WIN-63-0095-02

HITACHI

HITACHI INDUSTRIAL COMPUTER

HF-W100E

RAS FEATURES MANUAL

USER'S MANUAL

WIN-63-0095-02

HITACHI

HITACHI INDUSTRIAL COMPUTER

HF-W100E

RAS FEATURES MANUAL

Read and keep this manual.
• Read safety instructions carefully and understand them before starting your operation. • Keep this manual at hand for reference.

USER'S MANUAL

First Edition, September 2017, WIN-63-0095-01 (out of print)
Second Edition, November 2017, WIN-63-0095-02

TP<IC> (FL-MW2007)

⚠ SAFETY INSTRUCTIONS

Carefully read and fully understand the safety precautions below before operating the equipment.

- Operate the equipment by following the instructions and procedures described in this manual.

- Pay attention especially to safety precautions displayed on the equipment or in this manual. Make sure you follow them. Otherwise, personal injury and property damage including damage to the equipment may result.

- A safety precaution is indicated by a heading as shown below. 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 signify potential hazards that may result in personal injury or death. Make sure you follow the safety message that follows this symbol in order to avoid possible injury or death.
⚠ DANGER:	This symbol is used to indicate imminent hazards that will highly likely result in serious personal injury or death.
⚠ WARNING:	This symbol is used to indicate potential hazards that may result in serious personal injury or death.
⚠ CAUTION:	This symbol is used to indicate potential hazards that may result in minor or moderate personal injury.
NOTICE:	This symbol is used to indicate hazards that may result in equipment or property damage but not personal injury.

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

- Do not attempt to perform any operation that is not described in this manual. If there is any problem with the equipment, call your maintenance personnel.

- Read this manual carefully and fully understand the directions and precautions written in this manual before operating the equipment.

- Keep this manual nearby so that you can reference the manual anytime you need it.

- Every effort has been made to specify the best precautions on the equipment and in the manual. Nevertheless, unexpected incidents may occur. When you use the equipment, you are asked not only to follow the instructions but also to use your own judgement on safety.

SAFETY INSTRUCTIONS (Continued)

1. COMMON SAFETY PRECAUTIONS
   Carefully read and fully understand the following safety precautions.

1.1 WARNING

- This equipment is not designed and manufactured to be used for a life-critical system that requires extreme safety. If there is a possibility that the equipment may be used for this purpose, contact one of our sales representatives.

- In case of smoke, a burning smell, or the like, turn off the power to the equipment, disconnect the power cable from the outlet, and contact your sales representative or maintenance personnel. Using the faulty equipment without repair may result in a fire or an electric shock.

- Do not modify this equipment because that may result in a fire or an electric shock. If you do not, serious injury or death may result due to the equipment failure. The Manufacturer’s responsibility is exempted from any result arising out of the user’s modification of the Equipment.

⚠ SAFETY INSTRUCTIONS (Continued)

1.2 ⚠ CAUTION

- If the equipment drops or is tipped over, personal injury may result. Pay full attention when transporting the equipment.
- Make sure you do not catch or hit your fingers to cause personal injury when unpacking or carrying the equipment.
- Do not use this equipment for purposes other than its original purpose.
  If you do, personal injury or this equipment failure may result.
- Do not touch this equipment directly during operation and immediately after shutting down, since the equipment may become hot during operation. If you do not, injury of burn may result. In addition, install this equipment in a place where the user does not touch during operation.

SAFETY INSTRUCTIONS (Continued)

1.3 *NOTICE*

- This equipment alone cannot guarantee the system safety. In order to ensure sufficient safety of your system even when this equipment should fail, malfunction, or have program bugs, you must add systemic protections such as building external protective/safety circuits to facilitate safety measures to prevent personal injury and serious accidents.

- When you work on installation or replacement of hardware, wear an antistatic wrist strap to prevent the buildup of static electricity.

- When you tighten or remove a screw, use a screwdriver that fits the size and type of the head of the screw to avoid stripping the head.
  When you tighten a screw, drive a screw along the axis of a tapped hole without adding too much torque in order to avoid damaging the thread.

- Do not use the equipment in the environment full of dust or with corrosive gas because that may cause the equipment to fail.

- Do not give a shock to the equipment when unpacking or carrying the equipment. If you do, that may cause the equipment to fail.

- Be sure to secure operation and installation space when using this equipment. Otherwise, the temperature inside the equipment may rise and that may cause a failure or short life span of the equipment. In addition, you need to ensure sufficient clearance for maintenance work.

- Use the basic software that we specify. Operation is not guaranteed if any other basic software is used.

- Performing emergency shutdown (that is, unplugging the power cable from the outlet or shutting off the circuit breaker without proper shutdown of the OS) may cause the OS or applications not to work properly or may cause the saved data to be corrupted. Never perform emergency shutdown unless you must stop the system immediately due to some kind of error.

- Keep in mind that if the power supply is cut, the system may not be able to recover automatically.

⚠ SAFETY INSTRUCTIONS (Continued)

2. SAFETY WARNINGS IN THIS MANUAL

2.1 Safety Warning Indicated as "*NOTICE*"

- When failure of storage is anticipated, the storage may experience hardware failure in near future. We recommend you to back up the data and replace this equipment. Please contact the system administrator or maintenance personnel.
(See page 2-2.)

- When OS hung-up occurs, processes on the OS cannot run as scheduled, and the processing of the equipment may be delayed. As a result, the facility that uses this equipment may be affected. You must resolve the root cause of the OS hung-up as soon as possible.
(See page 2-4.)

- When failure of storage is anticipated, the storage may experience hardware failure in near future. We recommend you to back up the data and replace this equipment. Please contact the system administrator or maintenance personnel.
(See page 4-4.)

- When the **Hardware status** window shows an error in hardware, resolve the error immediately.
(See page 4-5.)

- Log function exits (asynchronously) without waiting for the data to be actually written to a log file. That means this function does not return an error even when writing to a log file fails for some reason. We recommend important information be recorded in the OS event log.
(See page 6-17.)

- CPU load increases while memory dump files are being collected. While CPU load is high, operation of user applications can be disturbed. Make sure that you do not collect memory dump files using the log information collection window while the applications for business use are running on this equipment.
(See page 7-1.)

⚠ SAFETY INSTRUCTIONS (Continued)

- While the equipment is running in simulation mode, monitoring of the actual hardware status is disabled. Errors including fan failure and abnormal temperature cannot be detected. You must never use this equipment in simulation mode for business use. Use the simulation function only for testing a user application and checking the notification interface of the RAS software.

  (See page 8-2.)

PREFACE

This manual describes how to use the Reliability, Availability, Serviceability (RAS) feature of the HITACHI INDUSTRIAL COMPUTER HF-W100E (hereinafter referred to as "this equipment").

<Organization of this manual>

This manual is organized as follows.

<Precautions when you use the RAS feature>

● Precautions about an event log entry at the startup of the SNMP service
When you enable the SNMP service, a standard feature of Windows®, in order to use remote notification, an error log entry with event ID 1500 may be recorded at the startup of the SNMP service. This event log entry is recorded when SNMP trap notification has not been set up. Set up the trap notification according to "4.4.3　Enabling the remote notification".

● User Account Control
If User Account Control (UAC) is enabled in Windows® configuration, User Account Control windows is displayed when you run application or command. In this case, click **OK** or **Continue**.

<Trademarks>

<Note for storage capacity calculations>

- Memory capacities and requirements, file sizes and storage requirements, etc. must be calculated according to the formula 2^n. The following examples show the results of such calculations by 2^n (to the right of the equals signs).

  1 KB (kilobyte) = 1,024 bytes
  1 MB (megabyte) = 1,048,576 bytes
  1 GB (gigabyte) = 1,073,741,824 bytes
  1 TB (terabyte) = 1,099,511,627,776 bytes

- As for disk capacities, they must be calculated using the formula 10^n. Listed below are the results of calculating the above example capacities using 10^n in place of 2^n.

  1 KB (kilobyte) = 1,000 bytes
  1 MB (megabyte) = $1{,}000^2$ bytes
  1 GB (gigabyte) = $1{,}000^3$ bytes
  1 TB (terabyte) = $1{,}000^4$ bytes

<Note for replacing terms for Windows® 10>

This manual describes user actions in Windows®. The terms “log on” and “log off” in Windows® 7 have been renamed “sign in” and “sign out” respectively in Windows® 10. If the OS you use is Windows® 10, replace the former terms in this manual with the latter terms as necessary.

In addition, this manual describes remote notification feature in Windows®. The terms “Allowed a program through Windows Firewall” and “Allowed programs” in Windows® 7 have been renamed “Allowed an app through Windows Firewall” and “Allowed apps” in Windows® 10. If the OS you use is Windows® 10, replace the former terms in this manual with the latter terms as necessary.

<Note for display images>

The display images of this manual assume Windows® 7 but it is similar under other OS.

CONTENTS

FIGURES

TABLES

CHAPTER 1 CAPABILITIES OF THE RAS FEATURE

The HF-W series come with the Reliability, Availability, Serviceability (RAS) feature that you would expect from a highly reliable industrial computer.
The following is an overview of the RAS feature.

Table 1-1 Overview of the RAS Feature

Category		Item
Monitoring		Hardware status monitoring
		Watchdog timer monitoring
GUI function setting		RAS setup window
Status check	GUI output	Hardware status window
	Notification	Event notification
		Popup notification
		Remote notification
		Status acquisition using library functions
Control	Shutdown	Automatic shutdown
		Shutdown using library functions
		Controlling the external general purpose I/O
Library functions		RAS library
Maintenance/ Failure analysis	Memory dump related	Memory dump collection
		Log information collection window
		Maintenance operation support commands
		Logging the trend of the temperature inside the chassis
Simulation		Hardware status simulation

<Monitoring>

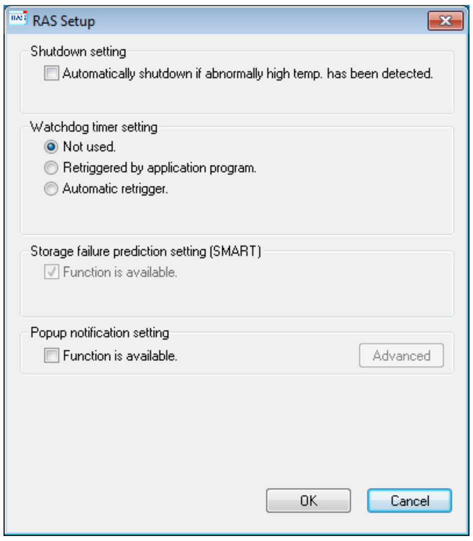
(1) Hardware status monitoring
Monitors the hardware status of this equipment including the status of the storages as well as the temperature inside the chassis.

(2) Watchdog timer monitoring
Monitors the operational state of the OS or user programs, using the watchdog timer implemented on this equipment. This feature also offers library functions to use the watchdog timer.

<GUI function setting>

(3) RAS Setup window

Provides a graphical user interface for configuring RAS Feature settings including the condition of automatic shutdown and the setting of the watchdog timer.



RAS Setup

Shutdown setting

- [ ] Automatically shutdown if abnormally high temp. has been detected.

Watchdog timer setting

- (●) Not used.
- ( ) Retriggered by application program.
- ( ) Automatic retrigger.

Storage failure prediction setting (SMART)

- [✓] Function is available.

Popup notification setting

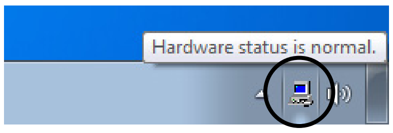
- [ ] Function is available.

Figure 1-1 RAS Setup Window

<Status check>

(4) Hardware status window

Displays the hardware status of this equipment using a graphical interface. There is always an icon in the notification area of the taskbar to display the hardware status.



Hardware status is normal.

Figure 1-2 Hardware Status Icon

(5) Event notification

Enables a user application to check the hardware status of this equipment by monitoring the status of event objects.

(6) Popup notification

Notifies a user that an error occurred in the hardware of this equipment by displaying popup messages.

(7) Remote notification

A remote device can check the hardware status of this equipment. A remote device can also be notified whenever the hardware status changes.

(8) Status acquisition using library functions

This function enables a user application to get the hardware status of this equipment by using the RAS library.

<Control>

(9) Automatic shutdown

Automatically shuts down the equipment when abnormal temperature inside the chassis is detected. Use "(3) RAS Setup window" to enable or disable the automatic shutdown function.

(10) Shutdown using library functions

This function enables a user application to shutdown this equipment by using the RAS library.

(11) Controlling the external general purpose I/O

Enables a user to control the external general purpose I/O with the RAS library.
Seven points input and eight points output in the external general purpose I/O are available for a user. If you use those I/O, 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>

(12) RAS library interface

Offers library functions for recording log information in addition to the library functions offered by (2), (8), (10), and (11).

<Maintenance / Failure analysis>

(13) Memory dump collection

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

(14) Log information collection window
Allows you to collect log data and memory dump files for this equipment using a graphical user interface.

(15) Maintenance operation support commands
These commands include a command used for saving failure information such as memory dump files and event log files to an external medium.

(16) Logging the trend of the temperature inside the chassis
This function periodically measures the temperature inside the chassis of this equipment and records the data into a file.

<Simulation>

(17) Hardware status simulation
Simulates the hardware status of this equipment. By using this function, you can test a user application and check the notification interface of the RAS software without actual hardware failure.

This manual explains functions (1) through (12), (14), (16), and (17). For details about other functions, refer to “HF-W100E INSTRUCTION MANUAL (manual number WIN-62-0069)”.

CHAPTER 2 ITEMS MONITORED BY THE RAS FEATURE

This chapter explains the items monitored by the RAS Feature.

2.1 Monitoring Temperature inside the Chassis

This function monitors the temperature inside the chassis using the temperature sensor in this equipment and notifies in the following methods when the temperature inside the chassis gets abnormally high.

(1) Hardware status window
(2) Event notification
(3) Popup notification
(4) Remote notification
(5) Automatic shutdown
(6) Alarm lamp

For details about (1) to (4), see "CHAPTER 4 CHECKING THE HARDWARE STATUS".
For details about (5), see "5.1 Automatic Shutdown of the Equipment".
If the temperature abnormality of the equipment is recovered, the alarm lamp is turn off.

2.2 Storage Failure Prediction Function (SMART Monitoring)

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

(1) Hardware status window
(2) Event notification
(3) Popup notification
(4) Remote notification
(5) The hfwDiskStat functions in the RAS library

For details about (1) to (4), see "CHAPTER 4 CHECKING THE HARDWARE STATUS".
For details about (5), see "6.1.10 Get function for the storage condition (hfwDiskStat)".

NOTICE

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

NOTE

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

2.3 Memory Monitoring

Memory with Error Checking and Correcting (ECC) is installed in this equipment. A single bit error in the memory can be automatically corrected without causing any problems to the operation of the equipment. On the other hand, if the frequency of memory error correction is high, that is considered to be caused by memory failure, and we recommend replacing this equipment from the viewpoint of preventive maintenance. Please contact the system administrator or maintenance personnel.
The memory monitoring function notifies in the following methods when the frequency of memory error correction is high.
(1) Event notification
(2) Popup notification
(3) Remote notification
(4) The GetMemStatus function in the RAS library

For details about (1) to (3), see “CHAPTER 4　CHECKING THE HARDWARE STATUS”.
For details about (4), see “6.1.9　Get function for the memory condition (GetMemStatus)”.

2.4 Watchdog Timer Monitoring

This equipment has a built-in watchdog timer and can monitor the operation of the OS and a user program.

- Automatic retriggering feature
  The watchdog timer monitoring process in the RAS software automatically retriggers the timer and monitors the operation of the OS.
- User program operational state monitoring
  A user can monitor the operational state of a user program by using dedicated library functions.

When this timer timeout occurs, a memory dump is generated. For information about memory dump, see “5.2 Memory Dump due to Timeout Detection”.
Note that you can configure the watchdog timer settings in the **RAS Setup** window. For information about how to use the **RAS Setup** window, see “3.1.3 Using the RAS Setup window”.

2.4.1 Automatic retriggering feature for a watchdog timer

The automatic retriggering feature can detects the situation that the watchdog timer monitoring process in the real-time priority class cannot run for a certain period of time (the situation that the watchdog timer expires) as an OS hung-up.
To use this feature, in the **RAS Setup** window, select **Automatic retrigger** under **Watchdog timer setting**.

NOTICE

When OS hung-up occurs, processes on the OS cannot run as scheduled, and the processing of the equipment may be delayed. As a result, the facility that uses this equipment may be affected. You must resolve the root cause of the OS hung-up as soon as possible.

NOTE

In this feature, OS hung-up is defined as a state that a process in the real-time priority class cannot run for a certain period of time.

2.4.2 Using a watchdog timer for monitoring a user program

When you use the watchdog timer for monitoring the operational state of a user program, the user program to be monitored must periodically retrigger the watchdog timer (that is, reset the timeout counter of the watchdog timer to the initial value). Figure 2-1 shows an example of monitoring the operational state of a user program.

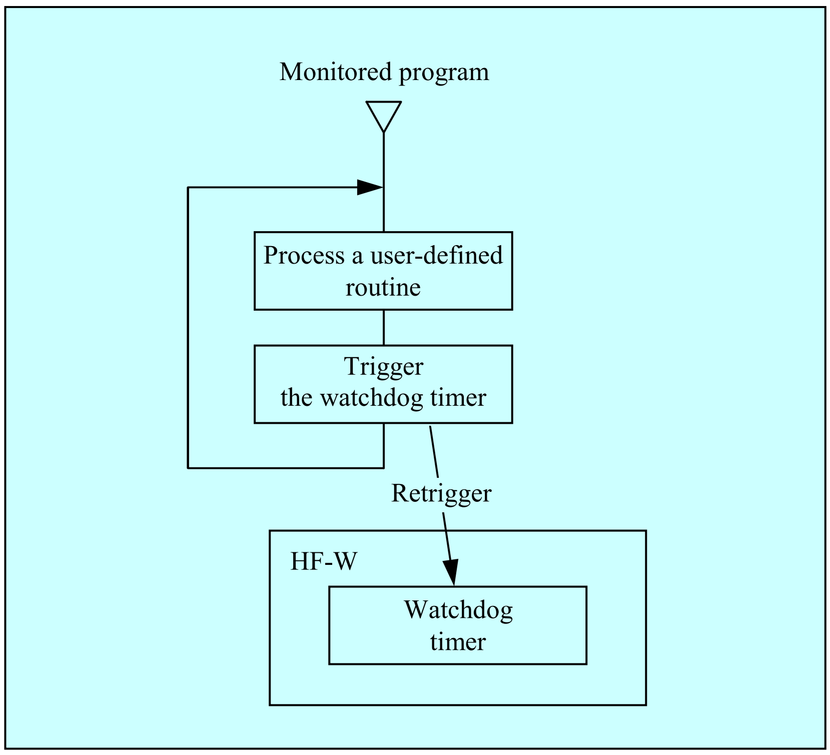


Figure 2-1 Example of a Flow Chart of Monitoring the Operational State of a User Program

If a watchdog timer timeout occurs, that means the monitored program was not be able to retrigger the watchdog timer within the configured timeout for some reasons.
A program can use a watchdog timer by calling the library function WdtControl. For information about how to use the WdtControl function, see “6.1.3 Watchdog timer control function (WdtControl)”.

NOTE

- If you want to stop monitoring using the watchdog timer when a user program exits or due to shutdown, you must stop the watchdog timer so that a timeout does not occur.
- The time it takes to generate timeout is longer than the timeout value set by the application. This is a correct operation.
  This situation occurs because it takes about 1.2 to 1.3 seconds for the hardware timer on this equipment to actually enable the timeout setting set by the application. If the timeout is set to 30 seconds, memory dump starts about 31 seconds after the timeout is set.

CHAPTER 3 SETTING UP THE RAS FEATURE

3.1 RAS Setup Windows

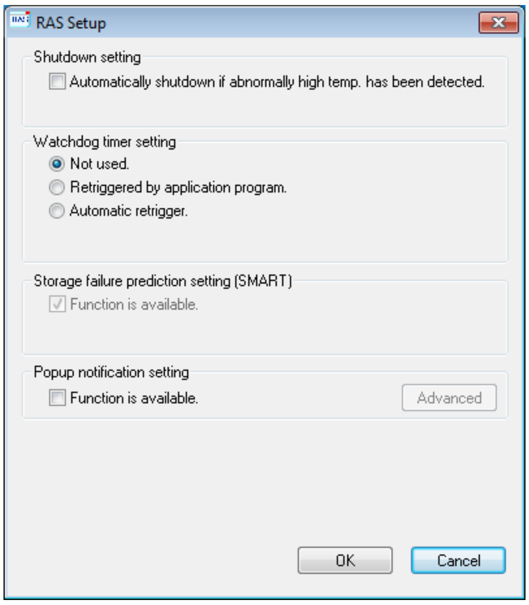
3.1.1 Overview

In the **RAS Setup** window, the following functions can be set up.

Table 3-1 Setup Items in the RAS Setup Window

Item
Shutdown setting
Watchdog timer setting
Popup notification setting

Figure 3-1 shows the **RAS Setup** window. The default factory-shipped settings are shown in the figure.



RAS Setup

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

Watchdog timer setting
● Not used.
○ Retriggered by application program.
○ Automatic retrigger.

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

Popup notification setting
☐ Function is available.

Figure 3-1 RAS Setup Window

3.1.2 Starting the RAS Setup window

To start the **RAS Setup** window, follow the procedure below.

Before you start this window, you need to log on to the computer as an administrator account.

＜For Windows® Embedded Standard 7＞

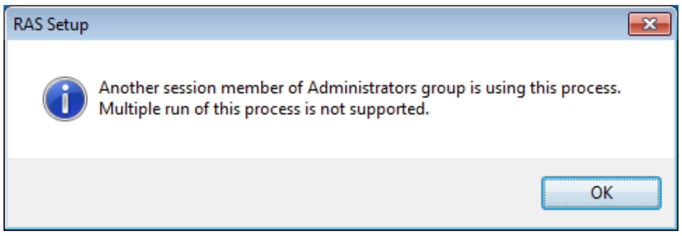
1. Click **Start**.
2. Point to **All Programs** > **RAS Software**. Click **RAS Setup**.

＜For Windows® 10＞

1. Click **Start**.
2. Click **RAS Software** from the list of applications.
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 on another console, and then start the **RAS Setup** window.



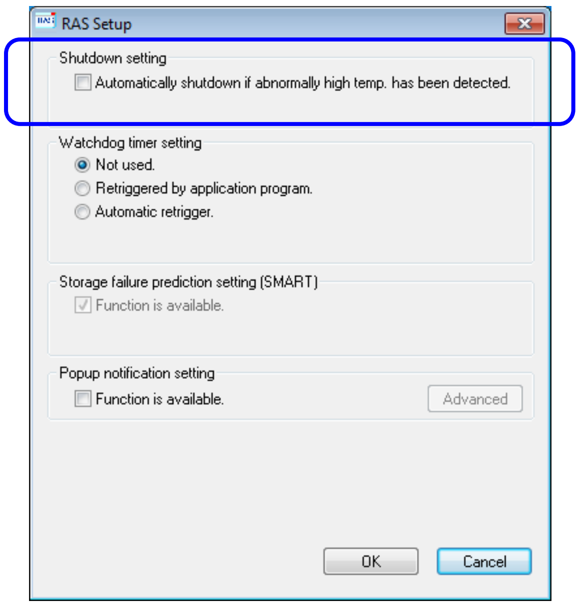
RAS Setup

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 setting

You can select whether this equipment is automatically shut down for an abnormally high temperature.



RAS Setup

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

Watchdog timer setting
● Not used.
○ Retriggered by application program.
○ Automatic retrigger.

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

Popup notification setting
☐ Function is available.

Figure 3-2 Items in the Shutdown Setting

- “**Automatically shutdown if abnormally high temp. has been detected**” check box
  - ON: This equipment will be automatically shutdown.
  - OFF: This equipment will not be automatically shutdown (default factory-shipped setting).

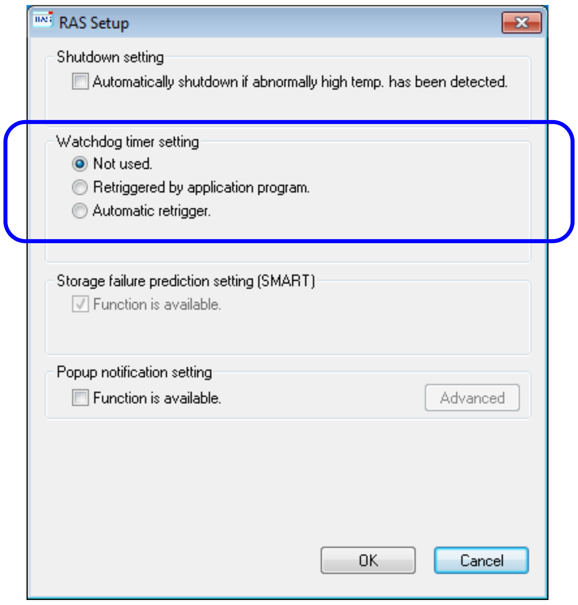
If you change the current setting, click the check box you want to change.

NOTE

After an automatic shutdown initiated by this function is complete, the power is turned off.

(2) Watchdog timer setting

You can set up the watchdog timer of this equipment.
You can select one of the following ways of using the watchdog timer:
• Not used (default factory-shipped setting)
• Retriggered by an application program
• Automatic retrigger
Click the radio button corresponding to the item you want to select.


RAS Setup

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

Watchdog timer setting
● Not used.
○ Retriggered by application program.
○ Automatic retrigger.

Storage failure prediction setting (SMART)
✓ Function is available.

Popup notification setting
☐ Function is available.

Figure 3-3 Items in Watchdog Timer Setting

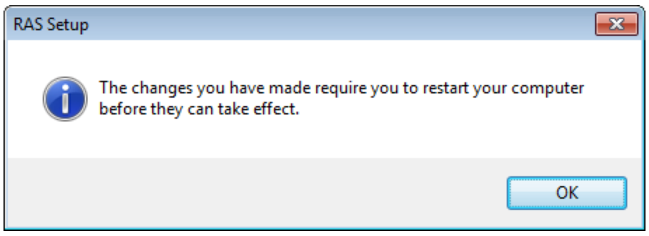
● **Not used:**
If you select this item, the watchdog timer is stopped. The watchdog timer will not time out. In addition, you cannot use the watchdog timer by calling the WdtControl function of the RAS library.

● **Retriggered by application programs:**
If you select this item, you can monitor the operational state of a user program by controlling the watchdog timer using the WdtControl function in the RAS library.

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

NOTE

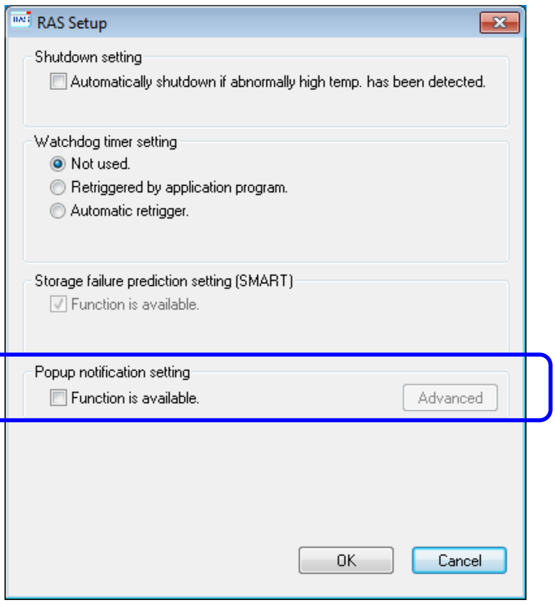
To make a new watchdog timer setting effective, you must restart the equipment.
When the watchdog timer setting is changed, the following message is displayed.
Click **OK** to close the box, and then make sure you restart the equipment.



The changes you have made require you to restart your computer before they can take effect.

(3) Popup notification setting

You can set up the popup notification setting. By clicking **Advanced**, you can set up the advanced settings of this function.


RAS Setup

Shutdown setting
- [ ] Automatically shutdown if abnormally high temp. has been detected.

Watchdog timer setting
- (●) Not used.
- ( ) Retriggered by application program.
- ( ) Automatic retrigger.

Storage failure prediction setting (SMART)
- [✓] Function is available.

Popup notification setting
- [ ] Function is available.

Figure 3-4 Items in the Popup Notification Setting

● **"Function is available"** check box
- ON: The popup notification is enabled.
- OFF: The popup notification is disabled (default factory-shipped setting).

If you change the current setting, click the check box.
If the popup notification is enabled, the **Advanced** becomes active.

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

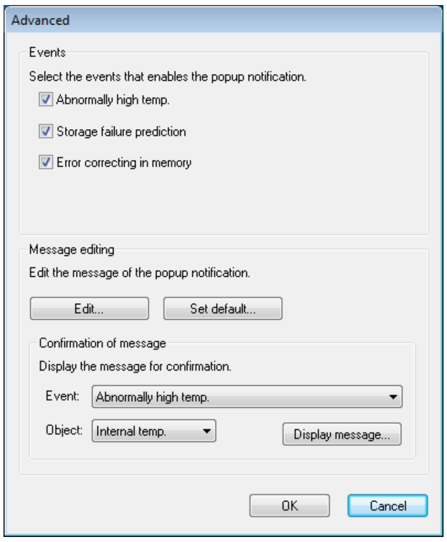

Advanced
Events
Select the events that enables the popup notification.
☑ Abnormally high temp.
☑ Storage failure prediction
☑ Error correcting in memory
Message editing
Edit the message of the popup notification.
Confirmation of message
Display the message for confirmation.
Event: Abnormally high temp.
Object: Internal temp.

Figure 3-5 Advanced Settings for the Popup Notification Setting

[Events]
- Abnormally high temp.
- Storage failure prediction (SMART)
- Error correcting in memory (When memory correction is detected frequently)

You can disable or enable popup notification for each of the items above.

● Check box for each item
- ON: The popup notification is enabled. (factory-shipped default setting)
- OFF: The popup notification is disabled.

If you change the current setting, click the check box.
If the monitoring function is disabled, it will not be notified even if the popup notification is enabled.

[Message editing]
You can edit popup notification messages. And you can check the messages after you edit them. For information about how to edit and check the messages, see "3.1.4 Editing popup notification messages".

If you changed the advanced setting and want to make the change effective, click **OK**. If you do not want the change, click **Cancel**.

(4) Making the change you made in (1) through (3) effective or discard the changes

If you changed the settings in (1) through (3) and want to make the change effective, click **OK**. The **RAS Setup** window is closed and the settings become effective immediately.

If you do not want the change the setting, click **Cancel**.

3.1.4 Editing popup notification messages

(1) Editing popup notification messages

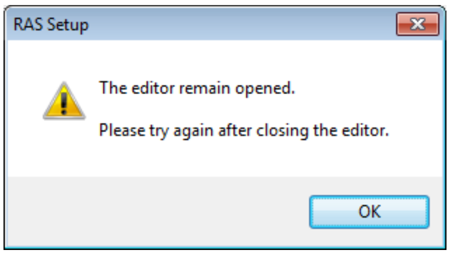
If you want to edit messages used for popup notification, click **Edit** in advanced settings for the popup notification setting. Notepad is launched and the message definition file for popup notification is opened. Edit the messages in the format described below.

NOTE

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

- Clicking **Edit**.
- Clicking **Set default**.
- Clicking **Display message**.
- Closing the **RAS Setup** window (clicking **OK** or **Cancel**).

If you do one of the above, the following cautionary message is displayed.


RAS Setup

The editor remain opened.

Please try again after closing the editor.

If you click **OK** when you receive the message, you go back to the **RAS Setup** window.

■ Format of a definition file

The format of a definition file is as follows.

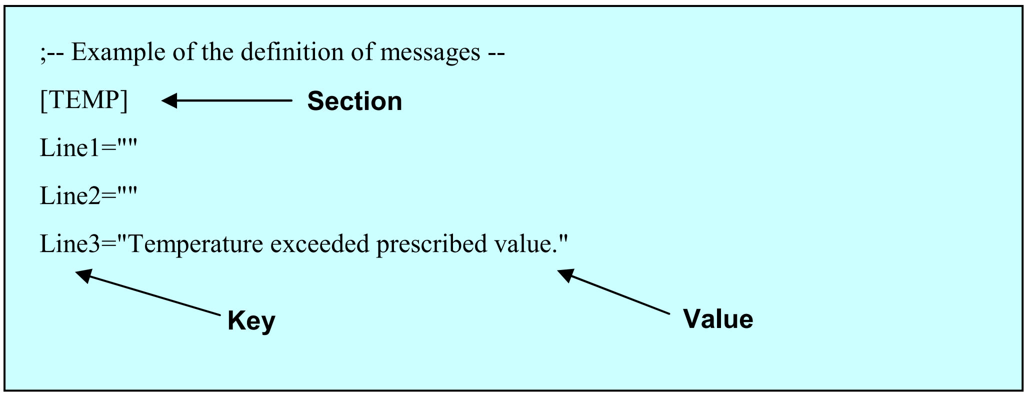

```
;-- Example of the definition of messages --
[TEMP]
Line1=""
Line2=""
Line3="Temperature exceeded prescribed value."
```

Figure 3-6 Format of a Definition File

A definition file consists of sections, keys, and the values of the keys.
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.

■ Description in a definition file

1. Section

   The table below shows a list of section names you can define for this function and an explanation of the message you define for each section.

Table 3-2 Section Names and Defined Messages

Section name	Defined message
[TEMP]	A message displayed when an abnormal temperature inside the chassis is detected.
[STR1-SMART] (*)	A message displayed when a storage failure prediction (SMART) is detected.
[DIMM1-ERR]	A message displayed when an error correction is detected frequently in DIMM 1.

(*) Regardless of the mounted storage, this section name is fixed.

2. Keys

   For a key, specify the line number of a line displayed as a part of the popup message.
   In this function, you can use the keys “Line1” through “Line5” for each section.
   If you specify Line6 and so on for keys, those keys are ignored.

3. Values

   Specify one line of message displayed as a part of the popup message for a value.
   For each key, a maximum of 50 bytes of characters (up to 25 double-byte characters) can be assigned. If you specify more than 50 bytes of characters, the characters at the 51th byte and after are ignored.
   If the line includes space characters, the whole value should be enclosed in double quotation marks (“ ”). If the value is empty, it is treated as a newline character.

NOTE

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

(2) Checking popup notification messages

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

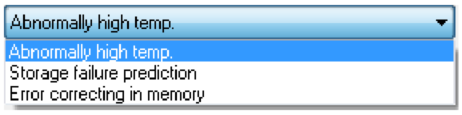
• Abnormally high temp.

• Storage failure prediction

• Error correcting in memory (When memory correction is detected frequently)

The following shows how to check the change you made in the message.

1. In the Event list (refer to Figure 3-5), select the event you want to check.
   In this list, only the items under **Events** selected using the check boxes are displayed. If no items under **Events** are selected using the check boxes, you cannot select an item from the **Event** list.



<Example of Selecting Abnormally high temp.>

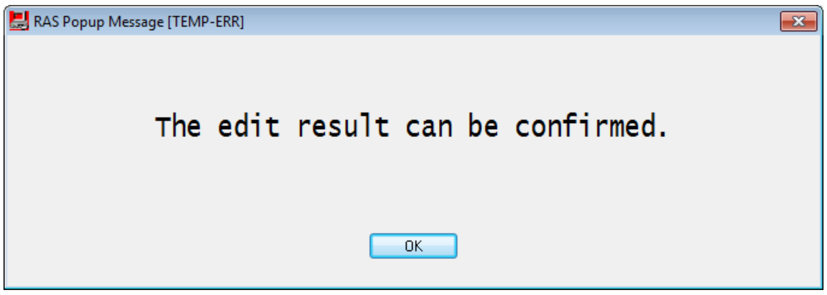
2. In the **Object** list (refer to Figure 3-5), select the object you want to check.
   Items in this list depend on the item you selected in step 1.

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

Table 3-3 Items Displayed in the Object List for Each Option Selected in the Event List

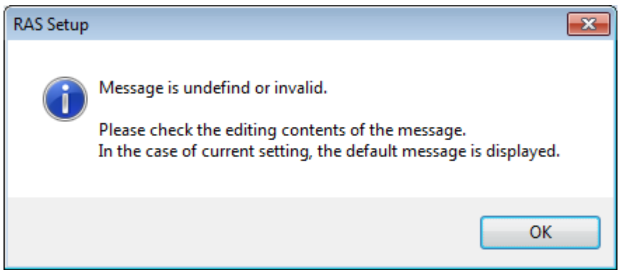
Option in the Event list	Items in the Object list
Abnormally high temp.	Internal temp.
Storage failure prediction	mSATA SSD
Error correcting in memory	DIMM 1

3. Click **Display message** (See Figure 3-5). A popup notification message is displayed based on the change you made. After you confirm that the message is OK, click **OK** in the popup.


RAS Popup Message [TEMP-ERR]

The edit result can be confirmed.

If the message is not edited or there is something wrong in the message definition file, the following message is displayed. Click **OK** to go back to the Advanced Settings. Set it again in the procedure of 3.1.4 (1).

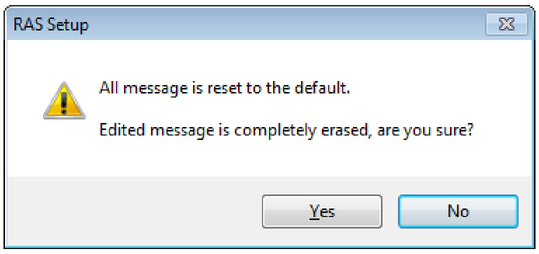

RAS Setup

Message is undefind or invalid.

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

(3) Restoring default messages

If you want to set the popup notification messages back to the default, click **Set default** in Advanced Settings. The following message will appear. Click **Yes**. Then the changes you made in the message definition file are discarded.


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 revert back to the default.

CHAPTER 4 CHECKING THE HARDWARE STATUS

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

(1) Check by using GUIs
You can check the hardware status of this equipment by using a graphical interface. For details, see “4.1 Hardware Status Window”.

(2) Check with 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 Notification”.
A user application can also get the hardware status of this equipment by using the RAS library. For details, see “4.5 Status Acquisition by Using the RAS Library”.

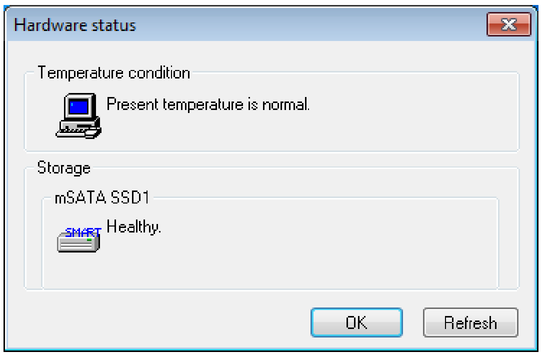
(3) Check on the desktop of this equipment
A popup message is displayed to notify that an error occurred in the hardware of this equipment. For details, see “4.3 Popup Notification”.

(4) Check from a remote device
A remote device can check the hardware status of this equipment. A remote device can also be notified whenever the hardware status changes. For details, see “4.4 Remote Notification”.

4.1 Hardware Status Window

4.1.1 Overview

After you log on 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 this icon or if you right-click the icon to display a popup menu and click **Display Hardware status**, detailed information about the hardware status of this equipment is displayed.
This window displays the following information:
• Temperature condition inside the chassis
• Storage failure prediction (SMART monitoring) condition



Present temperature is normal.

Healthy.

Figure 4-1 Hardware Status Window

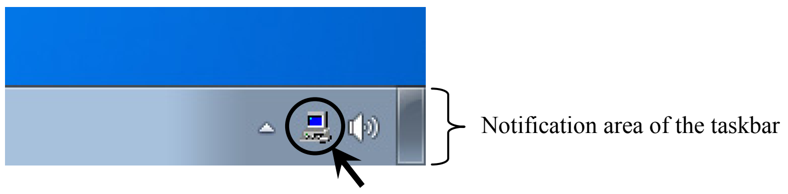
NOTE

This window only displays the internal storages that are recognized during OS startup. It may take a long time to recognize the new storage at the first startup after the new storage is connected, and the information about the storage may not be displayed. If this situation happens, restart this equipment.

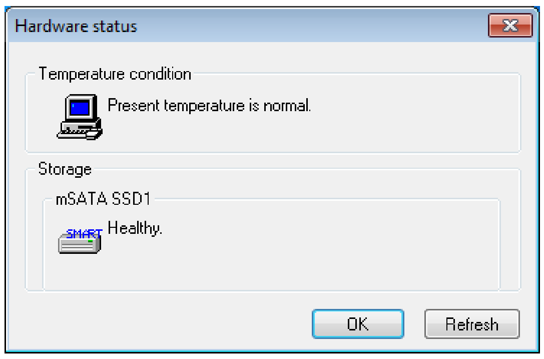
4.1.2 Hardware status window

The **Hardware status** window shows the details of the hardware status of this equipment. The following shows how to start the **Hardware status** window.

1. Double-click the **Hardware status** icon.
   Alternately, right-click the icon to display a popup menu and click **Display hardware status** on the menu.



2. The **Hardware status** window is displayed.



Temperature condition

Present temperature is normal.

Storage

mSATA SSD1

Healthy.

(1) Description of the window

1. Temperature condition

   Shows the current status of the temperature inside the chassis.

Table 4-1 Temperature Condition and Displayed Information

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

2. Storage condition
   Shows the current status of the storage.

Table 4-2 Storage Condition and Displayed Information

Drive condition	Icon	Information
Normal	🖴	Healthy.
Failure anticipation by SMART	⚠	A failure may be imminent.

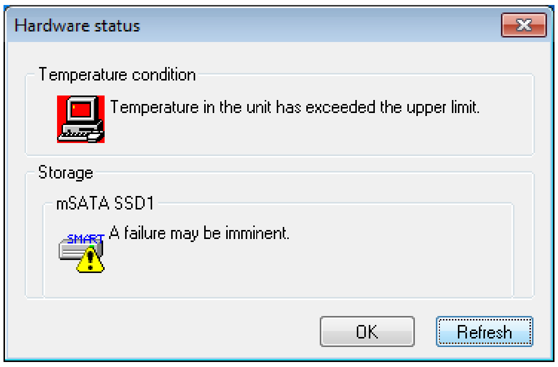
NOTICE

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

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

4. **OK** button
   Click this button to close the **Hardware status** window.

Figure 4-2 shows the **Hardware status** window when there is a hardware status error.


Hardware status

Temperature condition

Temperature in the unit has exceeded the upper limit.

Storage

mSATA SSD1

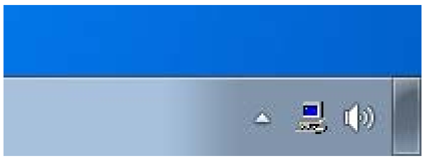
A failure may be imminent.

Figure 4-2 Hardware Status Window (Error Case)

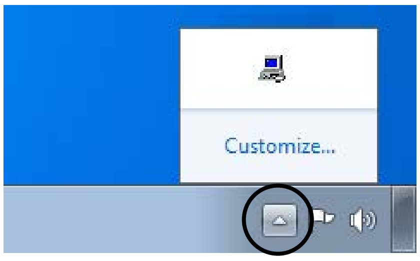
NOTICE
When the **Hardware status** window shows an error in hardware, resolve the error immediately.

4.1.3 Hardware status icon

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

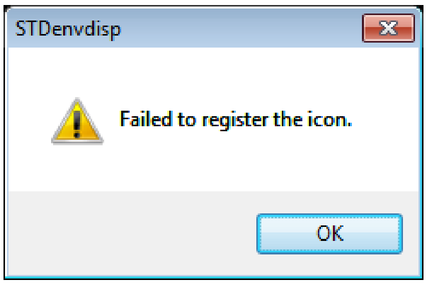


Note that, in the default factory settings, the icon is not shown. If you click the arrow at the side of the notification area, the icon will appear. If you want to always show the icon in the notification area of the taskbar, click **Customize** and configure the icon to be always shown in the notification area. (If you are using Windows® 10, right-click the arrow at the side of the notification area. A menu will appear. On the menu, click **Properties**. Then, a window will appear. In the window, click **Customize**, and then **Select which icons appear on the taskbar**. Then, configure the icon to be always shown in the notification area in the taskbar.)



NOTE

In rare cases, the registration of the hardware status icon fails and the following message is displayed. If this happens, follow the procedure below to retry the registration of the hardware status icon.



STDenvdisp

Failed to register the icon.

OK

＜For Windows® Embedded Standard 7＞

1. Click **OK** in the message above.
2. Point to **Start** > **All Programs** > **Startup**, and click **RAS Status**.

＜For Windows® 10＞

1. Click **OK** in the message above.
2. Click **Start**.
3. Click **RAS Software** from the list of applications, and then click **RAS Status**.

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

Table 4-3 shows a list of displayed icons and a description of each icon. A description of the displayed icon is shown when you point the mouse cursor to the icon.

Table 4-3 Hardware Status Icon

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

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

Figures 4-3 and 4-4 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.

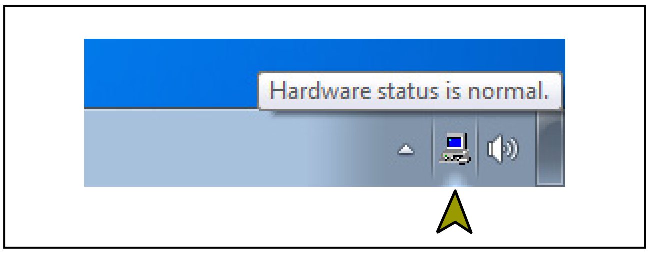

Hardware status is normal.

Figure 4-3 Example of Displaying the Description of an Icon (When the Hardware Status Is Normal)

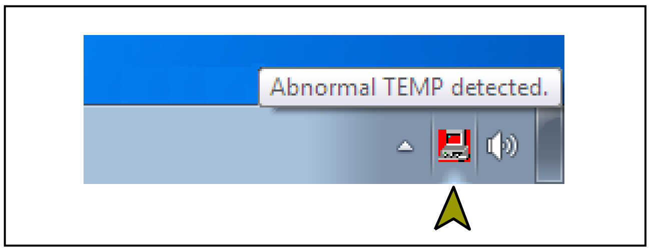

Abnormal TEMP detected.

Figure 4-4 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 popup menu is displayed.

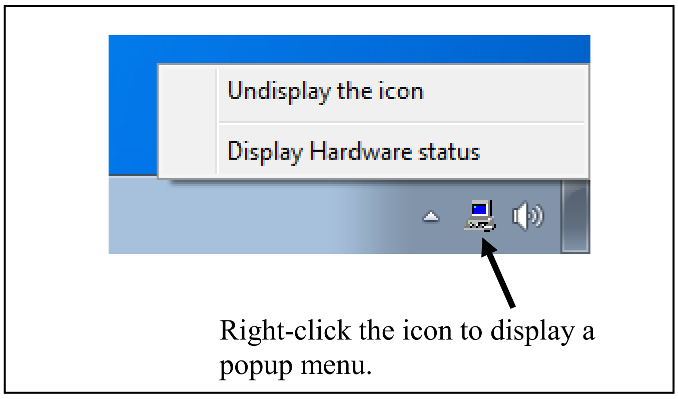


Figure 4-5 Menu of the Hardware Status Icon

● **Display Hardware status**

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

● **Undisplay the icon**

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

4.2 RAS Event Notification

4.2.1 Overview

When an event that must be reported to a user such as hardware failure occurs, this function notifies the event to an application by setting an event object to the signaled state. An application can detect an event such as hardware failure by monitoring when the event object is set to the signaled state.

The event object is reset to the nonsignaled state when the cause of the event is cleared.

4.2.2 Detecting an event

Follow the procedure below to detect an event:

1. Use the OpenEvent Windows API function to get the handle to the event object. Specify SYNCHRONIZE for the parameter dwDesiredAccess (the access to the event object).
2. Use the WaitForSingleObject or WaitForMultipleObject Windows API function to monitor when the event object is set to the signaled state.

Table 4-4 is a list of events to be reported to a user and their respective event objects.

Table 4-4 Reported Events

No.	Event	Event object name
1	The temperature inside the chassis became abnormal.	W2KRAS_TEMP_ERR_EVENT
2	SMART anticipates storage failure.	W2KRAS_HDD_PREDICT_EVENT
3	A frequent error correction occurredin one of the memory slots.	HFW_MEMORY_ERR_EVENT

No.2: This includes the case that getting the failure prediction status of the storage fails.

NOTE

When you use an event object in a program, you need to add “Global\” to the beginning of the name of the event object.

4.2.3 Example of using event objects

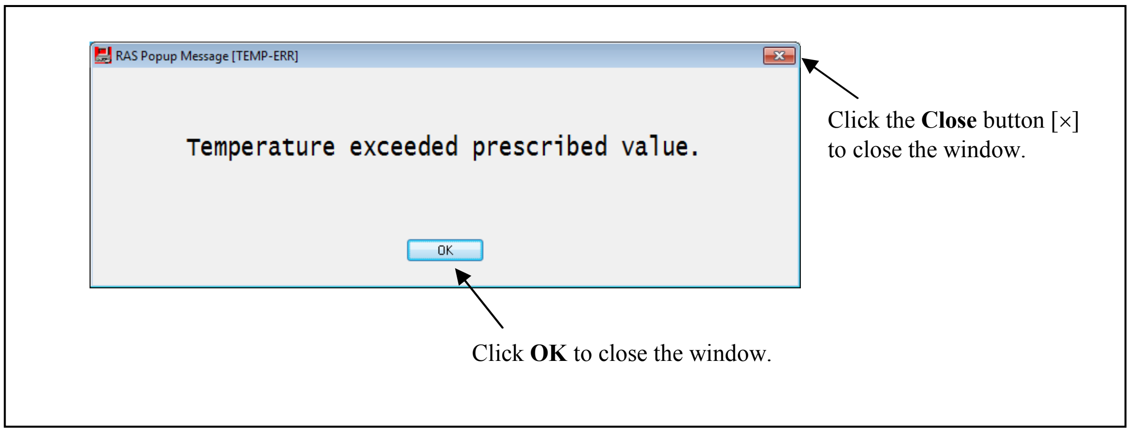
We provide a sample program in C (MemErr.c) to show how to monitor event objects. For information about the name of the sample program and where you can find it, see “6.2 Sample Programs”.

4.3 Popup Notification

4.3.1 Overview

When an event that must be reported to a user such as hardware failure occurred, this function notifies the event to a user by displaying a popup message on the desktop. Using this function, a user can know an event such as hardware failure occurred.
More specifically, a popup message is displayed in the following cases:
● The temperature inside the chassis becomes abnormal.
● SMART anticipates storage failure.
● A frequent error correction occurs in memory.

Figure 4-6 shows an example of popup message notification when the temperature inside the chassis becomes abnormal.


RAS Popup Message [TEMP-ERR]

Temperature exceeded prescribed value.

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

Figure 4-6 Example of Popup Message

4.3.2 Messages to be displayed

The following table shows a list of popup notification messages this function outputs. It should be noted that you can edit those messages. For information about how to edit messages, see “3.1.4 Editing popup notification messages”.

Table 4-5 Messages Displayed

No.	Event	Popup notification message
1	The temperature inside the chassis becomes abnormal.	Temperature exceeded prescribed value.
2	SMART anticipates storage failure.	A failure may be imminent on the storage of the mSATA SSD%1.
3	Frequent error correction occurs in memory.	In the %2, error corrections have occurred with high frequency.

No.2: %1 denotes the storage number.
No.2: This includes the case that getting the failure prediction status of the storage fails.
No.3: %2 denotes the name of DIMM.

4.3.3 Popup notification settings

This function can be enabled or disabled in the **RAS Setup** window. In the default factory-shipped setting, this function is disabled. If this function is disabled, popup messages are not displayed.
For details, see “3.1.3 Using the RAS Setup window”.

4.4 Remote Notification

4.4.1 Overview

If you use this function, from a remote device through the network, you can check hardware conditions that can only be checked beside this equipment without this function. Even when hardware conditions cannot be checked beside 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 function uses Simple Network Management Protocol (SNMP) to notify 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 one location.

NOTE

- The remote notification uses SNMP, a protocol in the application layer of the TCP/IP, and User Datagram Protocol (UDP) in the transport layer. That means if the network load is high, hardware conditions may not be received properly.
- The remote notification uses the SNMP service, a standard feature of Windows®. For information about how to enable the standard Windows® SNMP service, see “4.4.3　Enabling the remote notification”.

4.4.2 Hardware conditions that can be acquired by using remote notification

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

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

The following transitions of the hardware conditions are notified by using a trap.

(1) Temperature condition inside the chassis
- Normal → Error
- Error → Normal

(2) Storage condition
- Normal → Failure anticipated

(3) Memory condition
- Normal → Frequent error correction
- Frequent error correction → Normal

(4) Operational mode
- HF-W stopped → Started in normal mode
- Running in normal mode → Running in simulation mode

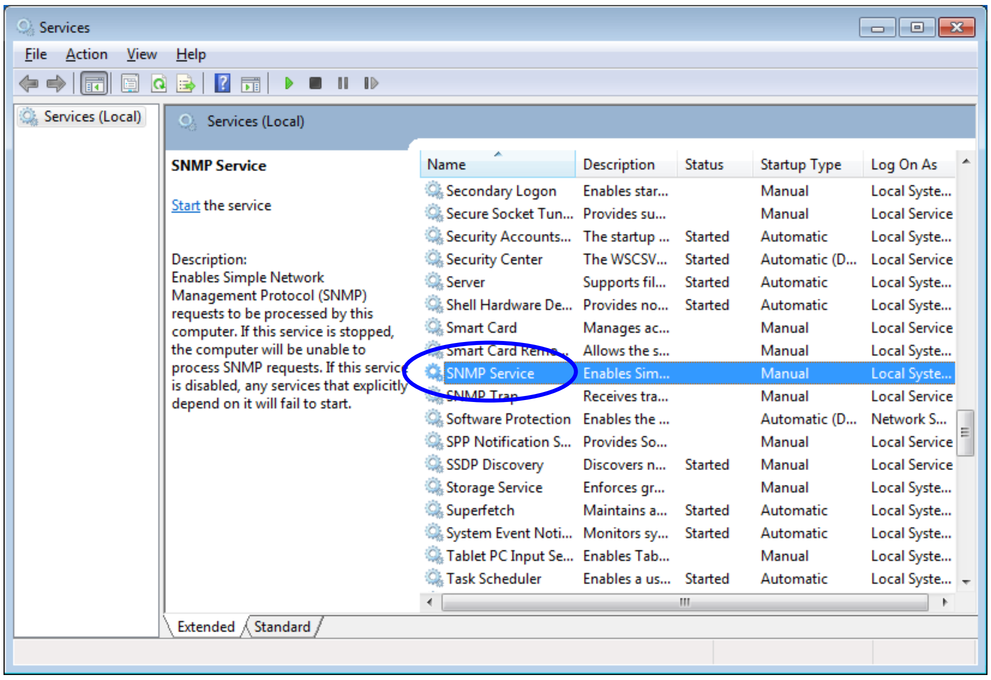
4.4.3 Enabling the remote notification

This function is disabled in the default factory-shipped settings. The remote notification uses the standard Windows® SNMP service. If you enable the SNMP service, the remote notification is enabled.

When you use the remote notification, follow the procedure below to enable the SNMP service:

(1) Starting the "**SNMP Service Properties**" window

1. If you are not logged on to the computer as an administrator, log on to the computer as an administrator.
2. Follow the procedure below to open **Control Panel**.
   If the OS is Windows® Embedded Standard 7, click **Start** > **Control Panel**.
   If the OS is Windows® 10, right-click **Start** to display menu **and** click **Control Panel**.
3. Click **System and Security** > **Administrative Tools**. Double-click **Services**.
4. Double-click **SNMP Service** to start the "**SNMP Service Properties**" window.



SNMP Service

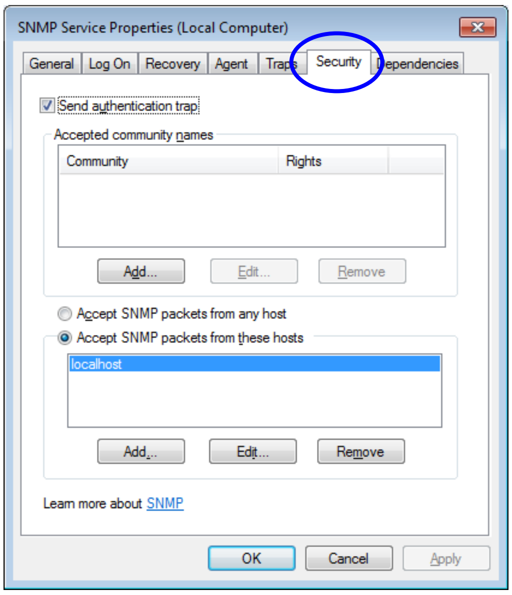
Start the service

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.

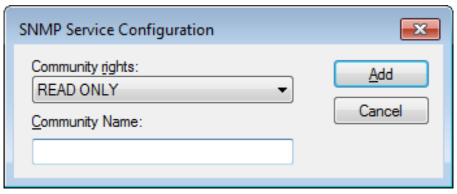
Name	Description	Status	Startup Type	Log On As
Secondary Logon	Enables star...		Manual	Local Syste...
Secure Socket Tun...	Provides su...		Manual	Local Service
Security Accounts...	The startup ...	Started	Automatic	Local Syste...
Security Center	The WSCSV...	Started	Automatic (D...	Local Service
Server	Supports fil...	Started	Automatic	Local Syste...
Shell Hardware De...	Provides no...	Started	Automatic	Local Syste...
Smart Card	Manages ac...		Manual	Local Service
Smart Card Remo...	Allows the s...		Manual	Local Syste...
SNMP Service	Enables Sim...		Manual	Local Syste...
SNMP Trap	Receives tra...		Manual	Local Service
Software Protection	Enables the ...		Automatic (D...	Network S...
SPP Notification S...	Provides So...		Manual	Local Service
SSDP Discovery	Discovers n...	Started	Manual	Local Service
Storage Service	Enforces gr...		Manual	Local Syste...
Superfetch	Maintains a...	Started	Automatic	Local Syste...
System Event Noti...	Monitors sy...	Started	Automatic	Local Syste...
Tablet PC Input Se...	Enables Tab...		Manual	Local Syste...
Task Scheduler	Enables a us...	Started	Automatic	Local Syste...

(2) SNMP security configuration

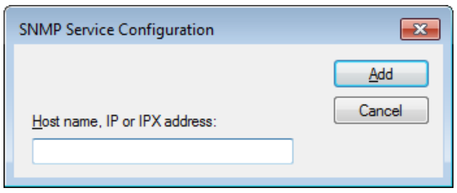
1. In the "**SNMP Service Properties**" window, select the **Security** tab.



2. If you want to send a trap message whenever authentication fails, select the "**Send authentication trap**" check box.
3. Under "**Accepted community names**," click **Add**. The "**SNMP Service Configuration**" window is displayed. In the "**Community rights**" list, select **READ ONLY**. In the **Community Name** box, type the community name you want, and click **Add**.



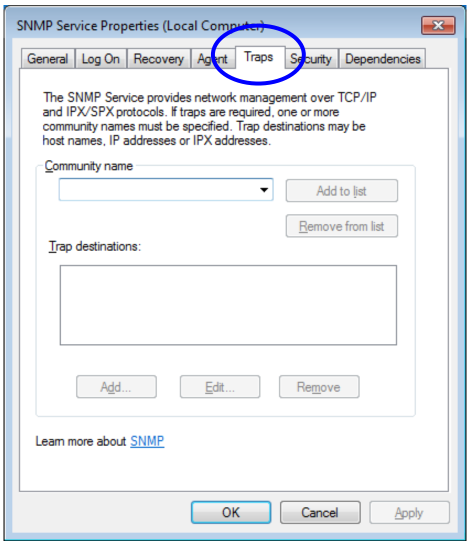
4. Specify whether to accept SNMP packets from hosts.
   If you want to accept SNMP packets from any manager on the network:
   • Select **Accept SNMP packets from any host**.
   If you want to restrict SNMP packets:
   • Select **Accept SNMP packets from these hosts**.
   • Click **Add**.
   • The "**SNMP Service Configuration**" window is displayed. Enter the host name, IP or IPX address of a host you want to accept the SNMP packets from, and click **Add**.



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

(3) SNMP trap configuration

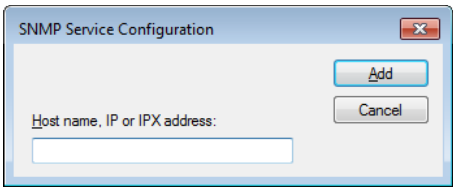
1. In the “**SNMP Service Properties**” window, select the **Traps** tab.



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.

Learn more about SNMP

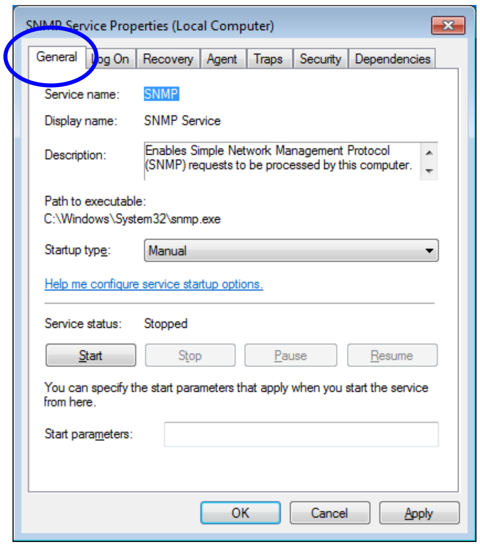
2. Under **Community name**, type the name of the community that trap messages are sent to, and click “**Add to list**.”
3. Under **Trap destinations**, click **Add**. The “**SNMP Service Configuration**” window is displayed. Enter the host name, IP or IPX address of a destination you want to send traps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



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

(4) Starting the SNMP service

1. In the “**SNMP Services Properties**” window, select the **General** tab.



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:

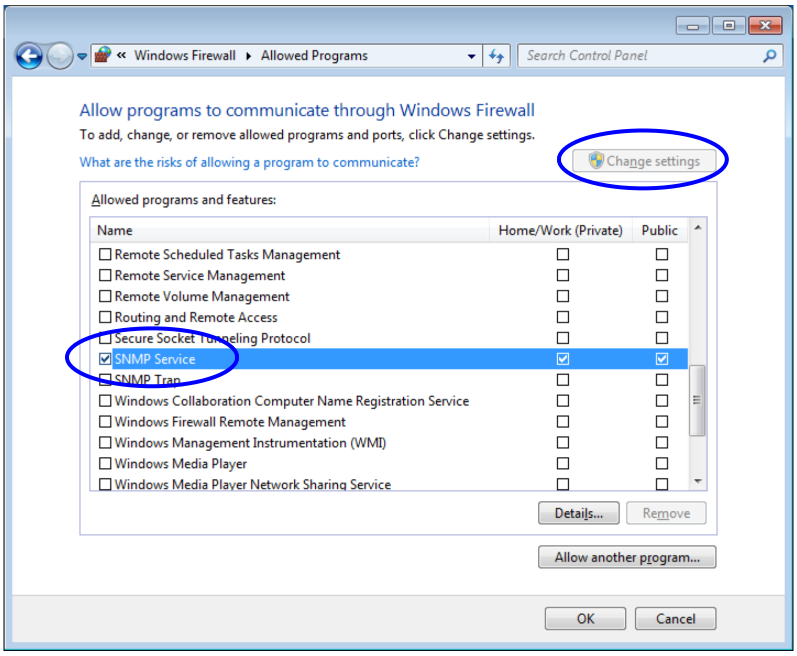
2. Click **Start**. The SNMP service starts and the remote notification for the hardware status is enabled.
3. In order to start the SNMP service automatically at the next OS startup, in the **Startup type** list, select **Automatic**.
4. In the “**SNMP Service Properties**” window, click **OK**.

NOTE

- If when the SNMP starts, there is an error that is configured to be notified by using a trap, the trap is sent at the timing when the SNMP starts.
- If Windows Firewall is configured to block the SNMP service, you cannot acquire the hardware status from a remote device. If you set up the firewall to block SNMP service, follow the procedure below to allow SNMP service to communicate via Windows Firewall.
  But, SNMP service can pass through Windows Firewall by default, you do not have to go through the following procedure.

1. If you are not logged on to the computer as an administrator, log on to the computer as an administrator.
2. Open **Control Panel**, click “**System and Security**.”
3. If the OS is Windows® Embedded Standard 7, click “**Allow a program through Windows Firewall**” under “**Windows Firewall**.”
   If the OS is Windows® 10, click “**Allow an app through Windows Firewall**” under “**Windows Firewall**.”

4. If the OS is Windows® Embedded Standard 7, the “**Allowed Programs**” window appears. Click **Change settings**, and then select the **SNMP Service** check box in “**Allowed programs and features**.
   If the OS is Windows® 10, the “**Allowed apps**” window appears. Click “**Change settings**” and then select the **SNMP Service** check box in “**Allowed apps and features**.”



Allow programs to communicate through Windows Firewall

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

What are the risks of allowing a program to communicate?

Allowed programs and features:

Name	Home/Work (Private)	Public
☐ Remote Scheduled Tasks Management	☐	☐
☐ Remote Service Management	☐	☐
☐ Remote Volume Management	☐	☐
☐ Routing and Remote Access	☐	☐
☐ Secure Socket Tunneling Protocol	☐	☐
☑ SNMP Service	☑	☑
☐ SNMP Trap	☐	☐
☐ Windows Collaboration Computer Name Registration Service	☐	☐
☐ Windows Firewall Remote Management	☐	☐
☐ Windows Management Instrumentation (WMI)	☐	☐
☐ Windows Media Player	☐	☐
☐ Windows Media Player Network Sharing Service	☐	☐

5. Click **OK**.

4.4.4 Objects in the extended MIB for HF-W

In order to acquire the hardware status of this equipment from a remote device, use the extended MIB for HF-W. This subsection provides a list of objects defined in the extended MIB for HF-W and a description of those objects.

(1) Objects related to the hardware conditions and settings

Table 4-6 shows a list of the objects related to the hardware status and a description of those objects. The object ID of each object is obtained by either 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.hfwRasStatus.x (x is an object in the table below)
or
.1.3.6.1.4.1.116.5.45.1.y (y is an object number in the table below)

Table 4-6 Objects Related to the Hardware Status

No.	Object	Object number	Description	Description of the values
1	hfwTemp	2	Temperature condition group	-
2	hfwTemp.tempNumber	2.1	Number of monitored temperatures	-
3	hfwTemp.tempTable.TempEntry.tempIndex	2.2.1.1	Index number of tempEntry	-
4	hfwTemp.tempTable.TempEntry.tempName	2.2.1.2	Name of the temperature to bemonitored	Internal temperature: Temperature inside the chassis
5	hfwTemp.tempTable.TempEntry.tempStatus	2.2.1.3	Temperature condition	1: Normal 2: Error
6	hfwHdd	3	Storage condition group	-
7	hfwHdd.hddNumber	3.1	Number of monitored storage	-
8	hfwHdd.hddTable.hddEntry.hddIndex	3.2.1.1	Index number of hddEntry	-
9	hfwHdd.hddTable.hddEntry.hddStatus	3.2.1.2	Storage condition	1: Healthy 3: Failure anticipated 99: Unknown
10	hfwMem	5	Memory condition group	-
11	hfwMem.memNumber	5.1	Number of monitored memory slots	-
12	hfwMem.memTable.memEntry.memIndex	5.2.1.1	Index of memEntry	-
13	hfwMem.memTable.memEntry.memName	5.2.1.2	Memory name	DIMM 1: DIMM 1 slot
14	hfwMem.memTable.memEntry.memStatus	5.2.1.3	Memory condition	1: Normal 2: Error (Frequent error correction) 3: Not mounted

No.2: For this equipment, the value is set to 1.
No.7: The number of storages that can be mounted in the HF-W is set for the number of monitored storages.
For this equipment, the number is 1.
No.11: The number of memory slots is set for the number of monitored memory slots.
For this equipment, the number is 1.

Table 4-7 shows a list of the objects related to the RAS function settings and a description of those objects. The object ID of each object is obtained by either 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 (x is an object in the table below)
or
.1.3.6.1.4.1.116.5.45.2.y (y is an object number in the table below)

Table 4-7 Objects Related to the RAS Function Settings

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

Table 4-8 shows a list of the objects related to operational modes and a description of those objects. The object ID of each object is obtained by either 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 (x is an object in the table below)
or
.1.3.6.1.4.1.116.5.45.3.y (y is an object number in the table below)

Table 4-8 Objects Related to Operational Modes

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

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

Table 4-9 Objects Related to the Extended MIB for HF-W

No.	Object	Object number	Description	Description of the values
1	Version	1	Version number of the extended MIB for HF-W	-
2	Revision	4	Revision number of the extended MIB for HF-W	-

(2) Objects related to the trap notification

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

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

Table 4-10 Objects Related to the Trap Notification (When an Error Occurs)

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

No.2: %1 denotes the storage number.
No.3: %2 denotes the name of the memory slot with frequent error correction.

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

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

Table 4-11 Objects Related to the Trap Notification (When the Equipment Has Recovered from an Error)

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

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

Table 4-12 shows a list of the objects related to the trap notification when the equipment has started in normal mode and a description of those objects. The enterprise ID for the trap notification related to operational modes is as follows.

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

Table 4-12 Objects Related to the Trap Notification (Operational Modes)

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

4.4.5 Extended MIB file for HF-W

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

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

4.5 Status Acquisition by Using the RAS Library

By using the RAS library functions, the following hardware conditions can be acquired. For details about the library functions, see “6.1 RAS Library Interface”.

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

CHAPTER 5 CONTROLLING THE HARDWARE

The RAS feature can have the following controls over this equipment.

(1) Automatic shutdown of the equipment
When a hardware error occurs, the equipment can be automatically shut down. For details, see “5.1 Automatic Shutdown of the Equipment”.

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

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

5.1 Automatic Shutdown of the Equipment

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

5.1.1 Automatic shutdown when detecting abnormally high temperature

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

- This function can be enabled or disabled in the **RAS Setup** window. In the default factory setting, this function is disabled. For details, see "3.1.3 Using the RAS Setup window".
- After an automatic shutdown initiated by this function is complete, the power is turned off.
- Alternatively; a user application can detect abnormally high temperature using a RAS event and shut down the equipment. For information about a RAS event, see "4.2 RAS Event Notification".

NOTE

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

5.2 Memory Dump due to Timeout Detection

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

• In the **RAS Setup** window, you can configure whether to enable memory dump using this feature. For details, see "(2) Watchdog timer setting" in "3.1.3 Using the RAS Setup window".

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

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

• STOP error code: 0x00009222

NOTE

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

5.3 Controlling the Hardware by Using the RAS Library

By using the RAS library functions, you can shut down the system and control the external general purpose I/O. For details about the library functions, see "6.1 RAS Library Interface".

● To shut down the system: Use the BSSysShut function.

● To control the watchdog timer: Use the WdtControl function.

● To control the external general purpose output: Use the GendoControlN functions.

● To control the external general purpose input: Use the GetGendiN, RegisterDICallback and UnRegisterDICallback functions.

This Page Intentionally Left Blank

CHAPTER 6 LIBRARY FUNCTIONS

A user application can get and control the hardware status of this equipment by using the RAS library.
For information about the hardware specifications of the external general purpose I/O described in this chapter and the usage of each contact, refer to “HF-W100E INSTRUCTION MANUAL (manual number WIN-62-0069)”.

6.1 RAS Library Interface

6.1.1 Overview

This chapter describes the interface to the functions provided by the RAS library.
Table 6-1 shows a list of RAS library functions. Those functions mentioned above are offered in the DLLs (w2kras.dll, hfwras.dll).

Table 6-1 RAS Library Functions

No.	Function name	Use	DLL
1	BSSysShut	Shuts down the equipment.	w2kras.dll
2	WdtControl	Retriggers and stops the watchdog timer.	
3	GendoControlN	Controls the external general purpose outputs.	
4	GetGendiN	Gets the status of the external general purpose inputs.	
5	RegisterDICallBack	Register a callback function to be executed when a rising edge (low level to high level) or falling edge (high level to low level) is detected in the external general purpose inputs.	
6	UnRegisterDICallBack	Unregister the callback function registered by RegisterDICallBack.	
7	MconWriteMessage	Records a specified message (characters) in this equipment’s own log files.	
8	GetMemStatus	Gets the condition of the memory built into this equipment.	
9	hfwDiskStat	Gets the condition of the storage.	hfwras.dll

NOTE

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

The functions number 1, 2, and 7 through 9 can be called from Visual Basic® (.NET is required for 64-bit operating systems). When you call functions number 1, 2, and 7 from Visual Basic®, add “_VB” to the end of the name of each function. Function parameters are the same. For example, when you call the WdtControl function from Visual Basic®, use the function name “WdtControl_VB”.

The following files are provided as import libraries:
%ProgramFiles%\HFWRAS\lib\w2kras.lib
%ProgramFiles%\HFWRAS\lib\hfwras.lib
When you use a library, link the corresponding import library.

The following files are provided as header files for the libraries:
%ProgramFiles%\HFWRAS\include\w2kras.h
%ProgramFiles%\HFWRAS\include\hfwras.h
When you use a library in C, include the corresponding header file.

6.1.2 Shutdown function (BSSysShut)

<Name>

BSSysShut - System shutdown

<Syntax>

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

<Description>

BSSysShut shuts down the system.

The reboot argument is used for specifying whether to reboot the system after the shutdown.

reboot = 0: The power to this equipment is turned off after the shutdown.

reboot ≠ 0: The system reboots after the shutdown.

<Diagnosis>

0: Successful completion (System shutdown process 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 information about the name of the sample program and where you can find it, see “6.2 Sample Programs”.

6.1.3 Watchdog timer control function (WdtControl)

<Name>

WdtControl - Watchdog timer control / 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.
In order to use this function, in the **RAS Setup** window, select **Retriggered by application program** under **Watchdog timer setting**. If the watchdog timer setting is different, this function terminates with an error. If you call the GetLastError Windows API function, an error code W2KRAS_WDT_NONMANUAL is returned.
Each parameter of this function is explained below.

dwCmd:

This parameter specifies which action is performed on the watchdog timer. The following options are available for this parameter.

Table 6-2 List of Actions of WdtControl Specified by dwCmd

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

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

pdwCount:

If dwCmd is WDT_SET, you can configure the timeout of the watchdog timer by specifying the timeout of the watchdog timer in the variable pointed by pdwCount and calling this function.
Specify a value between 1 and 63 in seconds. If a value other than the above is specified, this function terminates with an error. If you call the GetLastError Windows API function, an error code W2KRAS_INVALID_PARAMETER is returned.
When this function returns, the value of the variable pointed by pdwCount is undefined. Do not use the value.
If dwCmd is WDT_STOP, the value of pdwCount is ignored. When this function returns, the value of the variable pointed by pdwCount is undefined. Do not use the value.

NOTE

- If you want to stop monitoring using the watchdog timer when a user program exits or due to shutdown, you must stop the Watchdog timer so that a timeout does not occur.
- The time it takes to generate timeout is longer than the timeout value set by the application. This is a correct operation. This situation occurs because it takes about 1.2 to 1.3 seconds for the hardware timer on this equipment to actually enable the timeout setting set by the application. If the timeout is set to 30 seconds, memory dump starts about 31 seconds after the timeout is set.

<Diagnosis>

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

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

Error code (value)	Description
W2KRAS_INVALID_PARAMETER (0x2001)	There is an error in the specified parameters.
W2KRAS_WDT_NONMANUAL (0x2002)	This function cannot be used because **Retriggered by application program** is not selected under **Watchdog timer setting** in the **RAS Setup** window.
W2KRAS_NOT_INITIALIZE (0x2005)	Startup of the RAS software has not completed yet.
W2KRAS_INTERNAL_ERROR (0x2007)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

<Sample program>

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

For information about how to monitor the operational state of a user program using the watchdog timer, see “2.4.2 Using a watchdog timer for monitoring a user program”.

6.1.4 Control functions for the external general purpose outputs (GendoControlN)

<Name>

GendoControlN - Control for the external general purpose outputs (output1 to 8)

<Syntax>

```
#include <w2kras.h>
BOOL GendoControlN(UCHAR ucOutput, UCHAR ucMask);
```

<Description>

This function controls external general purpose outputs (output1 to 8).
The parameters of this function are explained below.

ucOutput:

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

ucMask:

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

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

bit0	output1
bit1	output2
bit2	output3
bit3	output4
bit4	output5
bit5	output6
bit6	output7
bit7	output8

<Diagnosis>

If this function completes successfully, the function returns TRUE. If this function terminates with an error, the function returns FALSE.
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
W2KRAS_INVALID_PARAMETER (0x2001)	There is an error in the specified parameters.
W2KRAS_INTERNAL_ERROR (0x2007)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

<Explanation>

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

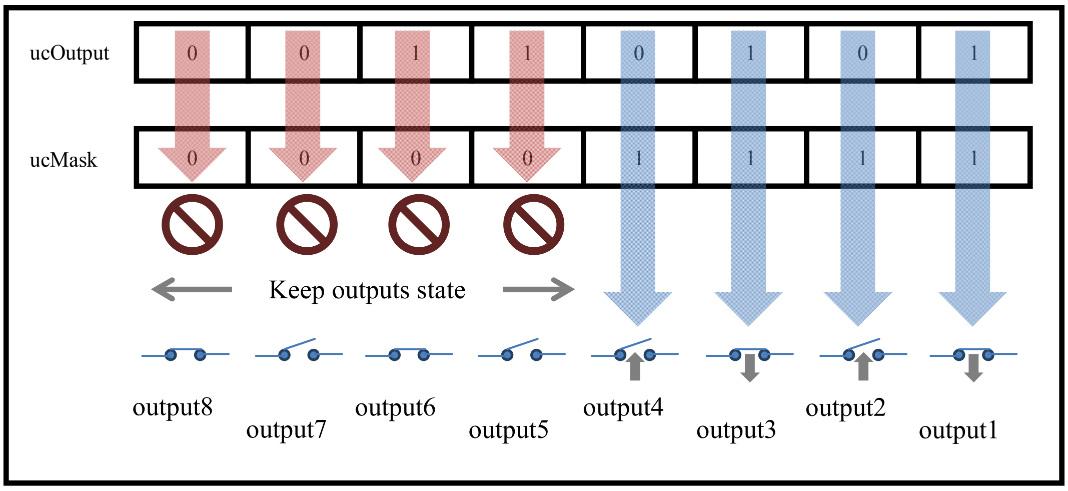


Figure 6-1 Operation Example of GendoControlN Function

<Supplementary information>

Figure 6-2 shows the operation of the external general purpose output1 when the GendoControlN function is used. The dashed lines show the general purpose output level, and the bold line show the transition of the general purpose output1.

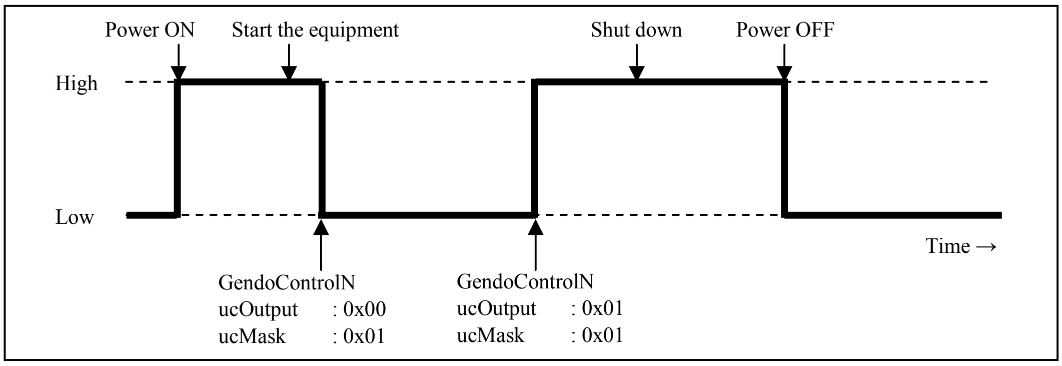


Figure 6-2 Operation Example of the External General Purpose Output1

<Sample program>

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

6.1.5 Get functions for the external general purpose inputs (GetGendiN)

<Name>

GetGendiN - Acquire status of external general purpose inputs (input1 to 7)

<Syntax>

```
#include <w2kras.h>
BOOL GetGendiN(PUCHAR pucInput);
```

<Description>

This function acquires the input state of external general purpose inputs (input1 to 7).
The parameters of this function are explained below.

pucInput:

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

Table 6-4 Bit Allocation of pcInput for GetGendiN

bit0	input1
bit1	input2
bit2	input3
bit3	input4
bit4	input5
bit5	input6
bit6	input7
bit7	Not used

<Diagnosis>

If this function completes successfully, the function returns TRUE. If this function terminates with an error, the function returns FALSE.
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
W2KRAS_NOT_INITIALIZE (0x2005)	Startup of the RAS software has not completed yet.
W2KRAS_INTERNAL_ERROR (0x2007)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

<Sample program>

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

6.1.6 Register function for the external general purpose inputs callback function (RegisterDICallback)

(1) Function interface (RegisterDICallback)

<Name>

RegisterDICallback - Register function for the external general purpose inputs callback function

<Syntax>

```
#include <w2kras.h>
DWORD RegisterDICallBack(USERFUNC userFunc, UCHAR ucMaskHigh,
                                UCHAR ucMaskLow, DWORD dwOption);
```

<Description>

This function is register a callback function to be executed when a rising edge (low level to high level) or falling edge (high level to low level) is detected in the external general purpose inputs.
The parameters of this function are explained below.

userFunc:
Specify a pointer to the callback function.

ucMaskHigh:
Specify general purpose input that detects a rising edge (low level to high level).
Set the bit corresponding to the general purpose input to 1 (refer to Table 6-5 for bit allocation for each general purpose input).

ucMaskLow:
Specify general purpose input that detects a falling edge (high level to low level).
Set the bit corresponding to general purpose input to 1 (refer to Table 6-5 for bit allocation for each general purpose input).

dwOption:
Specify the behavior of the callback function. Specify any combination of the values shown in the table below.

Value	Description
REGIDICB_DEFAULT	No option specified.
REGIDICB_INITIAL_DETECTION	It will notify at the first detection. Example: “ucMaskHigh = 1” setting will notify even if general purpose input1 is already in the high level state at the start of monitoring.

<Diagnosis>

If this function completes successfully, the function returns ID of Callback function. If this function terminates with an error, the function returns REGISTERDI_CALLBACK_ERROR (0x00000000).
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
W2KRAS_INVALID_PARAMETER (0x2001)	There is an error in the specified parameters.
W2KRAS_NOT_INITIALIZE (0x2005)	Startup of the RAS software has not completed yet.
W2KRAS_INTERNAL_ERROR (0x2007)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

<Sample program>

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

(2) Callback function to be registered

This section explains the callback function specified in userFunc with the RegisterDICallBack function.

<Function type>

USERFUNC

<Syntax>

```
Typedef VOID (* USERFUNC)(DWORD dwState, UCHAR ucEdgeHigh,
                          UCHAR ucEdgeLow, DWORD dwID);
```

<Description>

The parameters of this function are explained below.

dwState:

The operation state of external general purpose inputs is set.
Usually GPIOMTHREAD_RUN is set. If it is not GPIOMTHREAD_RUN, an error has occurred, execute error handling.

Value	Meaning	Description
GPIOMTHREAD_RUN (0x00000000)	Normal operation	This value is normally set.
GPIOMTHREAD_GET_ERROR (0xfffffffe)	Error occurrence (input value acquisition failure)	Indicates abnormality.
GPIOMTHREAD_WAIT_ERROR (0xffffffff)	Abnormal termination (monitoring thread termination)	Indicates abnormality.

ucEdgeHigh:

General purpose input that detects a rising edge (low level to high level) is set. With the general purpose input specified by ucMaskHigh / ucMaskLow, the bit at the edge detection point is set to 1 (refer to Table 6-5 for bit allocation for each general purpose input).

- Edge detection: 1
- No edge detection: 0

ucEdgeLow:

General purpose input that detects a falling edge (high level to low level) is set. With the general purpose input specified by ucMaskHigh / ucMaskLow, the bit at the edge detection point is set to 1 (refer to Table 6-5 for bit allocation for each general purpose input).

- Edge detection: 1
- No edge detection: 0

dwID:

The ID of callback function is set.

<Diagnosis>

If an error occurs during monitoring, it returns GPIOMTHREAD_GET_ERROR or GPIOMTHREAD_WAIT_ERROR in dwState of the callback function.
GPIOMTHREAD_GET_ERROR is a notification when an error occurs in a thread, and the monitoring thread continues to operate.
GPIOMTHREAD_WAIT_ERROR is a notification when an error that is difficult to continue monitoring thread occurs and the monitoring thread ends, so you need to register the callback function again.
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
W2KRAS_INTERNAL_ERROR (0x2007)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

(3) Operation explanation of external general purpose input monitoring

Execute the registered callback function when edge is detected with the general purpose input specified by ucMaskHigh / ucMaskLow.

In a realtime environment, response time from edge detection of general purpose input operates in 1 ms, but response time in Windows environment is not guaranteed.

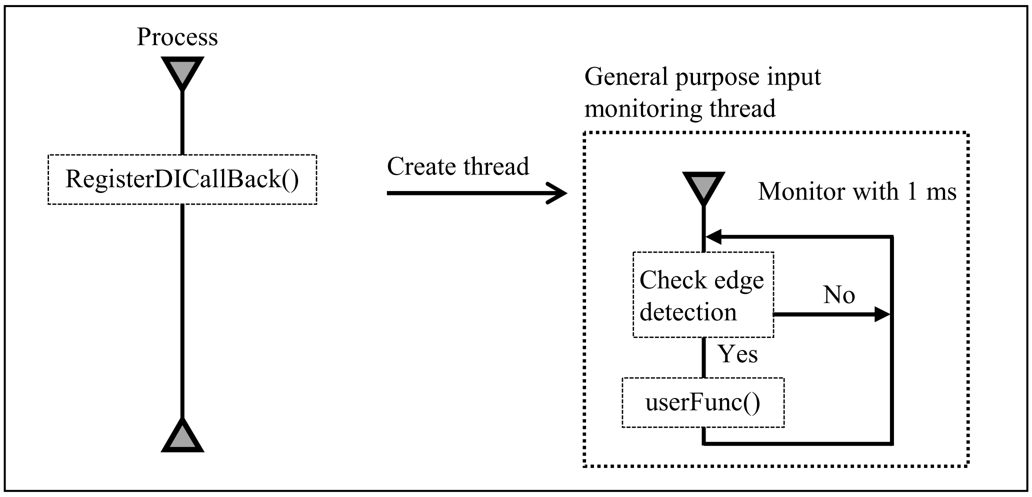


Figure 6-3 Overview of Monitoring Operation

ucMaskHigh and ucMaskLow of the RegisterDICallBack function and ucEdgeHigh and ucEdgeLow of the registered callback function correspond to each general purpose input bit as shown in Table 6-5.

Table 6-5 Bit Allocation

bit0	input1
bit1	input2
bit2	input3
bit3	input4
bit4	input5
bit5	input6
bit6	input7
bit7	Not used

Figure 6-4 shows the operation of the external general purpose input notification (when 0x03 is set in ucMaskHigh).
The dashed lines show the general purpose input level, and the bold line show the transition of the general purpose input1 and input2.

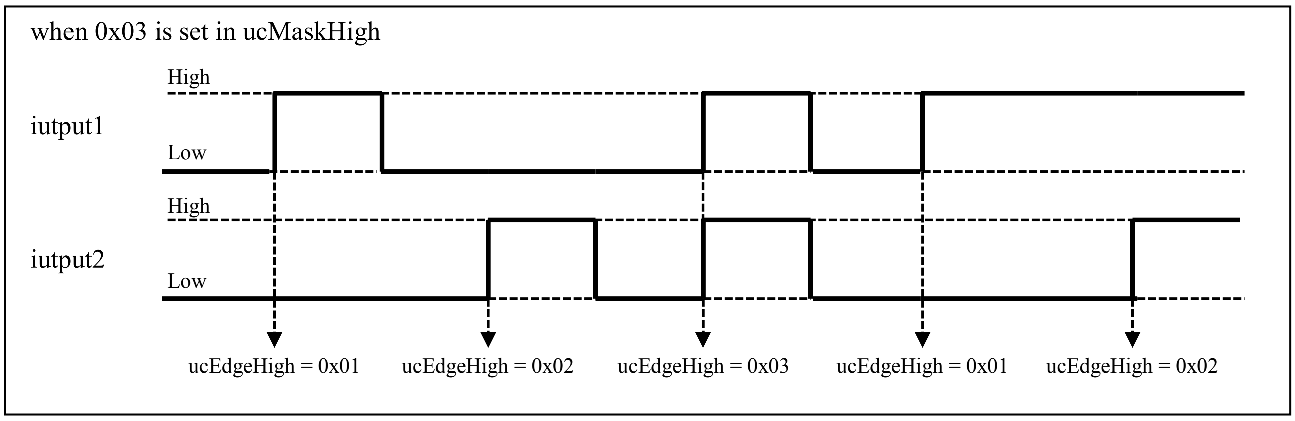


Figure 6-4 Operation Example of the External General Purpose Input Notification

<Others>

Register the callback function at the beginning of the process to be executed. (or when re-registration is necessary)
Since general purpose input monitoring thread is created each time a callback function is registered, the overhead is increased. Therefore, when there are multiple general purpose inputs that perform the same processing, for example, the corresponding general purpose inputs are grouped together and register a callback function.

6.1.7 Release function for the external general purpose inputs callback function (UnRegisterDICallBack)

<Name>

UnRegisterDICallBack - Unregister callback function registered with RegisterDICallBack function

<Syntax>

```
#include <w2kras.h>
BOOL UnRegisterDICallBack(DWORD dwID);
```

<Description>

This function unregisters the callback function registered with the RegisterDICallBack function.
Use this function to unregister callback functions registered with the RegisterDICallBack function.
Also, unregister the callback function at the end of the process to be executed.
The parameters of this function are explained below.

dwID:
Specify the callback function ID that acquired as the return value of the RegisterDICallBack function. (If more than one callback function is registered, pass the callback function ID to release.)

<Diagnosis>

If this function completes successfully, the function returns ID of Callback function. If this function terminates with an error, the function returns REGISTERDI_CALLBACK_ERROR (0x00000000).
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
W2KRAS_INVALID_PARAMETER (0x2001)	There is an error in the specified parameters.
W2KRAS_INTERNAL_ERROR (0x2007)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

<Sample program>

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

6.1.8 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 (file name: 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 currently used log file exceeds 64 KB, the log file used for logging is switched to another log file.

The parameters of this function are explained below.

lpBuffer:
This parameter specifies a pointer to a string that contains the message (characters) to be written.
In order to easily identify the application that records each log entry, specify a message that starts with the name of the application, for example.

<Checking log information>

This function records log information in the text format in the following files. When the size of the log recorded in the log file currently used exceeds 64 KB, the log file used for logging is switched to another log file.
• %ProgramFiles%\HFWRAS\log\hfwrasa.log
• %ProgramFiles%\HFWRAS\log\hfwrasb.log

You can check log information by opening the files above using an application such as Notepad.
The format of 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 (in AD)
MM: Month
DD: Day
hh: Hour (24-hour clock)
mm: Minute
ss: Second

Figure 6-5 Format of Log Information

Initially, each of the above files is filled with EOF (ASCII code: 0x1a).

<Sample program>

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

NOTICE

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

NOTE

- This function has the same name as the message console function provided by W2K-PLUS, software from Hitachi, but this function does not output to the message console.
- In order to reduce the amount of resources used, this function, for example, opens and closes a pipe every time the function is called. That means this function has relatively large overhead. Even when you record multiple lines of message, use one function call to output the message.
- This function does not support a Unicode string. Always use an ANSI string. The log entry for the message is stored in a text file. In a text file, the combination of two characters “\r\n” is recognized as a newline. If you want to include a newline in a string specified by lpbuffer, insert “\r\n” in the string.

6.1.9 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 stores the condition of the memory in this equipment to a structure pointed by pMemData. The parameters of this function are explained below.

pMemData:

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

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

When this function completes successfully, the maximum number of DIMMs is stored in Dimm_Number. This value varies depending on the model. Each element of Dimm_Status stores a value described in the following table. Note that the number of valid elements is Dimm_Number. (For example, if Dimm_Number is 2, the elements up to Dimm_Status[1] are valid.) The elements after that are reserved. The values of those reserved elements are undefined. Do not use those values.

Table 6-6 List of Values Stored in Each Element of Dimm_Status

Value	Description
MEMORY_NOMAL (0x00)	The memory is working properly.
MEMORY_ERR_DETECT (0x01)	Frequent error correction occurs.
MEMORY_NOT_MOUNTED (0x02)	DIMM is not mounted.

For this model, correspondence between elements of Dimm_Status and DMM names is as follows.

Element	DIMM name
Dimm_Status[0]	DIMM 1

<Diagnosis>

If this function completes successfully, the function returns TRUE. If this function terminates with an error, the function returns FALSE. When this function terminates with an error, the value stored in pMemData is invalid.
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
W2KRAS_INVALID_PARAMETER (0x2001)	There is an error in the specified parameters.
W2KRAS_NOT_INITIALIZE (0x2005)	Startup of the RAS software has not completed yet.
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 those error codes, refer to the Windows API help.

<Sample program>

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

6.1.10 Get function for the storage condition (hfwDiskStat)

<Name>

hfwDiskStat - Storage status acquisition

<Syntax>

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

<Description>

The hfwDiskStat function stores the storage conditions to a structure pointed by phfwDiskStatus.
The parameters of this function are explained below.

phfwDiskStatus:

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

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

Disk_Count stores “1”, the number of storage on this equipment.
Disk_Status[n] stores the condition of the mSATA SSD (n + 1). The following values are used to indicate a storage condition. Upper 16 bits are reserved. The values of those reserved bits are undefined. Do not use those bits.

Table 6-7 List of Values Stored in Disk_Status

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

<Diagnosis>

If this function completes successfully, the function returns TRUE. If this function terminates with an error, the function returns FALSE. When this function terminates with an error, the value stored in phfwDiskStatus is invalid.
When this function terminates with an error, call the GetLastError Windows API function to get the error code. Error codes returned by this function on its own are as follows.

Error code (value)	Description
HFWRAS_INVALID_PARAMETER (0x20000001)	There is an error in the specified parameters.
HFWRAS_NOT_INITIALIZE (0x20000002)	Startup of the RAS software has not completed yet.
HFWRAS_INTERNAL_ERROR (0x20000003)	An internal error has been generated.

Other error codes come from the Windows API functions used by this function. For details about those error codes, refer to the Windows API help.

<Sample program>

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

6.2 Sample Programs

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

The following table shows a list of sample programs.

Table 6-8 List of Provided Sample Programs

No.	File name	Description
1	shutd.c	A sample program for the BSSysshut function
2	wdt.c	A sample program for the WdtControl function
3	GendoControlN.c	A sample program for the GendoControlN function
4	GetGendiN.c	A sample program for the GetGendiN function
5	RegisterDICallback.c	A sample program for the RegisterDICallback function
6	UnRegisterDICallback.c	A sample program for the UnRegisterDICallback function
7	MCon.c	A sample program for the MconWriteMessage function
8	MemErr.c	A sample program for the GetMemStatus function and detecting a RAS event (For information about the RAS event notification, see “4.2 RAS Event Notification”.)
9	hfwDiskStat.c	A sample program for the hfwDiskStat function

CHAPTER 7 FEATURES RELATED TO MAINTENANCE AND FAILURE ANALYSIS

7.1 Log Information Collection Window

7.1.1 Overview

In the log information collection window, you can take the following actions by using a graphical user interface.

(1) Collecting log data

This function saves the data used for preventive maintenance and post-failure analysis of the problem. The data is compressed and saved as one file (File name: logsave.zip).

(2) Collecting memory dump files

This function collects the memory dump files saved by the OS. The data is saved as a compressed file (File name: memory.zip). At the same time, minimum memory dump files are also collected.

NOTICE

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

7.1.2 Starting the log information collection window

To start the log information collection window, follow the procedure below:

You need to have administrator privileges to use this window. Log on to the computer using an administrator account and start the window.

＜For Windows® Embedded Standard 7＞

1. Click **Start**.
2. Point to **All Programs** > **RAS Software**. Click **RAS Maintenance Support**.

＜For Windows® 10＞

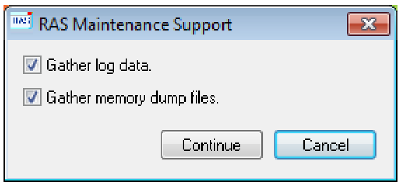
1. Click **Start**.
2. Click **RAS Software** from the list of applications.
3. Click **RAS Maintenance Support**.

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 may occur. If an error occurs, close the log information collection window on another console, and then start the window.

7.1.3 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 one of those two, clear the check box for the one you do not need, and then click **Continue**.

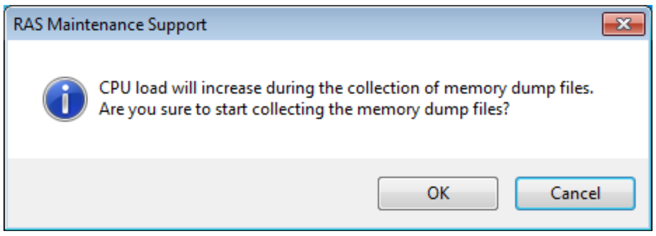


RAS Maintenance Support

☑ Gather log data.

☑ Gather memory dump files.

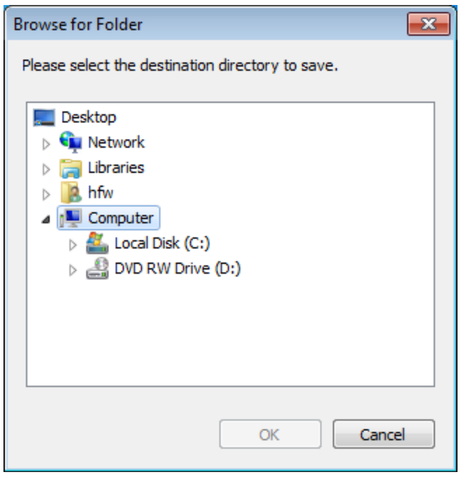
2. If the "**Gather memory dump files**" check box is selected, you will receive the following message. Click **OK**. If you click **Cancel**, you go back to the log information collection window without executing the maintenance operation.



RAS Maintenance Support

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

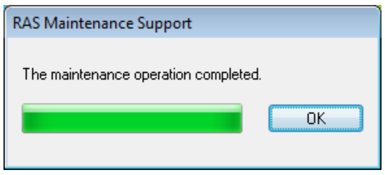
3. The following dialog box is displayed. Specify the destination directory to save and click **OK**. If you want to cancel the maintenance operation, click **Cancel**. If you click **Cancel**, you go back to the log information collection window without executing the maintenance operation.



Browse for Folder

Please select the destination directory to save.

4. The information selected at step 1 is collected. During the process, a window is displayed to show the progress. If the process finishes successfully, the following window appears.
   Do not do anything on the windows that appear during the process, and wait until the following window appears.



RAS Maintenance Support

The maintenance operation completed.

5. A directory is created under the directory specified as the destination directory to save. The name of the directory is assigned based on the date and time of the operation. Under the directory just created, collected data is saved.

```
Specified  ─ YYMMDD_        ┬ logsave.zip   : Log data file (*2)
directory    hhmm directory │
                  (*1)      ├ MEMORY.zip    : Memory dump file
                            │
                            └ Windows    ─── Minidump   : Minimum memory dump
                              directory      directory    files are stored
```

Figure 7-1　Organization of the Folders for Collected Data

(*1) The name of the directory is “YYMMDD_hhmm”.
YY: Lower two digits of the year, MM: Month, DD: Day, hh: Hour, mm: Minute
Example: If the date is saved at 13:59 on Jan 1, 2017
Directory name: “170101_1359”

(*2) The following data is saved.
- If **Gather log data** is selected: logsave.zip
- If **Gather memory dump files** are selected: MEMORY.zip and minimum memory dump files

7.1.4 Finishing the log information collection window

To close the log information collection window, click **Cancel.**

7.2 Logging the Trend of the Temperature inside the Chassis

7.2.1 Overview

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

7.2.2 Log files

The temperature inside the chassis is recorded in a log file at the specified logging intervals. When this equipment runs continuously for 8 hours or more, the highest and lowest temperatures in 8 hours are also recorded every 8 hours. For either file, if the log gets full, log entries are overwritten from the oldest entry.
Table 7-1 shows the name of each log file.

Table 7-1 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 intervals. (Maximum of 51200 entries)
	temp_mm.csv	The highest and lowest temperatures in 8 hours are recorded in this file. (Maximum of 1100 entries)

<Checking log information>

You can check log information by opening the log files using an application such as Notepad. The files are in the csv format. You can use spreadsheet or database software to load the log information and draw graphs.
You can also use the logsave command to collect those log files. For information about how to use the logsave command, refer to “8.3.1 Log information collection command (logsave)” in the “HF-W100E INSTRUCTION MANUAL (manual number WIN-62-0069)”.

<Log information format>

The format of 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 acquiring 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

7.2.3 Logging interval setup command

<Name>

tmplogset - Setting up the logging interval

<Syntax>

tmplogset

<Function>

This command configures the interval for logging the trend of the temperature inside the chassis.

The following shows how to use the command.

1. Start Command Prompt.
   You need to have administrator privileges to run this command. Log on to the computer as an administrator account and start Command Prompt.

2. At the command prompt, run the tmplogset command. The following initial screen is displayed along with the current setting. If you type 2 in the initial screen, the tmplogset command exits without changing the setting.

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

3. If you type 1 and then press **Enter**, the following message is displayed.

```
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. Type the number corresponding to the new interval you want to select and press **Enter**. If the number you typed is out of range, the following message is displayed to prompt you to type a correct number.

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

5. If you type a number between 1 and 3, the following message is displayed. The “×” below denotes the time you selected.

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

6. If you press **y** and then press **Enter**, the logging interval is updated to the new value and the command exits. The new setting becomes effective when the command exits. If you want to confirm the change, run this command again and check the initial screen.
   If you do not want to change the logging interval, press **n** and press **Enter**. The following message is displayed and the command exits without changing the setting.

```
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 is displayed and the command exits.

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

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

```
Error:   Systemcall failed. (API Name : Error Code)
```

In the message above, “API Name” actually shows the name of the Windows API with the error. “Error Code” shows the error code in hexadecimal. If this message is displayed, run the command again.

CHAPTER 8　SIMULATING THE HARDWARE STATUS

8.1　Hardware Status Simulation

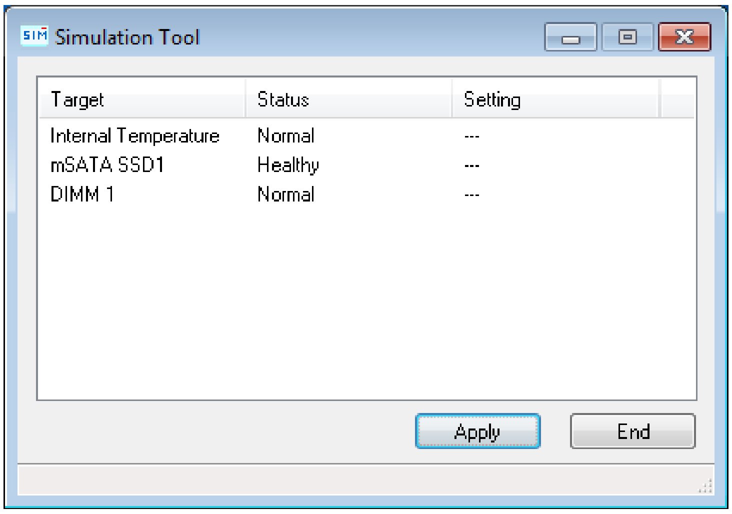
8.1.1　Overview

This function simulates the hardware status of this equipment. By simulating the hardware status, you can test a user application and check the notification interface of the RAS software without actual hardware failure.

When the hardware status is simulated, the RAS software is set to a special mode called "simulation mode". In "simulation mode", monitoring of the actual hardware status is disabled. You must never use this equipment in simulation mode for business use.

This function simulates the following hardware conditions.

・Temperature condition inside the chassis
・Storage failure prediction (SMART monitoring) condition
・Memory condition



Target	Status	Setting
Internal Temperature	Normal	---
mSATA SSD1	Healthy	---
DIMM 1	Normal	---

Figure 8-1　Simulation Tool Window

The monitoring function of the RAS software detects change in the simulated hardware status and notifies the change through various interfaces.

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

• Temperature condition inside the chassis:
"2.1　Monitoring Temperature inside the Chassis"
• Storage condition: "2.2　Storage Failure Prediction Function (SMART Monitoring)"
• Memory condition: "2.3　Memory Monitoring"

NOTICE

While the equipment is running in simulation mode, monitoring of the actual hardware status is disabled. Errors including fan failure and abnormal temperature cannot be detected. You must never use this equipment in simulation mode for business use. Use the simulation function only for testing a user application and checking the notification interface of the RAS software.

NOTE

・In simulation mode, the memory monitoring function records an event (Event ID: 525) in the event log only when memory error is detected for the first time. Afterward, even if memory error persists, the memory monitoring function does not record events in the event log.

・In simulation mode, when the storage failure prediction function records an event (Event ID: 265) of SMART detection in the event log, a string “XXXXXXXX” is used for the model name of the storage.
Also note that you cannot simulate a situation that the storage condition is “Unknown”.

8.1.2 Using the simulation function

Run the simulation mode start command at the command prompt to set the RAS software to the “simulation mode”. When the RAS software transitions to the “simulation mode”, the **Simulation Tool** window appears on the screen.

You can use this window to simulate the condition of hardware devices. Note that in order to exit the simulation mode, you need to restart this equipment.

This subsection explains the procedure of using the simulation mode.

(1) Overview of the procedure of using the simulation function

The figure below shows a rough flow chart of using this function. From when the simulation mode start command is executed until when OS shutdown completes, the RAS software runs in “simulation mode”.

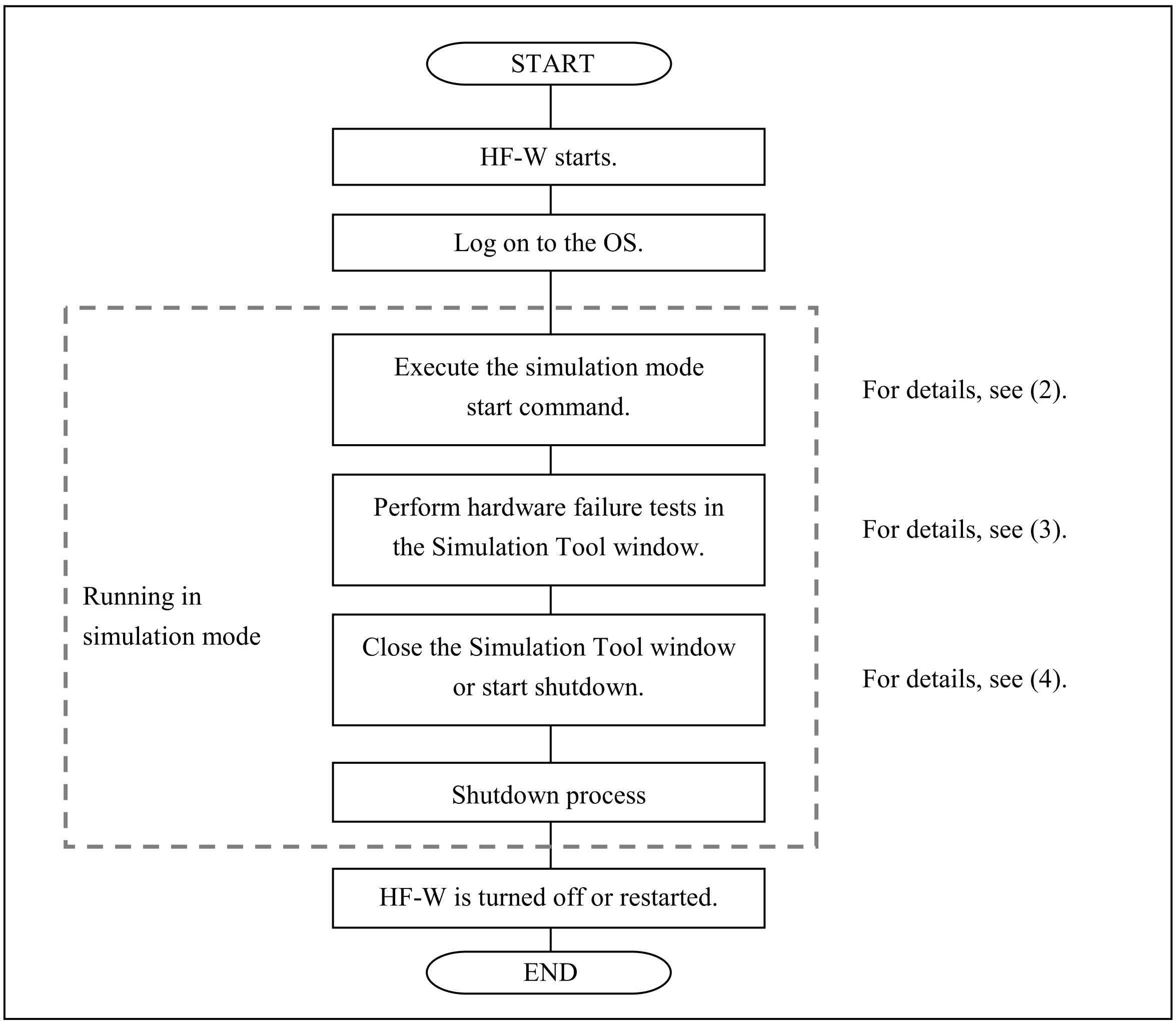

START
HF-W starts.
Log on to the OS.
Execute the simulation mode start command.
For details, see (2).
Perform hardware failure tests in the Simulation Tool window.
For details, see (3).
Running in simulation mode
Close the Simulation Tool window or start shutdown.
For details, see (4).
Shutdown process
HF-W is turned off or restarted.
END

Figure 8-2 Procedure to Use the Simulation Mode

(2) Starting the simulation mode

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

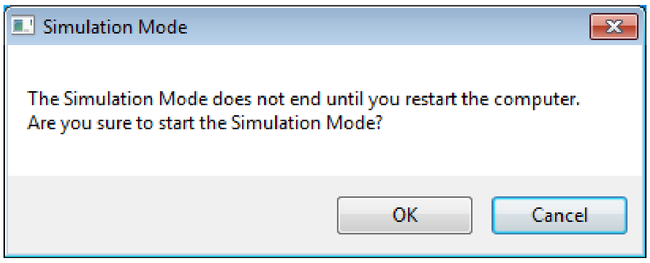
NOTE

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

1. Start Command Prompt.
   You need to have administrator privileges to run the simulation mode start command. Log on to the computer using an administrator account and start Command Prompt.

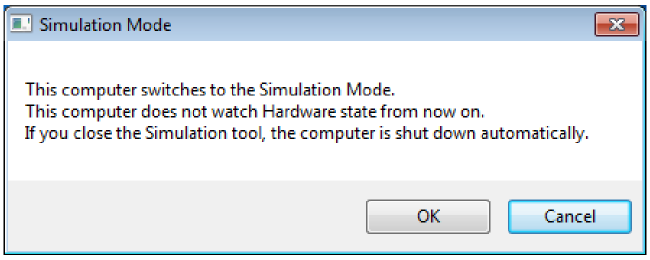
2. At the command prompt, type the following, and then press Enter.
   \%SystemDrive%\"Program Files"\HFWRAS\sim\simrasstart

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



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

4. You will receive the following message to notify 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 Tool** window appears. After this point on, this equipment runs in simulation mode. Monitoring hardware failure is now disabled.

NOTE

The simulation function performs the following actions while running in simulation mode.

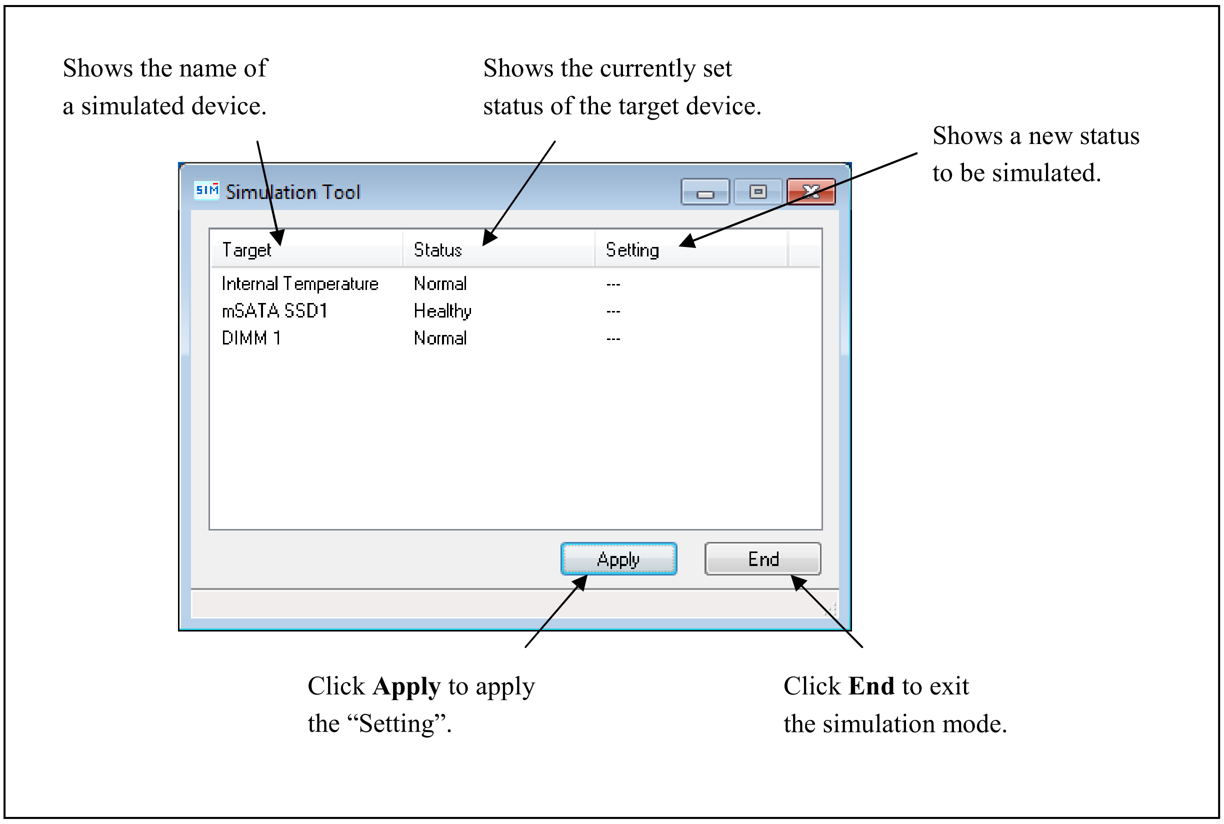
- Every 10 seconds, the Windows® Exclamation sound is played twice. (Only when speakers are connected)

(3) Using the Simulation Tool window

When the RAS software transitions to the simulation mode, the **Simulation Tool** window appears as shown in the following figure.

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

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


Shows the name of a simulated device.
Shows the currently set status of the target device.
Shows a new status to be simulated.
Simulation Tool
Apply
End
Click **Apply** to apply the “Setting”.
Click **End** to exit the simulation mode.

Target	Status	Setting
Internal Temperature	Normal	---
mSATA SSD1	Healthy	---
DIMM 1	Normal	---

Figure 8-3 Description of Each Part in the Simulation Tool Window

● Target

Shows the name of each simulated hardware device.

Category	Target
Temperature condition inside the chassis	Internal Temperature
Storage condition	mSATA SSD1
Memory condition	DIMM 1

- Status

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

Target	Status
Internal Temperature	Normal, Error
mSATA SSD1	Healthy, SMART Detected
DIMM 1	Normal, Error

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

- Setting

  Shows a new status to be simulated for each target hardware device.
  If no status to be simulated is set, “---” is displayed. (After the **Simulation Tool** window starts, “---” is displayed for all hardware devices.)

- Apply button

  If you click this button, all “Setting” are applied to the “Status” of hardware devices.
  The monitoring function of the RAS software detects change in the “Status” of the hardware devices and notifies the change through various interfaces.

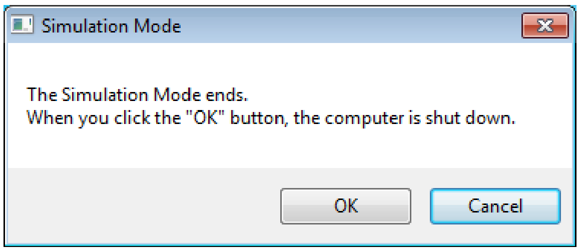
NOTE

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

• Temperature condition inside the chassis: About 15 seconds later
• Storage condition: About 5 seconds later
• Memory condition: About 10 seconds later

- End button

  If you click this button, shutdown is executed to exit the simulation mode. Before shutdown is executed, the following message is displayed. Save the data, for example, and then click **OK**. If you click **Cancel**, the **Simulation Tool** window does not exit.

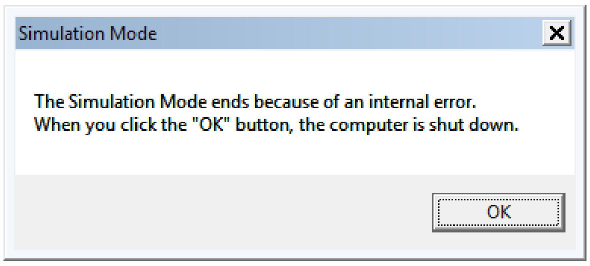


Simulation Mode

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

NOTE

Even if the **Simulation Tool** window exits due to an internal error or the like, shutdown is executed automatically to exit the simulation mode. Before shutdown is executed, the following message is displayed. Save the data, for example, and 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)

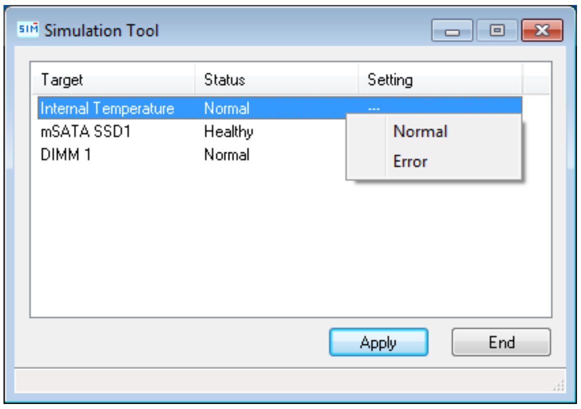
  Click the **Minimize** button at the upper right corner of the **Simulation Tool** window to minimize the **Simulation Tool** window.

- Close button ([×] button)

  Click the **Close** button at the upper right corner of the **Simulation Tool** window to execute shutdown and exit the simulation mode. The behavior after this button is clicked is the same as when you click **End**.

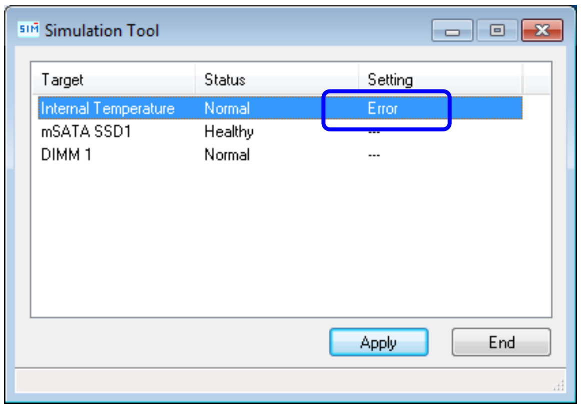
The following procedure shows how to simulate a hardware status using the **Simulation Tool** window.

1. Right-click a hardware item you want to simulate. A popup menu is displayed. The menu lists the statuses you can select based on the current hardware status.



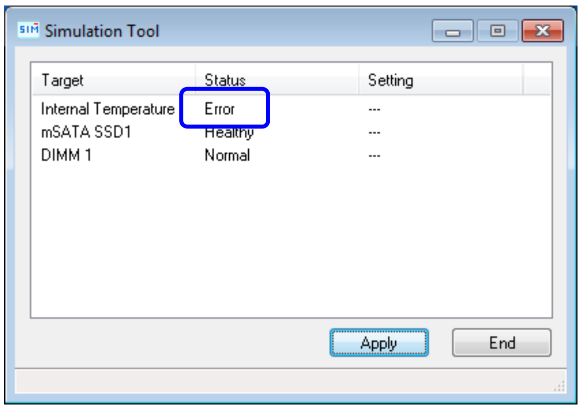
Target	Status	Setting
Internal Temperature	Normal	---
mSATA SSD1	Healthy	
DIMM 1	Normal	

2. If you select a status you want to simulate on the popup menu, the selected status is displayed in “Setting”.



Target	Status	Setting
Internal Temperature	Normal	Error
mSATA SSD1	Healthy	---
DIMM 1	Normal	---

3. To apply the status displayed in “Setting” to the hardware status, click **Apply**. As a result, the “Status” in the **Simulation Tool** window is updated.



Target	Status	Setting
Internal Temperature	Error	---
mSATA SSD1	Healthy	---
DIMM 1	Normal	---

NOTE

If the “Setting” is not selected (that is, “---” is shown) or is the same as the “Status” when you click **Apply**, the current “Status” does not change.

The following shows a list of items on the popup menu that are displayed when you right-click each hardware item.
On the popup menu, the current status and the statuses that the current status can transition to are displayed. Note that if the current status is the only status that can be displayed on the menu, “None” is displayed shaded on the menu.

● Internal Temperature

No.	Current status	Statuses in the popup menu	Note
1	Normal	Normal, Error	
2	Error		

● Storage

No.	Current status	Statuses in the popup menu	Note
1	Healthy	Healthy, SMART Detected	
2	SMART Detected	Healthy, SMART Detected	(*)

(*) Transition from “SMART Detected” to “Healthy” means that the target storage is replaced with a new one and that the new storage is connected.

● Memory

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

(4) Exiting the simulation mode

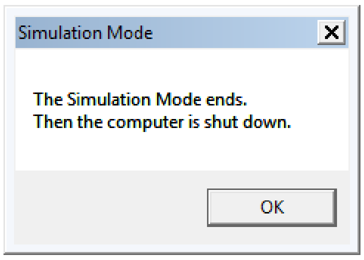
To exit the simulation mode, you need to restart this equipment.
For this reason, this equipment is automatically shut down in order to exit the simulation mode.

There is no limitation for how the equipment can restart in order to exit the simulation mode. In the same ways as normal mode, this equipment shutdown in the following cases (reasons). Restart the equipment after shutdown.

・Shutdown is executed on the **Start** menu.
・A system shutdown API function such as the BSSysshut and ExitWindowsEx functions is executed.
・The equipment is automatically shut down because abnormal temperature inside the chassis.
・Ctrl+Alt+Delete is pressed and using the power button at the lower right corner of the screen, the equipment is shut down.
・The **End** button or the **Close** button [×] in the **Simulation Tool** window is clicked.
・The **Simulation Tool** window is terminated with an error.

NOTE

• As explained above, the simulation mode exits when the equipment restarts after automatic shutdown caused by simulated fan failure or abnormal temperature inside the chassis. To start the simulation mode again, execute the simulation mode start command.
When you execute shutdown or logoff, the following message is displayed to notify that the simulation mode will exit and that the system will shut down.


Simulation Mode

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

• When a remote connection is used, shutdown process can be delayed when shutdown is executed due to a reason listed above.

8.1.3 Precautions when you use the Simulation Tool window

(1) When the new status to be simulated is finalized
From when you select a new status to be simulated from a popup menu on the **Simulation Tool** window until when you click **Apply**, the new status to be simulated is not finalized and you can change it. The actual status used for simulation is the one shown in "Setting" in the **Simulation Tool** window when you click **Apply**.

8.1.4 Event log entries

In order to clearly show which log entries for hardware failure originate from the simulation function, this function records event log entries listed in the following table. Note that a log entry with Event ID 252 is recorded at the timing when you click **Apply** in the **Simulation Tool** window. This log entry is recorded even when all "Settings" are not specified.

Table 8-1 Event Log Entries Recorded by This Function

Event ID	Source	Type	Category	Description
250	HFWSIM_SYS	Information	HFWSIM	The Simulation Mode started.
251	HFWSIM_SYS	Information	HFWSIM	The Simulation Mode ended.
252	HFWSIM_SYS	Information	HFWSIM	The following conditions were set at the Simulation Mode. Internal Temperature: %1 mSATA SSD1: %2 DIMM 1: %3

The %x above denotes the corresponding setting in the **Simulation Tool** window.

8.1.5 Remote notification

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

NOTE

The contents of the hardware status that can be acquired and a trap notification that can be notified are the same as in the normal mode. For details about the objects of the extended MIB for HF-W, see "4.4 Remote Notification".