles%¥HFWRAS¥include¥ctrl7seg.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)
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 = 0: The power to this equipment turns off after a shutdown.

reboot ≠ 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 file 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, select **Retriggered by applicatioin program** under **Watchdog timer setting** in the RAS Setup window. 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
`WDT_STAT (0x02)`	Acquires the status of 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 of 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 this 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 this value.
If *dwCmd* is `WDT_STAT`, the remaining time in seconds until the watchdog timer expires when the function is called is stored in the variable pointed to by *pdwCount*. If the value of the variable pointed to by *pdwCount* is 0 when the function returns, the watchdog timer has already timed out.
Note that the value of the variable pointed to by *pdwCount* when the function is called is ignored.

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

`W2KRAS_INVALID_PARAMETER`
`(0x2001)`
	The specified parameters contain an error.

`W2KRAS_WDT_NONMANUAL`
`(0x2002)`
	This function cannot be used because **Retriggered by applicatioin program** is not selected under **Watchdog timer setting** in the RAS Setup window.

`W2KRAS_NOT_INITIALIZE`
`(0x2005)`
	The startup of the RAS software is not yet complete.

`W2KRAS_INTERNAL_ERROR`
`(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 file 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 *2.7.2 Using a watchdog timer to monitor user programs*.

(2) Behavior of the WDTTO contact of the RAS external contact interface

This section describes the behavior of the WDTTO contact of the RAS external contact interface under the following conditions:

- When the equipment starts

  The WDTTO contact is closed.

- When the OS starts

  The WDTTO contact is closed when the OS starts. The WDTTO contact opens when the watchdog timer is retriggered by the automatic retriggering feature for the watchdog timer or by the `WdtControl` function.

- When the `WdtContol` function is called

  If *dwCmd* is `WDT_SET`:

  The WDTTO contact opens. If the watchdog timer is not retriggered afterward within the specified timeout period, the contact closes. If the watchdog timer is retriggered when the contact is closed, the WDTTO contact opens.

  If *dwCmd* is `WDT_STOP`:

  The WDTTO contact opens. In this case, the countdown of the watchdog timer is stopped and consequently the watchdog timer does not time out.

- When the automatic retriggering feature for the watchdog timer is used

  The WDTTO contact opens. This process for this function runs with the idle priority. If the amount of CPU time used by processes with a priority higher than the idle priority class exceeds a set value, the watchdog timer times out and the contact closes. Thereafter, when CPU time becomes available and the process for this function can run, the watchdog timer resumes being retriggered and the WDTTO contact opens.

Figure 6-1 shows an example of the behavior of the WDTTO contact when an application controls the watchdog timer by using the `WdtControl` function. In this example, the timeout of the watchdog timer specified by the `WdtControl` function is 10 seconds.

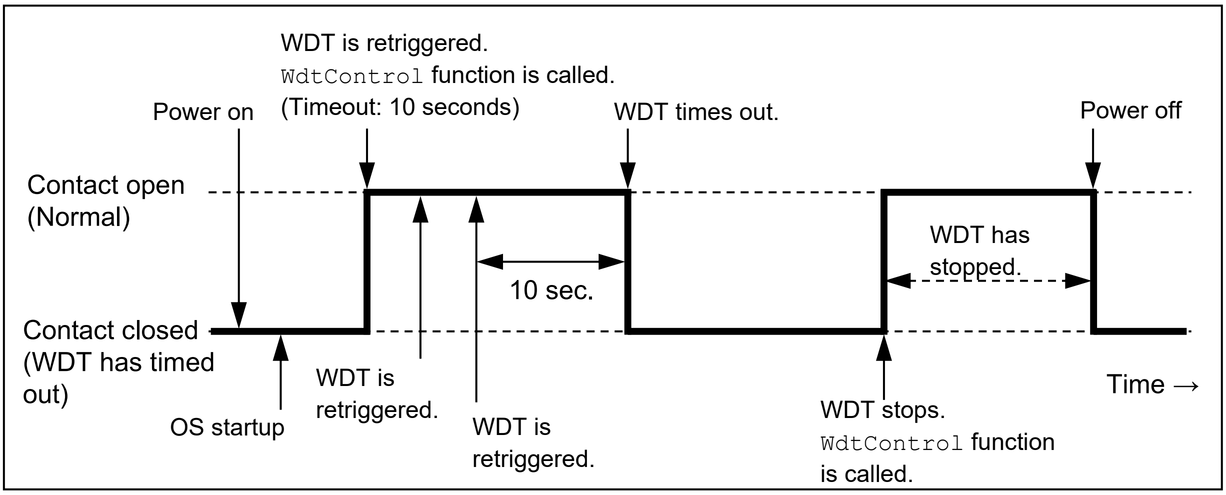


Figure 6-1 Behavior of the WDTTO contact

The dotted lines in Figure 6-1 denote the state of the contact, and the thick line denotes the transition of the state of the contact.

Figure 6-2 shows an example of the behavior of the WDTTO contact when the OS shuts down. In this example, the timeout of the watchdog timer is 60 seconds. The process that retriggers the watchdog timer terminates during shutdown, and consequently the watchdog timer times out.

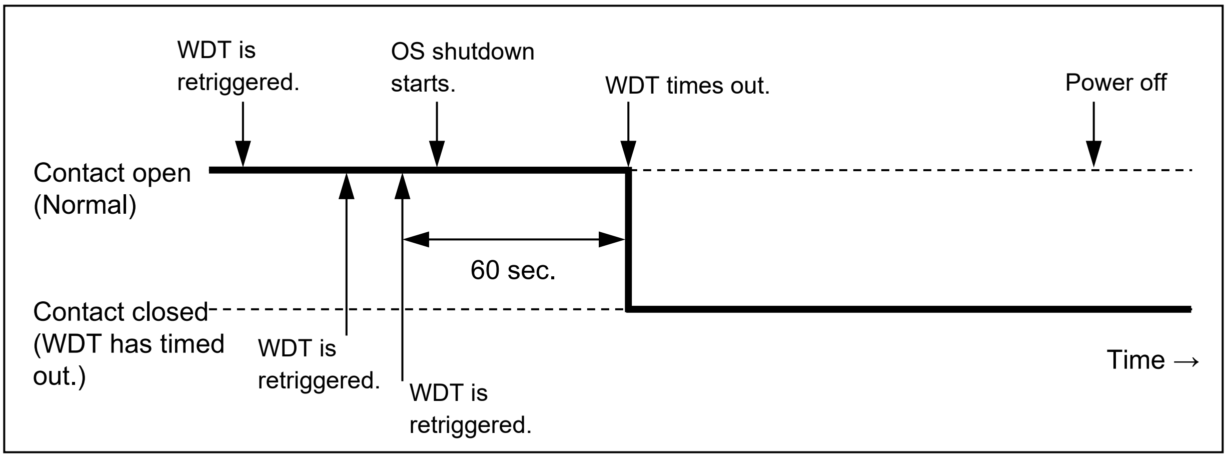


Figure 6-2 Behavior of the WDTTO contact (when the OS shuts down)

The dotted lines in Figure 6-2 denote the state of the contact, and the thick line denotes the transition of the state of the contact.

NOTE

This explanation assumes that the RAS external contact interface is HJ-F4070-11. If you are using HJ-F4070-12 as the RAS external contact interface, the behavior of the contact is the opposite of what is shown here. The contact is open when the contact described in the figure is closed, and vice versa.

6.1.4 Control function for the general-purpose external contact outputs (GendoControlEx)

You can use the GendoControlEx function to control the outputs to the general-purpose external contact of the RAS external contact interface.

(1) Function interface (GendoControlEx)

Name

GendoControlEx - Output control for the general-purpose external contact (GENDO0, GENDO1, GENDO2)

Syntax

```
#include <w2kras.h>
BOOL GendoControlEx(DWORD dwPort, DWORD dwCmd);
```

Description

This function performs the action specified by *dwCmd* on the general-purpose external contact (GENDO0, GENDO1, or GENDO2) of the RAS external contact interface specified by *dwPort*.

Table 6-3 shows a list of the actions that can be specified by *dwPort*.

Table 6-3 Actions of the GendoControlEx function specified by dwPort

dwPort	Explanation of the action
GENDO0_PORT (0x01)	Performs the action on the general-purpose external contact (GENDO0)
GENDO1_PORT (0x02)	Performs the action on the general-purpose external contact (GENDO1)
GENDO2_PORT (0x03)	Performs the action on the general-purpose external contact (GENDO2)

Table 6-4 shows a list of the actions that can be specified by *dwCmd*.

Table 6-4 Actions of the GendoControlEx function specified by dwCmd

dwCmd	Explanation of the action
GENDO_OPEN (0x00)	Opens the general-purpose external contact specified by *dwCmd*
GENDO_CLOSE (0x01)	Closes the general-purpose external contact specified by *dwCmd*

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.

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

`W2KRAS_INVALID_PARAMETER`
`(0x2001)`
	The specified parameters contain an error.

`W2KRAS_NOT_INITIALIZE`
`(0x2005)`
	The startup of the RAS software is not yet complete.

`W2KRAS_INTERNAL_ERROR`
`(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 file name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

(2) Behavior of the general-purpose external contact (`GENDO0`, `GENDO1`, or `GENDO2`) of the RAS external contact interface

These general-purpose external contacts open when the power turns on or off.

Figure 6-3 shows the behavior of the `GENDO0` contact when the `GendoControlEx` function is used.

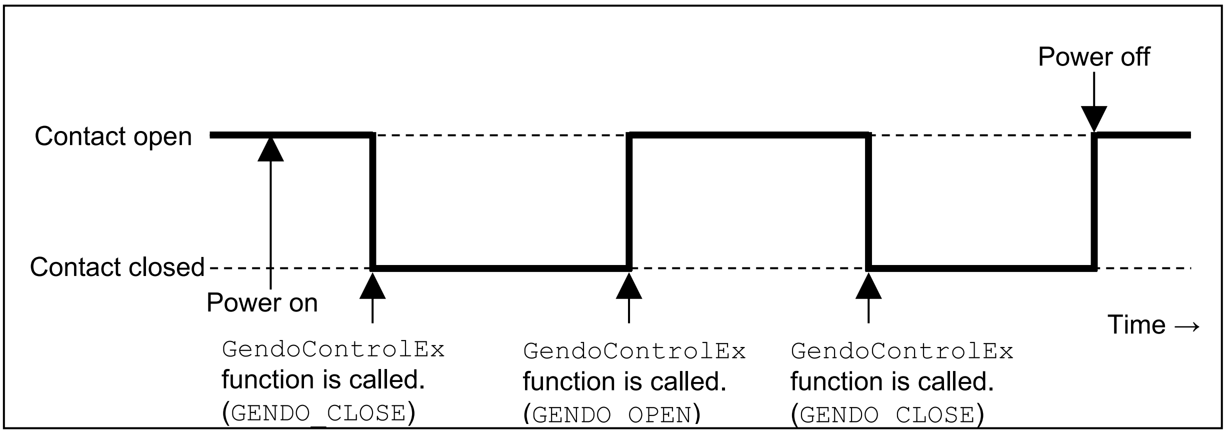

Power off
Contact open
Contact closed
Power on
Time →
GendoControlEx function is called. (GENDO_CLOSE)
GendoControlEx function is called. (GENDO_OPEN)
GendoControlEx function is called. (GENDO_CLOSE)

Figure 6-3 Behavior of the GENDO contact

The dotted lines in Figure 6-3 denote the state of the contact, and the thick line denotes the transition of the state of the contact.

The explanation in this section applies to the following RAS external contact interfaces: HJ-F4070-11, HJ-F4070-12, and HJ-F4070-13.

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

You can use the `GetGendiEx` function to get the inputs for the general-purpose external contact of the RAS external contact interface.

(1) Function interface (GetGendiEx)

Name

`GetGendiEx` - Status acquisition of the general-purpose external contact inputs (`GENDI`, `GENDI0`, `GENDI1`, and `GENDI2`)

Syntax

```
#include <w2kras.h>
DWORD GetGendiEx(DWORD dwPort);
```

Description

This function gets the status of the general-purpose external contact inputs (`GENDI`, `GENDI0`, `GENDI1`, and `GENDI2`) of the RAS external contact interface that was specified by *dwPort*.

Table 6-5 shows a list of the actions that can be specified by *dwPort*.

Table 6-5 Actions of the GetGendiEx function specified by dwPort

dwPort	Explanation of the action
`GENDI_PORT (0x00)`	Gets the status of the general-purpose external contact (`GENDI`)
`GENDI0_PORT (0x01)`	Gets the status of the general-purpose external contact (`GENDI0`)
`GENDI1_PORT (0x02)`	Gets the status of the general-purpose external contact (`GENDI1`)
`GENDI2_PORT (0x03)`	Gets the status of the general-purpose external contact (`GENDI2`)

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.

If you want to specify `GENDI_PORT` for *dwPort* and get the status of `GENDI`, in the RAS Setup window, clear the **Automatically shutdown if remote shutdown is requested.** check box under **Shutdown setting**. If this check box is selected, the function terminates with an error. If you call the `GetLastError` Windows API function, the error code `W2KRAS_RMTSTDN_ON` is returned.

Return values

`1`: The external contact specified by *dwPort* is closed.

`0`: The external contact specified by *dwPort* is open.

`0xffffffff`: The function terminates with an error.

Diagnosis

If the function terminates with an error, the function returns `0xffffffff`.

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

`W2KRAS_INVALID_PARAMETER`
`(0x2001)`
	The specified parameters contain an error.

`W2KRAS_RMTSTDN_ON`
`(0x2003)`
	This function cannot get the status of `GENDI` because the **Automatically shutdown if remote shutdown is requested.** check box is selected under **Shutdown setting** in the RAS Setup window.

`W2KRAS_NOT_INITIALIZE`
`(0x2005)`
	The startup of the RAS software is not yet complete.

`W2KRAS_INTERNAL_ERROR`
`(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 file name of the sample program and information about 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 the other log file.

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

You can check log information by opening the preceding files 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 6-4 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 file 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 output 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 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 line feed.
  If you want to include a new line feed 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.

```
typedefstruct 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 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 returned in Dimm_Number. (For example, if the value returned in Dimm_Number is 2, the elements up to Dimm_Status[1] are valid.) The elements after that are reserved. The values of these reserved elements are undefined. Do not use these values.

Table 6-6 Values stored in Dimm_Status

Value	Description
MEMORY_NOMAL (0x00)	The memory is working properly.
MEMORY_ERR_DETECT (0x01)	Frequent error corrections occur, or a possible failure is detected.
MEMORY_NOT_MOUNTED (0x02)	No DIMMs are installed.

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

Element	DIMM name
Dimm_Status[0]	DIMM1
Dimm_Status[1]	DIMM2

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
W2KRAS_INVALID_PARAMETER (0x2001)	The specified parameters contain an error.
W2KRAS_NOT_INITIALIZE (0x2005)	The startup of the RAS software is not yet complete.
W2KRAS_MEMST_INVALID (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 file 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 drive condition (hfwDiskStat)

Name

`hfwDiskStat` - Drive status acquisition

Syntax

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

Description

The `hfwDiskStat` function sets the drive 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 drive conditions.

```
typedefstruct HFW_DISK_STATUS{
  DWORD       Disk_Count;
  DWORD       Disk_Status[16];    //Drive condition
} HFW_DISK_STATUS, *PHFW_DISK_STATUS;
```

`Disk_Count` stores the number of arrays in the valid `Disk_Status`.

`Disk_Status[n]` stores the logical sum of the values for the type (the upper 16 bits) and condition (the lower 16 bits) of the drive in drive bay *n* + 1. Table 6-7 shows the values stored in `Disk_Status`.

Table 6-7 Values stored in Disk_Status

Disk_Status		Description
Upper 16 bits	0x0001	The drive type is HDD. (This model does not support HDD.)
	0x0002	The drive type is SSD.
Lower 16 bits	0x0001	The drive is working properly.
	0x0008	Drive failure prediction (SMART) is detected.
	0x0010	No drive is connected.
	0x0020	The drive condition could not be acquired.
	0x0040	The drive power-on (used) hours exceeded the threshold. (This value is not returned if drive usage monitoring is disabled.)

Note that if multiple conditions exist at the same time, the logical sum of the values shown in Table 6-7 is stored.

- If drive failure prediction (SMART) is detected and the drive power-on (used) hours exceeded the threshold at the same time:

  0x0048 is stored in the lower 16 bits of Disk_Status.

Table 6-8 shows a list of the defined values relating to the drive type and condition.

Table 6-8 Defined values

Defined value	Description
DRIVETYPE_HDD (0x00010000)	The drive type is HDD.
DRIVETYPE_SSD (0x00020000)	The drive type is SSD.
DISKSTAT_HEALTHY (0x00000001)	The drive is working properly.
DISKSTAT_SMART (0x00000008)	Drive failure prediction (SMART) is detected.
DISKSTAT_NOT_CONNECTED (0x00000010)	No drive is connected.
DISKSTAT_UNKNOWN (0x00000020)	The drive condition could not be acquired.
DISKSTAT_OVERRUN (0x00000040)	The drive power-on (used) hours exceeded the threshold.

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
HFWRAS_INVALID_PARAMETER (0x20000001)	The specified parameters contain an error.
HFWRAS_NOT_INITIALIZE (0x20000002)	The startup of the RAS software is not yet complete.
HFWRAS_INTERNAL_ERROR (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 file name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

6.1.9 Control functions for the digital LEDs for status indications (SetStCode7seg, TurnOff7seg, and SetMode7seg)

(1) Application status code display function (SetStCode7seg)

Name

`SetStCode7seg` - Displaying an application status code

Syntax

```
#include <ctrl7seg.h>
BOOL SetStCode7seg(DWORD dwStCode);
```

Description

This function outputs an application status code to the digital LEDs for status indications. On the digital LEDs for status indications, the value specified by this function is displayed in hexadecimal.

The following describes the parameters of this function.

dwStCode:

This parameter specifies an application status code to be displayed by using the LEDs. You can specify a value from 0 to 255. If a value outside of this range is specified, the function terminates with an error. If you call the `GetLastError` Windows API function, the error code `CTRL7SEG_INVALID_PARAMETER` is 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 code returned by this function itself is as follows:

Error code (value)	Description

`CTRL7SEG_INVALID_PARAMETER`
`(0x2001)`
	The specified parameters contain an error.

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 file name of the sample program and information about where you can find it, see *6.2 Sample Programs*.

(2) Application status code clear function (TurnOff7seg)

Name

`TurnOff7seg` - Clearing an application status code

Syntax

```
#include <ctrl7seg.h>
BOOL TurnOff7seg(VOID);
```

Description

This function clears the application status code currently displayed on the digital LEDs for status indications. When this function is called, the digital LEDs for status indications turn off.

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 code comes from the Windows API functions used by this function. For details about such error codes, see the Windows API Help.

Sample program

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

(3) Status display mode setup function (SetMode7seg)

Name

`SetMode7seg` - Setting up the status display mode

Syntax

```
#include <ctrl7seg.h>
BOOL SetMode7seg(DWORD dwMode);
```

Description

This function configures the status display mode of the digital LEDs for status indications.

The following describes the parameters of this function.

dwMode:

This parameter specifies the status display mode of the digital LEDs for status indications.

Table 6-9 shows a list of the values that can be specified by *dwMode*.

Table 6-9 Values specified by dwMode in the SetMode7seg function

dwMode	Explanation of the action
`APPST_MODE (0x00)`	Selects the application status display mode
`RASST_MODE (0x01)`	Selects the hardware status display mode

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 `CTRL7SEG_INVALID_PARAMETER` is 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 code returned by this function itself is as follows:

Error code (value)	Description

`CTRL7SEG_INVALID_PARAMETER`
`(0x2001)`
	The specified parameters contain an error.

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 file 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 directory %ProgramFiles%¥HFWRAS¥sample. Use these files for reference when you develop a program or to verify the operation of the functions.

Table 6-10 shows a list of the sample programs.

Table 6-10 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	`gendoex.c`	Sample program for the `GendoControlEx` function
4	`gendiex.c`	Sample program for the `GetGendiEx` 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
8	`7seg.c`	Sample program for the control functions of the digital LEDs for status indications (`SetStCode7seg`, `TurnOff7seg`, and `SetMode7seg`)
9	`FanErr.c`	Sample program for detecting RAS events (For information about RAS event notifications, see *4.2 RAS Event Notifications*.)

CHAPTER 7 FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

7.1 Notifications Regarding the Cause of a Bug Check (Blue Screen)

7.1.1 Overview

This equipment records the memory contents in a memory dump file when the system is forcibly recovered from an OS lockup, an NMI is generated due to a hardware failure, or an uncorrectable memory error occurs. At the same time, a blue screen appears and the identification name (`NMI_HARDWARE_FAILURE` or `WHEA_UNCORRECTABLE_ERROR`) is displayed.

In addition, the code corresponding to the cause of the bug check (blue screen) appears on the digital LEDs for status indications installed on the equipment.

This feature detects a bug check (blue screen) and records the cause of the bug check (blue screen) in the event log the next time that the OS starts.

NOTE

- If you perform an action (for example, a shutdown) that aborts processing during an analysis of the cause of a bug check (blue screen), the cause might not be recorded in the event log. In the case of a 4 GB memory dump file, analyzing the cause of a bug check (blue screen) takes about three minutes.

7.1.2 Supported causes of a bug check (blue screen)

This feature is activated when a bug check (blue screen) occurs because of one of the causes described in Table 7-1. Errors due to causes not shown in Table 7-1 are not supported, and 80 is displayed on the digital LEDs for status indications installed on the equipment.

Table 7-1 Supported causes of a bug check (blue screen)

Cause	Cause code#1	Contents recorded in the Description column of the event log#2
An NMI is generated by the **NMI** button.	Fb	A reset signal was input. The detail code is 0x9201.
An NMI is generated due to a parity error at the PCI bus, LPC bus, or elsewhere.	F8 or F9	A PCI bus parity error occurred. The detail code is 0x9202.
An uncorrectable memory error occurred.	FA	An uncorrectable memory error occurred in #3. The detail code is #4.

#1: This code is displayed on the digital LEDs for status indications installed on the equipment.

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

#3: The DIMM name (DIMM1 or DIMM2) is recorded.

#4: One of the following codes is displayed depending on the DIMM that contained the error:
DIMM1: 0x9218
DIMM2: 0x9217

For information about the actions to take when a blue screen appears, see Chapter 9 in the *HF-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

7.1.3 Event log

Table 7-2 shows a list of the event log entries recorded by this feature. 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* The detail code is *%2*.

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

%1	%2
A reset signal was input.	0x9201
A PCI bus parity error occurred.	0x9202
An uncorrectable memory error occurred in DIMM2.	0x9217
An uncorrectable memory error occurred in 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. Sign in to the computer as an administrator, and then open the window.

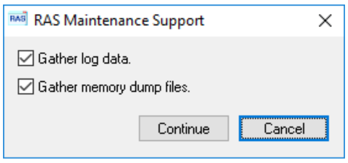
1. Click **Start**.
2. Click **RAS Software** under **Most used**.
3. Click **RAS Maintenance Support**.

NOTE

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

7.2.3 Using the log information collection window

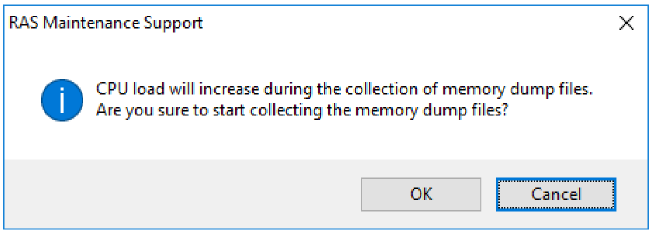
1. The log information collection window appears. By default, both the **Gather log data.** and **Gather memory dump files.** check boxes are selected. If you do not need to use either of one of these two options, clear the check box for the option that you do not need, and then click **Continue**.


RAS Maintenance Support

☑ Gather log data.

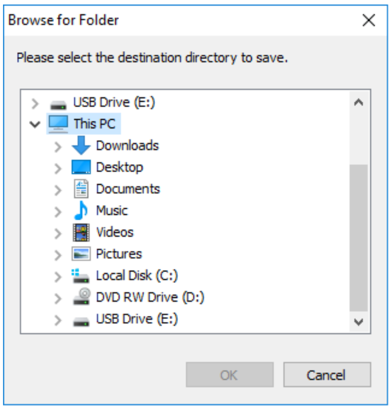
☑ Gather memory dump files.

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.


RAS Maintenance Support

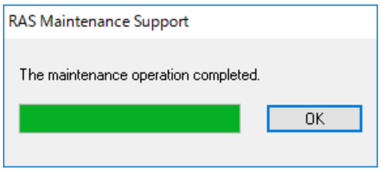
CPU load will increase during the collection of memory dump files.
Are you sure to start collecting the memory dump files?

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


Browse for Folder

Please select the destination directory to save.

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 appear during the process until `The maintenance operation completed.` appears in the following window. After this message appears, click **OK**.


RAS Maintenance Support

The maintenance operation completed.

5. A directory is created in the directory specified as the directory in which to save the files. 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, the collection of log information might have failed. In this case, collect the log information again.

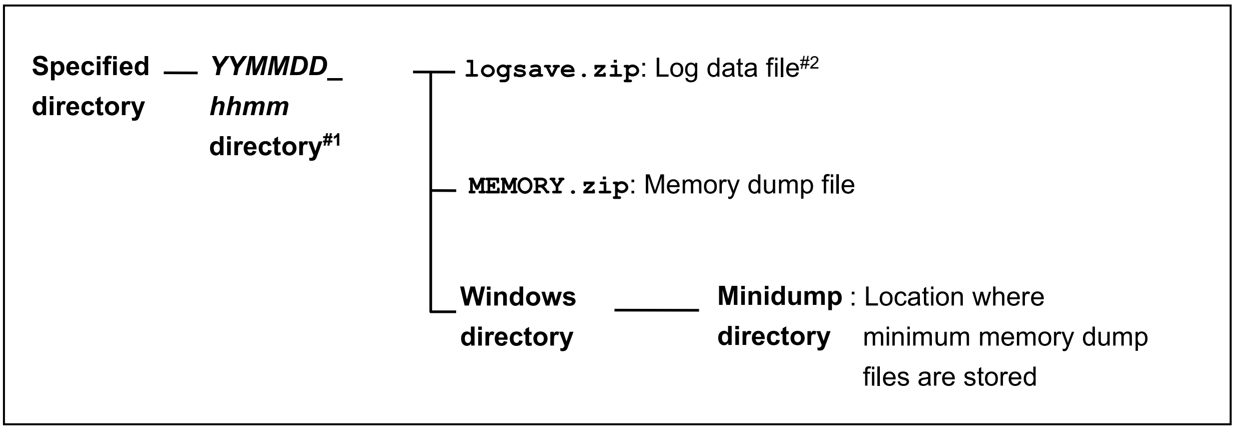

Specified directory
YYMMDD_ hhmm directory#1
`logsave.zip`: Log data file#2
`MEMORY.zip`: Memory dump file
Windows directory
Minidump directory
: Location where minimum memory dump files are stored

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 data is saved at 13:59 on Jan 1, 2014
Directory name: 140101_1359

#2: The following data is saved.
- If **Gather log data.** is selected:
  `logsave.zip` file
- If **Gather memory dump files.** is 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 the 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** in the window.

7.3 Logging Trends of 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 7-3 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 across 8 hours are recorded in this file. (The maximum number of entries is 1,100.)

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-W400E INSTRUCTION MANUAL* (manual number WIN-62-5001).

Format of log information

The format of the log information is as follows:

(1) `temp.csv`

```
YYYY/MM/DD hh:mm:ss, yxxx
                :
                :
```

YYYY: year, *MM*: month, *DD*: day, *hh*: hour (24-hour clock), *mm*: minute, *ss*: second,
y: sign (+ or -), *xxx*: temperature (°C)
If acquisition of the temperature fails, *xxx* is replaced with ---.

Figure 7-2 Format of log information (1)

(2) `temp_mm.csv`

```
YYYY/MM/DD hh:mm:ss, yxxx, yzzz
                :
                :
```

YYYY: year, *MM*: month, *DD*: day, *hh*: hour (24-hour clock), *mm*: minute, *ss*: second,
y: sign (+ or -), *xxx*: highest temperature (°C)
y: sign (+ or -), *zzz*: lowest temperature (°C)

Figure 7-3 Format of log information (2)

7.3.3 Logging frequency setup command

Name

`tmplogset` - Logging frequency setup

Syntax

`tmplogset`

Description

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 window.
   Administrator privileges are required to run this command. Sign in to the computer as an administrator, and then open a command prompt window.
2. In the command prompt window, 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
 :_
```

4. Enter the number corresponding to the new frequency that you want to select, and then press **Enter**. If the number you entered 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 frequency 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 then 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 then the command exits without changing the settings.

```
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 then 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 then 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.

CHAPTER 8 SIMULATING THE HARDWARE STATUS

8.1 Hardware Status Simulation

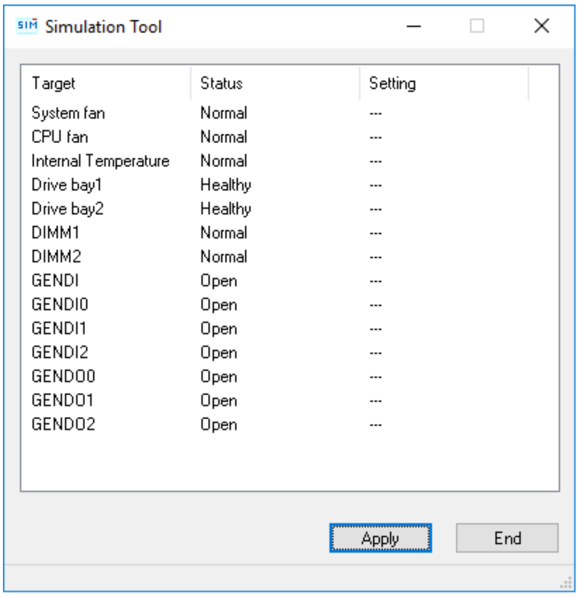
8.1.1 Overview

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 user applications 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:

- Fan condition
- Condition of the temperature inside the chassis
- Drive condition (including the drive failure prediction (SMART monitoring) condition and whether the power-on (used) hours have exceeded the threshold)
- Memory condition
- I/O status of the general-purpose external contacts



Target	Status	Setting
System fan	Normal	---
CPU fan	Normal	---
Internal Temperature	Normal	---
Drive bay1	Healthy	---
Drive bay2	Healthy	---
DIMM1	Normal	---
DIMM2	Normal	---
GENDI	Open	---
GENDI0	Open	---
GENDI1	Open	---
GENDI2	Open	---
GENDO0	Open	---
GENDO1	Open	---
GENDO2	Open	---

Figure 8-1 Simulation window

The monitoring feature of the RAS software detects change in the simulated hardware status and I/O status of the general-purpose external contacts, 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:

- Fan condition: *2.1 Fan Monitoring*
- Condition of the temperature inside the chassis: *2.2 Monitoring the Temperature Inside the Chassis*
- Drive condition: *2.3 Drive Failure Prediction (SMART monitoring)* and *2.4 Drive Usage Monitoring*
- Memory condition: *2.5 Memory Monitoring*
- Input status of the general-purpose external contact: (1) in *6.1.5 Get function for the general-purpose external contact inputs* and *4.5 Remote Notifications*
- Output status of the general-purpose external contact: (1) in *6.1.4 Control function for the general-purpose external contact outputs* and *4.5 Remote Notifications*

NOTICE

While the equipment is running in simulation mode, monitoring of the actual hardware status is disabled. Errors, including fan failures and 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

- While the equipment is running in simulation mode, OS deadlock monitoring cannot be used.
- 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 drive 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 drive. Also, note that you cannot simulate an "Unknown" drive condition or multiple drive 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 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 section 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 process is completed.

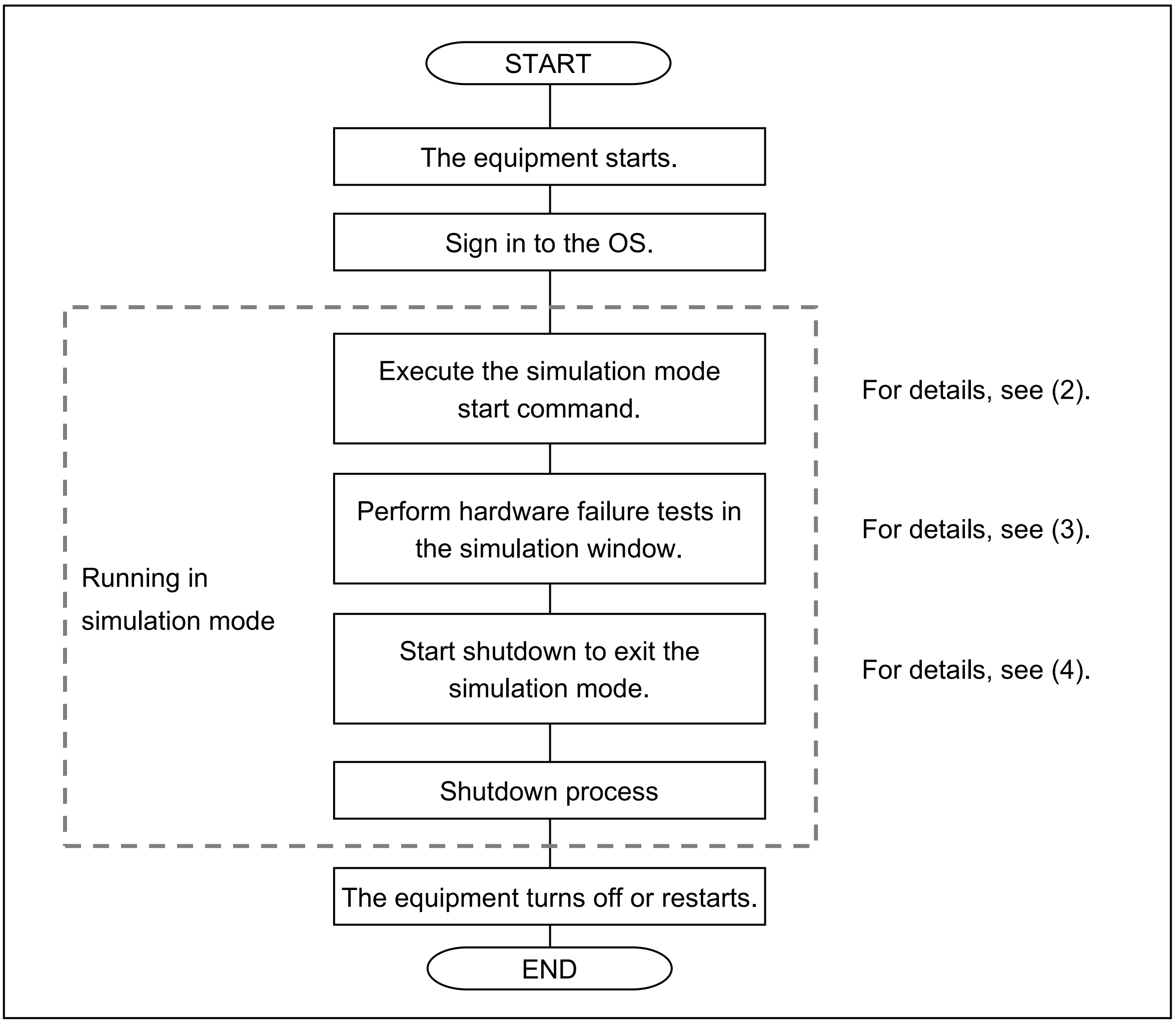

START
The equipment starts.
Sign in to the OS.
Running in simulation mode
Execute the simulation mode start command.
For details, see (2).
Perform hardware failure tests in the simulation window.
For details, see (3).
Start shutdown to exit the simulation mode.
For details, see (4).
Shutdown process
The equipment turns off or restarts.
END

Figure 8-2 Procedure for using the simulation mode

(2) Starting the simulation mode

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

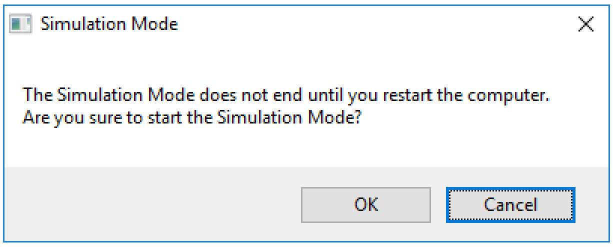
NOTE

- The simulation mode cannot start 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 window.

   Administrator privileges are required to run the simulation mode start command. Sign in to the computer as an administrator, and then open a command prompt window.
2. At the command prompt, enter the following, and then press **Enter**.

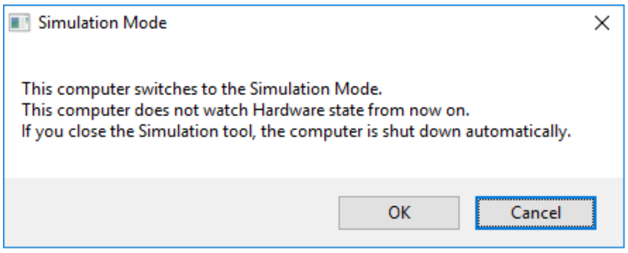
   ```
   >%SystemDrive%¥"Program Files"¥HFWRAS¥sim¥simrasstart
   ```
3. The following message about exiting the simulation mode appears. Click **OK**. If you click **Cancel**, the simulation mode does not start.



Simulation Mode

The Simulation Mode does not end until you restart the computer.
Are you sure to start the Simulation Mode?

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



This computer switches to the Simulation Mode.
This computer does not watch Hardware state from now on.
If you close the Simulation tool, the computer is shut down automatically.

5. The simulation 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:

- The status lamp is lit green or red alternately. The CPUSTOP contact switches between OPEN and CLOSE repeatedly.
- Every 10 seconds, the Windows® Exclamation sound is played twice. (This sound is audible only when speakers are connected.)

(3) Using the simulation window

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

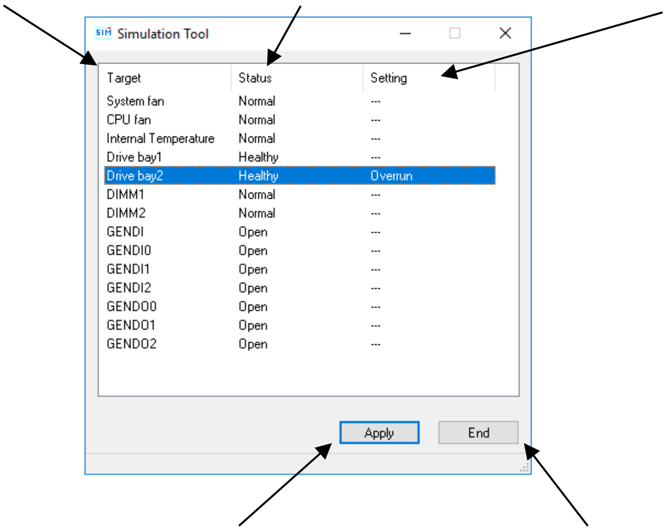
You can use this window to change the condition of hardware devices.

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

Shows the name of a simulated device.

Shows the currently set status of the target device.

Shows a new status to be simulated.



Target	Status	Setting
System fan	Normal	---
CPU fan	Normal	---
Internal Temperature	Normal	---
Drive bay1	Healthy	---
Drive bay2	Healthy	Overrun
DIMM1	Normal	---
DIMM2	Normal	---
GENDI	Open	---
GENDI0	Open	---
GENDI1	Open	---
GENDI2	Open	---
GENDO0	Open	---
GENDO1	Open	---
GENDO2	Open	---

Click **Apply** to apply the contents of the **Setting** column.

Click **End** to exit the simulation mode.

Figure 8-3 Description of the simulation window

- **Target**

Shows the name of each simulated hardware device.

Category	Target
Fan condition	System fan, CPU fan
Condition of the temperature inside the chassis	Internal temperature
Drive condition	Drive bay 1, drive bay 2
Memory condition	DIMM1, DIMM2
Input status of the general-purpose external contact	GENDI, GENDI0, GENDI1, GENDI2
Output status of the general-purpose external contact	GENDO0, GENDO1, GENDO2

● **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
Fan condition	System fan, CPU fan	Normal, Error
Condition of the temperature inside the chassis	Internal temperature	Normal, Error
Drive condition	Drive bay 1, drive bay 2	Healthy, SMART Detected, Overrun, Not Connected
Memory condition	DIMM1, DIMM2	Normal, Error, Failure, Not Mounted
Input status of the general-purpose external contact	GENDI, GENDI0, GENDI1, GENDI2	Open, Closed
Output status of the general-purpose external contact	GENDO0, GENDO1, GENDO2	Open, Closed

After the simulation 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 window starts, "---" is displayed for all target 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.

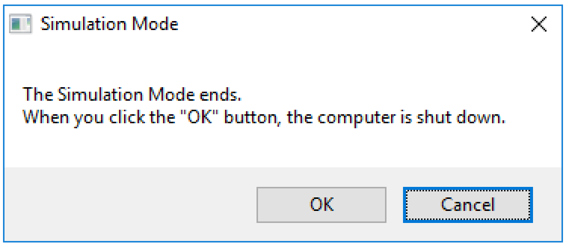
NOTE

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 window. Wait for the following times to elapse before you check the results of a simulation.

- Fan condition: About 10 seconds later
- Condition of the temperature inside the chassis: About 15 seconds later
- Drive condition: About 5 seconds later
- Memory condition: About 10 seconds later
- Input status of the general-purpose external contact: Immediately

- **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 window is not closed.

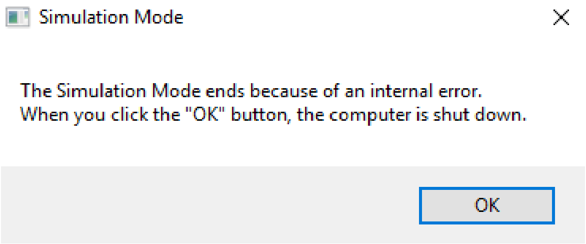


Simulation Mode

The Simulation Mode ends.
When you click the "OK" button, the computer is shut down.

NOTE

Even if the simulation window exits due to an internal error or other reasons, a shutdown is performed automatically 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**.



Simulation Mode

The Simulation Mode ends because of an internal error.
When you click the "OK" button, the computer is shut down.

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

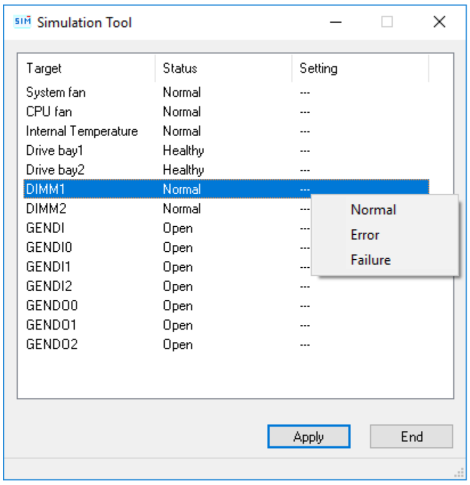
  Click the **Minimize** button in the upper-right corner of the simulation window to minimize the window.

- **Close** button ([×] button)

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

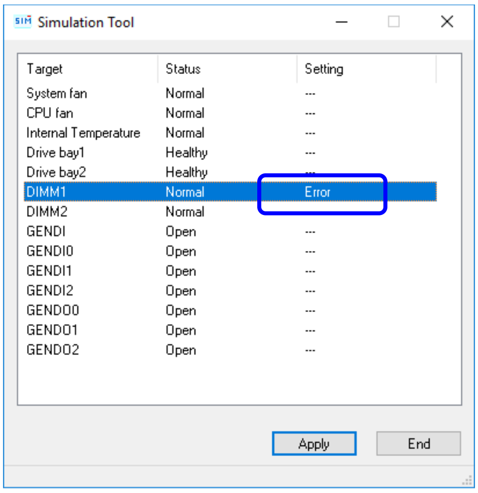
The following procedure shows how to simulate a hardware status by using the simulation window.

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.



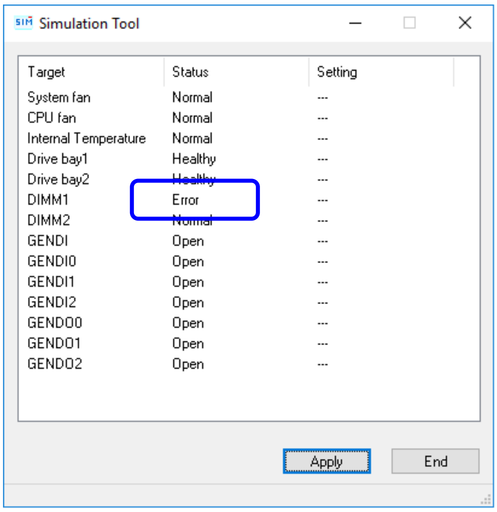
Target	Status	Setting
System fan	Normal	---
CPU fan	Normal	---
Internal Temperature	Normal	---
Drive bay1	Healthy	---
Drive bay2	Healthy	---
DIMM1	Normal	---
DIMM2	Normal	---
GENDI	Open	---
GENDI0	Open	---
GENDI1	Open	---
GENDI2	Open	---
GENDO0	Open	---
GENDO1	Open	---
GENDO2	Open	---

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.



Target	Status	Setting
System fan	Normal	---
CPU fan	Normal	---
Internal Temperature	Normal	---
Drive bay1	Healthy	---
Drive bay2	Healthy	---
DIMM1	Normal	Error
DIMM2	Normal	
GENDI	Open	---
GENDI0	Open	---
GENDI1	Open	---
GENDI2	Open	---
GENDO0	Open	---
GENDO1	Open	---
GENDO2	Open	---

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



Target	Status	Setting
System fan	Normal	---
CPU fan	Normal	---
Internal Temperature	Normal	---
Drive bay1	Healthy	---
Drive bay2	Healthy	---
DIMM1	Error	---
DIMM2	Normal	---
GENDI	Open	---
GENDI0	Open	---
GENDI1	Open	---
GENDI2	Open	---
GENDO0	Open	---
GENDO1	Open	---
GENDO2	Open	---

NOTE

If nothing appears in the **Setting** column, (that is, "---" is shown) or the status 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 なし is displayed in the menu.

- Fan condition

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

- Condition of the temperature inside the chassis

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

- Drive condition

No.	Current status	Statuses in the pop-up menu	Note
1	Healthy	Healthy, SMART Detected, Overrun, Not Connected	#1
2	SMART Detected	SMART Detected, Not Connected	#1, #2
3	Overrun	Overrun, Not Connected	#1, #2
4	Not Connected	Not Connected	#1, #3

#1: The equipment never starts without drive bay 1 being mounted, and consequently, **Not Connected** is not displayed for drive bay 1 in the pop-up menu.

#2: Transition from **SMART Detected** or **Overrun** to **Healthy** means that the target drive is replaced with a new one and that the new drive is connected.

#3: We assume that transition from **Not Connected** means a new drive is connected. Consequently, transition to **SMART Detected** or **Overrun** is not allowed.

● Memory condition

No.	Current status	Statuses in the pop-up menu	Note
1	Normal	Normal, Error, Failure, Not Mounted	#1
2	Error		#1, #2
3	Failure		#1, #2
4	Not Mounted		#1

#1: DIMM1 is required to start this equipment. Therefore, **Not Mounted** is not displayed for DIMM1 in the pop-up menu.

#2: For the memory condition, **Error** indicates a frequent memory error correction, and **Failure** indicates continued memory error corrections.

● Input status of the general-purpose external contact

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

● Output status of the general-purpose external contact

No.	Current status	Statuses in the pop-up menu	Note
1	Open	None	#
2	Closed		#

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

(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 shuts down or restarts in the following cases (for the following reasons):

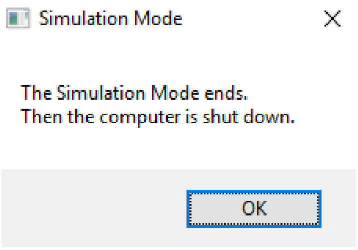
- A shutdown is executed from the **Start** menu.
- A system shutdown API function, such as `BSSysshut` or `ExitWindowsEx`, is executed.
- The equipment automatically shuts down because a remote shutdown signal input, a fan failure, or an abnormal temperature inside the chassis is detected.
- **Ctrl+Alt+Delete** is pressed and the power button in the lower-right corner of the screen is used to shut down the equipment.
- A blue screen appears because the **NMI** button was pressed or for other reasons.
- The power is forcibly turned off by using the power switch.

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

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

NOTE

- As explained above, 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 shut down or sign out of the system 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.



The Simulation Mode ends.
Then the computer is shut down.

- If 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 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 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 window when you click **Apply**. If nothing is set for **Setting**, (that is, "---" is shown), 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 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 starts.
251	HFWSIM_SYS	Information	HFWSIM	The simulation mode ends.
252	HFWSIM_SYS	Information	HFWSIM	The following conditions were set in the simulation mode. System fan: *%1* CPU fan: *%2* Internal temperature: *%3* Drive bay 1: *%4* Drive bay 2: *%5* DIMM1: *%6* DIMM2: *%7* GENDI: *%8* GENDI0: *%9* GENDI1: *%10* GENDI2: *%11*

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

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 in the simulation mode are the same as in the normal mode. For details about the objects of the extended MIB for HF-W, see *4.5 Remote Notifications*.