

# **ExpressCluster for Linux Ver3.1**

Internet Server Agent

2005.12.28  
2<sup>nd</sup> Edition



Revision History

Edition	Revision Date	Revision Page	Description
1	2005/05/30		The 1 <sup>st</sup> Edition was created.
2	2005/12/28	<p>6</p> <p>6</p> <p>8</p> <p>12</p> <p>20</p> <p>42 -</p> <p>48 -</p>	<p>Description is added and corrected to below by the agent correspondence for IA64 servers.</p> <p>The hardware requirement table of the Internet Server Agent is corrected.</p> <p>The installation procedure of the Internet Server Agent for IA64 servers is added.</p> <p>The description about the latest version of a script templates is added.</p> <p>The folder name uninstalled by the Windows version and trekking tool is corrected.(It united with the information on the newest trekking tool currently opened to Web.)</p> <p>Description is added to [Important].</p> <p>Bitmap is corrected.</p> <p>The description about [Start Monitor Wait Time] and [Re-active Threshold] is added.</p>

This construction guide describes “NEC ExpressCluster Internet Server Agent for Linux R3.0-1”.

ExpressCluster® is a registered trademark of NEC Corporation.

Linux is a trademark or registered trademark of Linus Torvalds in the United States and/or other countries.

Other system names, company names, and product names are trademarks or registered trademarks of their respective companies.

The latest information on system confirmation, system configuration guide, update, and tracking tool is provided in the following URL.  
Please obtain the latest version before configuring the system.

Usage on the NEC Internet:

<http://soreike.wsd.mt.nec.co.jp/>

Usage out of the NEC Internet:

<http://www.ace.comp.nec.co.jp/CLUSTERPRO/>

<b>Part I The Basics .....</b>	<b>6</b>
<b>1 Setting Up the Internet Server Agent.....</b>	<b>6</b>
1.1 Setting Up the Monitoring Module .....	6
1.1.1 Before Installation .....	6
1.1.2 Installing the Internet Server Agent on Linux System .....	6
1.2 Setting Up Script Templates .....	8
1.2.1 Installing Script Templates on Windows .....	8
1.2.2 Installing Script Templates on Linux System .....	10
<b>2 Uninstalling the Internet Server Agent .....</b>	<b>11</b>
2.1 Uninstalling the Monitoring Module.....	11
2.2 Uninstalling Script Templates .....	12
2.2.1 Uninstalling the Script Templates for Windows .....	12
2.2.2 Uninstalling the Script Templates for Linux .....	13
<b>3 License Registration .....</b>	<b>14</b>
3.1 Interactive License Registration.....	14
3.2 License Registration By Specifying A License File.....	15
<b>4 Using Internet Server Agent .....</b>	<b>16</b>
4.1 Setting Monitoring Commands .....	16
4.2 Setting PID Monitoring .....	18
<b>Part II Details .....</b>	<b>19</b>
<b>5 Internet Server Monitoring.....</b>	<b>19</b>
5.1 Monitoring Application.....	19
5.2 Overview of Monitoring .....	20
5.3 Writing Scripts to an EXEC Resource .....	22
5.4 Operation Check .....	23
5.4.1 Checking Behavior of Target Application to Be Monitored .....	23
5.4.1.1 Start up a Group .....	23
5.4.1.2 Stop a Group.....	23
5.4.1.3 Move a Group .....	23
5.4.1.4 Failover a Group .....	23
5.4.2 Checking Monitoring Command Operation .....	24
5.4.2.1 Start Up a Group.....	24
5.4.2.2 Stop a Group.....	24
5.4.2.3 Move a Group .....	24
5.4.2.4 Failover a Group .....	24
<b>6 Information Provided by Monitoring Commands .....</b>	<b>26</b>
6.1 Alert Messages .....	26
6.2 Log Collection in Case of an Error.....	26
<b>7 Internet Server Monitoring Commands .....</b>	<b>27</b>
7.1 List of Internet Server Monitoring Commands .....	27
7.2 Monitoring Chart .....	27

7.3	Interrupting and Resuming Monitoring.....	28
7.4	Command Syntax .....	28
	clp_httpmon.....	29
	clp_smtpmon.....	32
<b>8</b>	<b>Alert Messages .....</b>	<b>35</b>
8.1	Messages produced by clp_httpmon.....	35
8.1.1	Messages Indicating Normal Operation.....	35
8.1.2	Messages Due to Setting Error.....	35
8.1.3	Messages Produced when an Error is Detected in HTTP Daemon Monitoring .....	36
8.1.4	Messages Due to a System Error.....	37
8.2	Messages produced by clp_smtpmon .....	38
8.2.1	Messages Indicating Normal Operation.....	38
8.2.2	Messages Due to Setting Error.....	38
8.2.3	Messages Produced when an Error is Detected in SMTP Daemon Monitoring.....	40
8.2.4	Messages Due to a System Error.....	40
<b>9</b>	<b>Environment Construction using a Trekking Tool .....</b>	<b>41</b>
9.1	Adding an EXEC Resource for the Application to be Monitored .....	42
9.2	Adding an EXEC Resource for a Monitoring Command.....	45
9.3	Setting a Monitor Resource .....	48
<b>10</b>	<b>Script Templates.....</b>	<b>52</b>
10.1	For HTTP Daemon.....	53
10.1.1	start.sh .....	53
10.1.2	stop.sh.....	54
10.2	For HTTP Daemon Monitoring.....	55
10.2.1	start.sh .....	55
10.2.2	stop.sh.....	56
10.3	For SMTP Daemon .....	57
10.3.1	start.sh .....	57
10.3.2	stop.sh.....	58
10.4	SMTP Daemon Monitoring.....	59
10.4.1	start.sh .....	59
10.4.2	stop.sh.....	60

# Part I The Basics

## 1 Setting Up the Internet Server Agent

### 1.1 Setting Up the Monitoring Module

**Note:**

This document assumes that users use the ExpressCluster CD as the installation media. If you use a different kind of media or trial version, refer to the document that came with your media, and replace paths and others according to your environment settings..

#### 1.1.1 Before Installation

Check the followings before installing the Internet Server Agent on a server.

The Internet Server Agent requirements are as follows. Check each item on all servers on which the Internet Server Agent will be installed.

Requirements to use the Internet Server Agent (monitoring module)	
Hardware	IA32 server, x86_64 server, IA64 server
OS	The same as the ExpressCluster server and allowing operation of the internet server to be monitored.
ExpressCluster	ExpressCluster SE for Linux Ver3.1-3 or later ExpressCluster SE for Linux Ver3.1-3 or later ExpressCluster SX for Linux Ver3.1-3 or later ExpressCluster LX for Linux Ver3.1-3 or later
Memory size	5 M byte (per command)
Disk size	1 M byte

#### 1.1.2 Installing the Internet Server Agent on Linux System

Set up the Internet Server Agent after installing the ExpressCluster. Apply the latest update for the ExpressCluster.

Log in as a root user and follow the steps below when installing the Internet Server Agent on Linux system.

- (1) **Insert the CD (ExpressCluster CD) into a CD drive.**
- (2) **Mount the CD.**

```
# mount /dev/cdrom
```

**(3) Move to the following directory:**

```
# cd /mnt/cdrom/Linux/3.0/en/option
```

**(4) Install the Internet Server Agent by using the rpm command.**

For Linux IA32

```
# rpm -i --nodeps expresscls-ismon-3.0-1.i386.rpm
```

For Linux x86\_64

```
# rpm -i --nodeps expresscls-ismon-3.0-1.x86_64.rpm
```

For Linux IA64

```
# rpm -i --nodeps expresscls-ismon-3.0-1.ia64.rpm
```

\*You need to check the rpm file name since it may vary depending on the version of the Agent, etc.

**(5) After installing the Agent, register the license for the Internet Server Agent.**

Register the license by following the procedure in 3 "License Registration".

Supplement) The command description may vary depending on the types of Linux.

Note) If you set up Internet Server Agent while monitoring an application using the Internet Server Agent on the ExpressCluster, processes may fail to finish successfully. When setting up the Internet Server Agent, either stop the failover group which is monitoring internet server, or move to the server in which set up is not performed.

## 1.2 Setting Up Script Templates

Script templates are available for Windows version trekking tool and a Linux version trekking tool. Choose and setup scrip templates according to your requirements.

The latest version of a script templates should come to hand. Please apply with reference to the update procedure document for script templates.

### 1.2.1 Installing Script Templates on Windows

Set up script templates after Windows version trekking tool has been set up. This is because the script templates are installed in a folder having the trekking tool scripts. You cannot install script templates on a terminal where trekking tool is not installed.

When installing the script templates on Windows, log in as a user with administrator's authority.

**(1) Insert the CD (ExpressCluster CD) into a CD drive.**

The Setup Menu window appears automatically. If the window does not automatically start up, execute the menu.exe on the CD drive directly.

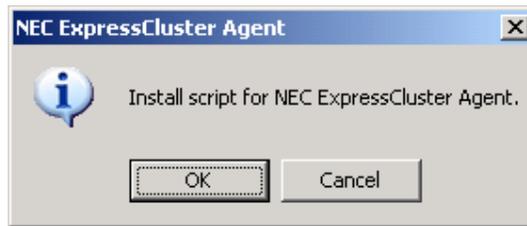


**(2) Click [NEC ExpressCluster for Linux].**



**(3) Click [NEC ExpressCluster Template Scripts for Agent].**

The setup message appears. Click [OK] to install.



Supplement) SETUP.EXE for script installation automatically searches the folder of trekking tool and then performs installation. Specify the folder name directly if the search cannot be executed correctly or if it takes time. Specify the following folder name when you specify a folder name.

Example) When the trekking tool is installed in C:\Program Files\NEC\clptrek (clptrek.htmlb is in the C:\Program Files\NEC\clptrek folder)

A:\SETUP.EXE **"C:\Program Files"**

When specifying the folder name where the script templates are to be installed

```
>CD Q:  
>CD \Linux\3.0\en\script\win  
>SETUP.EXE [installation_folder_of_the_trekking_tool]
```

\*Check the underlined CD drive name by using Windows Explorer, etc. since the name may vary depending on the environment of terminals.

## 1.2.2 Installing Script Templates on Linux System

Set up script templates after Linux version trekking tool has been set up. The script templates will be installed in the directory where the trekking tool scripts exist.

When installing the script templates on Linux, log in as a root user and follow the steps below.

- (1) **Insert the CD (ExpressCluster CD) into a CD drive.**
- (2) **Mount the CD.**

```
# mount /dev/cdrom
```

- (3) **Move to the following directory:**

```
# cd /mnt/cdrom/Linux/3.0/en/script/Linux
```

- (4) **Install the script templates using the rpm command.**

For Linux IA32

```
# rpm -i expresscls-monscript-3.0-1.i386.rpm
```

For Linux x86\_64

```
# rpm -i expresscls-monscript-3.0-1.x86_64.rpm
```

(\*) The rpm file name may vary depending on the version of the Agent.  
The script templates are for all Agents.

Supplement) The command description may vary depending on the type of Linux.

## 2 Uninstalling the Internet Server Agent

### 2.1 Uninstalling the Monitoring Module

When uninstalling a monitoring module, run the following command as a root user.

```
# rpm -e expresscls-ismon
```

Note) If you uninstall the Internet Server Agent while monitoring an application using the Internet Server Agent on the ExpressCluster, processes may fail to finish successfully. When uninstalling the Internet Server Agent, either stop the failover group that is performing internet server monitoring, or move to the server in which uninstallation is not performed.

When uninstalling the ExpressCluster, the Internet Server Agent will be uninstalled as well but the rpm module information will be left as is. Therefore, it is necessary to forcefully delete the package using the rpm command.

Example: rpm -e --force expresscls-ismon

## 2.2 Uninstalling Script Templates

### 2.2.1 Uninstalling the Script Templates for Windows

To uninstall script templates, run the following command as a user with administrator's authority.

```
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\db2v8"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\db2v8-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\http"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\http-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql3.23-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql4.0-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql4.1-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\ nfs"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\ nfs-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle9i"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle9i-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle10g"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle10g-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql7.2-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql7.3-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql8.0-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\powergres1.1"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\powergres1.1-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\samba"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\samba-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\smtp"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\smtp-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\sybase"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\sybase-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\tuxedo8.1"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\tuxedo8.1-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\weblogic8.1"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\weblogic8.1-mon"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\websphere6.0"
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\websphere6.0-mon"
```

(\*) Check the folder name, etc. since the underlined file name may vary depending on the terminal environment.

## 2.2.2 Uninstalling the Script Templates for Linux

To uninstall script templates, run the following command as a root user.

```
# rpm -e expresscls-monscript
```

Uninstalling the Linux Trekking Tool also uninstalls the script templates (for Linux TrekkingTool), however, rpm module information is not removed. Therefore, you have to remove the package forcibly with rpm command before reinstalling it.

i.e. rpm -e --force expresscls-monscript

## 3 License Registration

To use this product, register the license after installing the monitoring module.  
To register the license, log in as a root user from the server which has the monitoring module installed. You have to register a unique license for each server.  
Register the license by the following steps below.  
License registration is not required for script templates.

**Note 1:**

Depending on your ExpressCluster version, you may not be able to register the license properly. In such a case, you have to update your ExpressCluster.

**Note 2:**

You have to register a unique Internet Server Agent license key for each server in a cluster. Otherwise, license keys collide in a cluster, consequently, a license error occurs at the Internet Server Agent startup. See "8 Alert Messages" for details of logged messages.

### 3.1 Interactive License Registration

(1) Run the following command on a server.

```
# cplcncs -i -p ISMON30
```

(2) Enter 1 for the product division.

```
Selection of product division
1. Product
2. Trial
Select product division [1 or 2]...1
```

(3) Enter the product serial number.

```
Enter serial number [Ex. XX000000]... xxnnnnnn
```

Specify the number indicated on the license sheet as the serial number.

(4) Enter the product license key.

```
Enter license key
[Ex. XXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX]...
XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
```

Specify the number indicated on the license sheet as the license key. Enter the license sheet information exactly as indicated because the license key is case sensitive. Note that no "l"s and "O"s in upper case are used in the license key.

After running the command, check that the message “command was success” is indicated and the command is successfully completed. For the other ending messages, refer to a separate ExpressCluster configuration and installation guide, “*Commands.*”

## 3.2 License Registration By Specifying A License File

(1) Run the following command on a server.

```
# clplcncs -i filepath -p ISMON30
```

Specify the path to the distributed license file in the file path of `-i` option.

You can see on the console a message, "command was success." when the command is completed if the registration is successful. See ExpressCluster Construction Guide, "Commands" for details of other completion messages.

A license file may come with the trial version only.

## 4 Using Internet Server Agent

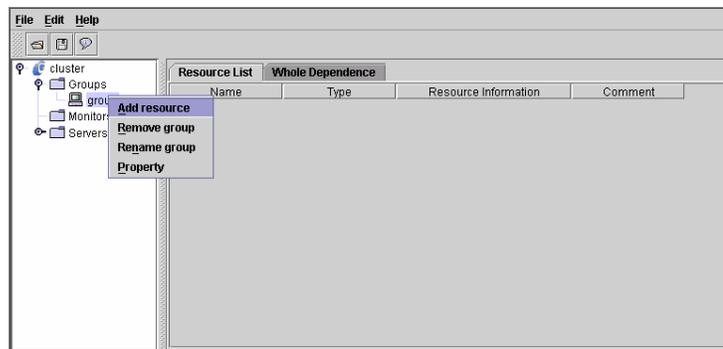
This section briefly explains how to use the Internet Server Agent. For the details, refer to “Part II Details” as necessary.

### 4.1 Setting Monitoring Commands

The description here assumes an internet server is already constructed.

To monitor an internet server, add an EXEC resource for monitoring to a failover group which starts up and stops the internet server. Use trekking tool to add a resource.

Adding an EXEC Resource



Write scripts to start and stop the monitoring command in an EXEC resource. The following is an example of HTTP daemon monitoring commands.

start.sh

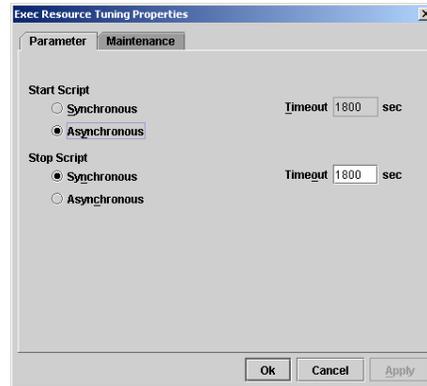
```
clp_httpmon httpwatch
```

stop.sh

```
clp_httpmon httpwatch --stop
```

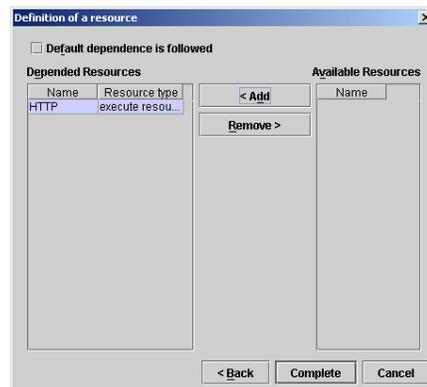
Configure the EXEC resource by choosing "Asynchronous" for start up script and leaving the stop script as "Synchronous."

### EXEC Resource Setting



Establish dependency with the EXEC resource of start-up and deactivation of the internet server system.

### Dependency Set Up

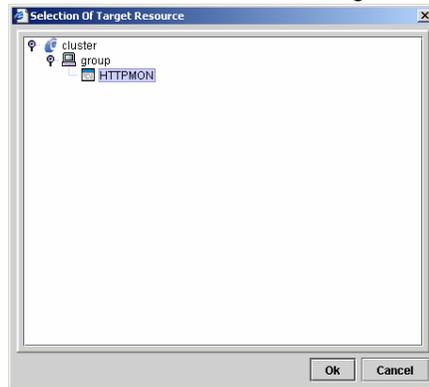


Due to the settings you made, the monitoring command starts up after the Internet server system starts up, and starts monitoring.

## 4.2 Setting PID Monitoring

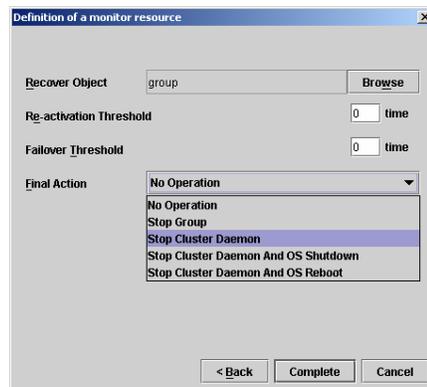
Set up the monitor resource for the EXEC resource which starts up monitoring commands. A monitoring command terminates its process when an error is detected in the internet server. Therefore, it is possible to detect an error in the internet server by looking at the termination of the monitoring command process through PID monitoring of a monitor resource.

Setting an EXEC Resource for monitoring command in monitoring PID



Configure how ExpressCluster behaves when an error is detected in the internet server by the PID monitoring resource. Generally set "0" for "Failover Threshold" and "Stop Cluster Daemon" for "Final Action."

Setting action to be taken when an error is detected



With the settings above, you can start monitoring an internet server by using monitoring commands.

# Part II Details

## 5 Internet Server Monitoring

### 5.1 Monitoring Application

The ExpressCluster Internet Server Agent monitors internet server working under the ExpressCluster environment. The following chart shows the version of the ExpressCluster Internet Server Agent and internet server application that can be monitored.

Internet Server	R3.0-1
HTTP daemon	+
SMTP daemon	+

+:Supported -:Not supported

To monitor internet servers, monitoring commands for each internet server are provided.

Refer to “7 Internet Server Monitoring Commands” for the detailed information of commands.

## 5.2 Overview of Monitoring

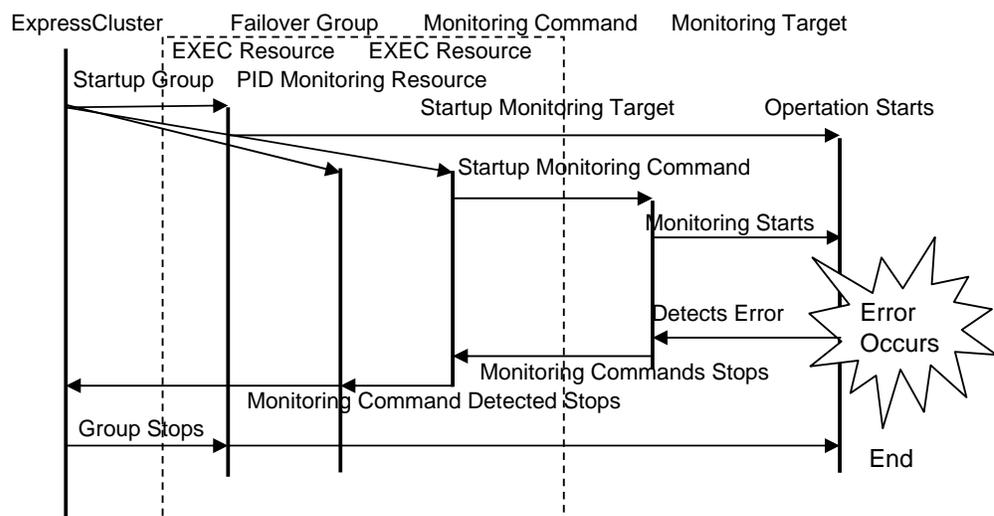
The ExpressCluster Internet Server Agent offers commands described in a script to monitor the operation of the internet server.

If you run these monitoring commands, monitoring is performed with the intervals specified by the parameter until an internet server error is detected. These commands stop when an internet server error is detected.

You use these commands by writing them in a script of an EXEC resource for an ExpressCluster failover group.

Since these commands stop when an internet server error is detected, you can achieve failover or server shutdown by configuring the EXEC resource.

### Overview of Monitoring Operation



A failover group will stop by failover and server shutdown, etc, by PID monitoring resource which detected an error of the EXEC resource which starts the monitoring command.

### [Important]

- (1) These commands can detect an error which does not cause the monitoring target to end abnormally (mainly stalling). Although it cannot detect an abnormal ending of a target application to be monitored, it can indirectly detect a failure through process of monitoring operation of the target (such as internet server access) .
- (2) The purpose of using these commands is to monitor the operation of a target application to be monitored, but not to investigate or diagnose the cause when an error occurs on the application. When an error occurs, you need to use other means, such as application logs, to see details of its cause.
- (3) Whenever it performs monitor processing depending on the application for

monitoring, an access log may be outputted to syslog etc., or a log may be outputted to the local directory of the application for monitoring. About these setup, since it is uncontrollable by this monitoring command, please set up suitably with the application for monitoring. However, when not outputting the log of the application for monitoring, the log at the time of obstacle generating is not outputted, either, but cause investigation may become difficult.

- (4) Use a stop command to stop these monitoring commands. If a process was stopped by using the kill command of Linux, the monitoring command may not be restarted since the monitoring command management information will not be initialized.
- (5) Since these monitoring commands run as a client application of an internet server, configuration to run client application on a server is necessary. Refer to the manuals for each internet server for the details.

## 5.3 Writing Scripts to an EXEC Resource

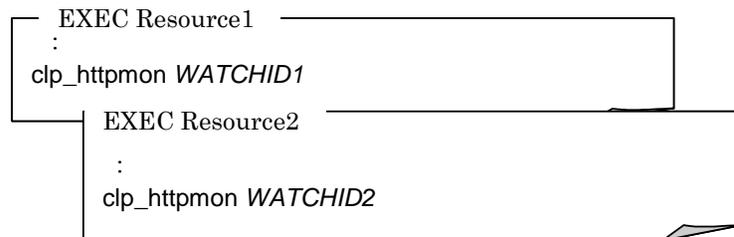
When you write a script on an EXEC resource, the following must be noted.

- Before creating an EXEC resource in which you will write start/end of the Agent, complete the EXEC resource in which start/end of a monitoring target application is written. Then, check if a failover group successfully starts, stops, moves, and fails over. If you create an EXEC resource where the start and end of the Agent are written without checking these issues, it would be difficult to determine the cause when an error was detected in the startup of a failover group by the Agent; If the error really occurred, the environment of the monitoring target application was not correctly set up, or a parameter value of the Agent is not appropriate.
- Configure the dependency relationship of resources so that the EXEC resource for start/stop of the Agent starts up after the EXEC resource in which start/end of an application to be monitored is written starts up. If you make incorrect settings, the Agent may consider the monitoring target application as an error.

These monitoring commands can also be written to monitor multiple internet servers in one failover group.

Example:

Failover Group A



See "10 Script Templates" for scripting examples.

## 5.4 Operation Check

Check on the Web Manager screen to see the failover group is correctly operating by following the steps below.

### 5.4.1 Checking Behavior of Target Application to Be Monitored

Before adding an EXEC resource for monitoring commands, check that a target application to be monitored is operating normally by following the steps below.

#### 5.4.1.1 Start up a Group

Start up a selected failover group.

After selecting the failover group which you wish to start on the tree view on the main screen, display a menu by right clicking, and then select [Start].

#### 5.4.1.2 Stop a Group

Stop the selected failover group.

After selecting the failover group which you wish to stop on the tree view on the main screen, display a menu by right clicking, and then select [Stop].

Check the group starts up and stops on all servers that start up a failover group.

#### 5.4.1.3 Move a Group

Move a failover group among servers.

Start a group and move it to other servers consecutively.

After selecting the failover group which you wish to move on the tree view on the main screen, display a menu by right clicking, and then select [Move].

**Depending on the script, it may take a few minutes to complete moving. See the tree view on the main screen to check completion of moving.**

#### 5.4.1.4 Failover a Group

Perform failover.

Start up a group and shut down its server. Check that the failover group has failed over to its failover destination server.

After selecting a server which you wish to shut down on the tree view on the main screen, display a menu by right clicking, and then select [reboot].

## 5.4.2 Checking Monitoring Command Operation

After making sure that the target application to be monitored operates normally, add an EXEC resource to start up the monitoring commands and a monitor resource which monitors the EXEC resource for starting up monitoring commands. After updating a failover group, check that monitoring commands are running normally by performing the following as the way you checked behavior of a target application to be monitored.

### 5.4.2.1 Start Up a Group

Start up a selected failover group.

After selecting a failover group, which you wish to start up, on the tree view on the main screen, display a menu by right clicking and then select [start].

If a monitoring command displays an error message in the ExpressCluster manager when the group is started, a parameter value of the monitoring command may not be appropriate. If the error message is displayed on a particular server, there may be an error in the environment settings of the target application to be monitored.

### 5.4.2.2 Stop a Group

Stop a selected failover group.

After selecting a failover group which you wish to deactivate on the tree view on the main screen, display a menu by right clicking and then select [Stop].

Check the group starts up and stops on all servers that start up a failover group.

### 5.4.2.3 Move a Group

Move a failover group among servers.

Start a group and move it to other servers consecutively.

After selecting a failover group which you wish to migrate on the tree view on the main screen, display a menu by right clicking and then select [Move].

### 5.4.2.4 Failover a Group

Perform failover.

Start up a group and shut down its server. Check that the failover group has failed over to its failover destination server.

After selecting a server, which you wish to shut down on the tree view on the main screen, display a menu by right clicking and select [Reboot].

**[CAUTION]**

Monitoring may not be performed normally depending on the system environment if you invoke the monitoring command with an extremely small value (e.g. 1) specified to the parameters of monitoring interval and response time. Run sufficient operational check before performing these operations.

## 6 Information Provided by Monitoring Commands

Monitoring commands display a monitoring status on the alert view of the ExpressCluster Web Manager.

### 6.1 Alert Messages

Property	Value
Name	samba
Comment	
Status	Online

Server Name	Status
server1	Online
server2	Online

Resource Name	Status
http	Online
http-mon	Online
ftp	Online

Receive Time	Time	Server Name	Module Name	Event ID	Message
2004/03/01 14:28:01	2004/03/01 14:28:00	server1	httpmon	2	The clp_httpmon is going to watch HTTP daemon. [ID:httpwatch]
2004/03/01 14:26:58	2004/03/01 14:26:58	server1	httpmon	1	The clp_httpmon has started watching HTTP. [ID:httpwatch]
2004/03/01 14:26:36	2004/03/01 14:26:35	server1	irc	1	Monitor pidw start.
2004/03/01 14:26:36	2004/03/01 14:26:35	server1	irc	11	The start processing of a group samba ended.
2004/03/01 14:25:34	2004/03/01 14:25:33	server1	irc	10	The start processing of a group samba started.

Displayed on an alert view of the Manager.

A single line in a message is up to 400 bytes. If information to be shown in a message is longer than 400 bytes, the message will be displayed in two or more lines. Other message may interrupt a message shown in multiple lines depending on timing.

The same information as shown in the alert message is output to the syslog.

See “8 Alert Messages” for the alert message details.

### 6.2 Log Collection in Case of an Error

Error logs of monitoring commands will be produced in the same folder where error logs of the ExpressCluster server are produced. Logs are collected in the same way as ExpressCluster logs are collected. For more information, see “ExpressCluster for Linux Ver3.0 Web Manager” and “ExpressCluster for Linux Ver3.0 Commands”.

# 7 Internet Server Monitoring Commands

## 7.1 List of Internet Server Monitoring Commands

The Internet Server Agent provides internet server monitoring commands to be written in script.

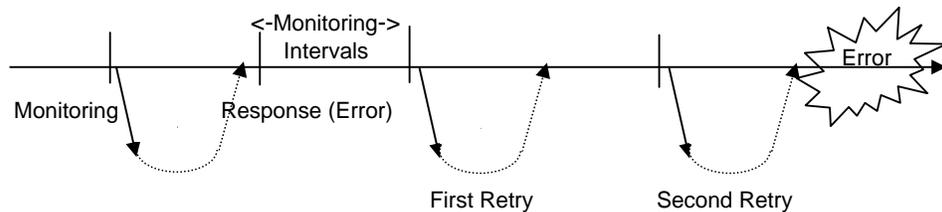
Command	Used to	See page
clp_httpmon	Monitor HTTP daemon	29
clp_smtpmon	Monitor SMTP daemon	32

[Important]

- (1) These commands should be run by a root user. If a internet server monitoring command is run by a user other than a root user, acquiring license and other process may fail. Therefore, the command fails to be run.
- (2) When running these commands, /usr/sbin must be added to a path. Typically /usr/sbin is added to a path.

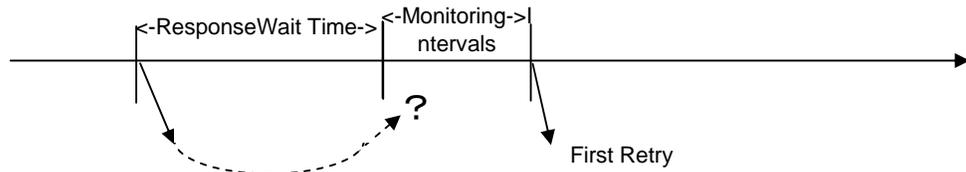
## 7.2 Monitoring Chart

Internet Server Agent detects an error by the following timing.



\*In this case, the retry count is 2 times.

Also, it is considered an error if the Agent does not get a response within the specified wait time, and will perform a retry.



## 7.3 Interrupting and Resuming Monitoring

You can interrupt/resume monitoring by the Internet Server Agent using the following method. While monitoring is suspended, you can perform internet server maintenance.

- (1) Start up the monitoring command to start monitoring.
- (2) To suspend monitoring, run the following command from a server console whenever you wish to suspend.  
"Monitoring Command **watchid** --pause"
- (3) The following message is displayed on ExpressCluster Web Manager to indicate that monitoring has been suspended.  
" The clp\_xxxxmon is going to stop monitoring. [ID:**watchid**]"
- (4) To resume monitoring, run the following command from the server console whenever you wish to resume.  
"Monitoring Command **watchid** --continue"
- (5)The following message is displayed on ExpressCluster Manager to indicate that monitoring was resumed.  
" The clp\_xxxxmon is going to monitor. [ID:**watchid**]"

## 7.4 Command Syntax

This section provides information on command functions. Note the following to understand functions of commands.

- Command line
  - \* Gives an idea what the user actually enters.
  - + [ ] indicates that the enclosed parameter may be skipped.
  - + | indicates one of parameters separated by this symbol should be chosen.
- Description
  - \* Gives information what is performed.
- Parameter
  - \* Explanation about parameters shown in the command line.
- Supplementary information
  - Gives supplementary information such as details on parameter settings
- Monitoring method
  - \* An explanation on how to monitor
- Note
  - Gives information on what should be noted when using the command.
- Command usage example
  - An example of scripting when using the command

<b>clp_httpmon</b>	Monitors the HTTP daemon
--------------------	--------------------------

Command line

Start

Monitoring

*Identifier* [-s *Server Name*] [-n *Port Number*]  
 clp\_httpmon [-i *Monitoring Interval*] [-c *Retry count*] [-r *Response Wait Time*]  
 [-v *HTTP Version*] [-h *Request-URL*]

Stop

Monitoring

*Identifier* --stop [*Stop Wait Time*]

Interrupt

Monitoring

*Identifier* --pause

Resume

Monitoring

*Identifier* --continue

Display

Information

*Identifier* --disp

Delete

Information

*Identifier* --del

Description

This monitoring command stops when an HTTP daemon error is detected.  
 The command also stops, interrupts and resumes monitoring.  
 Use a root privileged console to specify  
 --pause/--continue/--disp/--del.

Option

*Identifier* Specifies an identifier to uniquely identify the monitoring command.  
 You must set an identifier.

-s *Server Name* Specifies the name of an Internet server to be monitored.  
 Default value: localhost

-n *Port Number* Specifies a port number to connect to the Internet server.  
 Default value: 80

-i *Monitoring Interval* Specifies the Internet server monitoring interval in seconds (between 1 and 10000).  
 Default value: 60

-c *Retry count* Specifies how many retries will be made after a failure is detected when monitoring the Internet server (between 1 and 10000).  
 Default value: 3

-r *Response Wait Time* Specifies a response wait time (between 1 and 10000) in Internet server monitoring in seconds.  
 Default value: 10

<code>-v HTTP Version</code>	Specifies the HTTP version (1.0 or 1.1). Default value: 1.1
<code>-h Request-URI</code>	Specifies Request-URI (ex. /index.html). Default Value /.
<code>--stop</code>	Stops the monitoring command
<i>Stop Wait Time</i>	Specifies time to wait for the monitoring command to stop normally (between 1 and 10000) . Default value: 10
<code>--pause</code>	Temporary interrupts monitoring.
<code>--continue</code>	Resumes monitoring.
<code>--disp</code>	Displays a process ID (pid) of the monitoring command identifier managed by the monitoring command.
<code>--del</code>	Do not use this in ordinary use. (See note 1). Deletes information about the monitoring command identifier managed by the monitoring command. Do not use this in ordinary use. (See note 1).

Remarks

About identifier  
You need to specify an identifier that is unique on the system to control the monitoring command. You cannot use an identifier that is already used by a monitoring command for another monitoring command. Specify an identifier using alphanumeric characters. An identifier should be up to 255 bytes. An identifier is case sensitive.

About -s -h parameter  
A character string can be specified up to 255 bytes with the parameter noted above. The actual effective length varies by each parameter. However, with this monitoring command, no verification of the effecting length is performed. The effective length complies with the WWW service and the specification of the OS.

Monitoring method

This monitoring command monitors the following:  
By connecting to the HTTP daemon on the Internet server and issuing a HEAD request, the command determines the following results as an error.  
When performing monitoring by not specifying an -h parameter and:  
(1) No response to the HTTP daemon connection or the issued HEAD request within a response wait time (that is the -r parameter value)  
(2) an error is notified during the connection with HTTP daemon.  
(3) the response message to the issuing of a HEAD request is not started with "/HTTP"

If perform monitoring by specifying a URI other than "/" for the -h parameter and:  
(1) response to the connection to a HTTP daemon or the issuing

of a HEAD request is not received within the wait time (-r parameter value).

- (2) an error is notified during the connection with a HTTP daemon.
- (3) an error is noted in the response to the HEAD request (there are following two patterns) and
  - the response message is not started with "/HTTP"
  - the status code is within 400s and 500s.

Furthermore, the monitoring cannot be performed if the user authentication is set on the URI which specifies an -h parameter.

Note 1

Management information is not initialized properly when the monitoring command process is stopped by a kill command during monitoring. Because of this, a monitoring command with the same identifier may fail to start. If this happens, run the "clp\_httpmon identifier --disp" command and you will see the process ID corresponding to the specified identifier. Use the ps command to check the executable file of the process ID and if the process is other than monitoring command, remove management information by using the "clp\_httpmon identifier --del" command. Never run the "clp\_httpmon identifier --del" command by specifying an identifier which is operating normally; otherwise a monitoring command malfunctions.

Example of [start.sh]  
Command clp\_httpmon httpwatch  
Usage  
[stop.sh]  
clp\_httpmon httpwatch --stop

This command starts up from an EXEC resource.

<b>clp_smtpmon</b>	Monitors the SMTP daemon
--------------------	--------------------------

Command line

Start

Monitoring *Identifier* [-s *Server Name*] [-n *Port Number*]

clp\_smtpmon [-i *Monitoring Interval*] [-c *Retry count*] [*Response Wait Time*]

Stop

Monitoring *Identifier* --stop [*Stop Wait Time*]

clp\_smtpmon

Interrupt

Monitoring *Identifier* --pause

clp\_smtpmon

Resume

Monitoring *Identifier* --continue

clp\_smtpmon

Display

Information *Identifier* --disp

clp\_smtpmon

Delete

Information *Identifier* --del

clp\_smtpmon

Description This monitoring command stops when an SMTP daemon error is detected.  
The command also stops, interrupts and resumes monitoring.  
Use a root privileged console to specify --pause/--continue/--disp/--del.

Option	<i>Identifier</i>	Specifies an identifier to uniquely identify the monitoring command. You must set an identifier.
	-s <i>Server Name</i>	Specifies the name of an Internet server to be monitored. Default value: localhost
	-n <i>Port Number</i>	Specifies a port number to connect to the Internet server. Default value: 25
	-i <i>Monitoring Interval</i>	Specifies the Internet server monitoring interval in seconds (between 1 and 10000). Default value: 60
	-c <i>Retry count</i>	Specifies how many retries will be made after a failure is detected when monitoring the Internet server (between 1 and 10000). Default value: 3
	-r <i>Response Wait Time</i>	Specifies a response wait time (between 1 and 10000) in Internet server monitoring in seconds. Default value: 10
	--stop	Stops the monitoring command

<i>Stop Wait Time</i>	Specifies time to wait for the monitoring command to stop normally (between 1 and 10000) y. Default value: 10
<code>--pause</code>	Temporary interrupts monitoring.
<code>--continue</code>	Resumes monitoring.
<code>--disp</code>	Displays a process ID (pid) of the monitoring command identifier managed by the monitoring command.
<code>--del</code>	Do not use this in ordinary use. (See note 2). Deletes information about the monitoring command identifier managed by the monitoring command. Do not use this in ordinary use. (See note 2).

**Remarks**      **About identifier**  
 You need to specify an identifier that is unique on the system to control the monitoring command. You cannot use an identifier that is already used by a monitoring command for another monitoring command. Specify an identifier using alphanumeric characters. An identifier should be up to 255 bytes. An identifier is case sensitive.

**About -s parameter**  
 A character string can be specified up to 255 bytes with an -s parameter. The actual effective length varies by each parameter. However, with this monitoring command, no verification of the effecting length is performed. The effective length complies with the specification of the OS.

**Monitoring method**      This monitoring command monitors the following:  
 By connecting to the SMTP daemon on the Internet server and issuing the NOOP command, the command determines the following results as an error.

- (1) No response to the SMTP daemon connection or the issued NOOP command within a response wait time (that is the -r parameter value)
- (2) An error reporting in the response to the SMTP daemon or issued NOOP command.

**Note 1**      If a state of that the load average exceeds the RefuseLA value set in the sendmail.def file for a certain period of time, the Agent may consider this as an error and perform failover.

**Note 2**      Management information is not initialized properly when the monitoring command process is stopped by a kill command during monitoring. Because of this, a monitoring command with the same identifier may fail to start. If this happens, run the "clp\_smtpmon identifier --disp" command and you will see the process ID corresponding to the specified identifier. Use the ps command to check the executable file of the process ID and if the process is other than monitoring command, remove management information by using the "clp\_smtpmon identifier --del" command.

Never run the “clp\_smtpon identifier --del” command by specifying an identifier which is operating normally, otherwise a monitoring command malfunctions.

Example of [start.sh]  
Command clp\_smtpon smtpwatch  
Usage  
[stop.sh]  
clp\_smtpon smtpwatch --stop

This command starts up from an EXEC resource.

## 8 Alert Messages

### 8.1 Messages produced by clp\_httpmon

#### 8.1.1 Messages Indicating Normal Operation

ID	Message	Explanation	Supplement
1	The clp_httpmon has started watching HTTP. [ID:xx]	clp_httpmon has started.	-
2	The clp_httpmon is going to watch HTTP daemon. [ID:xx]	clp_httpmon has started to monitor HTTP daemon.	-
3	The clp_httpmon will stop watching HTTP daemon. [ID:xx]	clp_httpmon stops.	-
7	The clp_httpmon is going to stop monitoring. [ID:xx]	Clp_httpmon monitoring is suspended.	This message is displayed when monitoring is suspended by using the --pause command.
8	The clp_httpmon is going to monitor. [ID:xx]	Clp_httpmon monitoring is restarted.	This message is displayed when monitoring is restarted by using the --continue command.
52	Trial Period of clp_httpmon is till nn/nn/nn (mm/dd/yyyy).	Running with license of trial version.	-

#### 8.1.2 Messages Due to Setting Error

ID	Message	Explanation	Supplement
11	Invalid parameter in the clp_httpmon. [ID:xx]	The format for parameter value of clp_httpmon is invalid.	Check the parameter value of the monitoring command.
13	The monitor id is not specified at the clp_httpmon command.	Identifier is not specified in clp_httpmon.	Check the parameter value of the monitoring command.
14	The specified monitor id is already under use in other processes. [ID:xx]	clp_httpmon cannot be restarted because the same identifier is already running.	Check the parameter value of the monitoring command.
15	The clp_httpmon has not performed end processing. [ID:xx]	End process could not be performed with --stop parameter.	-
51	The license of clp_httpmon is not registered.	The license is not registered	Register the license.
53	The license of trial expired by nn/nn/nn (mm/dd/yyyy).	The license of trial version has expired.	-
55	The license of trial is valid from nn/nn/nn (mm/dd/yyyy).	The start day of the trial license is yet to come.	-
56	The registration license overlaps.	The license key you have registered overlaps with other key.	Register a different license key for each server.

ID	Message	Explanation	Supplement
72	The clp_httpmon has detected network error(connect error[nn]). [ID:xx]	Failure was detected during the connection process with HTTP server. The error code is the error number of socket system call.	IP address and port number may be invalid. In addition, httpd service may have stopped. Check the operating environment.
81	Failed to execute name resolution(error[nn]). [ID:xx]	Failed to obtain IP address. The error code is the error number of socket system call.	The server name specified to -s parameter may be incorrect.
83	The clp_httpmon has detected an error in HTTP(invalid status code[nn]). [ID:xx]	The status code from HTTP server is invalid.	The URI specified to -h parameter may be invalid. In addition, there may not be a permission to connect from local server to the monitoring target URI. Check the operating environment.

### 8.1.3 Messages Produced when an Error is Detected in HTTP Daemon Monitoring

ID	Message	Explanation	Supplement
6	The clp_httpmon will terminate. [ID:xx]	Terminating because a failure is detected.	Take action according to the message displayed immediately before.
72	The clp_httpmon has detected network error(connect error[nn]). [ID:xx]	Failure was detected during the connection process with HTTP server. The error code is the error number of socket system call.	Take an action based on the error code.
73	The clp_httpmon has detected network error(send error[nn]). [ID:xx]	Failure was detected during the data sending process to HTTP server. The error code is the error number of socket system call.	Take an action based on the error code.
74	The clp_httpmon has detected network error(select error[nn]). [ID:xx]	Failure was detected during the response wait process from HTTP server. The error code is the error number of socket system call.	Take an action based on the error code.
75	The clp_httpmon has detected network error(rcv error[nn]). [ID:xx]	Failure was detected during the data receiving process to HTTP server. The error code is the error number of socket system call.	Take an action based on the error code.
82	The clp_httpmon has detected an error in HTTP(invalid response). [ID:xx]	The response from HTTP server is invalid.	Take an action based on the error code.
83	The clp_httpmon has detected an error in HTTP(invalid status code[nn]). [ID:xx]	The status code from HTTP server is invalid.	Take an action based on the error code.

## 8.1.4 Messages Due to a System Error

ID	Message	Explanation	Supplement
42	The clp_httpmon has detected system error (xxx nn). [ID:xx]	A Linux system error occurred. xxx indicates a function name and nn indicates error code.	Check the status of the system based on error codes
54	Failed to check license of the clp_httpmon.	Verifying the license information has failed.	The license management module of ExpressCluster may be old. Check the module update.
71	The clp_httpmon has detected network error(socket error [nn]). [ID:xx]	Failed to create socket. The error code is the error number of socket system call.	Take an action based on the error code.
81	Failed to execute name resolution(error[nn]). [ID:xx]	Failed to obtain the IP address. The error code is the error number of socket system call.	Take an action based on the error code.

## 8.2 Messages produced by clp\_smtpmon

### 8.2.1 Messages Indicating Normal Operation

ID	Message	Explanation	Supplement
1	The clp_smtpmon has started watching SMTP. [ID:xx]	clp_smtpmon has started.	-
2	The clp_smtpmon is going to watch SMTP daemon. [ID:xx]	clp_httpmon has started to monitor SMTP daemon.	-
3	The clp_smtpmon will stop watching SMTP daemon. [ID:xx]	clp_smtpmon stops.	-
7	The clp_smtpmon is going to stop monitoring. [ID:xx]	clp_smtpmon monitoring is suspended.	The message is displayed when monitoring is suspended by using the --pause command.
8	The clp_smtpmon is going to monitor. [ID:xx]	clp_smtpmon monitor is restarted.	The message is displayed when monitoring is restarted by using the --continue command.
52	Trial Period of clp_smtpmon is till nn/nn/nn (mm/dd/yyyy).	Running with license of trial version.	-

### 8.2.2 Messages Due to Setting Error

ID	Message	Explanation	Supplement
11	Invalid parameter in the clp_smtpmon. [ID:xx]	The parameter value is invalid in its format.	Check the parameter value of the monitoring command.
13	The monitor id is not specified at the clp_smtpmon command.	Identifier is not specified in clp_smtpmon	Check the parameter value of the monitoring command.
14	The specified monitor id is already under use in other processes. [ID:xx]	The identifier cannot be newly started since the same identifier is already running.	Check the parameter value of the monitoring command
15	The clp_smtpmon has not performed end processing. [ID:xx]	End process could not be performed with --stop parameter.	-
51	The license of clp_smtpmon is not registered.	The license is not registered	Register the license.
53	The license of trial expired by nn/nn/nn (mm/dd/yyyy).	The license of trial version has expired.	-
55	The license of trial is valid from nn/nn/nn (mm/dd/yyyy).	The start day of the trial license is yet to come.	-
56	The registration license overlaps.	The license key you have registered overlaps with other key.	Register a different license key for each server.

ID	Message	Explanation	Supplement
72	The clp_smtprun has detected network error(connect error[nn]). [ID:xx]	Failure was detected during the connection process with SMTP server. The error code is the error number of socket system call.	IP address and port number may be invalid. In addition, sendmail service may have stopped. Check the operating environment.
81	Failed to execute name resolution(error[nn]) [ID:xx]	Failed to obtain IP address. The error code is the error number of socket system call.	The server name specified to -s parameter may be incorrect.

## 8.2.3 Messages Produced when an Error is Detected in SMTP Daemon Monitoring

ID	Message	Explanation	Supplement
6	The clp_smtpmon will terminate. [ID:xx]	Terminating because a failure is detected.	Take action according to the message displayed immediately before.
72	The clp_smtpmon has detected network error(connect error[nn]). [ID:xx]	Failure was detected during the connection process with SMTP server. The error code is the error number of socket system call.	Take an action based on the error code.
73	The clp_smtpmon has detected network error(send error[nn]). [ID:xx]	Failure was detected during the data sending process to SMTP server. The error code is the error number of socket system call.	Take an action based on the error code.
74	The clp_smtpmon has detected network error(select error[nn]). [ID:xx]	Failure was detected during the response wait process from SMTP server. The error code is the error number of socket system call.	Take an action based on the error code.
75	The clp_smtpmon has detected network error(recv error[nn]). [ID:xx]	Failure was detected during the data receiving process to SMTP server. The error code is the error number of socket system call.	Take an action based on the error code.
83	The clp_smtpmon has detected an error in SMTP (invalid status code[nn]). [ID:xx]	The response from SMTP server is invalid.	Take an action based on the error code.

## 8.2.4 Messages Due to a System Error

ID	Message	Explanation	Supplement
42	The clp_smtpmon has detected system error (xxx nn). [ID:xx]	A Linux system error occurred. xxx indicates a function name and nn indicates error code.	Check the status of the system based on error codes
54	Failed to check license of the clp_smtpmon.	Verifying the license information has failed.	The license management module of ExpressCluster may be old. Check the module update.
71	The clp_smtpmon has detected network error(socket error[nn]). [ID:xx]	Failed to create socket. The error code is the error number of socket system call.	Take an action based on the error code.
81	Failed to execute name resolution(error[nn]) [ID:xx]	Failed to obtain the IP address. The error code is the error number of socket system call.	Take an action based on the error code.

## 9 Environment Construction using a Trekking Tool

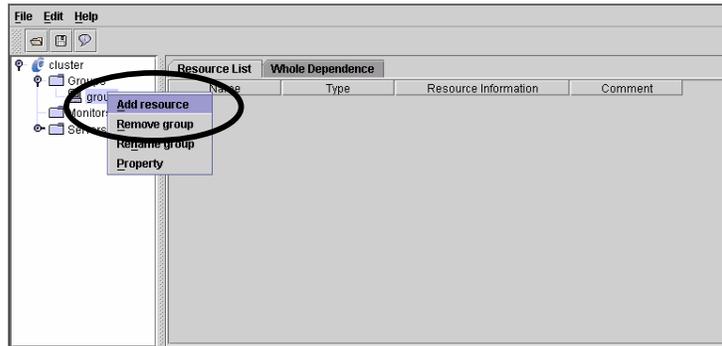
When you create a failover group of an internet server by using trekking tool, follow the steps below.

- 1) Add a failover group for the application to be monitored.
- 2) Add a disk resource or an IP resource
- 3) Add an EXEC resource for starting the application to be monitored.
- 4) Make what you have done take effect in the ExpressCluster server and check that the application to be monitored operates successfully. (See “5.4.1 Checking Behavior of Target Application to Be Monitored”)
- 5) Add an EXEC resource for starting the monitoring command.
- 6) Add a monitor resource to monitor an EXEC resource of the monitoring command.
- 7) Make what you have done take effect in the ExpressCluster server and check that the application to be monitored and the monitoring command operate normally. (See “5.4.2 Checking Monitoring Command Operation”)

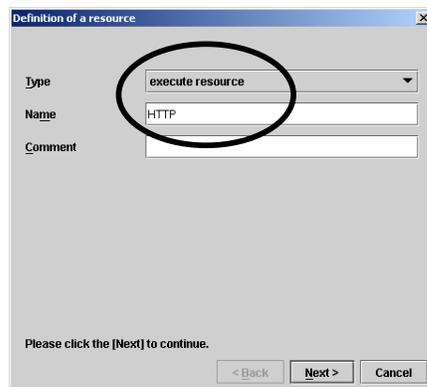
The following provides detailed information of Step 3), 5), and 6).

## 9.1 Adding an EXEC Resource for the Application to be Monitored

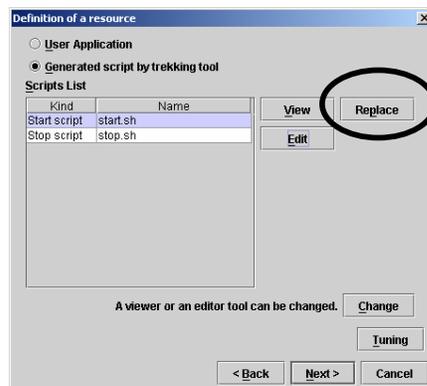
Run trekking tool and perform "Add resource" in the failover group of an application to be monitored.



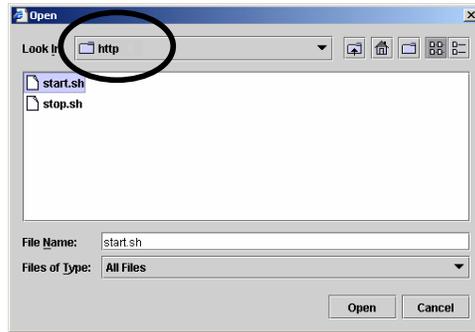
Add an EXEC resource for the monitored application. Select "execute resource" as a resource type.



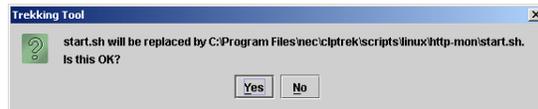
Click the [Replace] button and replace start.sh and stop.sh by script templates of the Agent.



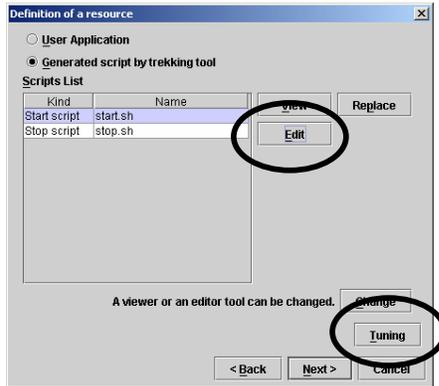
Specify and replace a script of the monitored application. Replace it by a template for monitoring application.



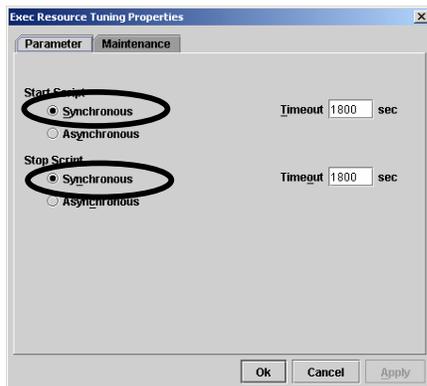
The following dialogue asking for confirmation appears. Check the original file to be replaced and then select "Yes."



If you click the "Edit" button, the editor will open and you can modify the script according to the environment. See "10 Script Templates" for what to be modified.



Click the "Tuning" button and check that both start script and stop script are "Synchronous" (The default value is Synchronous).



Configure items in the following window according to your environment. You do not have to change.

Definition of a resource

Restoration operation when detecting activity failure

Activity Retry Threshold: 0 time

Failover Threshold: 1 time

Final Action: No Operation (Next Resources Are Not Activated)

Restoration operation when detecting deactivity failure

Deactivity Retry Threshold: 0 time

Final Action: Stop Cluster Daemon And OS Shutdown

< Back Next > Cancel

No changes are made in the following window. Check that a disk resource and an IP resource are displayed in the [Depended Resources] box.

Definition of a resource

Default dependence is followed

Depended Resources

Name	Resource type
--	floating ip res...
--	disk resource
--	raw resource
--	VVM disk gro...
--	VVM volume r...
--	has resource

Available Resources

Name
------

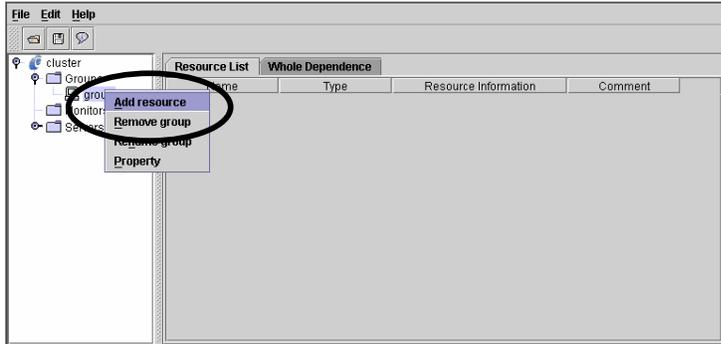
< Add Remove >

< Back Complete Cancel

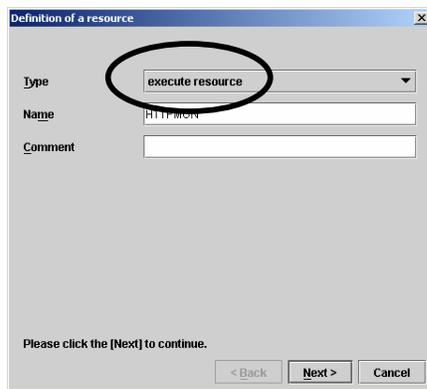
Click on the [Complete] button to create an EXEC resource for the monitored application. When the EXEC resource of the monitored application is added, make it take effect in the ExpressCluster server and check it works successfully.

## 9.2 Adding an EXEC Resource for a Monitoring Command

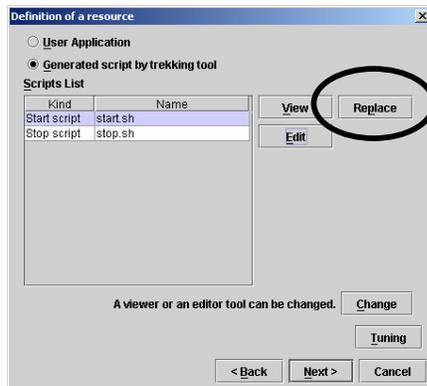
Add an EXEC resource for a monitoring command to a failover group of the monitored application.



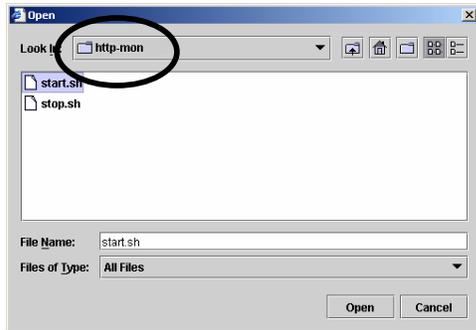
Select "execute resource" as a resource type. Specify a value that is different from the name selected previously for the [Name.]



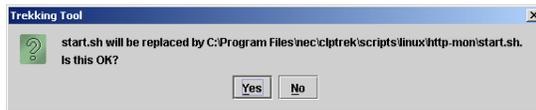
Click the [Replace] button and replace start.sh and stop.sh by script templates of the Agent.



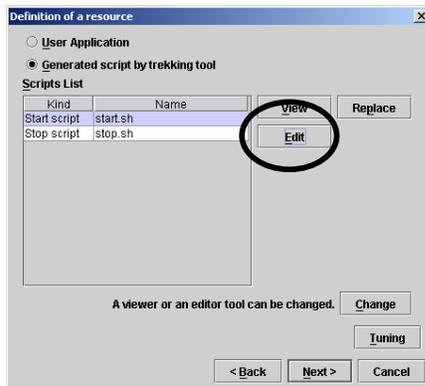
Specify and replace a script for the monitoring command. Replace it by a template for monitoring command.



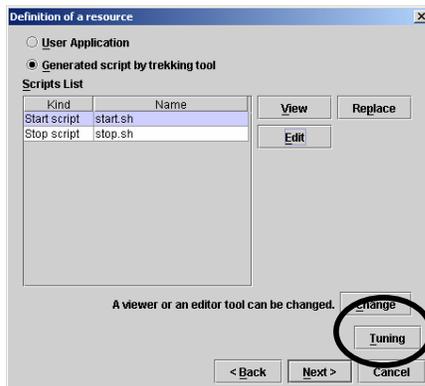
The following dialogue asking for confirmation appears. Check the original file to be replaced and then select "Yes."



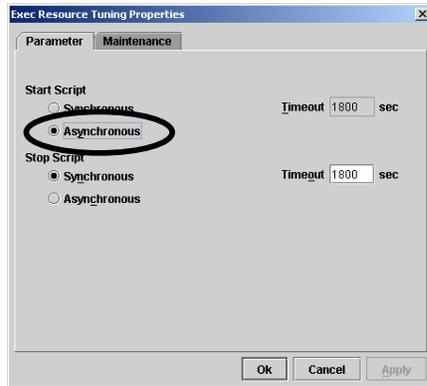
If you click the "Edit" button, the editor will open and you can modify the script according to the environment. See "10 Script Templates" for what to be modified.



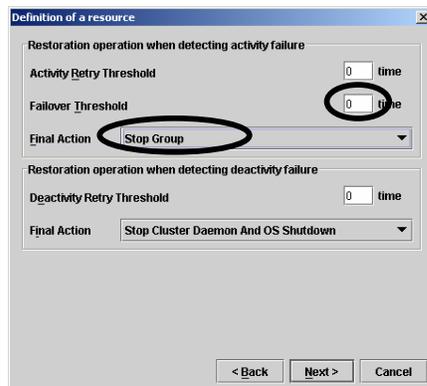
Click the "Tuning" button.



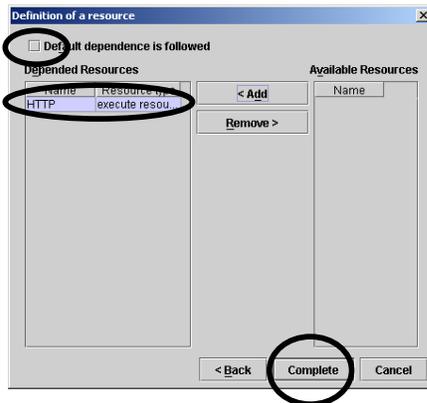
Select [Asynchronous] for the start script.



Set 0 for the [Failover Threshold] of [Restoration operation when detecting activity failure], and "Stop Group" for [Final Action.]



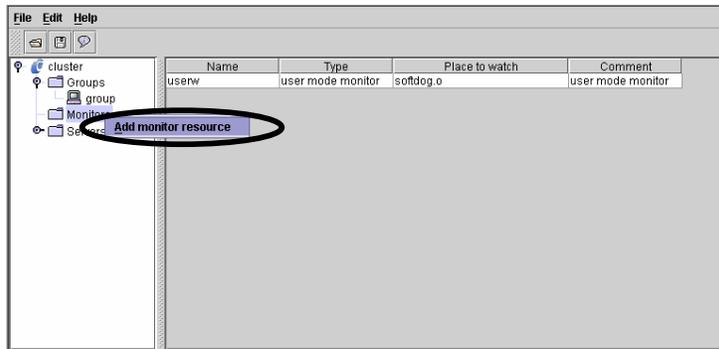
Clear the [Default dependence is followed] check box and add an EXEC resource of the monitored application to the depended resource.



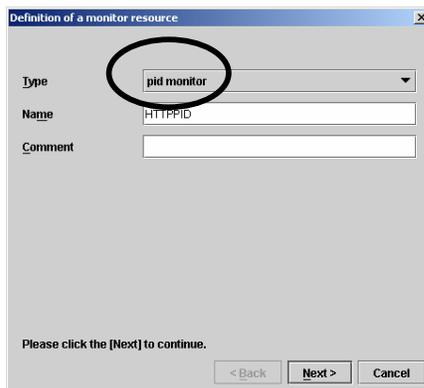
Click the [Complete] button and create an EXEC resource for the monitoring command.

## 9.3 Setting a Monitor Resource

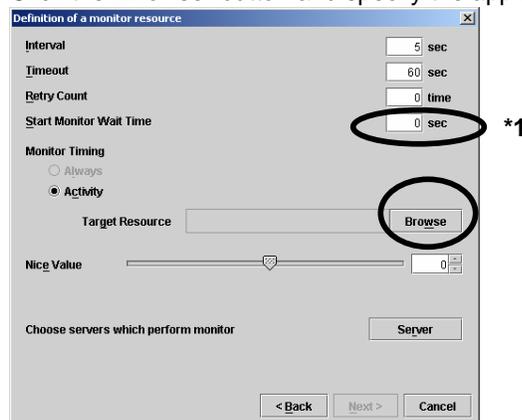
Add a monitor resource.



Select "pid monitor" as a monitoring type.



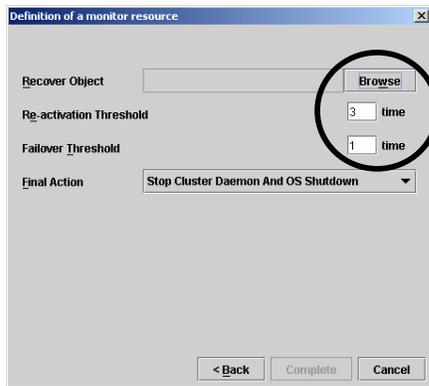
Click the "Browse" button and specify the application target for pid.



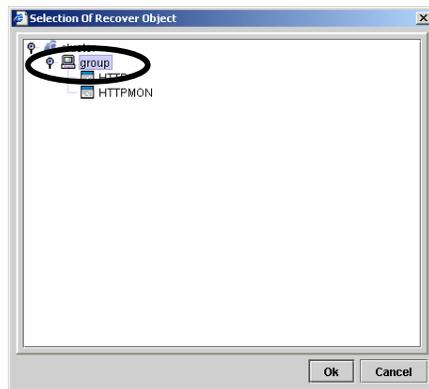
Select the EXEC resource of the monitoring command.



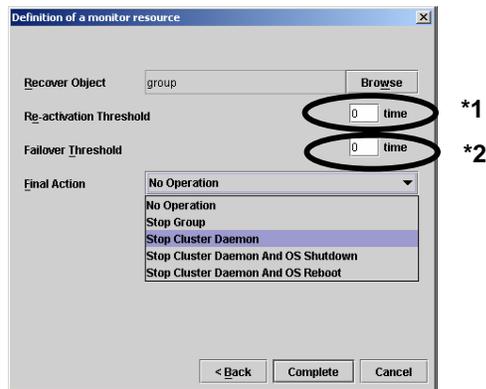
Click the [Browse] button and select a recovery target.



Select a failover group with a monitored application.



Select an action to be taken for [Final Action.] This selection determines the operation of ExpressCluster when the monitoring command detects an error in the monitored application.



**\*1**

When you detect an obstacle, please specify 0 as a [Re-active Threshold] to perform failover immediately. On the contrary, when you detect an obstacle, please specify values other than zero as a [Re-active Threshold] to revitalize a group.

When setting a [Re-active Threshold] as values other than zero, it is necessary to set up the value of [Start Monitor Wait Time] for a long time than the following time.

- Time after Agent starts until Agent is completed more unusually

Time until Agent is completed is influenced by the value of -i parameter (Monitoring Interval), -c parameter (Retry count), and -r parameter (Response Wait Time).

[Example] When a Agent is completed with the stole of a internet server

-i (Monitoring Interval) : 60 (seconds)  
 -c (Retry count) : 3 (Number of times)  
 -r (Response Wait Time) : 10 (seconds)

Near time until Agent is completed

$10(\text{seconds}) + (60(\text{seconds}) + 10(\text{seconds})) * 3(\text{Number of times}) = 220(\text{seconds})$

\*  $\text{Response Wait Time} + (\text{Monitoring Interval} + \text{Response Wait Time}) * \text{Retry count}$

With contents of the error, the time to the end of Agent changes a little.

Please refer to "7.2 Monitoring Chart" about operation of the Agent according to the kind of error.

**\*2**

If you want to perform failover without stopping a server as a cluster at the time of error detection, specify 1 in [Failover Threshold.] In this case, other failover groups continue operation in the server in which an error is detected. When an error is also detected in the failover destination server, failover occurs again and the failover group comes back.

If you want to perform failover by stopping a server as a cluster at the time of an error detection, specify 0 in [Failover Threshold.] In addition, specify "Stop Cluster Daemon", "Stop Cluster Daemon And OS Shutdown", or "Stop Cluster Daemon And OS Reboot" for [Final Action.] In this case, since the server which detected an error stops operating as a cluster,

other failover group terminates or failover occurs.

It is recommended to specify 0 for "Failover Threshold" and "Stop Cluster Daemon" for "Final Action" given recovery work will be done when an error occurs.

See the construction guide of the ExpressCluster server about the details of the final operation.

If a setup is completed, click the [Complete] button to complete creation of monitor resource.

## 10 Script Templates

This product is shipped with script templates for trekking tool. Since the templates may be improved and modified from time to time, make sure to check if there is a newer version of the templates on the ExpressCluster home page. Use the newer one if there is any.

Two templates, `start.sh` and `stop.sh`, are provided for the scripts.

Modify and use them according to the operational environment and other requirements. Where modification is made are indicated in underlined boldface italics in the scripts in the following section.

## 10.1 For HTTP Daemon

### 10.1.1 start.sh

```
#!/bin/sh
#*****
#*          start.sh          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"

if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi

date +"%Y/%m/%d %T"
echo "`/etc/init.d/httpd start`"
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"

if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi

date +"%Y/%m/%d %T"
echo "`/etc/init.d/httpd start`"
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.1.2 stop.sh

```
#!/bin/sh
#*****
#*                stop.sh                *
#*****

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"

if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi

date +"%Y/%m/%d %T"
echo "`/etc/init.d/httpd stop`"
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"

if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi

date +"%Y/%m/%d %T"
echo "`/etc/init.d/httpd stop`"
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.2 For HTTP Daemon Monitoring

### 10.2.1 start.sh

```
#!/bin/sh
#*****
#*          start.sh          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_httpmon start"
    clp_httpmon httpwatch
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_httpmon start"
    clp_httpmon httpwatch
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.2.2 stop.sh

```
#!/bin/sh
#*****
#*                stop.sh                *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_httpmon stop"
    clp_httpmon httpwatch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_httpmon stop"
    clp_httpmon httpwatch --stop
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.3 For SMTP Daemon

### 10.3.1 start.sh

```
#!/bin/sh
#*****
#*          start.sh          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

    date +"%Y/%m/%d %T"
    echo "`/etc/init.d/sendmail start`"
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

    date +"%Y/%m/%d %T"
    echo "`/etc/init.d/sendmail start`"
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.3.2 stop.sh

```
#!/bin/sh
#*****
#*          stop.sh          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

    date +"%Y/%m/%d %T"
    echo "`/etc/init.d/sendmail stop`"
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

    date +"%Y/%m/%d %T"
    echo "`/etc/init.d/sendmail stop`"
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.4 SMTP Daemon Monitoring

### 10.4.1 start.sh

```
#!/bin/sh
#*****
#*          start.sh          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_smtpmon start"
    clp_smtpmon smtpwatch
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_smtpmon start"
    clp_smtpmon smtpwatch
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

## 10.4.2 stop.sh

```
#!/bin/sh
#*****
#*                stop.sh                *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_smtpon stop"
    clp_smtpon smtpwatch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"

    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

    date +"%Y/%m/%d %T"
    echo "clp_smtpon stop"
    clp_smtpon smtpwatch --stop
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi

echo "EXIT"
exit 0
```