ExpressCluster® X Application Server Agent 1.0 *for Linux*

Administrator's Guide

8/31/2007
Fourth Edition

Revision History

Edition	Revised Date	Description
First	2006/09/08	New manual
Second	2006/12/12	EXPRESSCLUSTER logo has been changed.
Third	2007/6/30	Descriptions recommending using monitor resources have been added.
Fourth	2007/08/31	Added information about setup in "Preface".

Disclaimer

Trademark Information

Table of Contents

Preface

If you newly set up Agents as monitor resource with installation of version 1.1.0-1 or later rpm, please refer to the following manuals.

Getting Started Guide
- "Supported distributions and kernel versions"
- "Applications supported by monitoring options"

Reference Guide
- Chapter 6 Monitor resource details

Please refer to this manual in case that
- you set up Agents with installation of version 1.0.X-X rpm.
- you use a shell scripts to control Agents that created with version 1.0.X-X rpm for another cluster system with version 1.1.0-1 or later rpm.

Who Should Use This Guide

The *ExpressCluster X Application Server Agent Administrator's Guide* is intended for system engineers and administrators who want to build, operate, and maintain a cluster system with Application Server Agent. Instructions for setting up a cluster system with ExpressCluster X Appication Server Agent is covered in this guide.

How This Guide is Organized

Chapter 1 **Overview of Application Server Agent**
This chapter provides information on a product overview of the Application Server Agent and how to set it up.

Chapter 2 **Application Server Agent command reference**
This chapter provides detailed information on the script commands.

Chapter 3 **How to check monitor status**
This chapter describes the monitoring method and the messages.

Chapter 4 **Settings for ExpressCluster X Application Server Agent**
This chapter describes how to create cluster configuration data using the Application Server Agent and how to create a cluster.

Appendix A **Script templates**
The script template used in the ExpressCluster is introduced.

Appendix B **Glossary**

Appendix C **Index**

ExpressCluster X Documentation Set

The ExpressCluster X manuals consist of the following four guides. The title and purpose of each guide is described below:

Getting Started Guide
This guide is intended for all users. The guide covers topics such as product overview, system requirements, and known problems.

Installation and Configuration Guide
This guide is intended for system engineers and administrators who want to build, operate, and maintain a cluster system. Instructions for designing, installing, and configuring a cluster system with ExpressCluster are covered in this guide.

Reference Guide
This guide is intended for system administrators. The guide covers topics such as how to operate ExpressCluster, function of each module, maintenance-related information, and troubleshooting. The guide is supplement to the *Installation and Configuration Guide*.

Administrator's Guide (Add-on product)
This guide is intended for system administrators. The detailed information on each product package is described in this guide. There are five guides for each optional product and topics such as product overview, instruction for setting up are covered:

Alert Service Administrator's Guide
Application Server Agent Administrator's Guide
Database Agent Administrator's Guide
File Server Agent Administrator's Guide
Internet Server Agent Administrator's Guide

Conventions

In this guide, **Note**, **Important**, **Related Information** are used as follows:

Note:
Used when the information given is important, but not related to the data loss and damage to the system and machine.

Important:
Used when the information given is necessary to avoid the data loss and damage to the system and machine.

Related Information:
Used to describe the location of the information given at the reference destination.

The following conventions are used in this guide.

Convention	Usage	Example
Bold	Indicates graphical objects, such as fields, list boxes, menu selections, buttons, labels, icons, etc.	In **User Name**, type your name. On the **File** menu, click **Open Database**.
Angled bracket within the command line	Indicates that the value specified inside of the angled bracket can be omitted.	`clpstat –s[-h host_name]`
#	Prompt to indicate that a Linux user has logged on as root user.	`# clpcl -s -a`
Monospace (courier)	Indicates path names, commands, system output (message, prompt, etc), directory, file names, functions and parameters.	`/Linux/1.0/eng/server/`
Monospace bold (courier)	Indicates the value that a user actually enters from a command line.	Enter the following: **`# clpcl -s -a`**
Monospace italic (courier)	Indicates that users should replace italicized part with values that they are actually working with.	`rpm –i expressclsbuilder -<version_number>-<release_number>.i686.rpm`

Contacting NEC

For the latest product information, visit our website below:

http://www.ace.comp.nec.co.jp/CLUSTERPRO/clp/global-link.html

Chapter 1 Overview of Application Server Agent

This chapter provides an overview of ExpressCluster X Application Server Agent and instructions for how to register the license.

This chapter covers:

What is ExpressCluster X Application Server Agent?

When ExpressCluster monitors applications, failing over the application when detecting an error at application startup is possible; however, an application stalling and errors cannot be detected once application is started up.

With ExpressCluster X Application Server Agent, failover can be performed when an application is stalled or an error has occurred as well as at application startup. Thus you will be able to monitor applications on the cluster system.

ExpressCluster X Application Server Agent achieves the monitoring of an application as described above by using monitor resources[1] and monitoring commands.

For details on the function of this product of monitoring applications using monitor resources, see Chapter 6, "Monitor resource details" in the *Reference Guide*.

The remainder of this guide explains the monitoring function using monitoring commands.

Important:

It is recommended to perform monitoring using monitor resources since the WebManager and the – clpstat command enables you to monitor the status and to avoid describing dedicated scripts.

Overview of monitoring with ExpressCluster X Application Server Agent

ExpressCluster X Application Server Agent configures the following three settings to monitor the application server:

1. the exec resource for starting the target monitoring application (exec 1)
2. the exec resource for starting the monitoring command (exec 2)
3. the pid monitor resource that monitors exec 2

[1] Supported from ExpressCluster X version 1.1.0-1.

Mechanism of monitoring

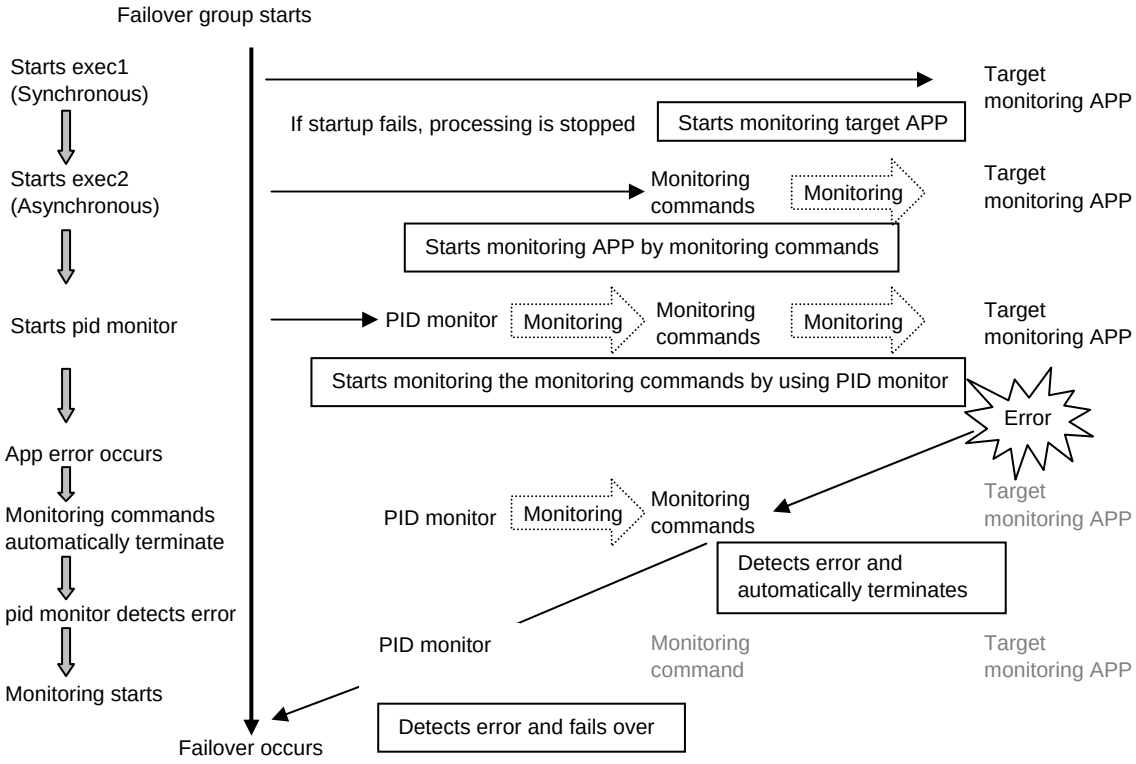
Start the target monitoring application in exec 1. Create an exec resource (exec 2) that differs from this exec 1, and in this exec 2 script, write the monitoring command that monitors the application started in exec 1. Two exec resources are created to distinguish whether an error has occurred at an application startup or after it has been started.

Configure the settings for the pid monitor resource to monitor exec 2. The monitoring commands specified in exec 2 are provided by ExpressCluster X Application Server Agent to monitor applications. These monitoring commands monitor the application server at the intervals specified by the parameter, and if they detect an application server error, they stop their own processes.

If these commands stop when an application server error is detected, configure the settings to execute failover or server shutdown by the pid monitor that monitors them.

Using the mechanism above, you can detect a stalling problem and an error of the target monitoring application that cannot be detected by the ordinary settings and failover and shutdown become possible. The following figure explains this mechanism:

See below for an overview of the application (APP) monitoring using the monitoring command:



Important:

- These commands can detect an error which does not cause the target monitoring application to end abnormally (mainly stalling problem). This is indirectly achieved, by monitoring not the process ending of target monitoring application, but the monitoring operations such as attempting to access to the application server.
- The purpose of using these commands is to monitor the operation of the target monitoring application, and 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 the details of its cause.
- Depending on the target monitoring application, access logs are output to syslog and other places, and logs are output to the local directory of target monitoring application every time you execute the monitoring processes. Since the settings cannot be controlled by these monitoring commands, configure these settings by the target monitoring application, if necessary. However, note that it may become difficult to investigate the cause because logs are not output when an error occurs, if the target monitoring application is set not to output logs.

Run the stop command (**`clp_`*`XXXXXX XXXXXX`*` --stop`**) to stop these monitoring commands. If a process is stopped by using the kill command of Linux, the monitoring command may not be restarted because management information of the monitoring command will not be initialized.

Related Information:
For details, see Note 4 at “Tuxedo monitoring command” on page 14 and other notes for commands in this guide.

Target monitoring application

ExpressCluster X Application Server Agent monitors application server working under the ExpressCluster environment. The following chart shows the versions of ExpressCluster X Application Server Agent and application server applications that can be monitored.

Application Server	ExpressCluster Application Server Agent 1.0-1
Tuxedo 8.1	Yes
WebLogic Server 8.1	Yes
WebSphere Application Server V6.0	Yes

Yes: Supported, No: Not supported

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

Refer to Chapter 2, “Application Server Agent command reference” for more information on commands.

Using the Application Server Agent

In this section, how to use ExpressCluster X Application Server Agent is introduced. The module for the Application Server Agent is installed together with the ExpressCluster X. You only need to register the license to use the Application Server Agent.

System requirements for the Application Server Agent

Check each item on all servers where the Application Server Agent will be installed. The system requirements for the Application Server Agent are as follows:

System Requirements for the Application Server Agent (monitoring module)	
Hardware	IA 32 server
OS	Same system requirements for the ExpressCluster Server. The application server system to be monitored need to be able to function.
ExpressCluster	ExpressCluster X 1.0 or later
Memory space	6 MB (per command)

Obtain the latest update of the Application Server Agent. Refer to the update instructions for how to update the Application Server Agent.

Before using the Application Server Agent

The following tasks need to be completed before registering the license for the Application Server Agent. If the tasks are not completed, refer to the Section II, “Installing and configuring ExpressCluster X” in the *Installation and Configuration Guide* for procedures and complete the tasks.

1. Installation of the ExpressCluster Server and ExpressCluster X Builder
2. Execution of the cluster creation command
3. License registration of the ExpressCluster Server

Upon completing 1 through 3 above, you can start operating the Application Server Agent by following the procedures below:

1. License registration of the Application Server Agent
   Register the license by following the procedure introduced in “License registration for the Application Server Agent” on the next page.

License registration for the Application Server Agent

License registration is necessary to use the Application Server Agent.

To register the license, log on as root user from the server which constitutes a cluster. You have to register a unique license key for each server.

Note:
When your ExpressCluster version does not support ExpressCluster X Application Server Agent, you may not be able to register the license properly. In such a case, you have to update your ExpressCluster.

Registering the license interactively

The explanation below is given using a product version as an example.

Check the following before installing a license:

- Verify that the ExpressCluster X is installed, cluster creation command is executed, and the license for the ExpressCluster X is registered.
- Prepare the license sheet for the Application Server Agent obtained from your sales agent. You will need to enter the value indicated in the license sheet.

1. Run the following command on a server:

   # clplcnsc –i –p ASAG10

2. The following is displayed. Enter 1 for the product version:

```
Software license
   1   Product version
   2   Trial version
Select the license version [1 or 2]…1
```

3. You will be prompted to enter the product serial number. Specify the number indicated in the license sheet:

   Enter serial number [Ex. XXX0000000]… ***xxxnnnnnnn***

4. You will be prompted to enter the product license key. Specify the number indicated in the license sheet.

Note:
Enter the license sheet information exactly as indicated because the license key is case-sensitive. To avoid any confusion with other letters and numbers, “I” and “O” in capital letter are not used in the ExpressCluster license key.

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

xxxxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxx

After running the command, the message “Command succeeded” is displayed in the console to indicate that the command is successfully completed. If other completion messages are displayed, refer to “License management command” in Chapter 4, “ExpressCluster command reference” in the *Reference Guide*.

Registering the license by specifying a license file

For a trial version, you may register the license using a license file in stead of a license sheet. License registration for trial version allows you to register the license for the entire cluster by registering to only one server in a cluster.

- Run the following command on a server:

  # clplcnsc –i *filepath* –p ASAG10

  Assign the file path to the license file specified by –i option.

  After running the command, the message “Command succeeded” is displayed in the console to indicate that the command is successfully completed. If other completion messages are displayed, refer to “License management command” in Chapter 4, “ExpressCluster command reference” in the *Reference Guide*.

  This completes the setup of ExpressCluster X Application Server Agent. How to use the monitoring commands is introduced in the next chapter.

Chapter 2 Application Server Agent command reference

This chapter provides information regarding how to set up and operate ExpressCluster X Application Server Agent. Application server monitoring commands and Application Server Agent commands are also introduced.

This chapter covers:

List of application server monitoring commands

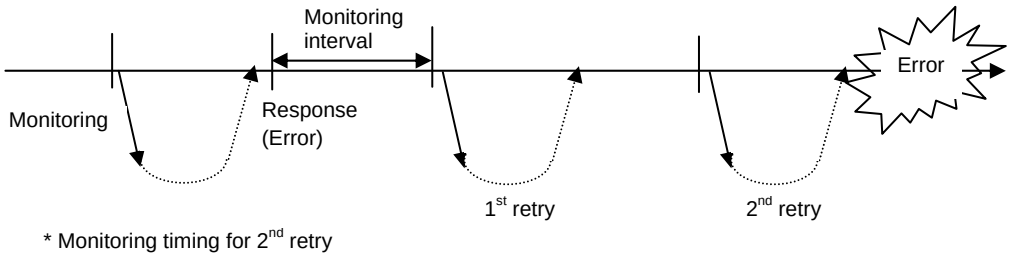
The Application Server Agent provides application server monitoring commands to be written in the script.

Command	Used for	Refer to
clp_tux81mon	Monitors the Tuxedo daemon	Page 22
clp_wls81mon	Monitors the WebLogic daemon	Page 25
clp_was60mon	Monitors WebSphere	Page 29

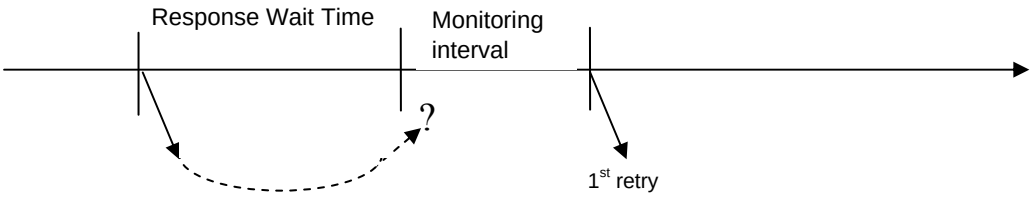
Note:
These commands should be run as root user. If a command is run by a user other than root user, you may not be able to acquire the license information and cannot run the command. When running these commands, /usr/sbin must be added to a path although it is normally added.

Monitoring chart

Application Server Agent detects an error in the following timing:



If the Agent does not get a response within the specified wait time, it is immediately considered as an error.



Writing scripts to exec resource

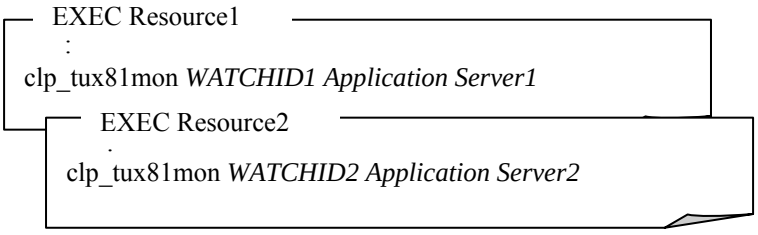
Note the following when you write a script to an exec resource:

- Before creating an exec resource in which you will write start/end of monitoring commands, complete the exec resource in which start/end of a target monitoring 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/end of the monitoring commands are written without checking these issues, it would be difficult to determine the cause of an error is detected by the monitoring commands at failover group startup. For example, if an error has actually occurred, various environments of the target monitoring application were set properly, or a parameter value of the monitoring commands is appropriate, is hard to distinguish.
- Configure the dependency of the resources so that the exec resource for start/end of the monitoring commands would start up after the exec resource in which start/end of a target monitoring application to be written starts up. If you make incorrect settings, the monitoring command may consider that an error has occurred in target monitoring application.

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

Example:

Failover Group A



See Appendix A “Script templates” for example of s writing a script.

Suspending and resuming monitoring

The Application Server Agent can suspend and resume monitoring. While monitoring is suspended, application server maintenance can be performed. Follow the procedure below:

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 the ExpressCluster WebManager to indicate that monitoring has been suspended:
   `clp_xxxxmon will stop monitoring..` [ID`:`***`watchid`***]
4. To resume monitoring, run the following command from the server console whenever you wish to resume:
   Monitoring Command watchid `--resume`
5. The following message is displayed on the ExpressCluster WebManager to indicate that monitoring is suspended:
   clp_xxxxmon will start monitoring.. [ID***:watchid***]

Application Server Agent command reference

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

- **Command line**

  Gives an idea what the user actually enters.

  "[ ]" indicates that the enclosed parameter may be skipped.

  "|" indicates one of the parameters separated by this symbol should be chosen.

- **Description**

  Provides information on function.

- **Option**

  Provides information on parameters used in the command line.

- **Remarks**

  Gives supplementary information such as details on parameter settings.

- **Monitoring method**

  Gives explanation on how monitoring is done.

- **Note**

  Gives information on what should be noted when using the command.

- **Example of command entry**

  Provides example of writing script when using the command.

Tuxedo monitoring command

clp_tux81mon: the clp_tux81mon monitors the Tuxedo daemon.

Command line:

Start Monitoring clp_tux81mon	*Identifier* [-s *Application_Server_name*] [-i *Monitoring_interval*] [-c *Retry_count*] [-r *Response_wait_time*]
Stop Monitoring clp_tux81mon	*Identifier* --stop [*Stop_wait_time*]
Interrupt Monitoring clp_tux81mon	*Identifier* --pause
Resume Monitoring clp_tux81mon	*Identifier* --resume
Display Information clp_tux81mon	*Identifier* --disp
Delete Information clp_tux81mon	*Identifier* --del

Description	This monitoring command stops when a Tuxedo error is detected. The command also stops, interrupts, and resumes monitoring. To specify --pause/--resume/--disp/--del, run from the root privileged console.

Option	*Identifier*	Specifies an identifier to uniquely identify the monitoring command. You must set an identifier.
	-s *Application_ server_name*	Specifies the name of an application server to be monitored. Default value: BBL
	-i *Monitoring_interval*	Specifies the application server monitoring interval in seconds (between 1 and 10000). Default value: 80
	-c *Retry_count*	Specifies the retry count of the application monitoring (between 1 and 10000). Default value: 120
	-r *Response_wait_time*	Specifies the wait time (between 1 and 10000) for application server monitoring in seconds. Default value: 120
	--stop	Stops the monitoring command
	Stop_wait_time	Specifies time to wait for the monitoring command to stop normally (between 1 and 10000). Default value: 60
	--pause	Temporarily interrupts monitoring.
	--resume	Resumes monitoring.
	--disp	Displays a process ID (pid) of the monitoring command identifier managed by the monitoring command. Not for ordinary use. (See Note 4)
	--del	Deletes information about the monitoring command identifier managed by the monitoring command. Not for ordinary use. (See Note 4)

Remarks	About identifier: You need to specify a unique identifier 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 and is case sensitive. About the -s –parameter: Unless specified otherwise, the –s parameter monitors BBL. In general, monitoring BBL is sufficient; however, if you want to monitor a certain application server, specify its name. Use the application server name defined by Tuxedo, not the server name on the network. A text can be specified up to 255 bytes. The actual effective length varies by each parameter. No verification for the effective length is performed with this monitoring command. The effective length complies with the Tuxedo specification. Behavior at the time when larger than 255 is specified is not guaranteed.

Monitoring Method	This monitoring command monitors the following: This monitoring command executes the application server monitoring by using the API of the Tuxedo. The command determines the following results as an error: (1) no response to the application server connection and/or to the acquisition of the status within a response wait time (–r parameter value) (2) an error is reported during the connection to the application server and/or the acquisition of the status.

Note 1	Set the install path of the Tuxedo to the environment variable before executing the monitoring command. Monitoring cannot be done properly if the path is not configured
Note 2	If any library of the Tuxedo such as libtux.so does not exist, monitoring cannot be performed.
Note 3	When an error is detected in a system call of the OS, the monitoring command determines it as a system error and stops without retrying.
Note 4	Management information is not initialized properly when the monitoring command process is stopped by a kill command during monitoring, and the monitoring command with the same identifier may fail to start. If this happens, run the "clp_tux81mon identifier --disp" command and you will see the process ID corresponding to the specified identifier. Use the ps command to check the process ID of the executable file and if the process is other than the monitoring command, remove management information by using the "clp_tux81mon identifier --del" command. NEVER run the "clp_tux81mon identifier --del" command by specifying a normally operating identifier. If you do this, the monitoring command malfunctions.

Example of command entry	[start.sh] **source /opt/bea/tuxedo8.1/tux.env** **export TUXCONFIG=/mnt/share/tuxconfig** **clp_tux81mon tuxwatch** [stop.sh] **source /opt/bea/tuxedo8.1/tux.env** **clp_tux81mon tuxwatch --stop** This command starts up from an exec resource.

WebLogic monitoring command

clp_wls81mon: the clp_wls81mon monitors the WebLogic.

Command line:

Start Monitoring clp_wls81mon	*Identifier* [-a *Host_name*] [-n *Port_number*] -u *User_name* –p *Password* | –f *Config _file* –k *key_ file* [-t demotrust | nonssl | customtrust –s *Key_ store_ file*] [-i *Monitoring_interval*] [-c *Retry_count*] [*Response_wait_time*]
Stop Monitoring clp_wls81mon	*Identifier* --stop [*Stop_wait_time*]
Interrupt Monitoring clp_wls81mon	*Identifier* --pause
Resume Monitoring clp_wls81mon	*Identifier* --resume
Display Information clp_wls81mon	*Identifier* --disp
Delete Information clp_wls81mon	*Identifier* --del

Description	This monitoring command stops when a WebLogic error is detected. The command also stops, interrupts and resumes monitoring. To specify --pause/--resume/--disp/--del, run from the root privileged console.

Option	*Identifier*	Specifies an identifier to uniquely identify the monitoring command. You must set an identifier.
	-a *Host_name*	Specifies the server that runs the application server to be monitored. Specify the server by IP address or the server name. Default value: localhost
	-n *Port_number*	Specifies a port number to connect to the application server. Default value: 7002
	-u *User_name*	Specifies the user name of the WebLogic. Default value: none
	-p *password*	Specifies the password for the WebLogic. Default value: none
	-f *Config_file*	Specifies the file in which the user information is saved. Specifies the full path of the file. Default value: none
	-k *Key_file*	Specified the file in which the password required to access to a config file is saved. Specify the full path of the file.

	-t *Authentication_ method*	Specifies the authentication method to be used to connect to the application server demotrust (default value): authentication that uses the authentication file of the WebLogic nonssl: no authentication customtrust: general SSL authentication
	-s *Key_store_file*	Specifies the authentication file when authenticating SSL. Specify the full path of the file. Specify this only when specifying customtrust with-t. Default value: none
	-i *Monitoingr_interval*	Specifies the monitor interval in seconds (between 1 and 10000) of the application server monitoring. Default value: 60
	-c *Retry_count*	Specifies the retry count (between 1 and 10000) when an error is detected by the application server monitor. Default value: 2
	-r *Response_wait_time*	Specifies the response wait time of the application server monitor process in seconds (between 1 and 10000) . Default value: 120
	--stop	Terminates the monitoring command.
	Stop_wait_time	Specifies time to wait for the monitoring command to stop normally in seconds (between 1 and 10000). Default value: 60
	--pause	Temporarily interrupts monitoring.
	--resume	Resumes monitoring.
	--disp	Displays a process ID (pid) of the monitoring command identifier managed by the monitoring command. Not for ordinary use. (See Note 4)
	--del	Deletes information about the monitoring command identifier managed by the monitoring command. Not for ordinary use. (See Note 4)

Remarks	About identifier: You need to specify a unique identifier 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 and is case sensitive. About the -n parameter: The default value is 7002. Change to an appropriate port number when operating without authentication and/or when starting more than one application server. About -u -p -f -k parameters: Use by combining -u and -p or -f and -k. You need to specify one of the combinations as a user to be connected to a server. About -a -u -p -f -k -s parameters: A text can be specified up to 255 bytes with these parameters. The actual effective length of text varies by each parameter. No verification for the effective length is performed with this monitoring command. The effective length complies with the Tuxedo specification. Behavior at the time when larger than 255 is specified is not guaranteed.

Monitoring Method	This monitoring command monitors the following: By executing Ping with webLogic.Admin command, this monitoring command monitors the application server. The command determines the following results as an error: (1) No response to the Ping issued to the application server within a response wait time (-r parameter value) (2) An error reporting as the response to the Ping.

Note 1	Set the install path of the WebLogic to the environment variable before executing the monitoring command. Monitoring cannot be done properly if the path is not configured.
Note 2	A Java environment is required to start monitoring with this command. The application server system uses Java functions. Therefore if Java stalls, it may be recognized as an error.
Note 3	When an error is detected in a system call of OS, the monitoring command determines it as a system error and stops without executing retry.
Note 4	Management information is not initialized properly when the monitoring command process is stopped by a kill command during monitoring and the monitoring command with the same identifier may fail to start. If this happens, run the "clp_wls81mon identifier --disp" command and you will see the process ID corresponding to the specified identifier. Use the ps command to check the process ID of the executable file and if the process is other than the monitoring command, remove management information by using the "clp_wls81mon identifier --del" command. NEVER run the "clp_tux81mon identifier --del" command by specifying a normally operating identifier because a monitoring command malfunctions if you do so.

Example of command entry	[start.sh] source /mnt/shared/domains/mydomain/setEnv.sh clp_wls81mon wlswatch [stop.sh] clp_wls81mon wlswatch --stop This command starts up from an exec resource.

WebSphere monitoring command

clp_was60mon: the clp_was60mon monitors the WebSphere.

Command line:

Start Monitoring clp_was60mon	*Identifier* [-s *Application_server_name*] [-u *User_name* -p *Password*] [-f *Profile_name*] [-i *Monitoring_interval*] [-c *Retry_count*] [-r *Response_wait_time*]
Stop Monitoring clp_was60mon	*Identifier* --stop [*Stop_wait_time*]
Interrupt Monitoring clp_was60mon	*Identifier* --pause
Resume Monitoring clp_was60mon	*Identifier* --resume
Display Information clp_was60mon	*Identifier* --disp
Delete Information clp_was60mon	*Identifier* --del

Description	This monitoring command stops when a WebSphere error is detected. The command also stops, interrupts and resumes monitoring. To specify --pause/--resume/--disp/--del, run from the root privileged console.

Option	*Identifier*	Specifies an identifier to uniquely identify the monitoring command. You must set an identifier.
	-s *Application_server_name*	Specifies the application server name to be monitored. Default value: server1
	-u *User_name*	Specifies the user name of the WebSphere. Default value: none
	-p *Password*	Specifies the password for the WebSphere. Default value: none
	-f *Profile_name*	Specifies the profile name of the application Server. Default value: default
	-i *Monitoring_interval*	Specifies the monitor interval of the application monitoring in seconds (between 1 and 10000). Default value: 60
	-c *Retry_count*	Specifies the retry count (between 1 and 10000) when an error is detected by the application server monitor. Default value: 2
	-r *Response_wait_time*	Specifies the response wait time of the application server monitor process in seconds (between 1 and 10000) . Default value: 120
	--stop	Terminates the monitoring command.

	Stop_wait_time	Specifies time to wait for the monitoring command to stop normally in seconds (between 1 and 10000). Default value: 60
	--pause	Temporarily interrupts monitoring.
	--resume	Resumes monitoring.
	--disp	Displays a process ID (pid) of the monitoring command identifier managed by the monitoring command. Not for ordinary use. (See Note 4)
	--del	Deletes information about the monitoring command identifier managed by the monitoring command. Not for ordinary use. (See Note 4)

Remarks	About identifier: You need to specify a unique identifier on the system to control the monitoring command. You cannot start up the monitoring command with an identifier that is used by a monitoring command that has been already started. Specify an identifier using alphanumeric characters. An identifier should be up to 255 bytes and is case sensitive. About -u -p parameters You need to specify both a user name and a password. -s -u -p –f parameter The length of a character string which you can specify with the parameters above is up to 255 bytes. The actual valid length is different depending on each parameter. This monitoring command does not confirm the valid length. The length complies with the specification of the WebLogic. The operation is not guaranteed when a value that exceeds 255 bytes is specified.

Monitoring Method	This monitoring command monitors the following. Execute monitoring of the application server by using the serverStatus.sh command. The command determines the following results as an error: (1) no response to the command execution within a response wait time (-r parameter value) (2) an error is reported with the state of the acquired application server.

Note 1	Set the install path of the WebSphere to the environment variable before executing the monitoring command. Monitoring cannot be done properly if the path is not configured.
Note 2	A Java environment is required to start monitoring with this command. The application server system uses Java functions. If Java stalls, it may be recognized as an error.
Note 3	When an error is detected in a system call of OS such as file open, the monitoring command determines it as a system error and stops without executing retry.

Note 4	Management information is not initialized properly when the monitoring command process is stopped by a kill command during monitoring and the monitoring command with the same identifier may fail to start. If this happens, run the “clp_was60mon identifier --disp” command and you will see the process ID corresponding to the specified identifier. Use the ps command to check the process ID of the executable file and if the process is other than the monitoring command, remove management information by using the “clp_was60mon identifier --del” command. NEVER run the “clp_was60mon identifier --del” command by specifying a normally operating identifier because a monitoring command malfunctions if you do so.

Example of command entry	[start.sh] clp_was60mon waswatch [stop.sh] clp_was60mon waswatch –stop This command starts up form an exec resource.

Each monitoring command may specify a user name and/or password. In that case, you need to write a name and a password as parameters of the monitoring command on an active script of a failover group explicitly. Because user names and passwords are important data for security reasons, it is desirable not to specify them explicitly.

The password management function notifies a user name and/or password to the monitoring command just by specifying only a user name if both user name and password are written in the password management function. A user name and password are managed per monitoring command.

Each monitoring command operates according to the registration status of a user name and password to a password management file. Therefore, no special application neither environment configuration to use the user name management function is required.

When using a management function of a password, you need to create a new file to manage a password.

* Create the following files to manage passwords if needed.

Monitoring command name	Password management file
clp_wls81mon	/opt/nec/clusterpro/work/clp_wls81mon
clp_was60mon	/opt/nec/clusterpro/work/clp_was60mon

A password management file can be accessed only by root user.

Separate a name and password with a comma “,” in the information of a password management file. When registering more than one user, add by starting a new line.

```
user1,password1
user2,password2
user3,password3
        :
        :
userN,passwordN
```

Note:
A user name and password can be specified up to 255 bytes.
Do not enter any unnecessary space and tab code.
Do not specify the same user name more than once.

When using a password management function, you have to write the -u parameter specified by a user name and do not write the -p parameter specified by a password in the description of the monitoring command.

The parameter specifications and operations of the monitoring command are as follows.

	With -u parameter	**Without -u parameter**
With -p parameter	Values specified by each parameter are valid.	The monitoring command stops due to a parameter error.
Without -p parameter	If a password corresponding to a user name specified by the –u parameter is written in a password management file, the password written in the password management file is valid. If the password is not written, the monitoring command stops due to a parameter error.	[clp_wls81mon] If both –f and –k are specified, acquire authentication information from the key file and config file. If other than the parameters written above are specified, the monitoring command stops due to a parameter error.
		[clp_was60mon] Default values of each command are valid. (Without a user name and a password)

Chapter 3 How to check monitor status

This chapter provides instructions for how to check the monitor status by using ExpressCluster X Application Server Agent. This chapter also includes information about results of commands and messages on the screen or in the log.

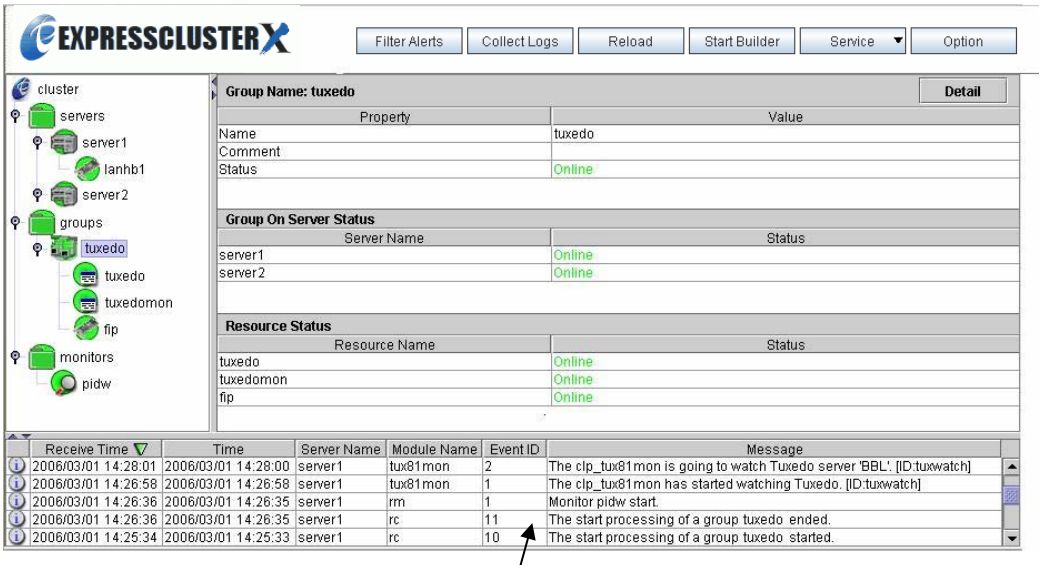
This chapter covers:

Verifying monitor information with monitoring commands

Verify the monitor status by using the monitoring commands on the ExpressCluster WebManager alert view.

Verifying alert messages on the WebManager

You can view alert messages that ExpressCluster X displays on the WebManager.



Displayed on the ExpressCluster WebManager alert view.

As shown above, messages are displayed on the alert view of the lower part of the WebManager window. If the messages shown are long, they are displayed in multiple lines. In such a case, other message may interrupt a message shown in multiple lines depending on the timing.

The same message displayed in the alert message is logged to the syslog.

Refer to “Alert messages” on page 37 for details on alert messages.

Log collection when an error has occurred

Error logs of monitoring commands will be produced in the same directory where error logs of the ExpressCluster Server are produced. Logs are collected in the same way as how ExpressCluster logs are collected. For more information, see “Collecting logs by using the WebManager” in Chapter 1, “Functions of the WebManager” and “Collecting logs (clplogcc command)” in Chapter 4, “ExpressCluster command reference” in the *Reference Guide*.

Alert messages

Messages being output by clp_tux81mon

Messages indicating normal operation

#	Message	Explanation	Remarks
1	clp_tux81mon started monitoring Tuxedo. [ID:xx]	clp_tux81mon has started up.	-
2	clp_tux81mon will start monitoring the Tuxedo server ‘xxx’. [ID:xx]	clp_tux81mon has started monitoring the Tuxedo.	-
3	clp_tux81mon will stop monitoring the Tuxedo server ‘xxx’. [ID:xx]	clp_tux81mon stops monitoring.	-
7	clp_tux81mon will stop monitoring. [ID:xx]	clp_tux81mon suspended monitoring.	This message is displayed when monitoring is suspended by using the --pause command.
8	clp_tux81mon will start monitoring. [ID:xx]	clp_tux81mon restarted monitoring.	This message is displayed when monitoring is restarted by using the --resume command.
52	clp_tux81mon trial version is effective till nn/nn/nn (mm/dd/yyyy).	Running with trial version license.	-

Messages due to settings error

#	Message	Explanation	Remarks
11	The parameter value of the clp_tux81mon is invalid. [ID:xx]	The format for parameter value of clp_tux81mon is invalid.	Check the parameter value of the monitoring command.
13	An identifier is not specified in the clp_tux81mon command.	The identifier is not specified in clp_tux81mon.	Check the parameter value of the monitoring command.
14	The specified identifier is already used. [ID:xx]	clp_tux81mon cannot be restarted because the same identifier is already used.	Check the parameter value of the monitoring command.
15	clp_tux81mon could not perform the end processing. [ID:xx]	End processing could not be performed with --stop parameter.	-
51	The license of clp_tux81mon is not registered.	The license is not registered	Register the license.
53	The trial version license has expired in nn/nn/nn (mm/dd/yyyy).	The trial version license has expired.	-
55	The trial version license is effective from nn/nn/nn (mm/dd/yyyy).	The trial version license cannot be used yet.	-
56	The registration license overlaps.	The license key you have registered overlaps with other key.	Register a different license key for each server.

Messages displayed when an error is detected by the application server monitoring

#	Message	Explanation	Remarks
6	clp_tux81mon detected an error and will be terminated. [ID:xx]	Terminating because clp_tux81mon detected an error.	Troubleshoot the problem by following the message displayed immediately before.
21	clp_tux81mon has detected a network error in connecting to Tuxedo server ‘xxx’(nn). [ID:xx]	An error was detected during the connection process to the application server. *nn* shows the error code of Tuxedo API.	Troubleshoot the error according to the error code.
31	clp_tux81mon has detected an error in Tuxedo server 'xxx' (stall). [ID:xx]	No response returns when executing the application server monitoring.	Check if there is any error on the application server.
33	The status of Tuxedo server ‘xxx’ is not active. [ID:xx]	The application server is not active.	Check if there is any error on the application server.
35	clp_tux81mon has detected an error in disconnecting from Tuxedo server ‘xxx’(nn]). [ID:xx]	An error was detected during the process to disconnect from the application server. *nn* shows an error code of Tuxedo API.	Troubleshoot the error according to the error code.
36	clp_tux81mon could not get status of Tuxedo server ‘xxx’(nn). [ID:xx]	An error is detected during the process of getting the status from the application server. *nn* shows an error code of Tuxedo API.	Troubleshoot the error according to the error code.

Messages due to system error

#	Message	Explanation	Remarks
42	clp_tux81mon has detected system error (xxx nn). [ID:xx]	A Linux system error has occurred. xxx indicates a function name and nn indicates an error code.	Check the system status based on error codes
54	Failed to check the license information of clp_tux81mon.	Failed to verify the license information.	The license management module of ExpressCluster may be old. Check the module update.

Messages being output by clp_wls81mon

Messages indicating normal operation

#	Message	Explanation	Remarks
1	clp_wls81mon started monitoring WebLogic. [ID:xx]	clp_wls81mon has started.	-
2	clp_wls81mon will start monitoring the WebLogic server ‘xxx’. [ID:xx]	clp_wls81mon has started monitoring the WebLogic.	-
3	clp_wls81mon will stop monitoring the WebLogic server ‘xxx’. [ID:xx]	clp_wls81mon stops monitoring.	-
7	clp_wls81mon will stop monitoring. [ID:xx]	clp_wls81mon suspended monitoring.	The message is displayed when monitoring is suspended by using the --pause command.
8	clp_wls81mon will start monitoring. [ID:xx]	clp_wls81mon restarted monitoring.	The message is displayed when monitoring is restarted by using the --resume command.
9	clp_wls81mon got the password. [ID:xx]	clp_wls81mon got the password information from the password management file.	-
52	clp_wls81mon trial version is effective till nn/nn/nn (mm/dd/yyyy).	Running with the trial version license.	-

Messages due to setting error

#	Message	Explanation	Remarks
11	The parameter value of clp_wls81mon is invalid. [ID:xx]	The parameter value is invalid in its format.	Check the parameter value of the monitoring command.
12	The combination of the parameter of -u, -p, -f, and -k of a clp_wls81mon command is not specified correctly. [ID:xx]	The combination of the -u, -p, -f, and -k parameter of clp_wls81mon command is not correct.	Check the parameter value of the monitoring command.
13	An identifier is not specified in the clp_wls81mon command.	The identifier is not specified in clp_wls81mon.	Check the parameter value of the monitoring command.
14	The specified identifier is already used. [ID:xx]	clp_wls81mon cannot be restarted because the same identifier is already used.	Check the parameter value of the monitoring command
15	clp_wls81mon could not perform the end processing. [ID:xx]	End processing could not be performed with --stop parameter.	-
16	clp_wls81mon has detected User authentication error in WebLogic server 'xxx'. [ID:xxx]	Authentication of a user failed.	Check the parameter value of the monitoring command
17	clp_wls81mon has detected SSL verification error in WebLogic server 'xxx'. [ID:xxx]	Authentication of SSL failed.	Check the parameter value of the monitoring command. Check the SSL setting of the monitoring command.
18	clp_wls81mon has detected Host name error in WebLogic server 'xxx'. [ID:xxx]	Checking of hostname failed.	Check the parameter value of the monitoring command.
51	The license of wls81mon is not registered.	The license is not registered	Register the license.

#	Message	Explanation	Remarks
53	The trial version license has expired in nn/nn/nn (mm/dd/yyyy).	The trial version license has expired.	-
55	The trial version license is effective from nn/nn/nn (mm/dd/yyyy).	The trial version license cannot be used yet.	-
56	The registration license overlaps.	The license key you have registered overlaps with other key.	Register a different license key for each server.

Messages being output when error is detected by the application server monitoring

#	Message	Explanation	Remarks
6	clp_wls81mon detected an error and will be terminated. [ID:xx]	Terminating because clp_ wls81mon detected an error.	Troubleshoot the problem by following the message displayed immediately before.
19	Failed to connect to WebLogic server ‘xxx’. [ID:xx]	Failed to connect to the application server.	Check if there is any error on the application server.
20	clp_wls81mon cannot find the path specified by WebLogic.	The path specified by WL_HOME, which is the environment variable of the WebLogic, does not exist.	Check if WL_HOME is set correctly.
31	clp_tux81mon has detected an error in WebLogic server 'xxx' (stall). [ID:xx]	No response returns after the application server monitoring is executed.	Check if there is any error on the application server.
32	clp_wls81mon has detected an error in WebLogic server 'xxx' (stopped). [ID:xxx]	The status of stop is notified through the response from the application server.	Check if there is any error on the application server.
36	clp_wls81mon could not get status of WebLogic server ‘xxx’. [ID:xx]	An error was detected during the process of acquiring the status from the application server.	Check if there is any error on the application server.

Messages due to system error

#	Message	Explanation	Remarks
42	clp_wls81mon has detected system error (xxx nn). [ID:xx]	A Linux system error has occurred. xxx indicates a function name and nn indicates an error code.	Check the system status based on error codes
54	Failed to check the license information of clp_wls81mon.	Failed to verify the license information.	The license management module of ExpressCluster may be old. Check the module update.

Messages being output by clp_was60mon

Messages indicating normal operation

#	Message	Explanation	Remarks
1	clp_was60mon started monitoring WebSphere. [ID:xx]	clp_was60mon has started.	-
2	clp_was60mon will start monitoring the WebSphere server ‘xxx’. [ID:xx]	clp_was60mon has started monitoring the WebSphere.	-

#	Message	Explanation	Remarks
3	clp_was60mon will stop monitoring the WebSphere server ‘xxx’. [ID:xx]	clp_was60mon stops monitoring.	-
7	clp_was60mon will stop monitoring. [ID:xx]	clp_was60mon suspended monitoring.	The message is displayed when monitoring is suspended by using the --pause command.
8	clp_was60mon will start monitoring. [ID:xx]	clp_was60mon restarted monitoring.	The message is displayed when monitoring is restarted by using the --resume command.
9	clp_was60mon got the password. [ID:xx]	clp_was60mon got the password information from the password management file.	-
52	clp_was60mon trial version is effective till nn/nn/nn (mm/dd/yyyy).	Running with trial version license.	-

Messages due to setting error

#	Message	Explanation	Remarks
11	The parameter value of clp_was60mon is invalid. [ID:xx]	The parameter value is invalid in its format.	Check the parameter value of the monitoring command.
12	'-u' parameter or '-p' parameter is not specified at the clp_was60mon command. [ID:xx]	The '-u' parameter or '-p' parameter is not specified in the clp_was60mon command.	Check the parameter value of the monitoring command.
13	An identifier is not specified in the clp_was60mon command.	No identifier is specified in the clp_was60mon command.	Check the parameter value of the monitoring command.
14	The specified identifier is already used. [ID:xx]	clp_was60mon cannot be started newly because the same identifier is already used.	Check the parameter value of the monitoring command
15	clp_was60mon could not perform the end processing. [ID:xx]	End processing could not be performed with the --stop parameter.	-
16	clp_was60mon has detected User Authentication error in WebSphere server xxx.[ID:xx]	User authentication failed.	Check the parameter value of the monitoring command.
20	clp_was60mon cannot find the path specified by WebSphere in system. [ID:xx]	The path specified by the environment variable of the WebSphere, “WAS_INST_PATH”, does not exist.	Check if WAS_INST_PATH is set correctly.
51	The license of clp_was60mon is not registered.	The license is not registered	Register the license.
53	The trial version license has expired in nn/nn/nn (mm/dd/yyyy).	The trial version license has expired.	-
55	The trial version license is effective from nn/nn/nn (mm/dd/yyyy).	The trial version license cannot be used yet.	-
56	The registration license overlaps.	The license key you have registered overlaps with other key.	Register a different license key for each server.

Messages being output when error is detected by the application server monitoring

#	Message	Explanation	Remarks
6	clp_was60mon detected an error and will be terminated. [ID:xx]	Terminating after clp_was60mon detected an error.	Troubleshoot the problem by following the message displayed immediately before.
31	clp_was60mon has detected an error in WebSphere server 'xxx' (stall). [ID:xx]	No response returns after the application server monitoring is executed.	Check if there is any error on the application server.
32	clp_was60mon has detected an error in WebSphere server 'xxx' (stopped). [ID:xx]	The stopping status is notified through the response from the application server.	Check if there is any error on the application server.
36	clp_was60mon could not get status of WebSphere server ‘xxx’. [ID:xx]	An error was detected during the process of acquiring the status from the application server.	Check if there is any error on the application server.

Messages due to system error

#	Message	Explanation	Remarks
42	clp_was60mon has detected system error (xxx nn). [ID:xx]	A Linux system error has occurred. xxx indicates a function name and nn indicates an error code.	Check the system status based on error codes
54	Failed to check the license information of clp_was60mon.	Failed to verify the license information.	The license management module of ExpressCluster may be old. Check the module update.

Chapter 4 Settings for ExpressCluster X Application Server Agent

This chapter provides an explanation on how to set ExpressCluster X Application Server Agent.

This chapter covers:

Setting up ExpressCluster X Application Server Agent

To monitor the applications by using ExpressCluster X Application Server Agent, follow the steps below.

Step 1 Creating a failover group (for the target monitoring application)

Create a failover group for monitoring the target monitoring application and performing failover when an error occurs. Then add a group resource excluding exec resource, such as disk resource and IP resource.

*If you have already created a failover group before setting up ExpressCluster X Application Server Agent and it is used for monitoring, use that group. In this case, you do not need to perform Step 1.

Step 2 Adding exec 1 (for starting the target monitoring application)

Add the exec resource for starting the target monitoring application to the failover group you have created in Step 1. In this guide, this exec resource is called exec 1.

Step 3 Confirmation test

If you have finished Step 2, reflect the settings information to the server, and check that the target monitoring application is started normally by exec 1.

Step 4 Adding exec 2 (for starting the monitoring command)

Add the exec resource for starting the monitoring command to the failover group. In this guide, this exec resource is called exec 2.

Step 5 Adding a pid monitor resource

Add a pid monitor resource for monitoring the exec 2 that has been added in Step 4.

Step 6 Verifying the settings for monitoring command and pid monitor resource

Reflect the settings information of the exec 2 and the pid monitor resource that you have added in Steps 4 and 5 to the server, and check that they can operate normally.

Step 1 Creating a failover group

Create a failover group for the target monitoring application. Add the group resource excluding exec resource such as this disk resource or IP resource.

Note:
If you have already created a failover group before setting up ExpressCluster X Application Server Agent, you can use this group. In this case, you do not have to perform Steps 1 to 3.

Step 1-1 Adding a group

Add a failover group to an existing cluster system. The explanation is given with an example of creating a failover group that includes FIP resource and disk resource to add to a cluster system that consists of two servers (server1, 2). Replace the value when you actually operate.

Note:
When creating a new cluster, refer to Chapter 3, "Creating the cluster configuration data using the Builder" in the *Installation and Configuration Guide*.

1. Start up the Builder.
   (Default path: C:\Program Files\CLUSTERPRO\clpbuilder-l\clptrek.html)
2. Open the cluster configuration data where a group will be added.
3. Click **Groups** in the tree view, and select **Add** on the **Edit** menu.
4. **Group Definition** dialog box is displayed. Enter the group name **failover1** in the **Name** box, and click **Next**.
5. Click **server1** in the **Available Servers** and click **Add**. **server1** is added to the **Servers that can run the Group**. Likewise, add server2.
6. Check that **server1** and **server2** are set in this order to the **Servers that can run the Group,** and then click **Finish**.

Proceed to the following Step 1-2.

Step 1-2 Adding a group resource (floating IP address)

Add a floating IP address to the group you created in Step 1-1.

1. Click **failover1** in the tree view, and click **Add** on the **Edit** menu.
2. The **Resource Definition** dialog box is displayed. Select the group resource type "**floating ip resource**" in the **Type** box, and enter the group name (floating ip address) in the **Name** box. Click **Next**.
3. Enter the IP address (ex. **10.0.0.12**) in the **IP Address** box. Click **Next**.
4. **Recovery Operation at Activation Failure Detection** and **Recovery Operation at Deactivation Failure Detection** are shown. Ignore them and click **Next**.
5. Click **Finish** in the screen shown.

Proceed to the following Step 1-3.

Step 1-3 Adding a group resource (disk resource)

Add a disk resource to the group you added a FIP resource in Step 1-2.

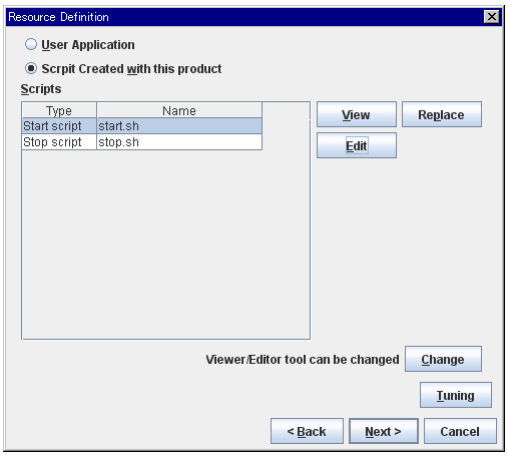
1. Click **failover1** in the tree view, and click **Add** on the **Edit** menu.
2. Enter the 2nd group resource information in the **Resource Definition** dialog box. Select the group resource type (**disk resource**) in the **Type** box, and enter the group name **disk1** in the **Name** box. Click **Next**.
3. Enter the device name (ex. **/dev/sdb2**), mount point (ex. **/mnt/sdb2**) to their corresponding boxes. Select the file system **ext3** on the **File System** box and the disk type **Disk** from the **Disk Type** box. Click **Next**.
4. **Recovery Operation at Activation Failure Detection** and **Recovery Operation at Deactivation Failure Detection** are shown. Ignore them and click **Next**.
5. Click **Finish** in the screen shown.

Proceed to the following Step 2.

Step 2 Adding the exec resource (exec 1) for starting the target monitoring application

Add the exec resource for starting the target monitoring application to the failover group that you have created in Step 1. In this guide, this exec resource is called exec 1. You can use the script templates for starting the target monitoring application. Here, the settings for starting the tuxedo daemon are given as an example.

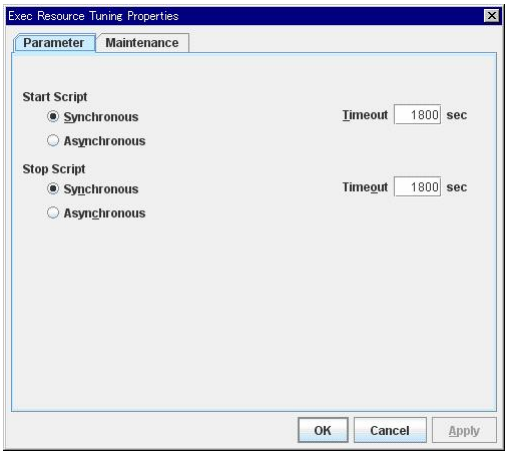
1. If the Builder has not been started yet, start it.
   (Default path: C:\Program Files\ CLUSTERPRO\clpbuilder-l\clptrek.html)
2. Right-click the target failover group, and click **Add Resource**.
3. The **Resource Definition** dialog box is displayed. Select **execute resource** in **Type**. Enter any name you like (enter **tuxedo** in this example), and click **Next**.
4. The following window is displayed. Confirm that **Script Created with this product** is selected. Select **Start script** in **Type** in the **Scripts** table, and click **Replace**.



5. Select the script to start the target monitoring application. Move to the folder of the script templates, and specify the script, “start.sh” in the “tuxedo” folder. Click **Open**.

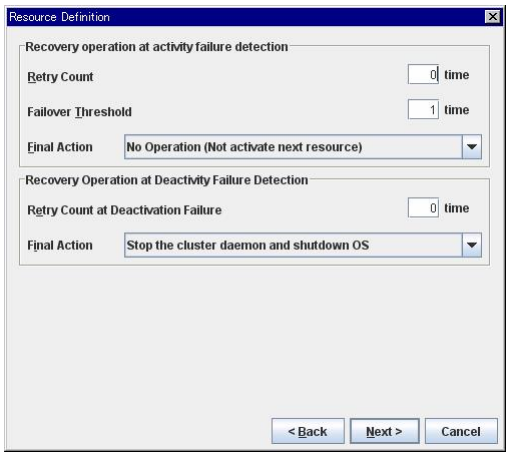
Note:
The following is the default installation folder of the script templates.
C:\Program Files\CLUSTERPRO\clpbuilder-l\scripts\linux

6. The replacement confirmation dialog box is displayed. Specify the original file to be replaced and the path, and select **Yes**.
7. In the same way, replace the end script. Select **Stop Script** in **Type** from the **Scripts** table and click **Replace**. Specify the script, “stop.sh” in the “tuxedo” folder, and click **Open**.
8. Modify the script. If you click **Edit**, the editor will open. You can modify the script according to the environment. See Appendix A “Script templates”, for what to be modified.
9. Click **Tuning**, and check that **Synchronous** is set for **Start Script** and **Stop Script** on the **Parameter** tab. (**Synchronous** is set by default.)

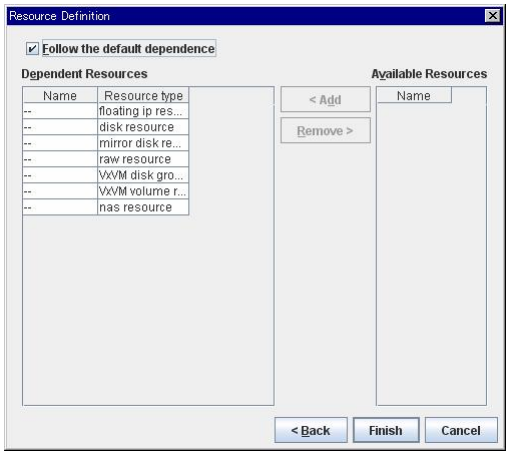


Note:
If you select **Synchronous**, the script is executed with the return value. Here, you need to set **Synchronous** to **Start Scrip** and **Stop Script** because it is required to confirm that the application has been started successfully.

10. In the following screen, you do not need to change the settings. If necessary, configure the settings according to your environment. Click **Next**.



11. In the following screen, you do not need to change the settings. Make sure that the disk resource and IP resource exist on **Dependent Resources**, and click **Finish**.



12. Exec resource for the target monitoring application (exec 1) has been created. Move to the Step 3.

Step 3 Confirmation test for target monitoring application startup

After completing the Steps 1 and 2, convert the configuration information to the server, and check that the monitored application is started normally by exec 1 after that.

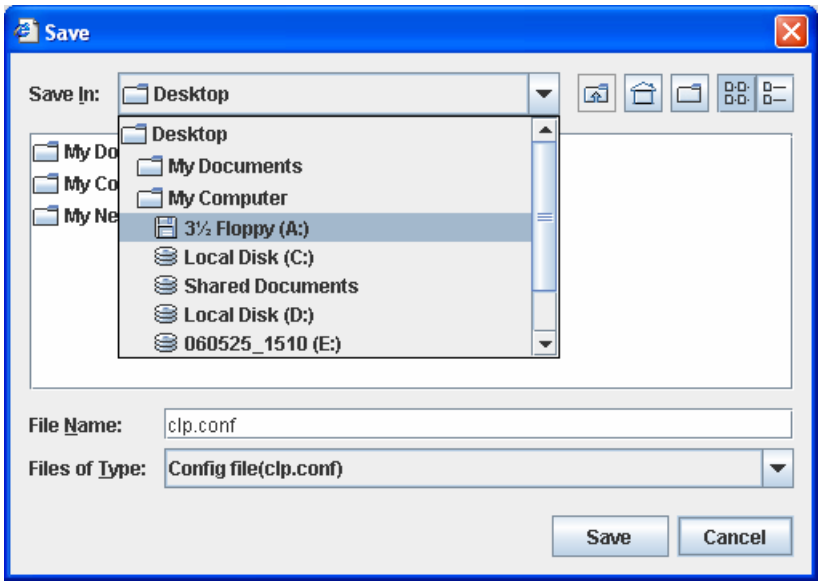
The following describes how to modify the settings to the server using an example of editing the cluster configuration data with the Builder installed on the Windows machine and saving it to the floppy disk.

Related Information:
For using the Builder installed on the Linux machine or saving the cluster configuration data on the file system, refer to "Saving the cluster configuration data" in Chapter 3 "Creating the cluster configuration data using the Builder" in the *Installation and Configuration Guide*.

Saving the cluster configuration data on a floppy disk (Windows)

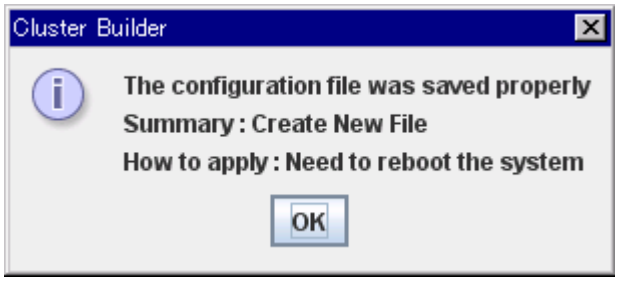
To save the cluster configuration data created with the Builder on Windows machine, follow the procedures below:

1. Insert the floppy disk into the floppy disk drive. Click **Save the configuration file** on the **File** menu. You can directly save it in the floppy disk without creating a directory in the floppy disk.
2. The following dialog box is displayed. Select the floppy disk drive in the **Save** box and click **Save**.



Note:
- Two files (clp.conf and clp.conf.rep) and the directory (scripts) are saved. If any of these are missing, the command to create a cluster does not run successfully. Make sure to treat these three as a set. If you have changed a newly created configuration data, clp.conf.bak is also created as well as those three files.

When the cluster configuration data is saved, the following message is displayed:



3. View what is contained in the floppy disk to check the two files (clp.conf and clp.conf.rep) and the directory (scripts) are located in the floppy disk. If they are in a directory in the floppy disk, you need to specify the directory when you run the command to create a cluster.

Creating a cluster using the data in a floppy disk

To create a cluster using the cluster configuration data saved on a floppy disk, follow the procedures below. If the data is saved in a file system, refer to Chapter 4, "Installing ExpressCluster" in the *Installation and Configuration Guide*.

Note:
- You have to restart all servers after installing the server RPM.
- The server used to create a cluster is the one specified as a master server at the time of creating the cluster configuration data.

1. Insert the floppy disk that contains the cluster configuration data created with the Builder into the floppy disk drive of the server specified as a master server.

Note:
In the clpcfctrl command, /dev/fd0 is used as the floppy disk device, and /mnt/floppy as the mount point by default. If your environment is different from these default values, specify them using the option. For details on option, refer to the *Reference Guide*.

2. Distribute the configuration data saved in the floppy disk to the servers.
   - To use the floppy disk that contains the data saved on Linux, run the following command:
     `clpcfctrl --push -l`
   - To use the floppy disk (1.44-MB formatted) that contains the data saved on Windows, or to use the Windows-formatted floppy disk that contains the data on Linux, run the following command:
     `clpcfctrl --push -w`

Note:
If the two files (clp.conf and clp.conf.rep) and one directory (scripts) that are requisite to create a cluster are saved directly in the floppy disk, you do not need to specify any directory.

The message below is displayed:

```
Need to shutdown system and reboot
please shutdown system after push. (hit return) :
```

3. Press **ENTER**.
   When the cluster is successfully created, the message below is displayed:

```
success.(code:0)
```

Related Information:
For troubleshooting of the clpcfctrl command, refer to the *Reference Guide*.

Verifying the behavior of the target monitoring application

Execute the operations below from the WebManager or the command line to verify that the target monitoring application is running normally.

Note:
Be sure to execute Steps 1 to 4 below before you move to the next step (adding exec 2). If you execute these steps after adding exec 2, it becomes difficult to distinguish whether an error is caused when the application is being started or after it has been started.

Execute the following steps to confirm the monitoring target operation with the WebManager. Start the WebManager (http://*FIP_address_for_the WebManager_group*: *port_number* (default value is 29003)), and execute the following four verification steps.

(Verification 1) Starting up a group

1. Right-click the icon of the failover group that you want to start in the WebManager tree view, and click **Start**.
2. Check that the group icon has turned green and the application is running normally.

(Verification 2) Stopping a group

1. Right-click the icon of the failover group that you want to start in the WebManager tree view, and click **Stop**.
2. Check that the group icon has turned gray and the application has stopped.

Note:
Check that the group has been started and/or stopped among all the servers that start the failover group.

(Verification 3) Moving a group

Move a failover group among servers. Check that the started group properly moves by moving the group to other servers in order.

1. Right-click the icon of the failover group that you want to move in the WebManager tree view.
2. Click the group icon to check if the group has been started on the destination server in the table view. Check that the application is running normally.

Note:
Depending on the application startup/stop process described in the script, it may take a few minutes to complete moving a group. See the tree view on the main screen to check that the group has been moved successfully.

(Verification 4) Failing over a group

Shut down the server on which the failover group is running, and check that the group fails over on the destination server.

1. In the WebManager tree view, right-click the server on which the group you want to fail over is running, and click **Reboot OS**.
2. Restart the server, and check that the failover group fails over to the destination server. Check that the application is running normally on the destination server.

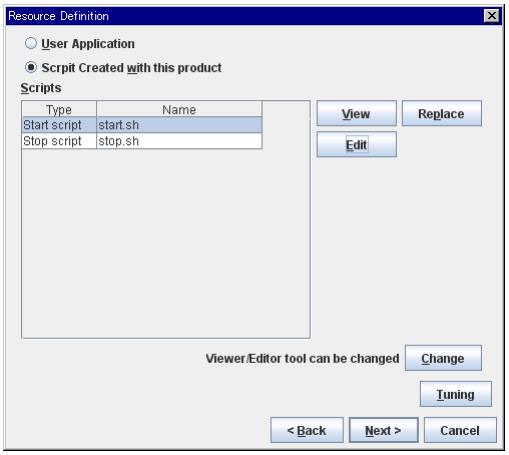
You have completed verifying that the application is running normally by exec 1. Move to Step 4.

Note:
To verify the operation of the application from the command line, run the following commands.
(Verification 1) clpgrp -s group_name
(Verification 2) clpgrp -t group_name
(Verification 3) clpgrp -m group_name
Run the clpstat command to confirm the cluster status.

Step 4 Adding the exec resource (exec 2) for starting monitoring command

Add the exec resource for starting the monitoring command to the failover group. In this guide, this exec resource is called exec 2.

1. If the Builder has not been started, start it.
   (Default path: C:\Program Files\CLUSTERPRO\clpbuilder-l\clptrek.html)
2. Right-click the target failover group, and click **Add Resource**.
3. The **Resource Definition** dialog box is displayed. Select **execute resource** in **Type**. Enter any name you like (enter “tuxedomon” in this example), and click **Next**.
4. The following window is displayed. Confirm that **Script Created with this product** is selected. Select **Start script** in **Type** in the **Scripts** table, and click **Replace**.



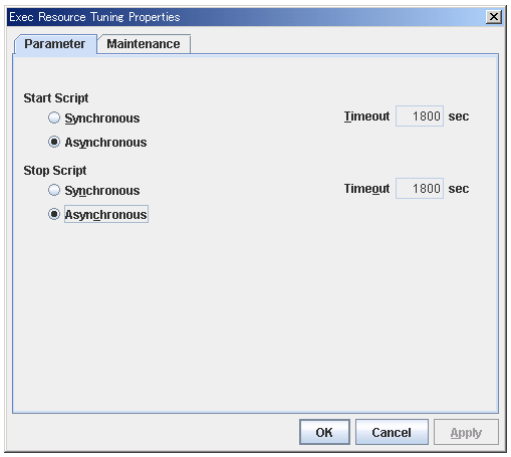
5. Specify the script to start the monitoring command. Move to the folder of the script templates, and specify the script “start.sh” in the “`tuxedo-mon`” folder. Then click **Open**.

Note:
The following is the default installation folder of script template:
C:\Program Files\CLUSTERPRO\clpbuilder-l\scripts\linux

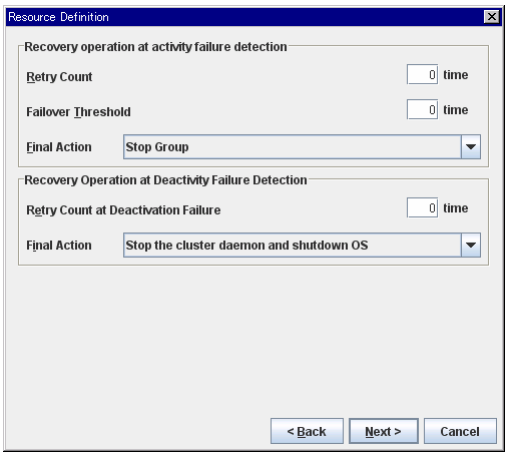
The target monitoring application name is given to the folder of the script templates for starting up the target monitoring application. “-mon” is added after the target monitoring application name in the folder of the script templates for starting up the target monitoring application.

6. The confirmation dialog box asking for replacement is displayed. Specify the original file to be replaced and the path, and select **Yes**.
7. In the same way, replace the end script. Select **Stop Script** in **Type** and click **Replace**. Specify the script “stop.sh” in the “`tuxedo-mon`” folder, and click **Open**.
8. Modify the script. Click **Edit** and open the editor to modify the script according to your environment. See Appendix A “Script templates” for what to be modified.
9. Click **Tuning**, and check that **Asynchronous** is selected for **Start Script** and **Synchronous** for **Stop Script** on the **Parameter** tab.

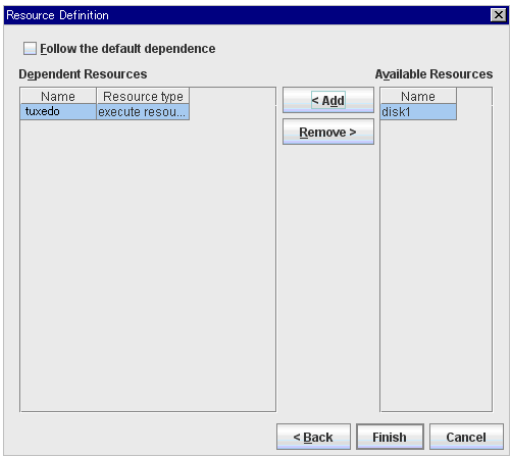
Important:
If you select **Asynchronous**, the script is not started synchronously by the return value, and it is run by itself. The monitoring command enters the monitoring mode and will not take the return value. Be sure to set **Start Script** of the exec resource for running monitoring commands to **Asynchronous**.



10. The following window is displayed. Set **0** to **Failover Threshold** of **Recovery Operation at Activation Failure Detection**, and set **Stop Group** to **Final Action**. Click **Next**.



11. In the following window, uncheck **Follow the default dependence** and set the exec resource (tuxedo) created in Step 2 to **Dependent Resources**.

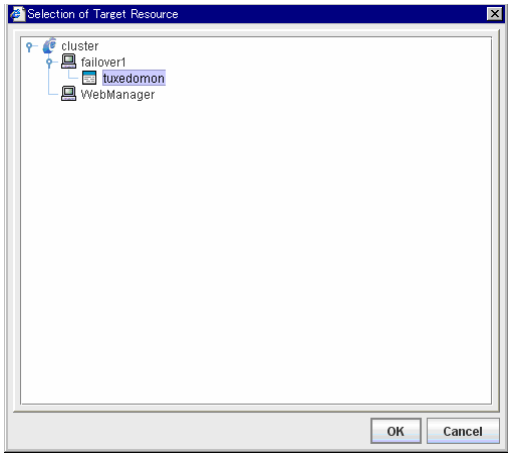


Click the **Finish** button. You have completed adding the exec resource (exec 2) for starting the monitoring command. Move to Step 5.

Step 5 Adding the pid monitor resource

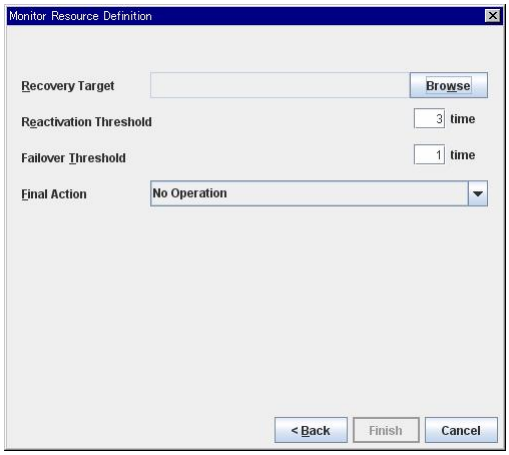
Add the pid monitor resource for monitoring the exec 2 that has been added in Step 4.

1. If the Builder has not been started yet, start it.
   (default path: C:\Program Files\CLUSTERPRO\clpbuilder-l\clptrek.html)
2. Right-click **Monitors** on the tree view, and click **Add Monitor Resource**.
3. The **Monitor Resource Definition** dialog box is displayed. Select the monitor resource type (pid monitor) and enter a monitor resource name in the **Name** box. Click **Next**.
4. Click **Browse** in the window displayed. The **Selection of Target Resource** window is displayed.

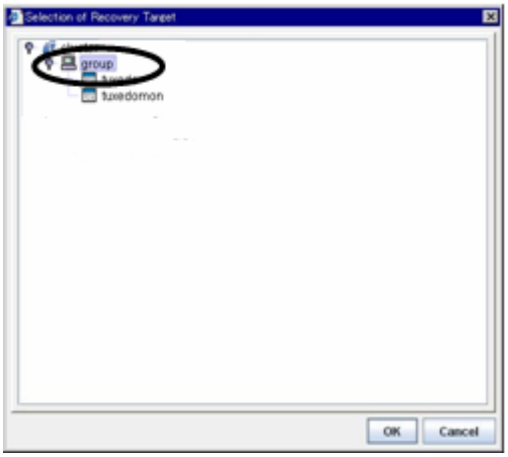


Note:
In the **Selection of Target Resource** window, only the exec resource set to **Asynchronous** in **Exec Resource Tuning Properties** is displayed as selectable target. If nothing can be selected, open the setting window for target exec resource and set the exec resource to **Asynchronous**.

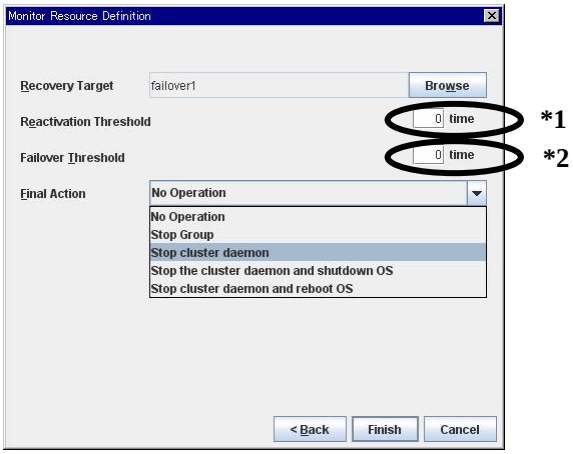
5. Specify the exec resource (exec 2) that starts the monitoring command and click **OK.**
6. Click **Next** to move to the next window and then click **Browse** of **Recovery Target**.



7. The **Selection of Recovery Target** dialog box is displayed. Select the failover group in which the target monitoring application exists.



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



9. When you finish configuring the settings, click **Finish**. You have finished creating the monitor resource. Move to Step 6.

Failover soon after detecting an error

If you want to execute failover when an error is detected, set **0** to **Reactivation Threshold**. If you want to reactivate the group when an error is detected, specify a value other than **0** to **Reactivation Threshold**.

When you set a value other than 0 to **Reactivation Threshold**, you need to set a value longer than the time below to **Wait Time to Start Monitoring**.

From when a monitoring command is started to when it is terminated due to an error:

The time to terminate the monitoring command is determined by the values of -i parameter (monitoring interval), -c parameter (retry count) and -r parameter (response wait time).

Example: When the monitoring command is terminated because an error is detected in the application server:

- -i (monitoring interval)
- : 60 (sec.)
- -c (retry count)
- : 2 (times)
- -r (response wait time)
- : 120 (sec.)

Approximate time to finish the monitoring command:

60(sec.) × 2(times) + monitoring period = 120 + α (sec.)

* Monitoring interval × retry count + monitoring period

Depending on the type of error, the time to end the monitoring command slightly differs. For the monitoring command operation depending on the error type, see “Alert messages” on page 37.

Executing failover without stopping a server

If you want to perform failover without stopping a server as a cluster when an error is detected, set **1** to **Failover Threshold**. In this case, other failover groups are operated continuously on the server that detected the error. When an error is also detected in the failover destination server, a failover occurs again and the failover group returns. If you want to perform failover by stopping a server as a cluster when an error is detected, set **0** to **Failover Threshold**.

In addition, specify either **Stop cluster daemon**, **Stop the cluster daemon and shut down OS,** or **Stop cluster daemon and reboot OS** to **Final Action**.

In this case, other failover group is terminated or a failover occurs because the server that detected an error will not be operated as a cluster.

Considering that the recovery will be done when an error occurs, it is recommended to specify as follows:

Failover Threshold 0

Final Action Stop cluster daemon

Step 6 Verifying behaviors of the monitoring command

In Step 2, you have added the exec resource for starting target monitoring application to the failover group created in Step 1. In Step 3, you have checked that the target monitoring application runs normally.

Then in Step 4, you have added the exec resource for starting monitoring commands. Finally, in Step 5, you have added the monitor resource for monitoring the exec resource for starting monitoring commands.

In Step 6, after updating a failover group, check that monitoring commands run normally by performing the following steps in the same way as you checked the operation of the target monitoring application.

Note:

You need to reflect the cluster configuration data edited in Steps 4 and 5 to the cluster system. For how to reflect the data, see "

Saving the cluster configuration data on a floppy disk (Windows)," "Creating a cluster using the data in a floppy disk" in Step 3.

Execute the following operation from the WebManager, and check that the target monitoring application runs normally.

Start the WebManager (http://*IP_address_for_the WebManager_group*: *port_number* (default value: 29003)), and execute the following four verification steps.

(Verification 1) Starting up a group

1. Right-click the icon of the failover group that you want to start in the WebManager tree view, and click **Start**.
2. Check that the group icon has turned green.
3. Check that the messages of the Event ID1 and ID2 of the monitoring command are displayed in the WebManager alert view.

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

(Verification 2) Stopping a group

1. In the WebManager tree view, right-click the icon of the failover group you want to stop, and click **Stop**.
2. Check that the group icon has turned gray.
3. Check that the message of the Event ID 3 of the monitoring command is displayed in the WebManager alert view.

Note:
Check that the group has been started and/or stopped among all the servers that start the failover group.

(Verification 3) Moving a group

Move a failover group among servers. Check that the started group properly moves by moving the group to other servers in order.

1. In the WebManager tree view, right-click the icon of the failover group that you want to move, and click **Move**.
2. Click the group icon, and check that the group has been started on the destination server in the table view.
3. Check that the monitoring command message is displayed in the WebManager alert view.

   Server name: *Name_of_the_server_from_which_the_group_is_moved*

   Event ID: 3

   Server name: *Name_of_the_server_to_which_the_group_is_moved*

   Event ID: 1, 2

Note:
Depending on the application startup/stop process described in the script, it may take a few minutes to complete moving a group. See the tree view on the main screen to check that the group has been moved.

(Verification 4) Failing over a group

Shut down the server on which the failover group is running, and verify that the group fails over to the destination server.

1. In the WebManager tree view, right-click the server in which the group that you want to fail over is running, and click **Reboot OS**.
2. Check that the server is restarted and the failover group fails over to the destination server.
3. Verify that the message of the monitoring command is displayed in the WebManager alert view.

   Server name: *Name_of_the_server_from_which_the_failover_group_fails_over*

   Event ID: 3

   Server Name: *Name_of_the_server_to_which_the_failover_group_fails_over*

   Event ID: 1, 2

Note:
If you set an extremely small value such as 1 (one) to the monitoring interval and response time parameters, you may not be able to perform monitoring normally. Check the operation carefully beforehand.

Related Information:
For more information on messages, see “Alert messages” on page 37.

You have successfully completed the settings for ExpressCluster X Application Server Agent.

Note:
Run the following commands to confirm the group failover from the command line.
(Verification 1) clpgrp -s *group_name*
(Verification 2) clpgrp -t *group_name*
(Verification 3) clpgrp -m *group_name*
Run the clpstat command to check the cluster status.

Appendix A. Script templates

Setting up script templates

ExpressCluster provides various script templates to monitor applications. Write the necessary information into templates to monitor applications effectively.

The script templates can be used from the ExpressCluster X Builder. There are Windows version and Linux version for the Builder and the templates are prepared for each version. Choose and set up the script templates according to your environment.

Obtain the latest version of the script templates and follow the instruction that comes with the product to update.

Installing the script templates on Windows machine

Log on as administrator when installing the script templates on Windows machine.

Follow the procedures below:

1. Insert the ExpressCluster X CD into a CD drive.
   The Setup Menu window is displayed automatically. If the window is not displayed, execute the menu.exe on the CD drive directly.
2. Click **ExpressCluster for Linux**.

3. Click **ExpressCluster Template Scripts**.

4. A dialog box to verify where to install the template scripts is displayed. Check the path and click **Extract**.

Installing script templates on Linux machine

Set up the script templates after the Builder for the Linux version has been set up. The script templates are installed in a folder having the scripts for the Builder. Refer to Chapter 3, "Creating the cluster configuration data using the Builder" in the *Installation and Configuration Guide* for how to install the Builder.

Log on as root user when installing the script templates on Linux machine.

Follow the procedures below:

1. Insert the ExpressCluster X CD into a CD drive.
2. Mount the CD.

   # mount /dev/cdrom
3. Move to the following directory:

   # cd /mnt/cdrom/Linux/1.0/en/script/Linux
4. Install the script templates using the rpm command.

   # rpm –i expresscls-script-1.0.0-1.linux.i686.rpm

Note:
The rpm file name may vary depending on the version of the Agent. The command description may also vary depending on the type of Linux.

Uninstalling the script templates

Uninstalling the script templates for Windows

To uninstall the script templates, log on as an administrator and delete all folders except defaultl and defaultw under C:\Program Files\CLUSTERPRO\clpbuilder-l\scripts\linux.

Note:
Check the directory because the underlined part may vary depending on the terminal environment.

Uninstalling the script templates for Linux

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

 # rpm -e expresscls-script

If you uninstall the Builder for Linux, the script templates (the Builder for Linux) will be uninstalled as well, but the rpm module information will be remained. Therefore, it is necessary to forcefully delete the package using the rpm command:

(Example): **rpm –e --force expresscls-script**

Details on script templates

Two templates, start.sh and stop.sh, are provided for the scripts. Modify and use them according to your system requirements. Modifications to the scripts need to be made are underlined boldface italics.

For Tuxedo startup

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 "start Tuxedo"
   su - bea -c "source /opt/bea/tuxedo8.1/tux.env;
   export TUXCONFIG=/mnt/share/tuxconfig;
   export PMID=SERVER1;
   export LANG=C;
   tmipcrm -y;
   tmboot -y"

   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 "start Tuxedo"
   su - bea -c "source /opt/bea/tuxedo8.1/tux.env;
   export TUXCONFIG=/mnt/share/tuxconfig;
   export PMID=SERVER1;
   export LANG=C;
   tmipcrm -y;
   tmboot -y"

else
   echo "ERROR_DISK from FAILOVER"
fi
```

```
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

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 "Stop Tuxedo"

   su - bea -c "source /opt/bea/tuxedo8.1/tux.env;
   export TUXCONFIG=/mnt/share/tuxconfig;
   export PMID=SERVER1;
   export LANG=C;
   tmshutdown -s WSL -y;
   sleep 2;
   tmshutdown -k TERM -y;
   sleep 2;
   tmshutdown -k KILL -y;
   sleep 2;
   tmshutdown -c -y;
   tmipcrm -y"

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 "Stop Tuxedo"

   su - bea -c "source /opt/bea/tuxedo8.1/tux.env;
   export TUXCONFIG=/mnt/share/tuxconfig;
   export PMID=SERVER1;
   export LANG=C;
   tmshutdown -s WSL -y;
   sleep 2;
   tmshutdown -k TERM -y;
   sleep 2;
   tmshutdown -k KILL -y;
   sleep 2;
   tmshutdown -c -y;
   tmipcrm -y"
```

```
else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

For Tuxedo monitoring

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 "start Tuxedo monitor"

   source /opt/bea/tuxedo8.1/tux.env
   export TUXCONFIG=/mnt/share/tuxconfig
   clp_tux81mon tuxmon

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 " start Tuxedo monitor "

   source /opt/bea/tuxedo8.1/tux.env
   export TUXCONFIG=/mnt/share/tuxconfig
   clp_tux81mon tuxmon

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

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 "stop Tuxedo monitor"

   source /opt/bea/tuxedo8.1/tux.env
   clp_tux81mon tuxmon --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 "stop Tuxedo monitor"

   source /opt/bea/tuxedo8.1/tux.env
   clp_tux81mon --stop

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

For WebLogic startup

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

   cd /mnt/shared/domains/mydomain

   if [ -f ./myserver/ldap/ldapfiles/EmbeddedLDAP.lok ]
   then
      rm -f ./myserver/ldap/ldapfiles/EmbeddedLDAP.lok
   fi

   date +"%Y/%m/%d %T"
   echo "start WebLogic"
   source setEnv.sh
   ./startWebLogic.sh &
   sleep 10

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

   cd /mnt/shared/domains/mydomain

   if [ -f ./myserver/ldap/ldapfiles/EmbeddedLDAP.lok ]
   then
      rm -f ./myserver/ldap/ldapfiles/EmbeddedLDAP.lok
   fi

   date +"%Y/%m/%d %T"
   echo "start WebLogic`"
   source setEnv.sh
   ./startWebLogic.sh &
   sleep 10

    else
   echo "ERROR_DISK from FAILOVER"
```

```
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

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 "Stop WebLogic Server"
  /mnt/shared/domains/mydomain/stopWebLogic.sh username password

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 "Stop WebLogic Server
  /mnt/shared/domains/mydomain/stopWebLogic.sh username password

else
  echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

For WebLogic monitoring

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_wls81mon start"
   source /mnt/shared/domains/mydomain/setEnv.sh
   clp_wls81mon weblogic -a localhost -t demotrust -n 7002
                         -f
/mnt/shared/domains/mydomain/userconfig.secure
                         -k
/mnt/shared/domains/mydomain/userkey.secure

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_wls81mon start"
   source /mnt/shared/domains/mydomain/setEnv.sh
   clp_wls81mon weblogic -a localhost -t demotrust -n 7002
                         -f
/mnt/shared/domains/mydomain/userconfig.secure
                         -k
/mnt/shared/domains/mydomain/userkey.secure

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

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_wls81mon stop"
  clp_ wls81mon weblogic --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_wls81mon stop"
  clp_wls81mon weblogic -stop

else
  echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi

echo "EXIT"
exit 0
```

For WebSphere startup

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 "Start WebSphere Application Server"
   /opt/IBM/WebSphere/AppServer/bin/startServer.sh server1
-profileName default

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 "Start WebSphere Application Server"
   /opt/IBM/WebSphere/AppServer/bin/startServer.sh server1
-profileName default

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

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 "Stop WebShphere Application Server"
   /opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1
                                -username user -password pass
-profileName default

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 "Stop WebShphere Application Server"
   /opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1
                                -username user -password pass
-profileName default

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

For WebSphere monitoring

start.sh

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

export WAS_INST_PATH=/opt/IBM/WebSphere/AppServer

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 "start WebSphere monitor"
   clp_was60mon wasmon -s server1 -f default -u user -p pass

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 "start WebSphere monitor"
   clp_was60mon wasmon -s server1 -f default -u user -p pass

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

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 "stop WebSphere monitor"
   clp_was60mon wasmon --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 "stop WebSphere monitor"
   clp_was60mon wasmon -stop

else
   echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

Appendix B. Glossary

Cluster partition	A partition on a mirror disk. Used for managing mirror disks. (Related term: Disk heartbeat partition)
Interconnect	A dedicated communication path for server-to-server communication in a cluster. (Related terms: Private LAN, Public LAN)

Virtual IP address	IP address used to configure a remote cluster.
Management client	Any machine that uses the WebManager to access and manage a cluster system.
Startup attribute	A failover group attribute that determines whether a failover group should be started up automatically or manually when a cluster is started.
Shared disk	A disk that multiple servers can access.
Shared disk type cluster	A cluster system that uses one or more shared disks.
Switchable partition	A disk partition connected to multiple computers and is switchable among computers. (Related terms: Disk heartbeat partition)
Cluster system	Multiple computers are connected via a LAN (or other network) and behave as if it were a single system.
Cluster shutdown	To shut down an entire cluster system (all servers that configure a cluster system).
Active server	A server that is running for an application set. (Related term: Standby server)
Secondary server	A destination server where a failover group fails over to during normal operations. (Related term: Primary server)
Standby server	A server that is not an active server. (Related term: Active server)
Disk heartbeat partition	A partition used for heartbeat communication in a shared disk type cluster.
Data partition	A local disk that can be used as a shared disk for switchable partition. Data partition for mirror disks. (Related term: Cluster partition)
Network partition	All heartbeat is lost and the network between servers is partitioned. (Related terms: Interconnect, Heartbeat)
Node	A server that is part of a cluster in a cluster system. In networking terminology, it refers to devices, including computers and routers, that can transmit, receive, or process signals.
Heartbeat	Signals that servers in a cluster send to each other to detect a failure in a cluster. (Related terms: Interconnect, Network partition)
Public LAN	A communication channel between clients and servers. (Related terms: Interconnect, Private LAN)

Failover	The process of a standby server taking over the group of resources that the active server previously was handling due to error detection.
Failback	A process of returning an application back to an active server after an application fails over to another server.
Failover group	A group of cluster resources and attributes required to execute an application.
Moving failover group	Moving an application from an active server to a standby server by a user.
Failover policy	A priority list of servers that a group can fail over to.
Private LAN	LAN in which only servers configured in a clustered system are connected. (Related terms: Interconnect, Public LAN)
Primary (server)	A server that is the main server for a failover group. (Related term: Secondary server)
Floating IP address	Clients can transparently switch one server from another when a failover occurs. Any unassigned IP address that has the same network address that a cluster server belongs to can be used as a floating address.
Master server	The server displayed on top of the **Master Server** in **Cluster Properties** in the Builder.
Mirror connect	LAN used for data mirroring in a data mirror type cluster. Mirror connect can be used with primary interconnect.
Mirror disk type cluster	A cluster system that does not use a shared disk. Local disks of the servers are mirrored.

Appendix C. Index

V

W