ExpressCluster® X *for Linux*

VMware vSphere™ System Configuration Guide

12/10/2012
10th Edition

Revision History

Edition	Revised Date	Description
1	07/14/2009	Created
2	08/21/2009	Correspond to the internal version 2.1.1-1
3	02/26/2010	Added the procedure for adding a second virtual NIC Added the procedure for associating a physical NIC with a virtual NIC Added the firewall settings of vSphere Added the settings for detecting an NIC error on a host Added the settings of userw
4	07/09/2010	Changed the document construction such as chapter Corrected broken links, typo errors
5	10/01/2010	Corresponds to ExpressCluster X3.0
6	12/06/2010	Deleted Shared-disk type cluster construction by virtual disk
7	08/01/2011	Corrected notational errors
8	10/11/2011	Corresponds to ExpressCluster X 3.1 and VMware vSphere 5
9	09/30/2012	Added the settings of management OS
10	12/10/2012	Corresponds to VMware vSphere 5.1.

Disclaimer

Information in this document is subject to change without notice.
NEC Corporation is not liable for technical or editorial errors or omissions in the information in this document.
You are completely liable for all risks associated with installing or using the product as described in this manual to obtain expected results and the effects of such usage.
The information in this document is copyrighted by NEC Corporation.
No part of this document may be reproduced or transmitted in any form by any means, electronic or mechanical, for any purpose, without the express written permission of NEC Corporation.

Trademark Information

ExpressCluster® X is a registered trademark of NEC Corporation.
AMD and AMD Virtualization are trademarks of Advanced Micro Devices, Inc.
Intel, Pentium, Xeon, and Intel VT are registered trademarks or trademarks of Intel Corporation.
VMware and vMotion are registered trademarks of VMware, Inc. in the United States.
Linux is a registered trademark of Linux Torvalds in the United States and other countries.
Other product names, company names, and products written in this manual are trademarks or registered trademarks of their respective companies.

CONTENTS

INTRODUCTION

Target Readers and Purpose

This manual is intended for administrators who want to build a cluster system, system engineers who want to provide user support, and maintenance personnel.
This manual introduces software whose operation in an ExpressCluster environment has been checked. The software and setup examples introduced here are for reference only. They are not meant to guarantee the operation of each software product.

ExpressCluster X Documentation Set

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

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

   ExpressCluster X 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.

   ExpressCluster X 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 *ExpressCluster X Installation and Configuration Guide*.

   ExpressCluster X Integrated WebManager Administrator's Guide
   This guide is intended for system administrators who manage cluster system using ExpressCluster with ExpressCluster Integrated WebManager and for system engineers who introduce the Integrated WebManager. In this guide, details on required items for introducing the cluster system using the Integrated WebManager are explained in accordance with the actual procedures.

   ExpressCluster X WebManager Mobile Administrator's Guide
   – ExpressCluster X 3.1 or later
   This guide is intended for system administrators who manage cluster system using ExpressCluster with ExpressCluster WebManager Mobile and for system engineers who introduce the WebManager Mobile. In this guide, details on required items for introducing the cluster system using the WebManager Mobile are explained in accordance with the actual procedures.

   For details about the ExpressCluster manuals, see the following web site.
   ExpressCluster Web Site
   http://www.nec.com/expresscluster/

2. VMware vSphere document
   For details about the VMware vSphere, see VMware vSphere document.

   VMware vSphere document
   http://www.vmware.com/support/pubs/

Organization of This Manual

Chapter 1 **Configuration**
Provides feasible cluster systems by combining VMware vSphere and ExrpressCluster.

Chapter 2 **Operating environment**
Provides operation environments when combining VMware vSphere and ExrpressCluster.

Chapter 3 **Notes**
Provides notes when combining VMware vSphere and ExrpressCluster.

Chapter 4 **Virtual environment**
Provides instructions on how to construct a virtual environment when combining VMware vSphere and ExpressCluster.

Chapter 5 **Cluster Environment Setup Procedure**
Provides instructions on how to construct the cluster system when combining VMware vSphere and ExpressCluster.

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`*`]`
Monospace (`courier`)	Indicates path names, commands, system output (message, prompt, etc), directory, file names, functions and parameters.	`c:\Program files\EXPRESSCLUSTER`
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.	`clpstat -s [-h` *`host_name`*`]`

Terms used in This Document

The meanings of terms used in this guide are provided below.

Term	Abbreviation	Explanation
Physical server	SV	Server on which VMware ESX or another OS is running
Standalone OS	Operating system	Normal OS used on its own, not with a virtualization platform
Host OS	Host	OS installed on a physical server as the virtualization platform, such as VMware ESX
Virtual machine	VM	Virtual server or client created on a host OS
Guest OS	Guest	OS installed on a virtual machine
Management OS	←	Service Console is not offered for VMware ESXi. The management OS is a guest OS that is used to manage the host OS, in place of Service Console.
ExpressCluster X	CLS	ExpressCluster X
ExpressCluster X SingleServerSafe	SSS	ExpressCluster X SingleServerSafe
Application	AP	Business application
VMware vSphere Management Assistant	vMA	Management OS offered by VMware to manage the host OS
VMware vSphere CLI (Command Line Interface)	CLI	Command interface offered to manage and control the host OS

Chapter 1 Configurations

By combining VMware vSphere and ExpressCluster, clusters that have the configurations below can be set up.

Inter-host OS clusters

ExpressCluster X is installed on the VMware ESX Service Console to cluster physical servers together. Both normal applications and guest OSs can be failed over.

By associating guests with hosts, applications in guest OSs can be monitored.

For details about how to set up the cluster, see "Setting up an inter-host OS cluster" (on page 32). If using a guest-to-host association, see "Using a guest-to-host association in an inter-host OS cluster" (on page 38).

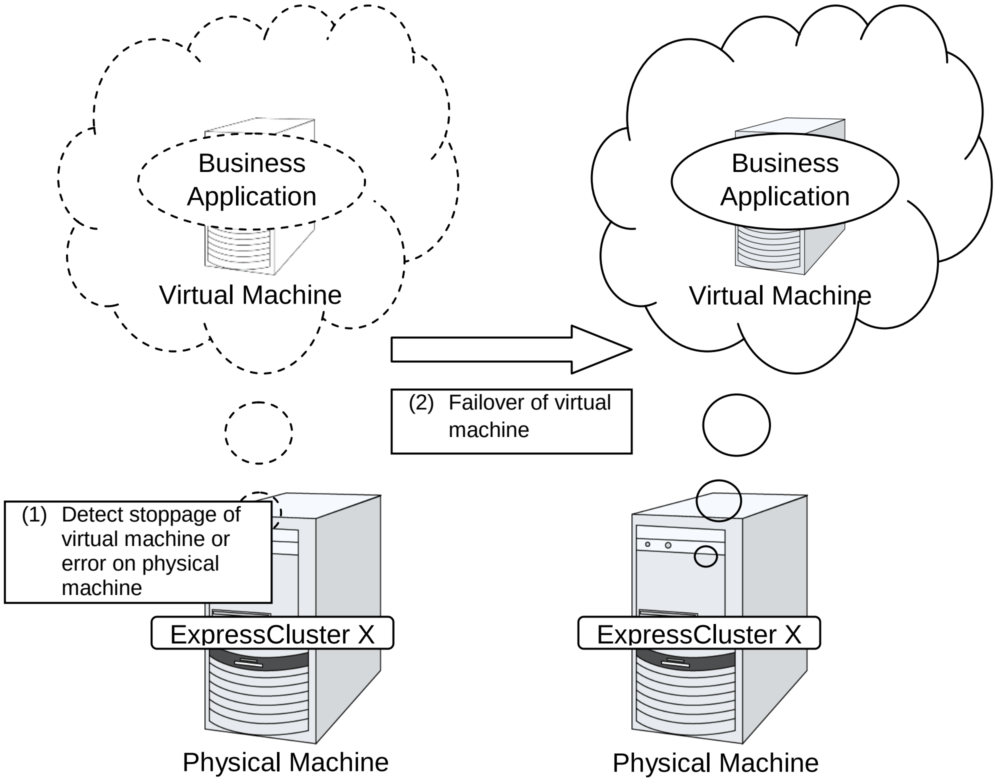


Figure 1: Overview of an inter-host OS cluster

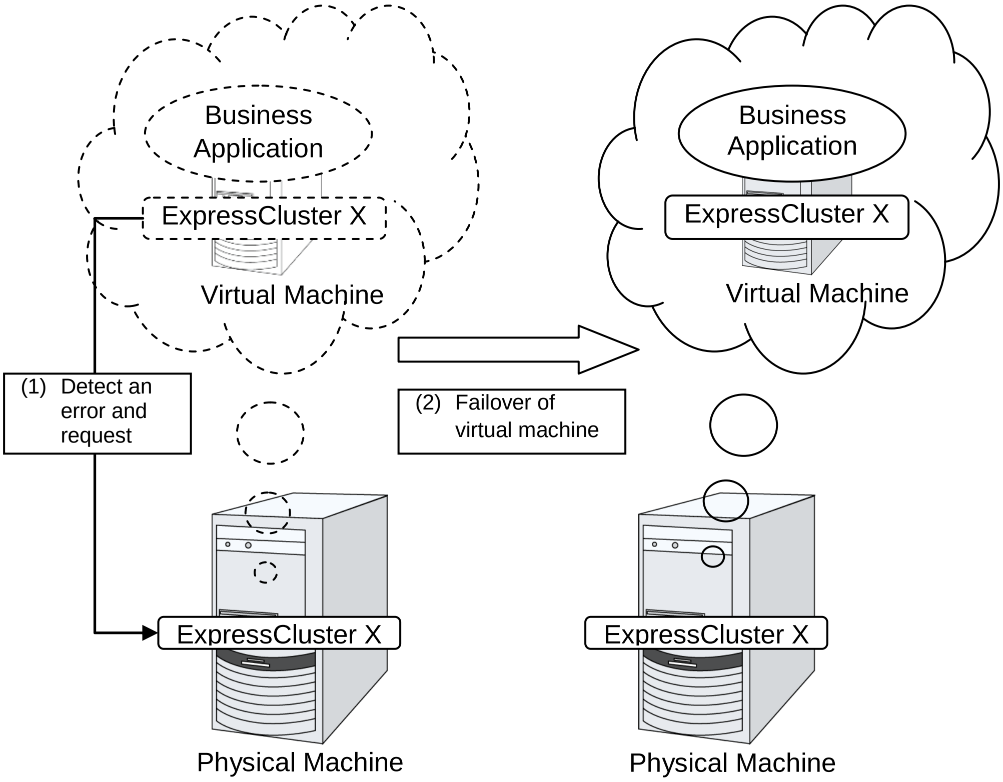


Figure 2: Overview of an inter-host OS cluster when guest-to-host associations are used

Inter-guest OS clusters

ExpressCluster X is installed on guest OSs to cluster virtual machines together. As in normal cluster systems, applications can be failed over, improving operational availability.

For details about how to set up the cluster, see “Setting up an inter-guest OS cluster” (on page39).

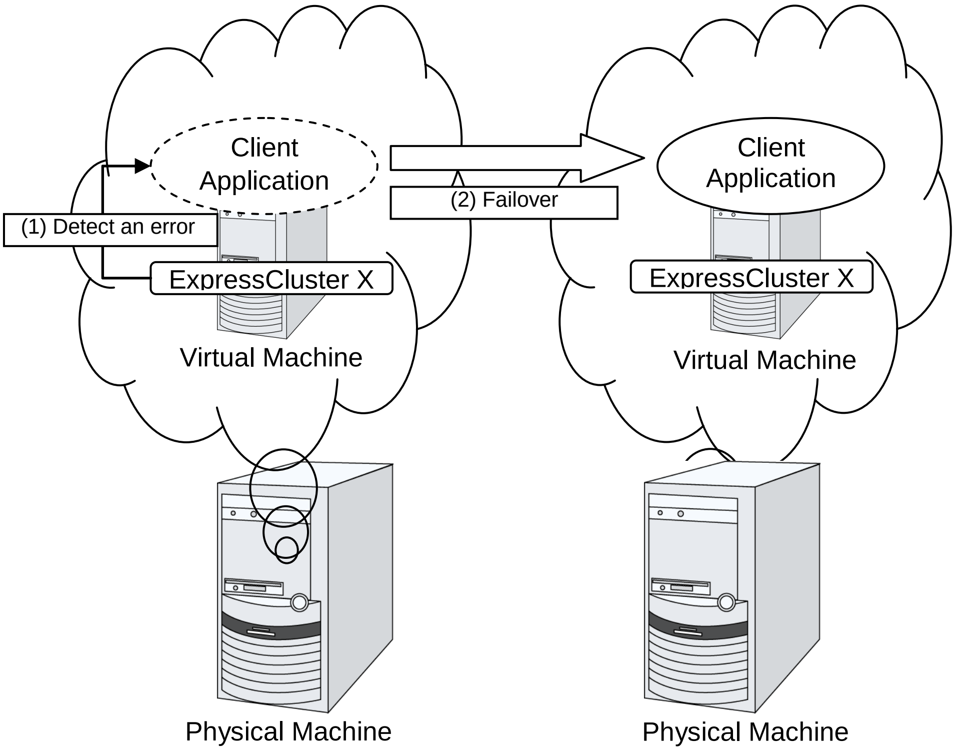


Figure 3: Overview of an inter-guest OS cluster of the mirror disk type

Inter-management OS clusters

HyperVisor offered by VMware vSphere 5 is integrated into VMware ESXi, so that Service Console, offered with the conventional VMware ESX, cannot be used.

The management OS (virtual machine) is provided for each host as a substitute for Service Console to manage the virtual machine. Installing ExpressCluster X under this management OS enables the guest OS to be subjected to failover in the VMware ESXi environment, in addition to ordinary business applications.

Moreover, when using vMA (VMware vSphere Management Assistant) as the management OS, you can monitor the physical machine in exactly the same way as with Service Console by using Remote CLI and vSphere API for managing and controlling VMware ESXi.

For details on the cluster configuration procedure, see “Setting up inter-management OS clusters” (on page 41).

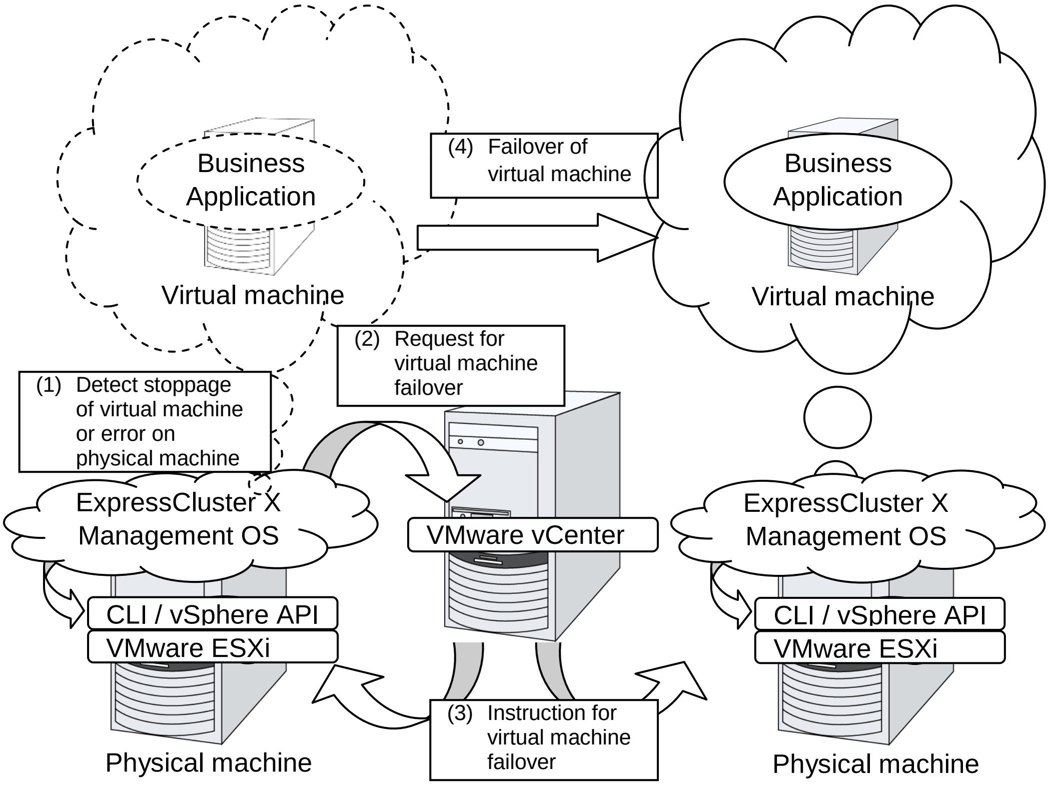

(4) Failover of virtual machine
Business Application
Virtual machine
Business Application
Virtual machine
(2) Request for virtual machine failover
(1) Detect stoppage of virtual machine or error on physical machine
ExpressCluster X Management OS
VMware vCenter
ExpressCluster X Management OS
CLI / vSphere API
VMware ESXi
CLI / vSphere API
VMware ESXi
(3) Instruction for virtual machine failover
Physical machine
Physical machine

Figure 4: Outline of an inter-management OS cluster

Physical server-to-virtual machine cluster

ExpressCluster X is installed on the OS running on a physical server and on the OS running on a virtual machine to cluster the physical server and virtual machine together. As in normal cluster systems, applications can be failed over.

For details about how to set up the cluster, see “Setting up a physical server-to-virtual machine cluster” (on page 52).

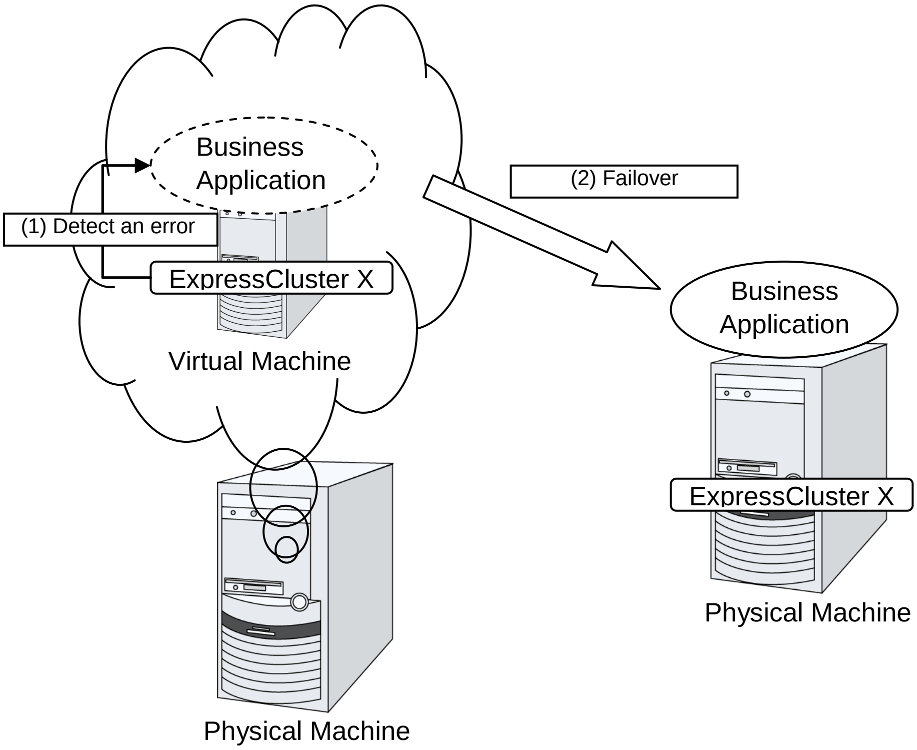


Figure 5: Overview of a physical server-to-virtual machine cluster of the mirror disk type

VMware HA associations

A virtual machine sets up VMware HA, and, if a guest OS fails or the vCenter-to-ESX network is disconnected, the guest OS is failed over.

By installing ExpressCluster X SingleServerSafe on the VMware ESX Service Console, physical hardware errors that cannot be detected by VMware can be detected. If, in particular, an error is detected in the shared disk in which a guest OS is stored, ESX is shut down immediately so operations do not continue on the guest OS, which runs unstably. (Figure 6)

Also, by installing ExpressCluster X on a guest OS, guest OS errors (including virtual hardware errors and application errors) can be detected, so operations can be failed over. (Figure 7)

For details about how to set up this configuration, see “Associating with VMware HA” (on page 68).

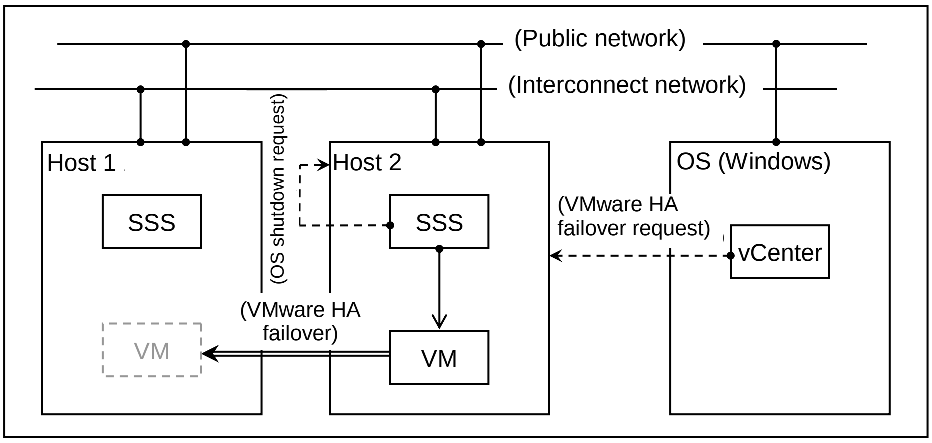


Figure 6: Overview of the configuration of associations between ExpressCluster X SingleServerSafe on hosts and VMware HA

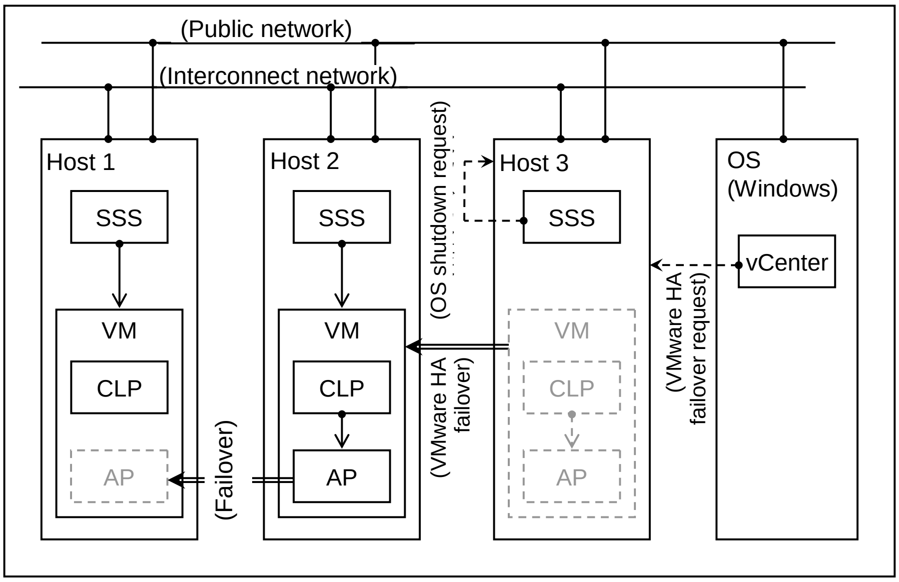


Figure 7: Overview of the configuration of associations among ExpressCluster X on guests, ExpressCluster X SingleServerSafe on hosts, and VMware HA

Chapter 2 Operating environment

The versions of VMware vSphere and ExpressCluster covered in this chapter are as follows:

Table 1: Configuration

VMware vSphere Version	Cluster configuration	Associated ExpressCluster version	Construction procedure
VMware vSphere 4.0 VMware vSphere 4.0 update 1 VMware vSphere 4.0 update 2 VMware vSphere 4.1 VMware vSphere 4.1 update 1 VMware vSphere 4.1 update 2	Inter-host OS cluster	ExpressCluster X 2.1 for Linux	Setting up an inter-host OS cluster (on page 32) The script needs to be used to control and monitor the virtual machine.
		ExpressCluster X 3.0 for Linux ExpressCluster X 3.1 for Linux	Setting up an inter-host OS cluster (on page 32)
	Inter-guest OS cluster	ExpressCluster X 2.1 for Linux ExpressCluster X 2.1 for Windows ExpressCluster X 3.0 for Linux ExpressCluster X 3.0 for Windows ExpressCluster X 3.1 for Linux ExpressCluster X 3.1 for Windows	Setting up an inter-guest OS cluster (on page 39)
VMware vSphere 5 VMware vSphere 5.1	Inter-guest OS cluster	ExpressCluster X 2.1 for Linux ExpressCluster X 2.1 for Windows ExpressCluster X 3.0 for Linux ExpressCluster X 3.0 for Windows ExpressCluster X 3.1 for Linux ExpressCluster X 3.1 for Windows	Setting up an inter-guest OS cluster (on page 39)
	Inter-management OS cluster	ExpressCluster X 3.1 for Linux	Setting up inter-management OS clusters (on page 41) To monitor the host hardware, it is necessary to use VMware vSphere Management Assistant 5.0 as the management OS, together with the Sample Script shown in this manual.

Chapter 3 Notes

Notes on using ExpressCluster on Service Console

- Install x86_64 version rpm (`clusterpro-X.X.X-X.x86_64.rpm`) on Service Console.[1]
- Select keepalive as monitor method when using user-mode monitor resource (userw) or shutdown stall monitor.
- The following functions are not available.
  - Mirror disk resource (md)
  - Hybrid disk resource (hd)
- When using ExpressCluster on Service Console, the firewall is needed to be configured so that the ExpressCluster communication port can be accessed.
- When using the second NIC, **Service Console Port** is needed to be added on the network configuration on the host.
- The same IP address cannot be set to each port of the network configuration of the host. When you want to use **VMkernel** and **Service Console** at the same time, set different IP addresses.
- Because the output destination of VMware ESX core is `/var/core` by default, the core cannot be collected by log collection function of ExpressCluster.
- VMware vSphere 5 cannot use Service Console because it can use only VMware ESXi 5.0 as HyperVisor.

Notes on using ExpressCluster on virtual machine

- When you want to use the second physical adaptor, the virtual switch associated to it is needed to be created on the network configuration on the host.
- The NIC Link Up/Down monitor resource cannot be used. Use the IP monitor resource instead.

[1] Be careful not to use the rpm for VMware ESX 3.5 Service Console (`clusterpro-2.1.X-X.vmware.i386.rpm`) for X2.1.

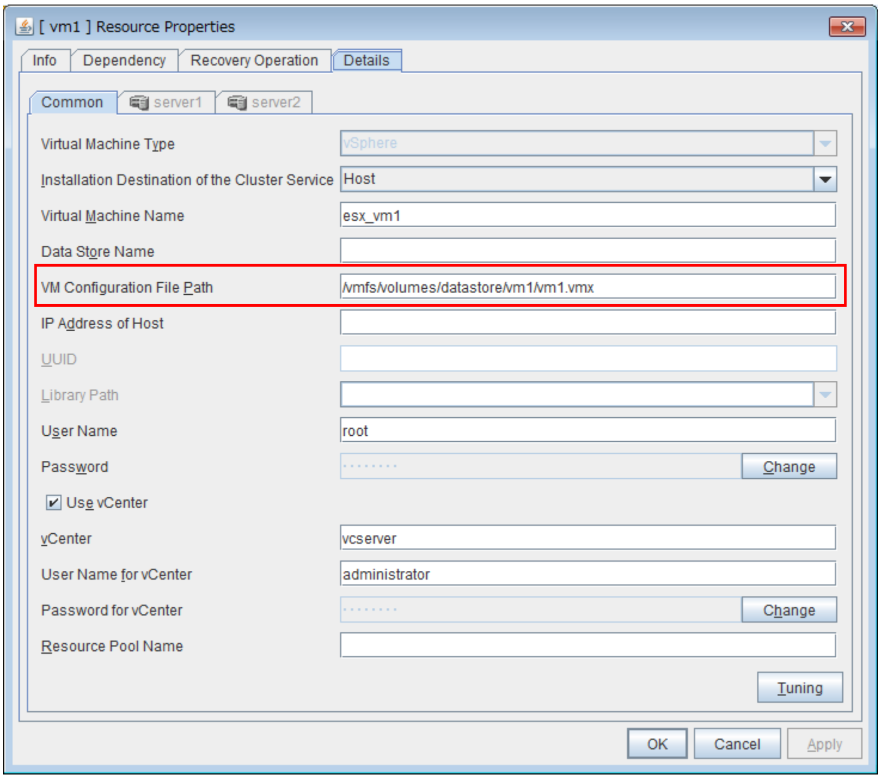
Notes on using ExpressCluster on the management OS

- ExpressCluster X 3.1 and later can be used.
  - Install x86_64 version rpm in vMA.
- Red Hat Enterprise Linux 6 to be used as a management OS
  - Internal version 3.1.4 or earlier: Cannot be used
  - Internal version 3.1.5 or later: Can be used
- The following OSs have already been verified for management use:
  - VMware vSphere Management Assistant 5.0
  - VMware vSphere Management Assistant 5.1
  - Red Hat Enterprise Linux 5.5 (IA32 version/x86_64 version)

  For the internal version 3.1.5 or later:
  - Red Hat Enterprise Linux 6.0 (IA32 version/x86_64 version)
- To perform “Setting the host monitoring monitor” (on page 46), vMA must be used as the management OS.
- To use the second physical adaptor, it is necessary to create a virtual switch connected to the second physical adapter in the host network configuration.
- The name of the virtual machine to be controlled by the virtual machine resource must be the same as its directory name in the data store. See “Changing the virtual machine name” (on page 30).

Notes on inter-host OS clusters

- During cluster operation, do not execute **Suspend** for the virtual machine to be managed by ExpressCluster.
- If using a guest-to-host association, ExpressCluster X 2.1 or later must be installed in the guest OS.
- Make sure to set a **VM Configuration File Path** when setting the virtual machine resource.



Virtual Machine Type	vSphere
Installation Destination of the Cluster Service	Host
Virtual Machine Name	esx_vm1
Data Store Name	
VM Configuration File Path	/vmfs/volumes/datastore/vm1/vm1.vmx
IP Address of Host	
UUID	
Library Path	
User Name	root
Password	········
☑ Use vCenter	
vCenter	vcserver
User Name for vCenter	administrator
Password for vCenter	········
Resource Pool Name	

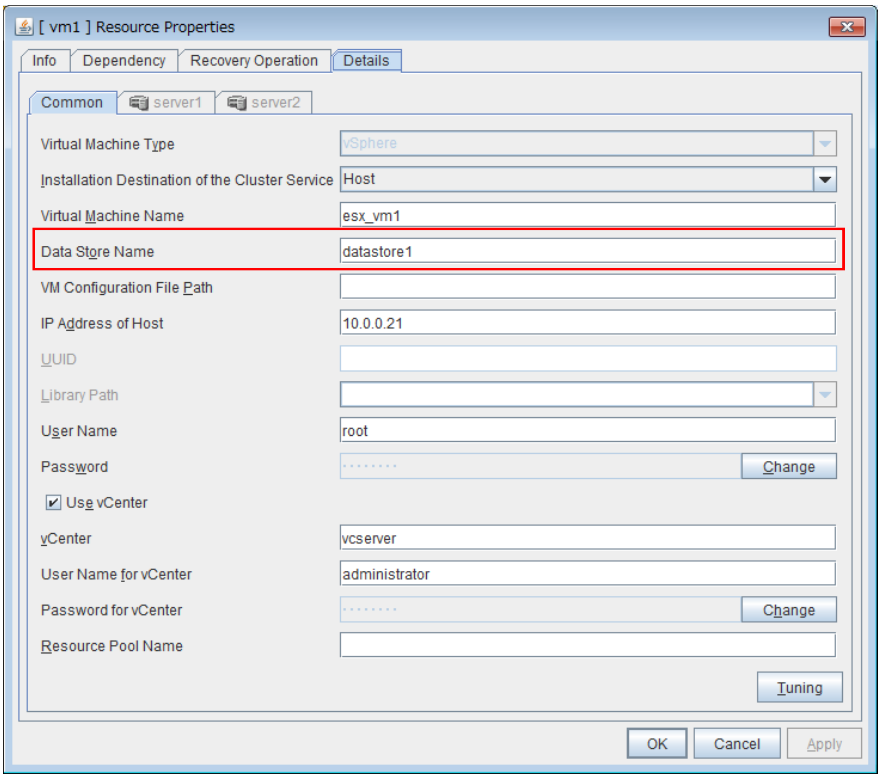
Notes on inter-guest OS clusters

- The COM heartbeat cannot be used.
- The following functions that use the IPMI function cannot be used.
  - Forced stop function
  - Chassis ID lamp linkage function
  - Monitoring using user mode monitoring method “ipmi"
  - Monitoring using shutdown stall monitoring method “ipmi"
  - Final action of group resources when an activation/deactivation error is detected
    “BMC Reset”, “BMC Power Off”, “BMC Power Cycle”, “BMC NMI”
  - Final action of monitoring resources when an error is detected
    “BMC Reset”, “BMC Power Off”, “BMC Power Cycle”, “BMC NMI”
- To use a shared disk in this configuration, the following conditions must be met:
  - You must set the shared disk type of the virtual machine to **Raw device mapping**. In this case, it is recommended that “Physical” be selected for “Compatible mode”.
  - Do not specify an iSCSI connection disk in **Raw device mapping**.
  - You must set the **Share the SCSI bus** setting of the SCSI controller of the virtual machine to **Physical** or **Virtual**. If you select **Virtual**, you cannot set up an inter-guest OS cluster across an ESX host as shown in Figure 3.
    Physical: Virtual disks can be shared among the VMs on all ESX hosts.
    Virtual: Virtual disks can be shared among the VMs on the same ESX host.
- During cluster operation, do not suspend a virtual machine by using **Suspend**. If you suspend a virtual machine by using **Suspend**, ExpressCluster will detect a heartbeat timeout, and a failover group will be activated on another server. If, in this state, you use **Resume** to resume the virtual machine suspended using **Suspend**, both systems become active, and both virtual machines on which the failover group is running are shut down to protect data.

Notes on using inter-management OS clusters

- The management OS must be installed together with VMware ESXi in a one-to-one configuration.
- To use the virtual machine resource, VMware vCenter must be provided.
- The COM heartbeat cannot be used.
- The following functions that use the IPMI function cannot be used.
  - Monitoring using user mode monitoring method “ipmi”
  - Monitoring using shutdown stall monitoring method “ipmi”
  - Final action of group resources when an activation/deactivation error is detected
    “BMC Reset”, “BMC Power Off”, “BMC Power Cycle”, “BMC NMI”
  - Final action of monitoring resources when an error is detected
    “BMC Reset”, “BMC Power Off”, “BMC Power Cycle”, “BMC NMI”

- During cluster operation, do not execute **Suspend** for the virtual machine on which the management OS is installed. If you execute **Suspend** for the virtual machine, ExpressCluster detects a heartbeat timeout and then starts a failover group on another server. In this status, if you execute **Resume** for the virtual machine suspended by using **Suspend**, both systems become active. Both virtual machines on which the related failover group is active are shut down to ensure data protection.
- If OS shutdown is selected for monitor resource recovery or the like, the management OS, rather than the host, is shut down.
- If vMA is used as the management OS, the host can be shut down by using sample script `shutdownhost.pl` (on page 91), given in this manual.
- If the management OS goes down due to shutdown, panic, power-off, or the like while the host is operable, the other guests running on the host on which the management OS was running are live-migrated and the standby server is used.
- During cluster operation, do not execute **Suspend** for the virtual machine to be managed by ExpressCluster.
- Make sure to set a **Data Store Name** when setting the virtual machine resource.



[vm1] Resource Properties

Virtual Machine Type	vSphere
Installation Destination of the Cluster Service	Host
Virtual Machine Name	esx_vm1
Data Store Name	datastore1
VM Configuration File Path	
IP Address of Host	10.0.0.21
UUID	
Library Path	
User Name	root
Password	••••••••
☑ Use vCenter	
vCenter	vcserver
User Name for vCenter	administrator
Password for vCenter	••••••••
Resource Pool Name	

Notes on physical server-to-virtual machine clusters

- See “Notes on inter-guest OS clusters” (on page 22).

Notes on using vMotion concurrently

- When using vMotion, the following conditions are needed to be met.
  - VMware vCenter Server is introduced.
  - **VMkernel port** which can be used by vMotion is configured to each ESX network.
  - The shared storage device[1] is configured and the virtual machine is stored in the data store on the shared storage.
  - The setting of **Share the SCSI bus** of the SCSI controller of the virtual machine is **None**.
  - The DNS is configured correctly, and vCenter and service console of all ESX can resolve each other's names.
  - The host name configured in each ESX and the host name registered to the DNS are the same.
- vMotion and an inter-guest OS cluster that uses the shared disk cannot be used at the same time. Because of the VMware specifications, of virtual machines which have the same virtual disk, only one virtual machine can be executed when the **Share the SCSI bus** setting for the SCSI controller of the virtual machine is not **None**.

Table 2: Whether vMotion can be used at the same time

Cluster configuration	**Inter-host clusters, inter-management OS clusters,** VMware HA, and the like			**Inter-guest OS clusters**	
vMotion target	Guest OS of the non-cluster configuration	Guest OS of **the inter-guest OS cluster** configuration		Guest OS constituting the cluster	
		Shared disk	Mirror disk	Shared disk	Mirror disk
Whether to use vMotion together	Yes	No	Yes	No	Yes

* The **bold face** indicates the cluster configuration by ExpressCluster.

[1] Fibre Channel SAN, iSCSI SAN and NFS are available as connection method to the shared storage device.

Notes on using VMware HA concurrently

➢ To enable the use of VMware HA, the following conditions must be satisfied:

- VMware vCenter Server has already been installed.
- Each ESX has license which can use VMware HA.
- The DNS is correctly configured, and vCenter and the service consoles of all ESXs can resolve each other's names.
- The host name configured for each ESX matches the host name registered to the DNS.
- The Gateway address of Service Console is set correctly on each ESX and the Gateway responds to ping.
- When enabling monitoring of the virtual machine on VMware HA, VMware Tools are installed to the virtual machine to be monitored.
- The management OS is set so that it is not targeted.

Chapter 4 Virtual environment setup procedures

Settings for inter-host OS clusters

Installing VMware ESX

Follow the installation guide provided by VMware, Inc. to install VMware ESX.

- Resource documents supported by VMware
  http://www.vmware.com/support/pubs/

Setting the firewall

When using ExpressCluster on the Service Console, set up the firewall so that the communication ports of ExpressCluster can be accessed.

To disable the firewall:

```
# esxcfg-firewall --allowIncoming --allowOutgoing
```

For details about the port numbers used by ExpressCluster, refer to the following:

- *ExpressCluster X for Linux Getting Started Guide*
  Chapter 5 Notes and Restrictions
  > Before installing ExpressCluster and after installing OS
  >> Communication port number

Setting the virtual machine

Set the virtual machine to be controlled by using virtual machine resources while referring to “Settings for the virtual machine” (on page 31).

Setting the virtual switch

Set the virtual switch as required while referring to “Adding the virtual switch” (on page 31).

Adding the service console port

To use the second virtual switch on the Service Console, add **Service Console port**.

Note: Different IP settings must be specified for **For VMkernel** and **For Service Console**.

(1) Click **Properties** of the second virtual switch, and then select the **Port** tab.

(2) Click **Add**, select **Service Console**, and then click **Next**.

(3) Enter any name for **Network label** of the port group properties, and then click **Next**.

(4) Select **Use the next IP setting**, enter values for **IP address** and **Subnet mask**, and then click **Next**.

(5) Check the settings, and then click **Finish**.

(6) Click **Close** in **Properties** of the virtual switch.

Settings for inter-guest OS clusters

Setting the virtual switch

Set the virtual switch as required while referring to "Adding the virtual switch" (on page 31).

Creating virtual machines

Create virtual machines according to the cluster system to be set up. Examples of creating a virtual machine are described below for reference.

For details about creating virtual machines, refer to the *Basic System Management Guide* provided by VMware, Inc.

- Resource documents supported by VMware
  http://www.vmware.com/support/pubs/

Creating a virtual machine that uses a shared disk[1]

(1) Execute **Create a new virtual machine**.

(2) Select **Standard** from **Configuration**, and then click **Next**.

(3) Enter any name, and then click **Next.**

(4) Select an appropriate host, and then click **Next.**

(5) Select an appropriate data store, and then click **Next.**

(6) Select the guest OS to install, and then click **Next.**

(7) Enter any virtual disk size, and then click **Next**. In this step, specify the size of the disk in which the OS is to be installed. Specify the settings for the shared disk of the cluster system later.

(8) Check the settings, and then click **Finish.**

(9) Right-click the added virtual machine, and then execute **Edit settings**.

(10) Click **Add** on the **Hardware** tab.

(11) To add the second VM Network, select **Ethernet adapter**, and then click **Next**.

(12) Specify **Adapter type**, select **Network label of the second virtual switch**, and then click **Next**.

(13) Check the settings, and then click **Finish.**

(14) Select **Hard disk**, and then click **Next.**

(15) Select **Create a new virtual disk** or **Raw device mapping**, and then click **Next.**
To set up a physical server-to-virtual machine cluster, be sure to select **Raw device mapping**.
For the second or subsequent virtual machine, select **Use the existing virtual disk**, click **Next**, and then select the ***.vmdk** file created for the first one.

(16) If you selected **Create a new virtual disk**, specify the size of the virtual disk.
Select **Support clustering functions such as fault tolerance** of disk provisioning, and then select a data store on the shared disk.
If you selected **Raw device mapping**, specify the disk LUN to use. Next, click **Next**.

[1] If you set up a shared disk on a virtual machine, you can no longer use vMotion. For details, see "Notes on using vMotion " on page 24.

(17) For **Virtual device node**, select a node whose X value in **SCSI (X:Y)** or **IDE (X:Y)** differs from that of the disk for installing OSs, and then click **Next**.

(18) Check the settings, and then click **Finish**.

(19) Make sure that **SCSI controller** has been added to the hardware list, in addition to **Hard disk**. If the SCSI controller is not found, the **Virtual device node** setting of the added **Hard disk** might be incorrect.

(20) Select **SCSI controller**, change **Share the SCSI bus** to **Physical** or **Virtual**, and then click **OK**.

(21) A virtual machine that uses a shared disk is now created. If you have not installed an OS, install one.

Creating a virtual machine that uses a mirror disk

(1) Follow the steps up to (14) in “Creating a virtual machine that uses a shared disk”.

(2) Select **Create a new virtual disk**, and then click **Next**. Also select **Create a new virtual disk** for the second and subsequent machines.

(3) If you selected **Create a new virtual disk**, specify the virtual disk size, and then click **Next**. It is recommended that the same size be specified for all virtual machines.

(4) Click **Next** without specifying anything.

(5) Check the settings, and then click **Finish**.

(6) A virtual machine that uses a mirror disk is now created. If you have not installed an OS, install one.

Settings for inter-management OS clusters

Installing VMware ESXi

Install VMware ESXi as described in the installation guide provided by VMware, Inc.

- Resource documents supported by VMware
  http://www.vmware.com/jp/support/pubs/

Installing the management OS

To use vMA as the management OS, install it as described in the installation guide provided by VMware, Inc.

- Resource documents supported by VMware
  http://www.vmware.com/jp/support/pubs/

To use other OSs, install the guest OS supported by ExpressCluster.

Setting SSH authentication

To enable the management OS to control the virtual machine, make settings so that the management OS and host can be connected by key authentication over SSH.

(1) Enable SSH authentication for the host.

(2) Generate a key for SSH authentication on each management OS.

(3) Add the key created above to the host on which each management OS runs.

(4) Make sure that each management OS can connect to the host by key authentication over SSH.

Setting the virtual switch

Set the virtual switch as required while referring to “Adding the virtual switch” (on page 31).

Setting the virtual machine

Set the virtual machine to be controlled by using virtual machine resources while referring to “Settings for the virtual machine” (on page 31).

Changing the virtual machine name

Use the following procedure to change the name of the virtual machine to be controlled by the virtual machine resource.

(1) Make a connection from VMware vSphere Client, select the virtual machine, and then select **Rename** from the shortcut menu.

(2) Change the virtual machine name.

(3) Open **Inventory** - **Data store and data store cluster**.

(4) Select the data store containing the virtual machine to be changed.

(5) Click **Reference this data store** to open **Data store browser**.

(6) Select the directory containing the configuration file of the virtual machine to be changed and then select **Rename** from the shortcut menu.

(7) Change the name to that changed in (2).

(8) From **Setting mode** of **Cluster Manager**, change **Virtual machine name** of **Virtual machine resource** to the changed virtual machine name.

Settings for the virtual machine

By using the following steps, add configuration parameters to the virtual machine to be controlled by using the virtual machine resource.

(1) On the vSphere Client, open the **Virtual Machine Properties** window for the target virtual machine.

(2) Select **Details - General** in the left pane of the **Options** tab, and then click **Configuration Parameters** at the lower right.

Click **Add Line** on the **Configuration Parameters** window, and then enter the following parameters:
Name: `answer.msg.uuid.altered`
Value: `I moved it`

Adding the virtual switch

For a server that contains more than one physical NIC, a virtual switch related to only one of them is created by default. Therefore, to use the other physical NIC, you must add a new virtual switch.

Adding a second virtual switch

(1) Make a connection from VMware vSphere Client, select the host (physical server), and then click **Network** on the **Configuration** tab.

(2) Click **Add network**

(3) Select **Virtual machine** and then click **Next**.

(4) Select **Create a virtual switch** and then click **Next**.

(5) Enter any name for **Network label** of the port group properties, and then click **Next**.

(6) Check the settings, and then click **Finish**.

Adding the VMkernel port

To use the second virtual switch with vMotion, add “VMkernel port”.

(1) Click **Properties** of the second virtual switch, and then select the **Port** tab.

(2) Click **Add**, select **VMkernel**, and then click **Next**.

(3) Enter any name for **Network label** of the port group properties, check **Use this port group in vMotion**, and then click **Next**.

(4) Select **Use the next IP setting**, enter the values for **IP address** and **Subnet mask**, and then click **Next**.

(5) Check the settings, and then click **Finish**.

(6) If the warning **The default gateway is not set. To use this network interface, you may need to set the default gateway. Do you want to set it up right now?** is output, click **No**.

(7) Click **Close** in **Properties** of the virtual switch.

Chapter 5 Cluster Environment Setup Procedure

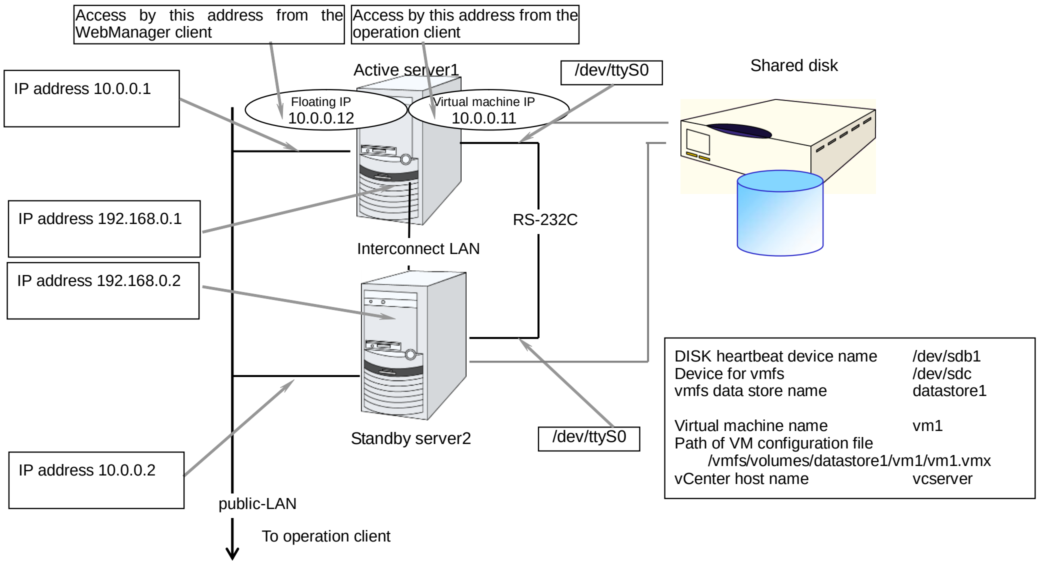
Setting up an inter-host OS cluster

Setting example of an inter-host OS cluster

When constructing an inter-host OS cluster, available control methods for virtual machine differ depending on the version of ExpressCluster on the host OS.

Table 3: Correspondence table of inter-host OS clusters construction

Type	Control method of virtual machine	ExpressCluster X2.1 for Linux	ExpressCluster X3.x for Linux
Pattern 1	Using EXEC resource and Custom monitor resource	Yes	Yes
Pattern 2	Using VM resource and VM monitor resource	No	Yes



DISK heartbeat device name /dev/sdb1
Device for vmfs /dev/sdc
vmfs data store name datastore1

Virtual machine name vm1
Path of VM configuration file
/vmfs/volumes/datastore1/vm1/vm1.vmx
vCenter host name vcserver

Example of configuration of two nodes

Target	Parameter	Value (Pattern 1)	Value (Pattern 2)
Cluster configuration	Cluster name	cluster	←
	Number of servers	2	←
	Number of failover groups	2	←
	Number of monitor resources	5	←
Heartbeat resources	Number of LAN heartbeats	2	←
	Number of COM heartbeats	1	←
	Number of disk heartbeats	1	←
First server	Server name*1	server1	←

Target	Parameter	Value (Pattern 1)	Value (Pattern 2)
information (Master server)	Interconnect IP address (Dedicated)	192.168.0.1	←
	Interconnect IP address (Backup)	10.0.0.1	←
	Public IP address	10.0.0.1	←
	COM heartbeat device	/dev/ttyS0	←
	Disk heartbeat device	/dev/sdb1	←
Second server information	Server name*1	server2	←
	Interconnect IP address (Dedicated)	192.168.0.2	←
	Interconnect IP address (Backup)	10.0.0.2	←
	Public IP address	10.0.0.2	←
	COM heartbeat device	/dev/ttyS0	←
	Disk heartbeat device	/dev/sdb1	←
Group resources for management (For the WebManager)	Type	failover	←
	Group name	ManagementGroup	←
	Startup server	All servers	←
	Number of group resources	1	←
Group resources for management *2	Type	floating ip resource	←
	Group resource name	ManagementIP	←
	IP address	10.0.0.12	←
Group resources for operation	Type	failover	Virtual machine
	Group name	failover-vm	virtualmachine1
	Startup server	All servers	←
	Number of group resources	1	←
First Group resources (Virtual machine controlling resource) *3	Type	execute resource	virtual machine resource
	Group resource name	exec-vm	vm1
	Script	Standard script	-
	Virtual machine type	-	vSphere
	Installation Destination of the Cluster Service	-	Host
	Virtual machine name	-	esx_vm1
	VM configuration file path	-	/vmfs/volumes/datastore1/vm1/vm1.vmx
	Use vCenter	-	On
	vCenter host name	-	vcserver
	Request timeout	-	120
	Virtual machine start waiting time	-	30

Target	Parameter	Value (Pattern 1)	Value (Pattern 2)
First monitor resources (Created by default)	Type	user mode monitor	←
	Monitor resource name	userw	←
Second monitor resources	Type	diskw monitor	←
	Monitor resource name	diskw1	←
	Monitor target	/dev/sdc	←
	Monitor method	TUR	←
	When an error is detected	Stop the cluster daemon and shut down the OS.	←
Third monitor resources	Type	NIC Link Up/Down monitor	←
	Monitor resource name	miiw1	←
	Monitor target	vmnic0 (Virtual switch interface of Public LAN)	←
	When an error is detected	“ManagementGroup” group’s failover	←
Fourth monitor resources	Type	NIC Link Up/Down monitor	←
	Monitor resource name	miiw2	←
	Monitor target	vmnic0 (Virtual switch interface of Public LAN)	←
	Recovery action	-	Executing failover to the recovery target
	Execute migration before failing over	-	On
	When an error is detected	“failover-vm” group’s failover	←
Fifth monitor resource (Virtual machine monitoring monitor) *3*4	Type	Custom monitor	virtual machine monitor
	Monitor resource name	genw-vm	vmw1
	Script	Standard script	-
	VM machine resource	-	vm1
	Wait time for external migration	-	15 (seconds)
	Monitor timing	Active	Always
	Target resource	exec-vm	-
	When an error is detected	“failover-vm” group’s failover	“vm1” resource’s retry (three times) then failover (once)

*1: In principle, host name is a short name of the FQDN form from which the domain name was removed.

*2: Prepare a floating IP for connecting to the WebManager. This IP enables the Web browser to access to the WebManager on the server which is running when an error occurs.

*3: Items which are hatched are the setting items about virtual machine.

*4: Automatically created after setting “virtual machine resource” when using X3.0 or later.

Setting up an inter-host OS cluster

(1) Install ExpressCluster X on each host OS as described in the *ExpressCluster X Installation and Configuration Guide*[1]. The RPM of the x86_64 version is installed here.
If using the user space monitor resource and shutdown stalling monitoring, set the monitoring method to **keepalive**.

(2) Create a cluster as described in the *ExpressCluster X Installation and Configuration Guide*, referring to “Setting example of an inter-host OS cluster” (on page 32).

(3) With regard to the virtual machine resource in step 2, configure the resource referring to “Setting up Virtual machine controlling resource” (on page 54) in this guide.

(4) With regard to the virtual machine monitor in step 2, configure the monitor referring to “Setting up Virtual machine monitoring monitor” (on page 63) in this guide.

(5) Add other settings if needed.

(6) Apply the changes. From **File**, which is on the Builder menu, select **Apply the Configuration File** or **Upload the Configuration File** to upload the configuration data.

(7) Before starting the cluster, turn off the power to the virtual machine to be subjected to failover. For a configuration in which vCenter is not used, execute the following command on Service Console of VMware ESX where the virtual machine to be subjected to failover exists so as to unregister that virtual machine.

```
# vmware-cmd -s unregister vm_path
```

(8) Start ExpressCluster and make sure that the virtual machine starts up correctly.

(9) These steps above conclude the construction of the cluster.

[1] Available from http://www.nec.com/en/global/prod/expresscluster/.

Checking the operation of an inter-host OS cluster

(1) Start the cluster by using either the WebManager or `clpcl` command.

(2) Activate the virtual machine group by using either the WebManager or `clpgrp` command. Make sure that the guest OS is running on the server on which the group is activated.

(3) Move the virtual machine group by using either the WebManager or `clpgrp` command. Make sure that the guest OS is running on the server to which the group is moved.

(4) Execute live migration of the virtual machine group by using either the WebManager or `clpgrp` command. Make sure that the guest OS is running on the server to which the group is migrated.

(5) Shut down or reboot the physical server on which the virtual machine group is running by using either the WebManager or `clpdown` command. At this time, make sure that the group has moved to another server and the guest OS is running.

(6) Shut down the guest OS, and then make sure that the virtual machine monitoring monitor detects an error and reactivates the recovery target or performs a failover. Make sure that the guest OS is restarted after the failover.

(7) Turn off the physical server from other than ExpressCluster, and make sure that the other server detects the stoppage of the server, activates the virtual machine group, and that the guest OS is restarted.

(8) In addition to the above, implement the items described in “Operation tests”, in Chapter 8, “Verifying operation” in the *ExpressCluster X Installation and Configuration Guide*, as appropriate.

Using a guest-to-host association in an inter-host OS cluster

(1) Create the inter-host OS cluster as described in “Setting up an inter-host OS cluster” (on page 36).

(2) Install ExpressCluster X SingleServerSafe in the guest OS as described in the *ExpressCluster X SingleServerSafe Installation and Configuration Guide*.

(3) Edit the cluster configuration data in the guest OS by using the ExpressCluster Builder. For details about scripts, see the Appendix.

A) Add the monitor resources to be monitored (such as the pid, appli, and oracle monitor resources).

B) Enable **Execute Script before Final Action** for the monitor resources, and then select **Settings**.

C) If the guest OS is Linux, click **Replace** to replace the contents of `preaction.sh` with those of `vmpreaction.sh`.
If the guest OS is Windows, click **Replace** to replace the contents of `preaction.bat` with those of `vmpreaction.bat`.

D) Specify other settings as appropriate.

(4) Upload the configuration data created using the ExpressCluster Builder. From **File**, which is on the Builder menu, select **Apply the Configuration File** or **Upload the Configuration File** to upload the configuration data.

Setting up an inter-guest OS cluster

Setting up an inter-guest OS cluster

(1) If you have not created a virtual machine, create one as described in “Creating virtual machines” (on page 28).

(2) Install a guest OS supported by ExpressCluster on the virtual machine.

(3) Install ExpressCluster in the guest OS as described in the *ExpressCluster X Installation and Configuration Guide*.

(4) Set up a cluster by using the ExpressCluster Builder, following the *ExpressCluster X Installation and Configuration Guide*.

(5) Apply the changes. From **File**, which is on the Builder menu, select **Apply the Configuration File** or **Upload the Configuration File** to upload the configuration data.

Checking the operation of an inter-guest OS cluster

(1) Activate the cluster by using either the WebManager or `clpcl` command.

(2) Move the failover group by using either the WebManager or `clpgrp` command. Make sure that the failover group is running on the server to which the failover group is moved by using either the WebManager or `clpstat` command.

(3) Shut down or reboot the virtual machine on which the failover group is running by using either the WebManager or `clpdown` command. At this time, make sure that the failover group is running on another server by using either the WebManager or `clpstat` command.

(4) Turn off the physical server from other than ExpressCluster, and make sure that the other server detects the stoppage of the server and activates the failover group by using either the WebManager or `clpstat` command.

(5) In addition to the above, implement the items described in “Operation tests”, in Chapter 8, “Verifying operation” in the *ExpressCluster X Installation and Configuration Guide*, as appropriate.

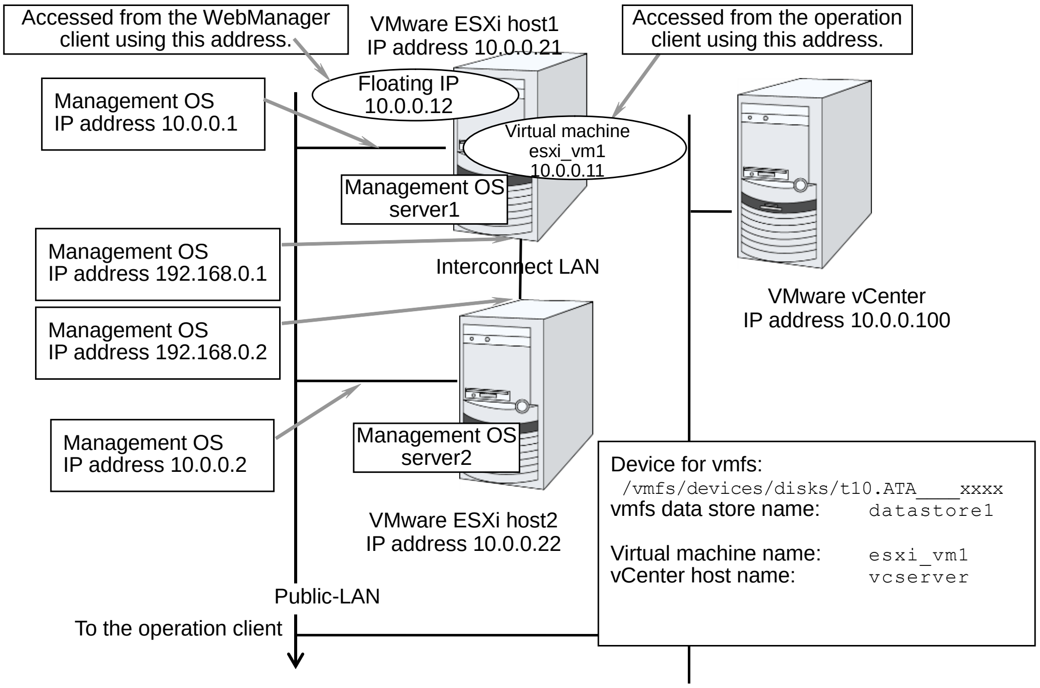
Setting up inter-management OS clusters

Example of setting an inter-management OS cluster

The inter-management OS cluster can be used only with ExpressCluster X 3.1 and later.

Table 4: Correspondence table for inter-management OS cluster construction

Virtual machine control method	**ExpressCluster X 3.1 for Linux**
Using the EXEC resource and custom monitor resource	No
Using the virtual machine resource and virtual machine monitor resource	Yes



Example of configuration of two nodes

Target	**Parameter**	**Value**
Cluster configuration	Cluster name	cluster
	Number of servers	2
	Number of failover groups	2
	Number of monitor resources	5
Heartbeat resources	Number of LAN heartbeats	2
First server information (Master server)	Server name*1	server1
	Interconnect IP address (Dedicated)	192.168.0.1
	Interconnect IP address (Backup)	10.0.0.1

Target	Parameter	Value
Second server information	Server name*1	server2
	Interconnect IP address (Dedicated)	192.168.0.2
	Interconnect IP address (Backup)	10.0.0.2
Group resources for management (For the WebManager)	Type	failover
	Group name	ManagementGroup
	Startup server	All servers
	Number of group resources	1
Group resources for management *2	Type	floating ip resource
	Group resource name	ManagementIP
	IP address	10.0.0.12
Group resources for operation	Type	Virtual machine
	Group name	virtualmachine1
	Startup server	All servers
	Number of group resources	1
First Group resources (Virtual machine controlling resource) *3	Type	virtual machine resource
	Group resource name	vm1
	Virtual machine type	vSphere
	Installation Destination of the Cluster Service	Guest
	Virtual machine name	esxi_vm1
	Data store name	datastore1
	VM configuration file path	-
	Host IP address (common)	10.0.0.21
	Host IP address (Server separate setting: server1)	10.0.0.21
	Host IP address (Server separate setting: server2)	10.0.0.22
	Use vCenter	On
	vCenter host name	vcserver
	Request timeout	120
	Virtual machine start waiting time	30
First monitor resources (Created by default)	Type	user mode monitor
	Monitor resource name	userw
Second monitor resources (Host: Dist monitoring monitor) *4	Type	custom monitor
	Monitor resource name	genw_disk
	Script	clphostmon_wrap.sh
	Monitor timing	Always
	Set to clphostmon.pl Monitor target	/vmfs/devices/disks/t10.ATA____xxxx

Target	Parameter	Value
	Recovery action	Execute only the final action
	Recovery script	`stdnhost_wrap.sh`
	Execute script before final action	On
Third monitor resources (Host: NIC monitoring monitor) *4	Type	custom monitor
	Monitor resource name	genw_nics1
	Script	clphostmon_wrap.sh
	Monitor timing	Always
	Set to `clphostmon.pl` Monitor target	vmnic0 (Virtual switch interface of Public LAN)
	Recovery action	Executing failover to the recovery target
	When an error is detected	“ManagementGroup” group’s failover
Fourth monitor resources (Host: NIC monitoring monitor) *4	Type	custom monitor
	Monitor resource name	genw_nics2
	Script	clphostmon_wrap.sh
	Monitor timing	Always
	Set to clphostmon.pl Monitor target	vmnic0 (Virtual switch interface of Public LAN)
	Recovery action	Executing failover to the recovery target
	Execute migration before failing over	On
	When an error is detected	“virtualmachine1” group’s failover
Fifth monitor resource (Virtual machine monitoring monitor) *3*5	Type	virtual machine monitor
	Monitor resource name	vmw1
	VM machine resource	vm1
	Wait time for external migration	15 (seconds)
	Monitor timing	Always
	When an error is detected	“vm1” resource’s retry (three times) then failover (once)

*1: In principle, “host name” is a shortened version of the FQDN form from which the domain name has been removed.

*2: A floating IP is provided which is connected to WebManager. With this IP, upon the occurrence of a failure, access can be made from the Web browser to WebManager to be executed by the active server.

*3: The hatched items are the setting items for the virtual machine.

*4: Items related to host monitoring that can be set only when vMA is in use.

*5: Automatically created when “virtual machine resource” is set.

Constructing an inter-management OS cluster

(1) Install ExpressCluster X 3.1 for Linux on each management OS as described in the *ExpressCluster X Installation and Configuration Guide*. If vMA is in use, the x86_64 version of the RPM is to be installed.

(2) Construct the cluster while referring to the example in “Example of setting an inter-management OS cluster” (on page 41) in the *ExpressCluster X Installation and Configuration Guide*.

(3) With regard to the virtual machine resource described in step (2), configure the resource as described in “When using the virtual machine resource under the management OS (X3.1 or later)” (on page 60) of this guide.

(4) With regard to the virtual machine monitor described in step (2), configure the resource as described in “Setting up Virtual machine monitoring monitor” (on page 63) of this guide.

(5) With regard to the host monitoring monitor described in step (2), configure the monitor as described in “Setting the host monitoring monitor” (on page 46) of this guide.

(6) Add other settings as needed.

(7) Apply the changes. From **File** of the Builder menu, select **Apply the Configuration File** or **Upload the Configuration File** to upload the configuration data.

(8) Start ExpressCluster and ensure that the virtual machine starts up correctly.

(9) The above steps conclude the construction of the cluster.

Checking the operation of an inter-management OS cluster

(1) Start the cluster by using the WebManager or `clpcl` command.

(2) Activate the virtual machine group by using the WebManager or `clpgrp` command. Check that the guest OS is running on the server on which the group is active.

(3) Move the virtual machine group by using the WebManager or `clpgrp` command. Check that the guest OS is running on the server to which the group is moved.

(4) Execute live migration of the virtual machine group by using the WebManager or `clpgrp` command. Check that the guest OS is running on the server to which the group is moved.

(5) Shut down the guest OS, and then check that the monitor for virtual machine monitoring detects any errors and either reactivates the recovery target or performs failover. Moreover, make sure that the guest OS is restarted after failover.

(6) Shut down the physical server from VMware vCenter, and then check that the other server detects the stoppage of the remote server, activates the virtual machine group, and that the guest OS is restarted.

(7) In addition to the above, implement the items described in “Operation tests", in Chapter 8, “Verifying operation” in the *ExpressCluster X Installation and Configuration Guide* as appropriate.

Setting the host monitoring monitor

To use this function, vMA must be used as the management OS.

(1) Execute the following command on each vMA console to register VMware ESXi to be monitored.

```
# vifp addserver VMware ESXi’s hostname
root@esxi_host1's password: login password for VMware ESXi
```

Execute the following command to check that VMware ESXi has been registered normally.

```
# vifp listservers
esxi_host1 ESXi
```

(2) Obtain hardware information from the target VMware ESXi.
Execute the following command to specify the target VMware ESXi as the subsequent command execution target.

```
# vifptarget --set registerd VMware ESXi’s hostname
```

Execute the following command to obtain the storage information. From the listed storage, record path **Devfs Path** of the device file of the storage to be monitored.

```
#[esxi_host1]# vicfg-scsidevs --list
t10.ATA_____xxxx
   Device Type: disk
   Size: 238419 MB
   Display Name: Local ATA Disk
   Plugin: NMP
   Console Device: /vmfs/devices/disks/t10.ATA_____xxxx
   Devfs Path: /vmfs/devices/disks/t10.ATA_____xxxx
   Vendor: ATA      Model: WDC WD2502ABYS-0  Revis: 02.0
   SCSI Level: 5  Is Pseudo:  Status:
   Is RDM Capable:  Is Removable:
   ……
```

Execute the following command to obtain the network adapter information. Record device name **Name** for the network adaptor to be monitored.

```
#[esxi_host1]# vicfg-nics --list
Name      PCI      Driver  Link  Speed    ……
vmnic0    01:00.0  igb     Up    100Mbps  ……
vmnic1    01:00.1  igb     Up    1000Mbps ……
```

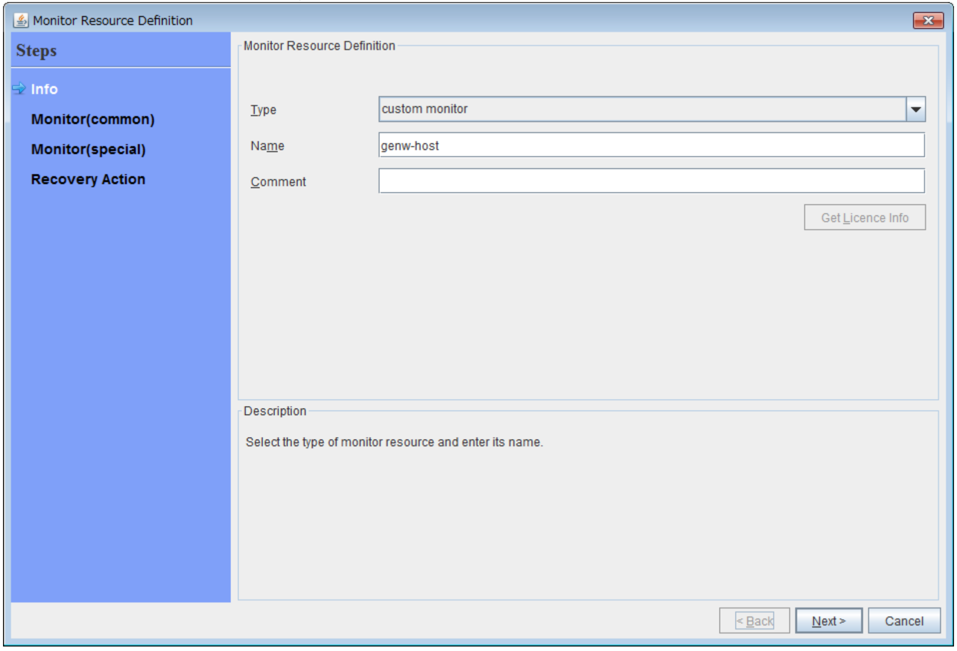
Finally, execute the following command to release VMware ESXi from the command execution target.

```
# vifptarget –clear
```

(3) Edit “Sample Script clphostmon.pl” (on page 85) of this guide based on the information obtained in the previous step.

(4) From the ExpressCluster Builder window, right-click **Monitors** in the tree view and then click **Add monitor resource**.

(5) Register the custom monitor resource used for virtual machine monitoring. From **Type**, select **custom monitor resource**. Set any name (`genw-host`) for the custom monitor resource, and then click **Next**.



Monitor Resource Definition

Type: custom monitor

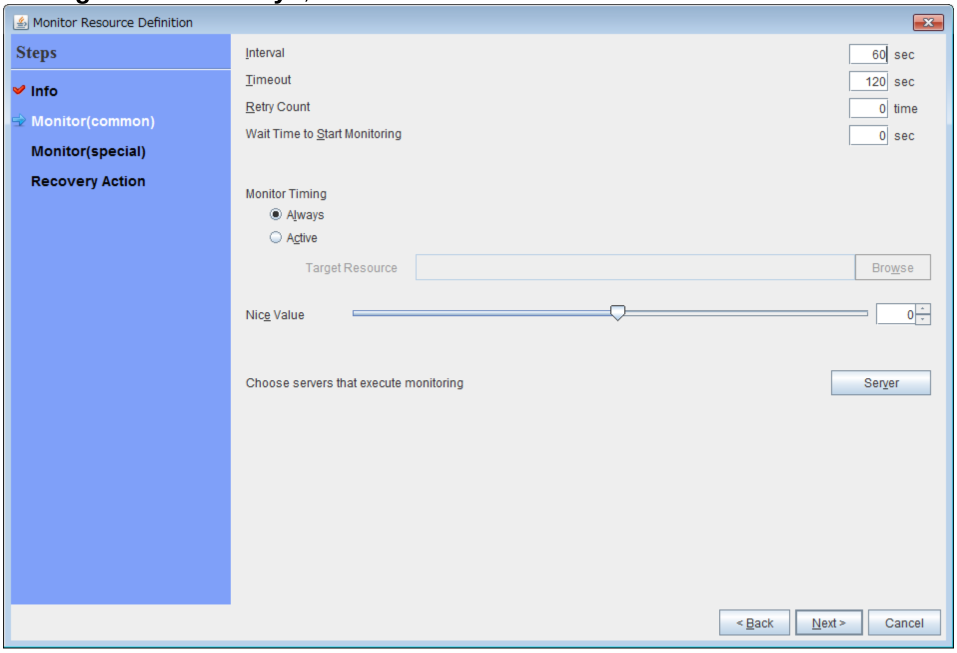
Name: genw-host

Comment:

Description

Select the type of monitor resource and enter its name.

(6) To avoid a misdetection resulting from a host connection error (busy or the like), it is recommended that **Retry Count** be set to one or more. Make sure that **Monitor Timing** indicates **Always**, and then click **Next**.



Interval: 60 sec

Timeout: 120 sec

Retry Count: 0 time

Wait Time to Start Monitoring: 0 sec

Monitor Timing

● Always

○ Active

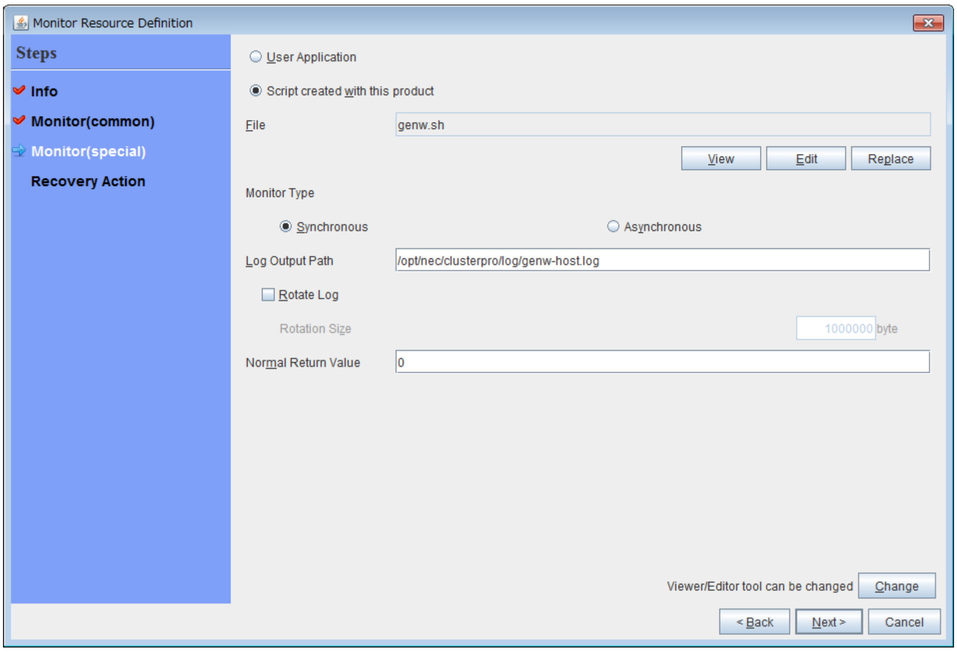
Target Resource:

Nice Value: 0

Choose servers that execute monitoring

(7) Set the script for host monitoring. Store “Sample Script `clphostmon.pl`” (on page 85) of this guide in the same path for each vMA. (*`/opt/nec/clusterpro/work/clphostmon/clphostmon.pl`* or the like)

(8) Click **Replace** to replace `genw.sh` with “Sample Script `clphostmon_wrap.sh`” (on page 84) of this guide, make sure that **Monitor Type** is set to **Synchronous**, and then click **Next**.



○ User Application

● Script created with this product

File: genw.sh

Monitor Type

● Synchronous ○ Asynchronous

Log Output Path: /opt/nec/clusterpro/log/genw-host.log

□ Rotate Log

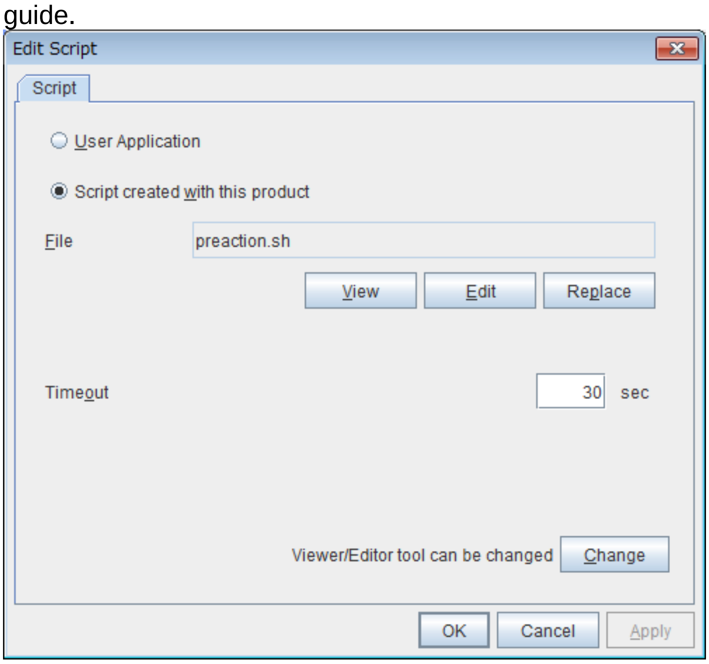
Rotation Size: 1000000 byte

Normal Return Value: 0

(9) Set **Recovery Action**. See (9) to (12) for the procedure for setting the second monitor resource (host: disk monitoring monitor) of “Example of setting an inter-management OS cluster” (on page 41).
See (13) for the procedure for setting the third monitor resource (host: NIC monitoring monitor).
See (14) for the procedure for setting the fourth monitor resource (host: NIC monitoring monitor).

(10) The following explains the procedure for setting **Recovery Action** for the second monitor resource. Set **Recovery Action** so that the host to which the management OS belongs is shut down upon error detection. Store “Sample Script `shutdownhost.pl`” (on page 91) of this guide in the same path of each vMA. (*`/opt/nec/clusterpro/work/shutdownhost/shutdownhost.pl`* or the like)

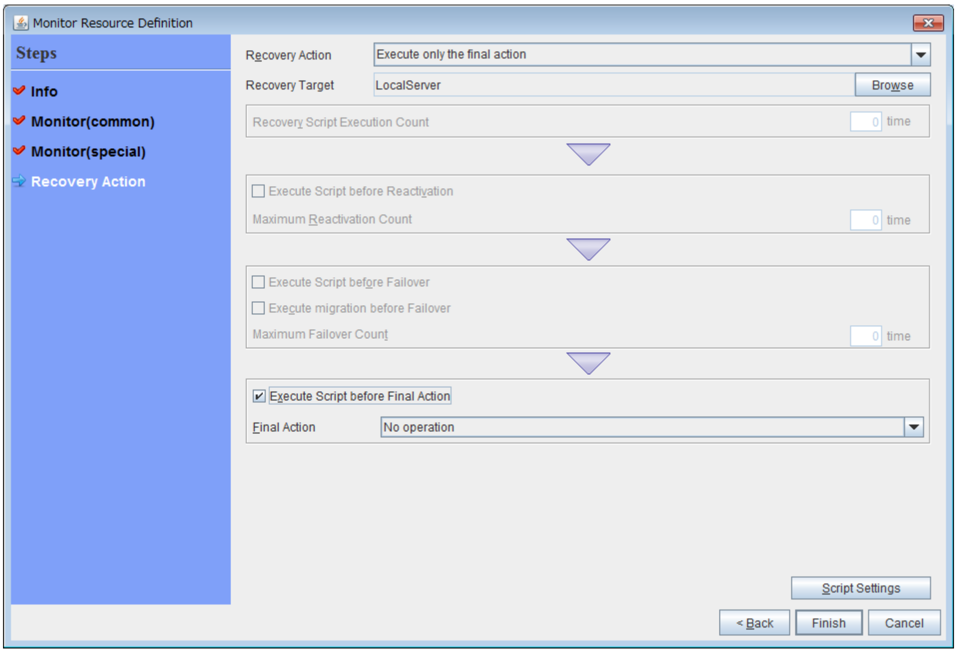
(11) Click **Script Settings** to open **Edit Script**. Click **Replace** to replace `preaction.sh` with “Sample Script `stdnhost_wrap.sh`” (on page 90) of this guide.



File: preaction.sh

Timeout: 30 sec

(12) Set **Recovery Action** to **Execute only the final action** and then select **LocalServer** as **Recovery Target**. Check **Execute Script before Final Action** and then click **Finish**.



Monitor Resource Definition

Steps
- ✓ Info
- ✓ Monitor(common)
- ✓ Monitor(special)
- Recovery Action

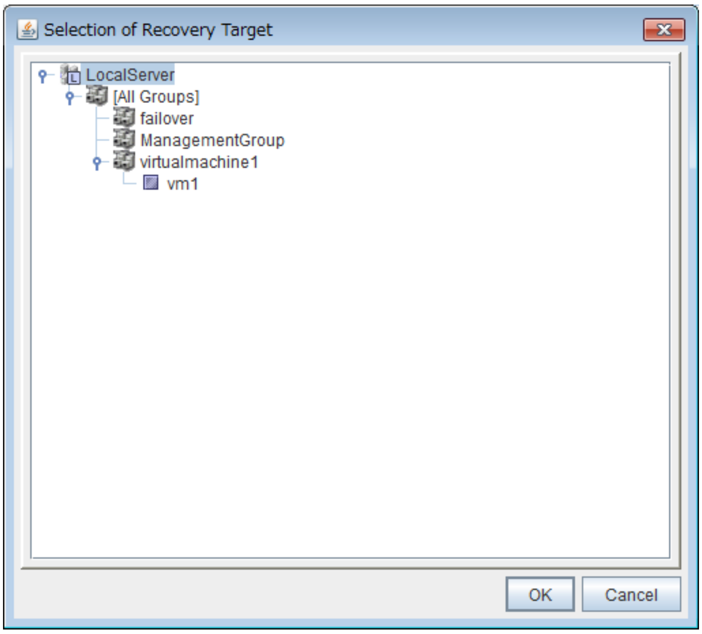
Recovery Action: Execute only the final action
Recovery Target: LocalServer [Browse]

Recovery Script Execution Count 0 time

□ Execute Script before Reactivation
Maximum Reactivation Count 0 time

□ Execute Script before Failover
□ Execute migration before Failover
Maximum Failover Count 0 time

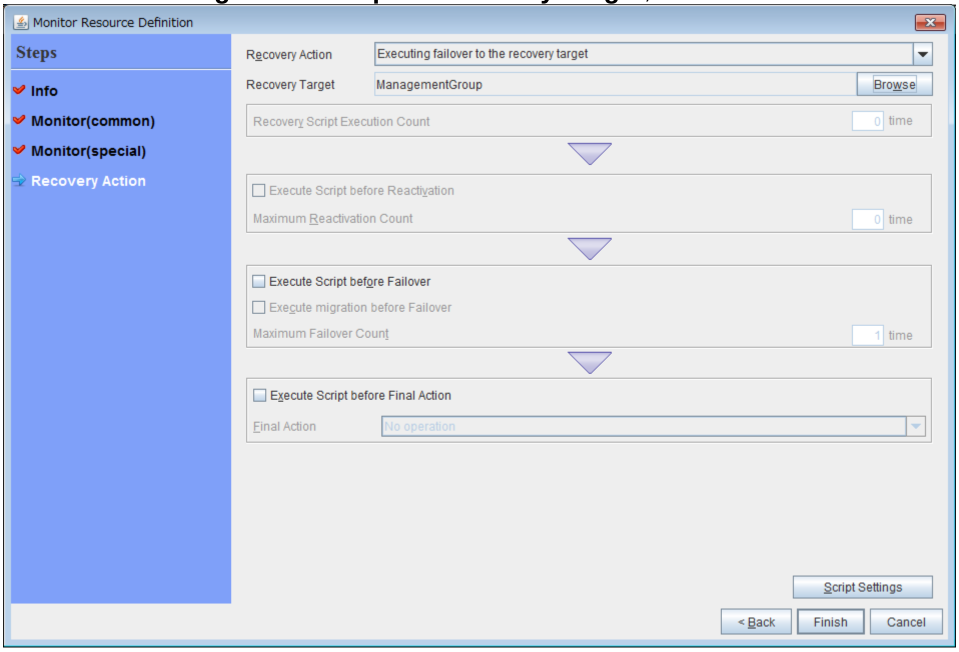
☑ Execute Script before Final Action
Final Action: No operation



Selection of Recovery Target

- LocalServer
  - [All Groups]
    - failover
    - ManagementGroup
    - virtualmachine1
      - vm1

(13) The following explains the procedure for setting **Recovery Action** for the third monitor resource. Select **Executing failover to the recovery target** for **Recovery Action** and **ManagementGroup** for **Recovery Target**, and then click **Finish**.



Monitor Resource Definition

Steps
- Info
- Monitor(common)
- Monitor(special)
- Recovery Action

Recovery Action: Executing failover to the recovery target

Recovery Target: ManagementGroup [Browse]

Recovery Script Execution Count: 0 time

□ Execute Script before Reactivation

Maximum Reactivation Count: 0 time

□ Execute Script before Failover

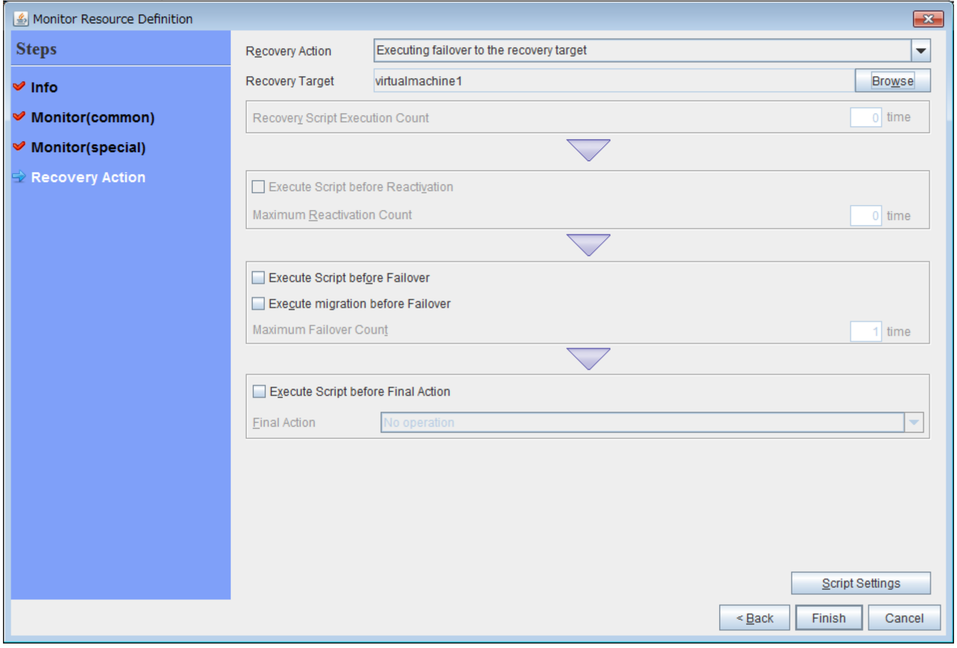
□ Execute migration before Failover

Maximum Failover Count: 1 time

□ Execute Script before Final Action

Final Action: No operation

(14) The following explains the procedure for setting **Recovery Action** for the fourth monitor resource. Select **Executing failover to the recovery target** for **Recovery Action** and **virtualmachine1** for **Recovery Target**. Next, check **Execute migration before failing over** and then click **Finish**.



Monitor Resource Definition

Steps
- Info
- Monitor(common)
- Monitor(special)
- Recovery Action

Recovery Action: Executing failover to the recovery target

Recovery Target: virtualmachine1 [Browse]

Recovery Script Execution Count: 0 time

□ Execute Script before Reactivation

Maximum Reactivation Count: 0 time

□ Execute Script before Failover

□ Execute migration before Failover

Maximum Failover Count: 1 time

□ Execute Script before Final Action

Final Action: No operation

Setting up a physical server-to-virtual machine cluster

Setting up a physical server-to-virtual machine cluster

(1) If you have not created a virtual machine, create one as described in “Creating virtual machines” (on page 28).

(2) Install a guest OS supported by ExpressCluster on the virtual machine.

(3) Install ExpressCluster on the physical server OS and guest OS as described in the *ExpressCluster X Installation and Configuration Guide*.

(4) Set up a cluster by using the ExpressCluster Builder, following the *ExpressCluster X Installation and Configuration Guide*.

(5) Apply the changes. From **File**, which is on the Builder menu, select **Apply the Configuration File** or **Upload the Configuration File** to upload the configuration data.

Checking the operation of a physical server-to-virtual machine cluster

(1) Activate the cluster by using either the WebManager or `clpcl` command.

(2) Move the failover group by using either the WebManager or `clpgrp` command. Make sure that the failover group is running on the server to which the failover group is moved by using either the WebManager or `clpstat` command.

(3) Shut down or reboot the server on which the failover group is running by using either the WebManager or `clpdown` command. At this time, make sure that the failover group is running on another server by using either the WebManager or `clpstat` command.

(4) Turn off the physical server from other than ExpressCluster, and make sure that the other server detects the stoppage of the server and activates the failover group by using either the WebManager or `clpstat` command.

(5) In addition to the above, implement the items described in “Operation tests”, in Chapter 8, “Verifying operation” in the *ExpressCluster X Installation and Configuration Guide*, as appropriate.

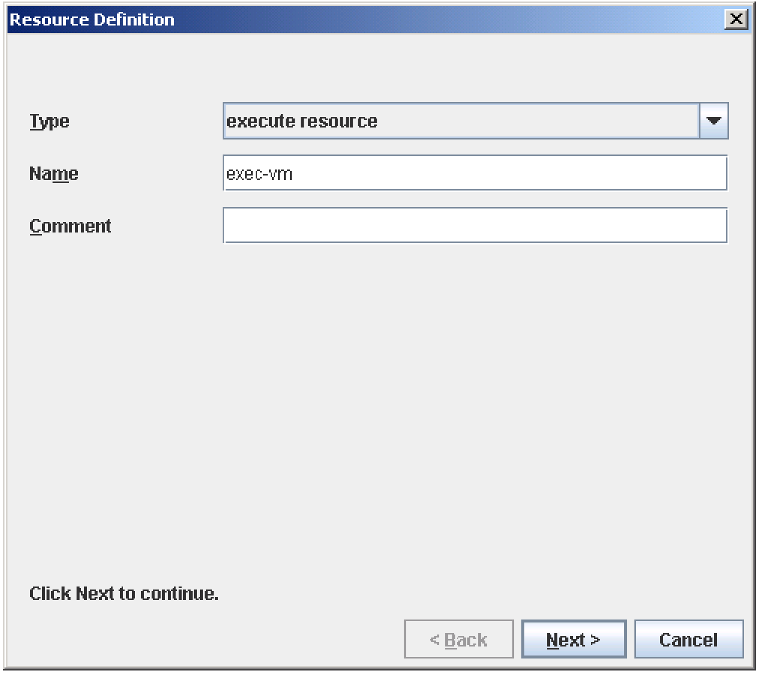
Setting up Virtual machine controlling resource

When using EXEC resources

* If you are using ExpressCluster X 3.0 or later, we recommend that you use the virtual machine resource.

(1) On the Expresscluster Builder window, right-click the failover group name (`failover-vm`) on the tree view, then click **Add resource**.

(2) Register EXEC resource for virtual machine control. Select **execute resource** from **Type**. Set the name (`exec-vm`) to exec resource, and then click **Next**.



Resource Definition

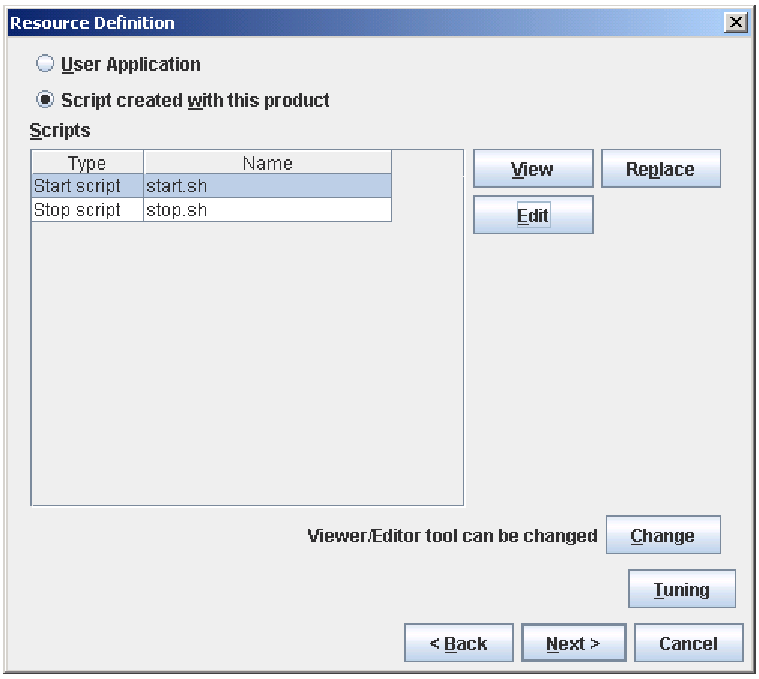
Type: execute resource

Name: exec-vm

Comment:

Click Next to continue.

(3) Select **Start script** and click **Replace**. File selection window is displayed. Select the sample script `vmpower.start.pl` (on page 71) in this guide, and then replace `start.sh`.



Resource Definition

○ User Application

● Script created with this product

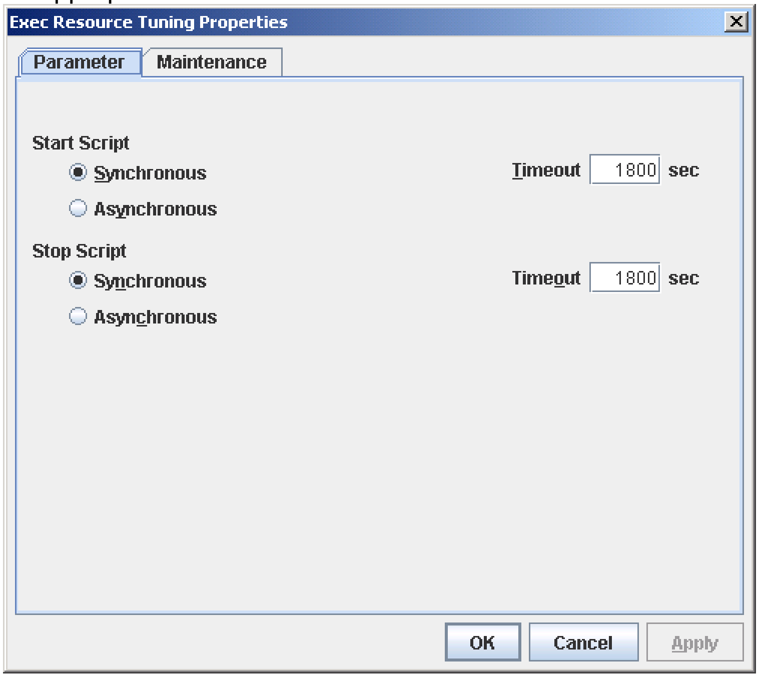
Scripts

Type	Name
Start script	start.sh
Stop script	stop.sh

Viewer/Editor tool can be changed

(4) Select **Stop script** and click **Replace**. File selection window is displayed. Select the sample script `vmpower.stop.pl` (on page 75) in this guide, and then replace `stop.sh`.

(5) Click **Tuning** and change activation of non-global zone and timeout of deactivation, as appropriate. Click **Next**.



Exec Resource Tuning Properties

Start Script

● Synchronous

○ Asynchronous

Timeout 1800 sec

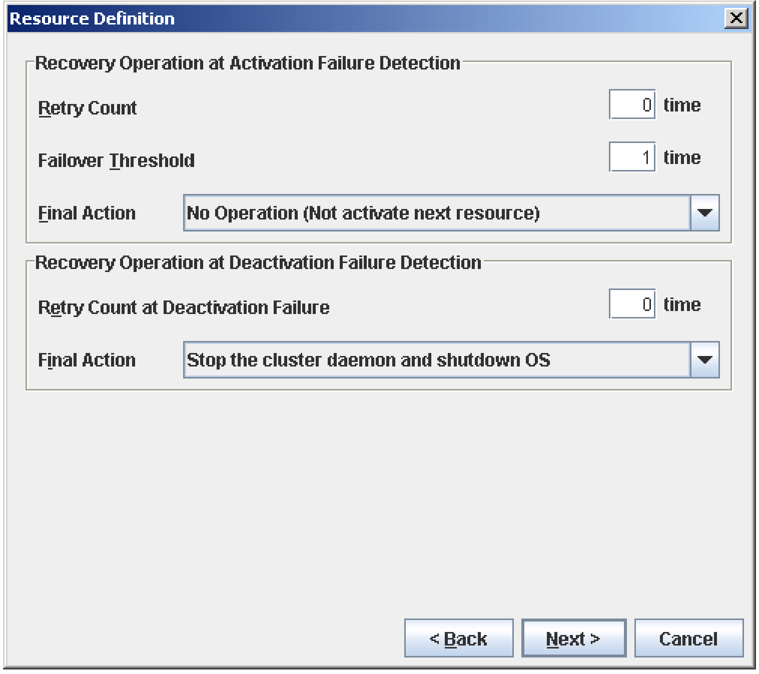
Stop Script

● Synchronous

○ Asynchronous

Timeout 1800 sec

(6) On the **Recovery Operation at Activation Failure Detection** and **Recovery Operation at Deactivation Failure Detection** window, click **Next**.



Recovery Operation at Activation Failure Detection

Retry Count 0 time

Failover Threshold 1 time

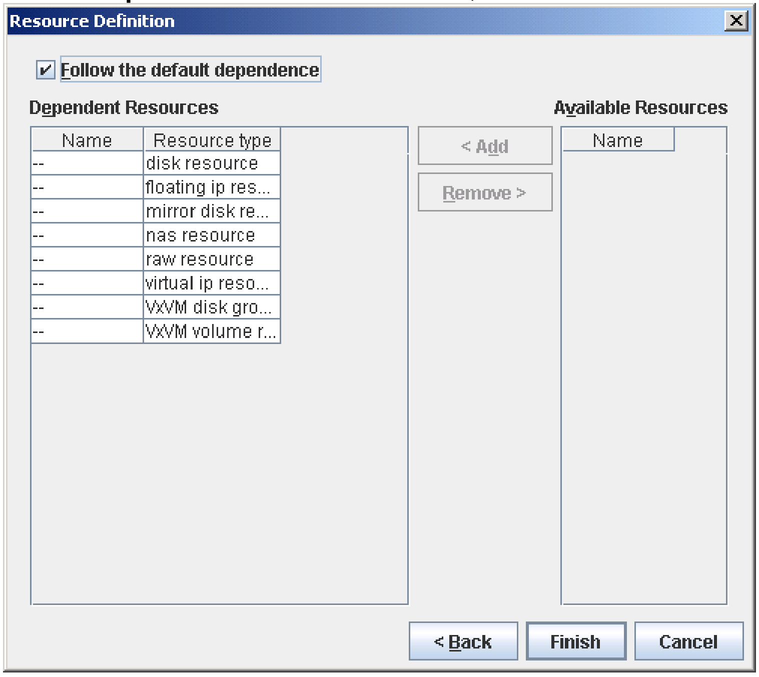
Final Action No Operation (Not activate next resource)

Recovery Operation at Deactivation Failure Detection

Retry Count at Deactivation Failure 0 time

Final Action Stop the cluster daemon and shutdown OS

(7) On the **Dependent Resources** window, click **Finish**.



☑ Follow the default dependence

Dependent Resources

Name	Resource type
--	disk resource
--	floating ip res...
--	mirror disk re...
--	nas resource
--	raw resource
--	virtual ip reso...
--	VXVM disk gro...
--	VXVM volume r...

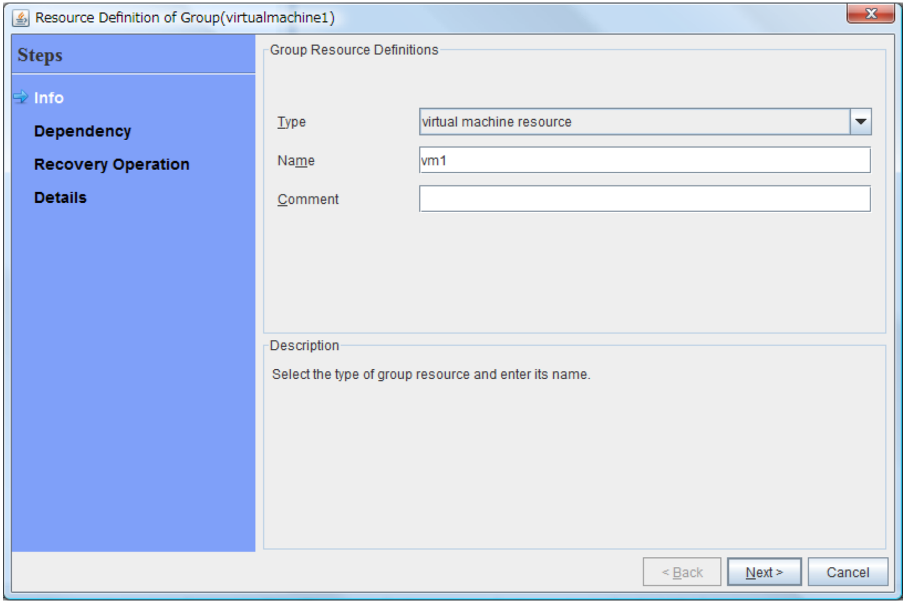
Available Resources

Name

When using VM resources (X3.0 or later)

(1) On the Expresscluster Builder window, right-click the virtual machine group name (`virtualmachine1`) on the tree view, and then click **Add resource**.

(2) Register VM resource. Select **virtual machine resource** from **Type**. Enter the name, and then click **Next**.



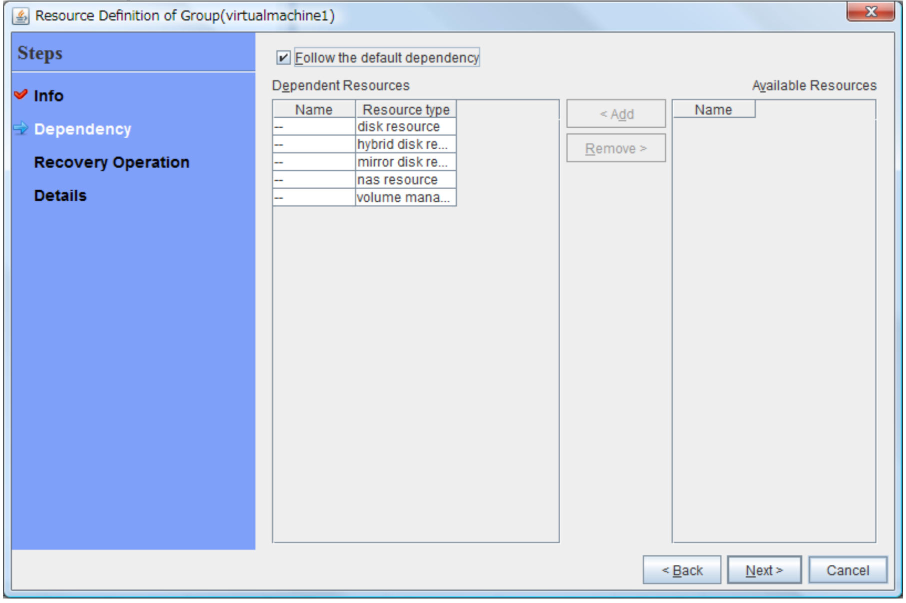
Type: virtual machine resource

Name: vm1

Comment:

Select the type of group resource and enter its name.

(3) On the **Dependent Resources** window, click **Next**.



☑ Follow the default dependency

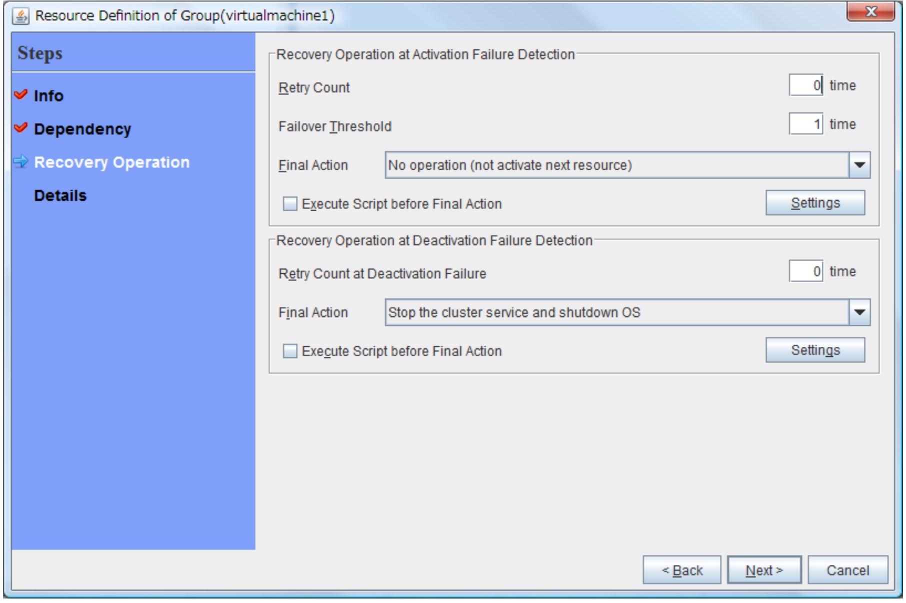
Dependent Resources

Name	Resource type
--	disk resource
--	hybrid disk re...
--	mirror disk re...
--	nas resource
--	volume mana...

Available Resources

Name

(4) On the **Recovery Operation at Activation Failure Detection** and **Recovery Operation at Deactivation Failure Detection** window, click **Next**.



Resource Definition of Group(virtualmachine1)

Steps
- ✓ Info
- ✓ Dependency
- Recovery Operation
- Details

Recovery Operation at Activation Failure Detection

Retry Count 0 time

Failover Threshold 1 time

Final Action: No operation (not activate next resource)

□ Execute Script before Final Action [Settings]

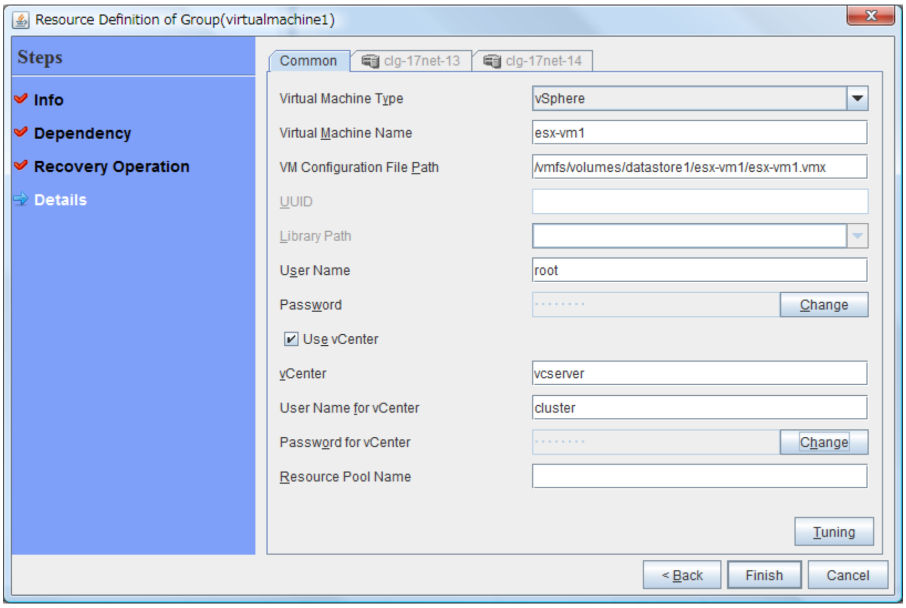
Recovery Operation at Deactivation Failure Detection

Retry Count at Deactivation Failure 0 time

Final Action: Stop the cluster service and shutdown OS

□ Execute Script before Final Action [Settings]

(5) Select **vSphere** for **Virtual Machine Type**, and set **Virtual Machine Name**, **VM Configuration File Path**, **User Name** and **Password**.
For X 3.1 and later, select **Host** for **Installation Destination of the Cluster Service**.
When not using vCenter, make sure that the **Use vCenter** checkbox is cleared.
When using vCenter, select the **Use vCenter** check box, and enter **vCenter**, **User Name for vCenter** and **Password for vCenter**.



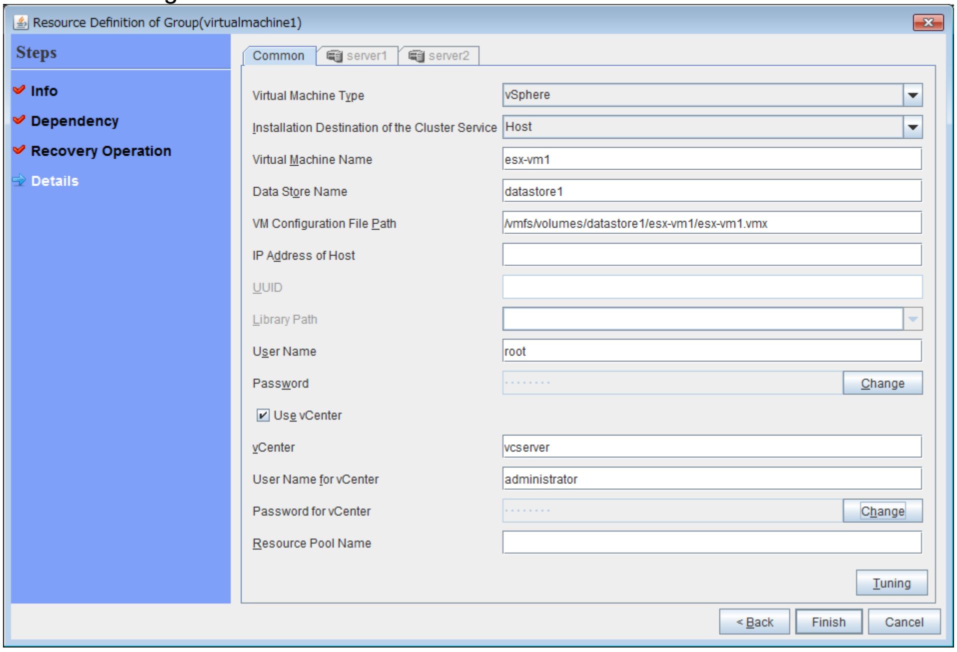
Resource Definition of Group(virtualmachine1)

Steps
- ✓ Info
- ✓ Dependency
- ✓ Recovery Operation
- Details

Common | clg-17net-13 | clg-17net-14

Virtual Machine Type	vSphere
Virtual Machine Name	esx-vm1
VM Configuration File Path	/vmfs/volumes/datastore1/esx-vm1/esx-vm1.vmx
UUID	
Library Path	
User Name	root
Password	········ [Change]
☑ Use vCenter	
vCenter	vcserver
User Name for vCenter	cluster
Password for vCenter	········ [Change]
Resource Pool Name	

* The following window is for X3.1 and later.



Resource Definition of Group(virtualmachine1)

Virtual Machine Type: vSphere

Installation Destination of the Cluster Service: Host

Virtual Machine Name: esx-vm1

Data Store Name: datastore1

VM Configuration File Path: /vmfs/volumes/datastore1/esx-vm1/esx-vm1.vmx

IP Address of Host:

UUID:

Library Path:

User Name: root

Password: ········ Change

☑ Use vCenter

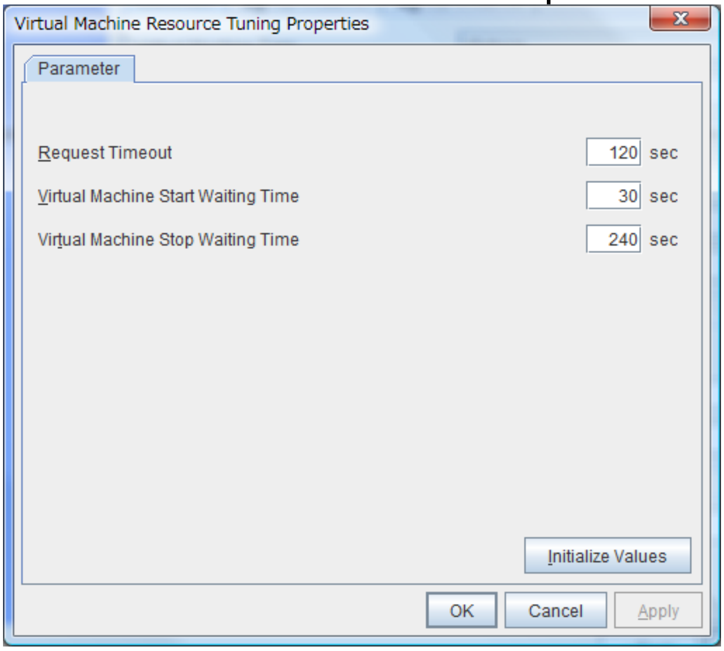
vCenter: vcserver

User Name for vCenter: administrator

Password for vCenter: ········ Change

Resource Pool Name:

(6) Click **Tuning** and display tuning properties. Enter **Request Timeout**, **Wait Time To Start Virtual Machine** and **Wait Time To Stop Virtual Machine**, and then click **OK.**



Virtual Machine Resource Tuning Properties

Request Timeout: 120 sec

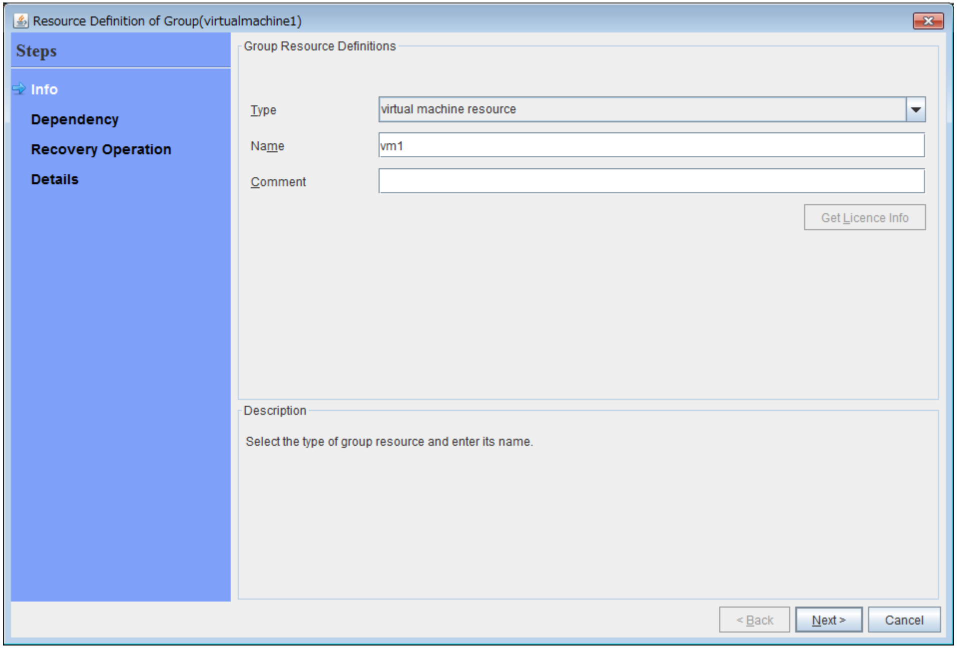
Virtual Machine Start Waiting Time: 30 sec

Virtual Machine Stop Waiting Time: 240 sec

When using the virtual machine resource under the management OS (X3.1 or later)

(1) From the ExpressCluster Builder window, right-click the virtual machine group name (`virtualmachine1`) in the tree view and then click **Add Resource**.

(2) Register the virtual machine resource. Select **virtual machine resource** for **Type**, enter **Name**, and then click **Next**.

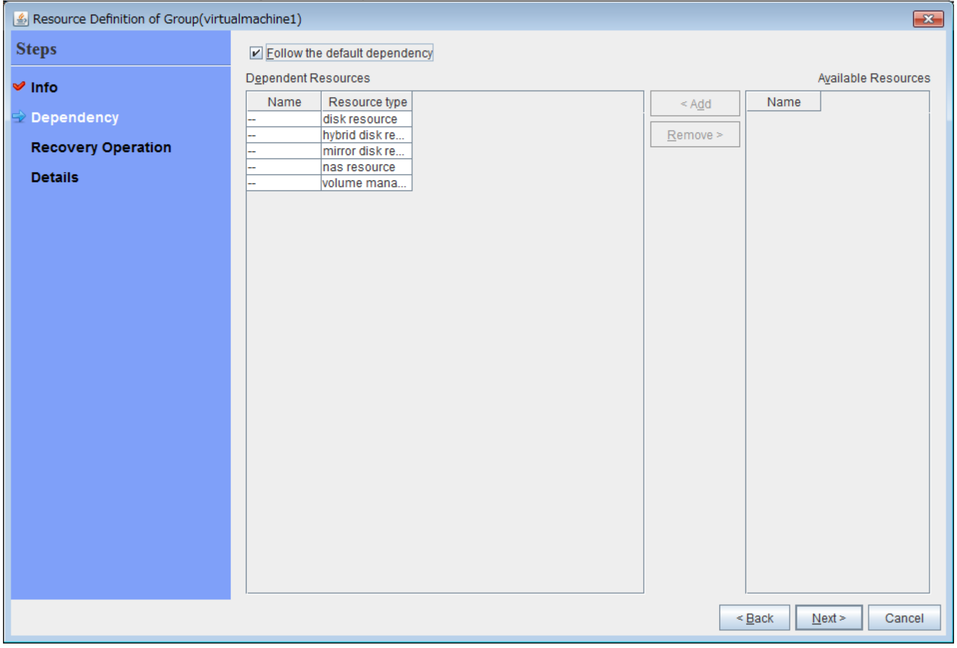


Type: virtual machine resource

Name: vm1

Select the type of group resource and enter its name.

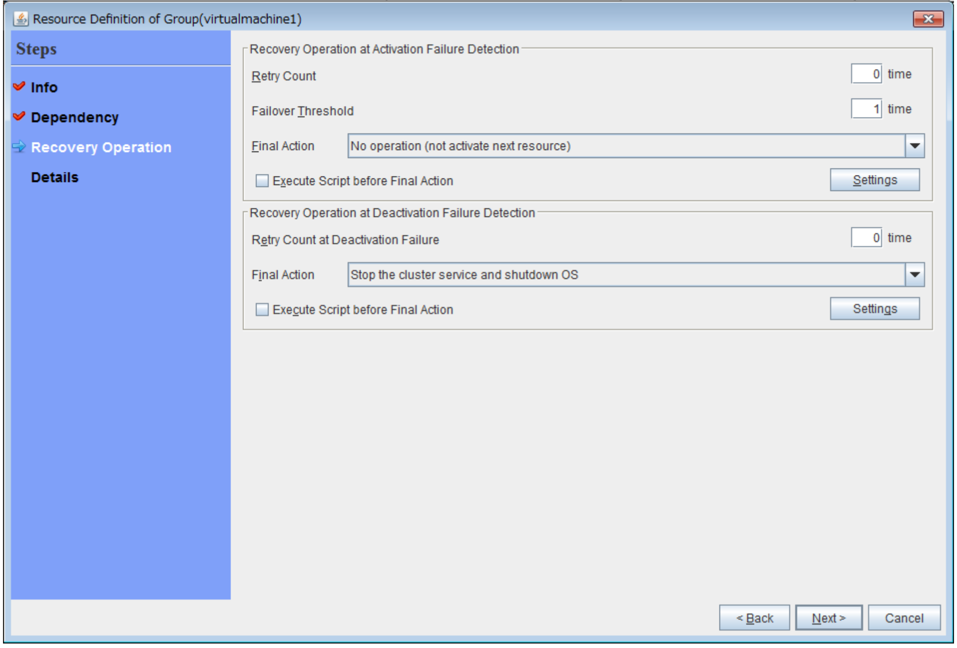
(3) From the resource dependency window, click **Next**.



☑ Follow the default dependency

Name	Resource type
--	disk resource
--	hybrid disk re...
--	mirror disk re...
--	nas resource
--	volume mana...

(4) Set the recovery operation window for an activation/deactivation error, and then click **Next**.
If OS shutdown is selected, the management OS, rather than the host, is shut down. To shut down the host (VMware ESXi) to which the management OS belongs, check **Execute Script before Final Action** and then click **Settings**.
* The host can be shut down only when vMA is being used as the management OS.



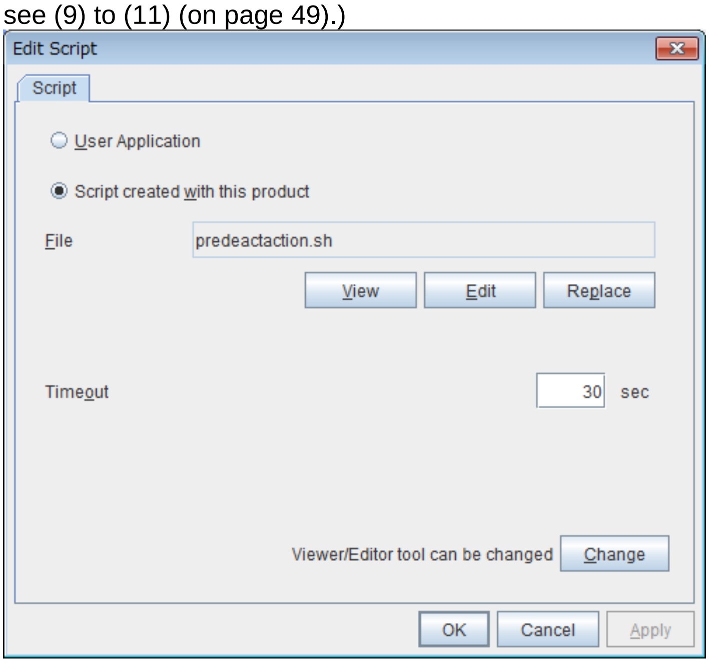
Resource Definition of Group(virtualmachine1)

Steps: Info, Dependency, Recovery Operation, Details

Recovery Operation at Activation Failure Detection
- Retry Count: 0 time
- Failover Threshold: 1 time
- Final Action: No operation (not activate next resource)
- □ Execute Script before Final Action [Settings]

Recovery Operation at Deactivation Failure Detection
- Retry Count at Deactivation Failure: 0 time
- Final Action: Stop the cluster service and shutdown OS
- □ Execute Script before Final Action [Settings]

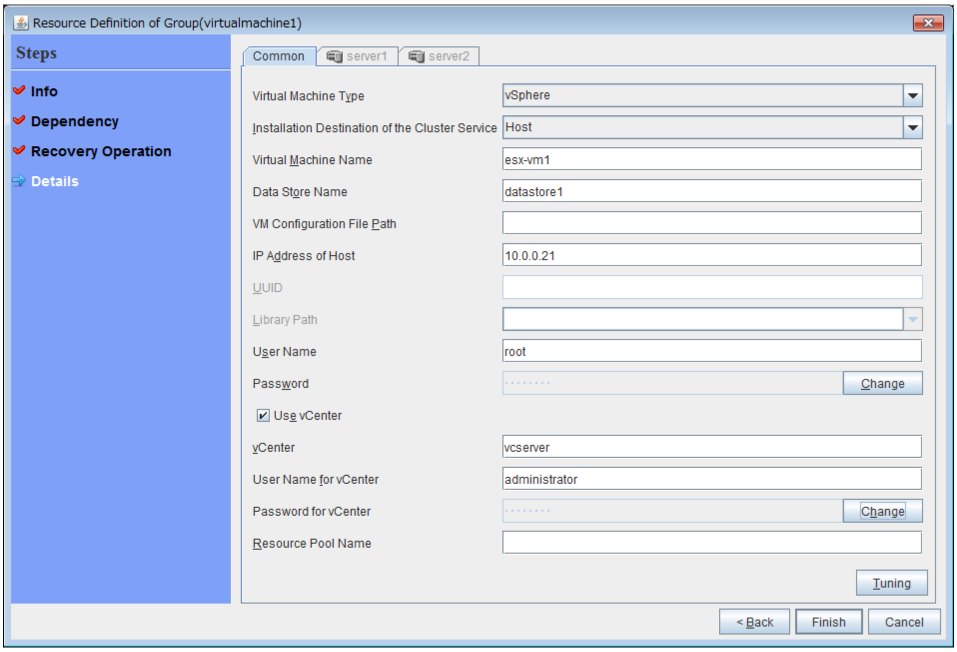
Click **Replace** in the **Edit Script** window and then replace `preactaction.sh` and `predeactaction.sh` with “Sample Script `stdnhost_wrap.sh`” (on page 90) of this guide. It is necessary to register the host in each vMA (see (1) (on page 46)) beforehand and to store `shutdownhost.pl` (on page 3) in the same path. (Also, see (9) to (11) (on page 49).)



Edit Script

Script
- ○ User Application
- ● Script created with this product
- File: predeactaction.sh
- Timeout: 30 sec

(5) Select **vSphere** for **Virtual Machine Type** and **Guest** for **Installation Destination of the Cluster Service**.
From the **Common** tab, enter **Virtual Machine Name**, **Data Store Name**, **IP Address of Host**, **vCenter host name**, **User name for vCenter**, and **Password for vCenter**.



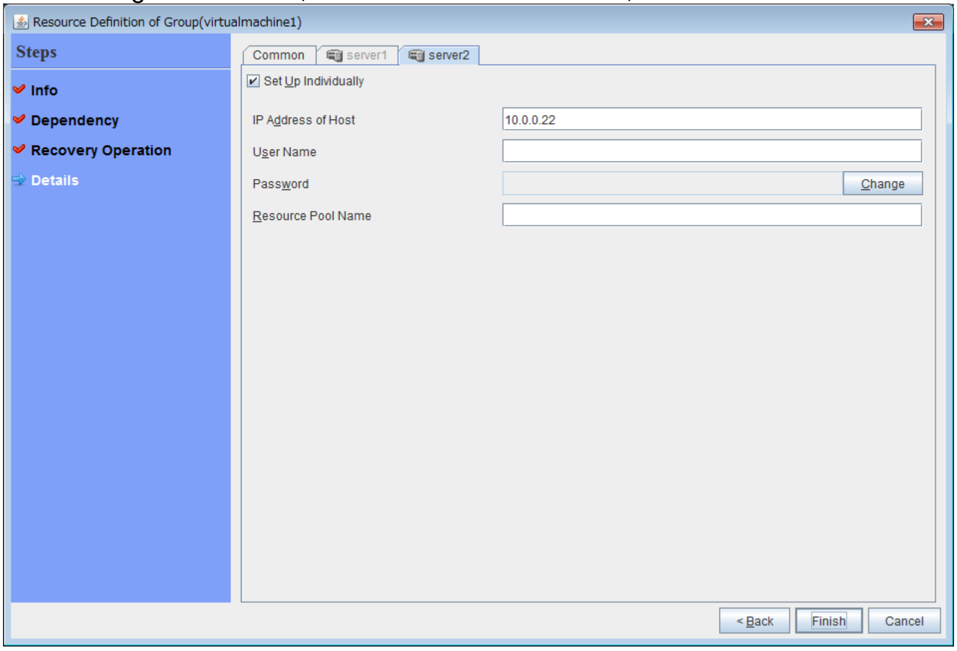
Resource Definition of Group(virtualmachine1)

Steps
- ✓ Info
- ✓ Dependency
- ✓ Recovery Operation
- Details

Common | server1 | server2

Virtual Machine Type	vSphere
Installation Destination of the Cluster Service	Host
Virtual Machine Name	esx-vm1
Data Store Name	datastore1
VM Configuration File Path	
IP Address of Host	10.0.0.21
UUID	
Library Path	
User Name	root
Password	•••••••• Change
☑ Use vCenter	
vCenter	vcserver
User Name for vCenter	administrator
Password for vCenter	•••••••• Change
Resource Pool Name	

Next, open the tab for each server and then enter **IP Address of Host** for VMware ESXi to which each server belongs
After setting both servers, return to the **Common** tab, and then click **Finish**.



Resource Definition of Group(virtualmachine1)

Steps
- ✓ Info
- ✓ Dependency
- ✓ Recovery Operation
- Details

Common | server1 | server2

☑ Set Up Individually

IP Address of Host	10.0.0.22
User Name	
Password	Change
Resource Pool Name	

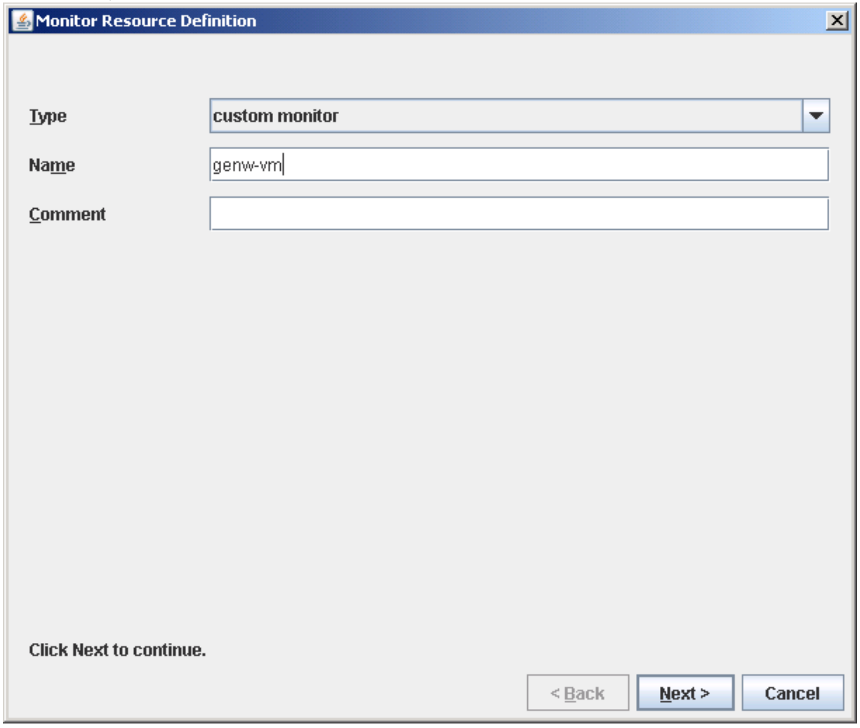
Setting up Virtual machine monitoring monitor

When using custom monitor resource

* If you are using ExpressCluster X 3.0 or later, we recommend that you use the virtual machine monitor resource.

(1) On the Expresscluster Builder window, right-click **Monitors** on the tree view, click **Add monitor resource**.

(2) Register custom monitor resource for virtual machine monitoring. Select **custom monitor resource** from **Type**. Enter any name (`genw-vm`) for custom monitor resource, and then click **Next**.



Monitor Resource Definition

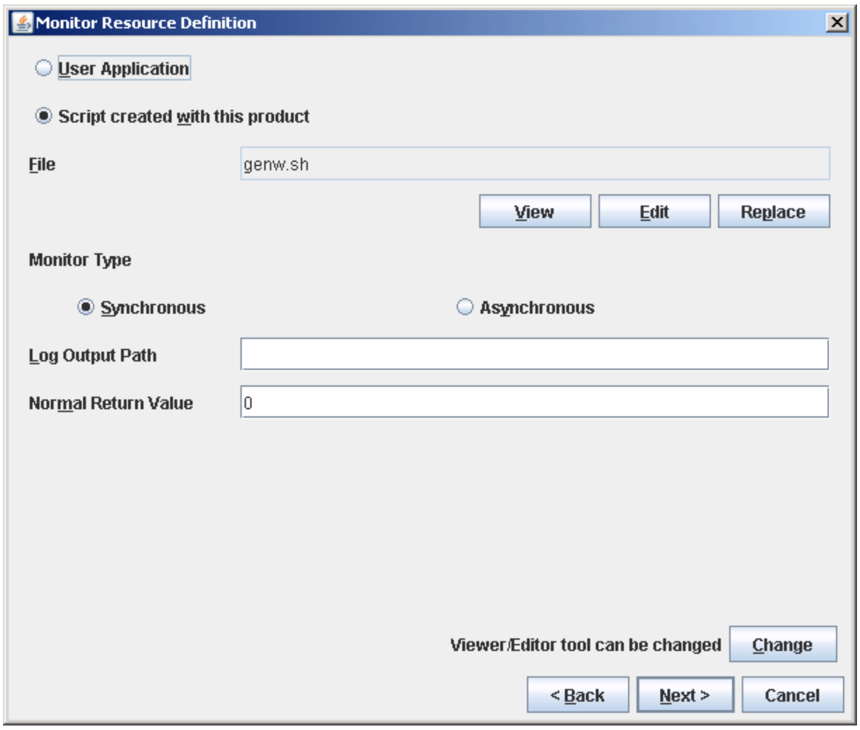
Type: custom monitor

Name: genw-vm

Comment:

Click Next to continue.

(3) Click **Replace** and replace `genw.sh` with the sample script `clpvmmon.pl` (on page 79) in this guide. Make sure that **Monitor script type** is **Synchronous**, and then click **Next**.



Monitor Resource Definition

○ User Application

● Script created with this product

File: genw.sh

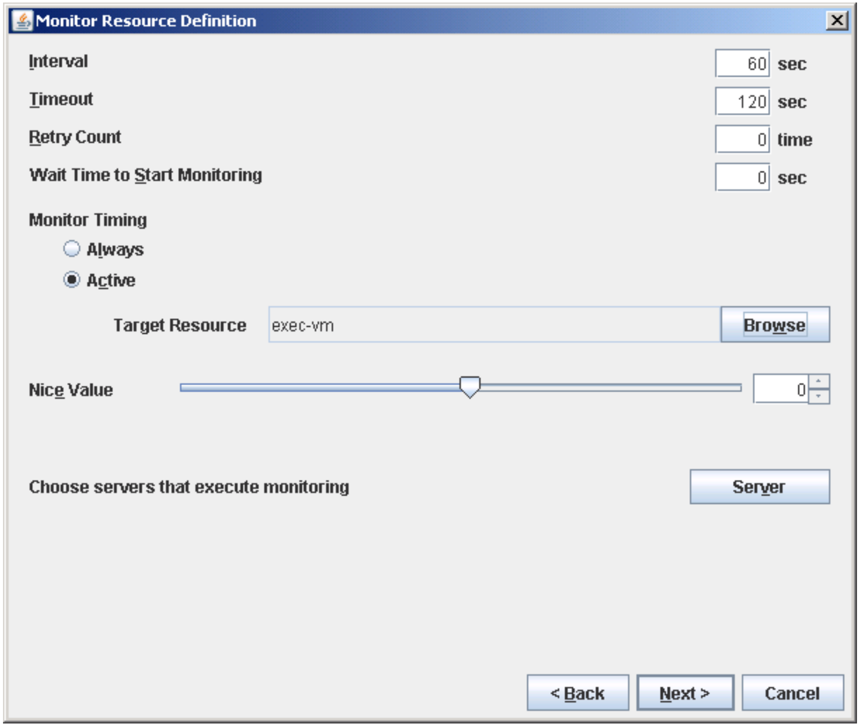
Monitor Type

● Synchronous ○ Asynchronous

Log Output Path:

Normal Return Value: 0

(4) Select **Active** for **Monitor Timing**, and select the virtual machine controlling resource **exec-vm** for the target resource, and change **Timeout**, **Interval**, **Retry Count** in accordance with the environment, and then click **Next**.



Monitor Resource Definition

Interval: 60 sec

Timeout: 120 sec

Retry Count: 0 time

Wait Time to Start Monitoring: 0 sec

Monitor Timing

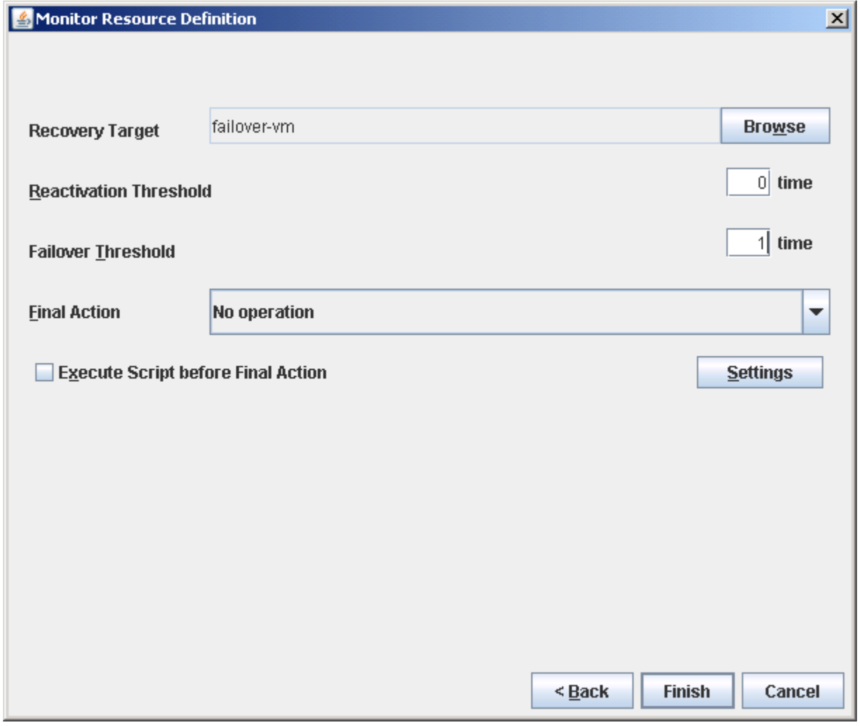
○ Always

● Active

Target Resource: exec-vm

Nice Value: 0

(5) Select the failover group name (`failover-vm`) to which the virtual machine controlling resource (`exec-vm`) belongs for **Recovery Target**, click **OK**. Edit the parameters such as **Reactivation Threshold** as appropriate[1], and then click **Finish**.


Monitor Resource Definition

Recovery Target: failover-vm

Reactivation Threshold: 0 time

Failover Threshold: 1 time

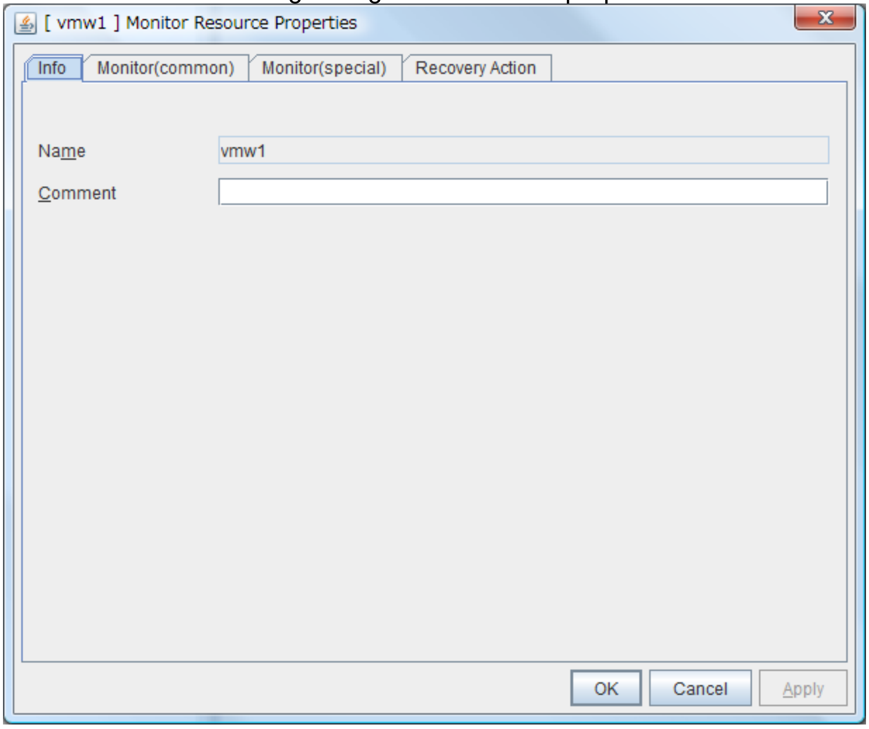
Final Action: No operation

□ Execute Script before Final Action

[1] Tune each parameter as described in Chapter 6, "Monitor Resource Details" in *ExpressCluster X Reference Guide*.

When using VM monitor resource (X3.0 or later)

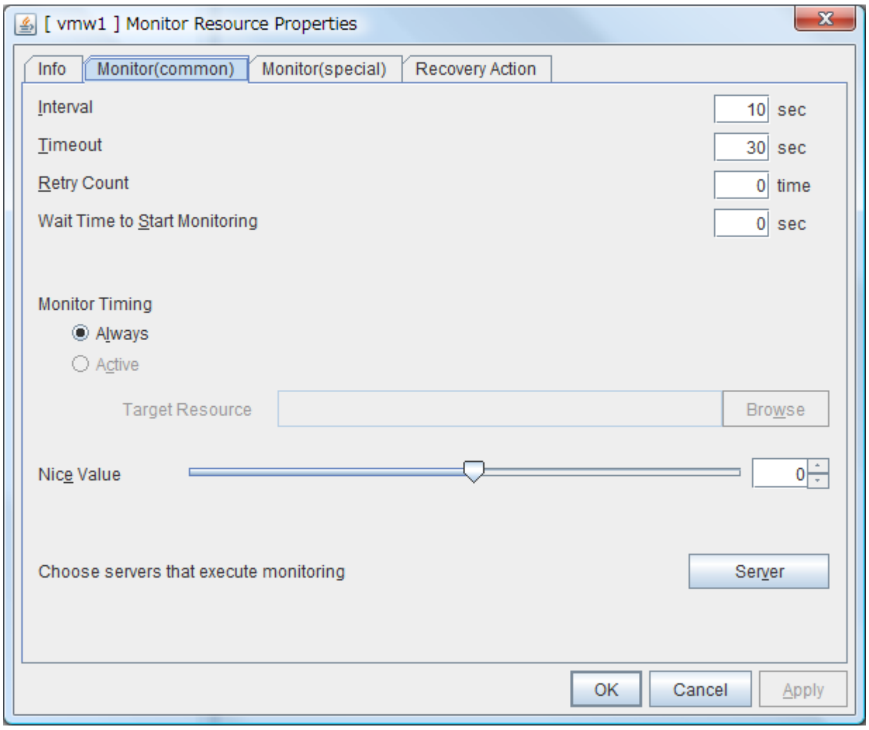
(1) When VM resource is created, VM monitor resource is created automatically. Modify the items of which setting change is needed on properties of VM monitor resource.



[vmw1] Monitor Resource Properties

Name: vmw1

Comment:



[vmw1] Monitor Resource Properties

Interval: 10 sec

Timeout: 30 sec

Retry Count: 0 time

Wait Time to Start Monitoring: 0 sec

Monitor Timing

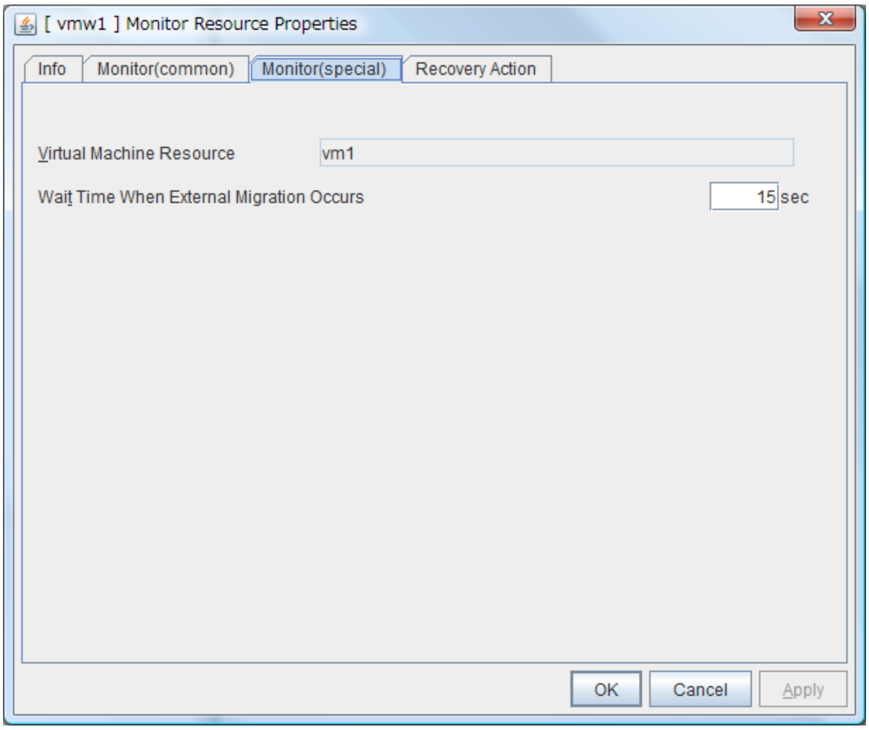
● Always

○ Active

Target Resource:

Nice Value: 0

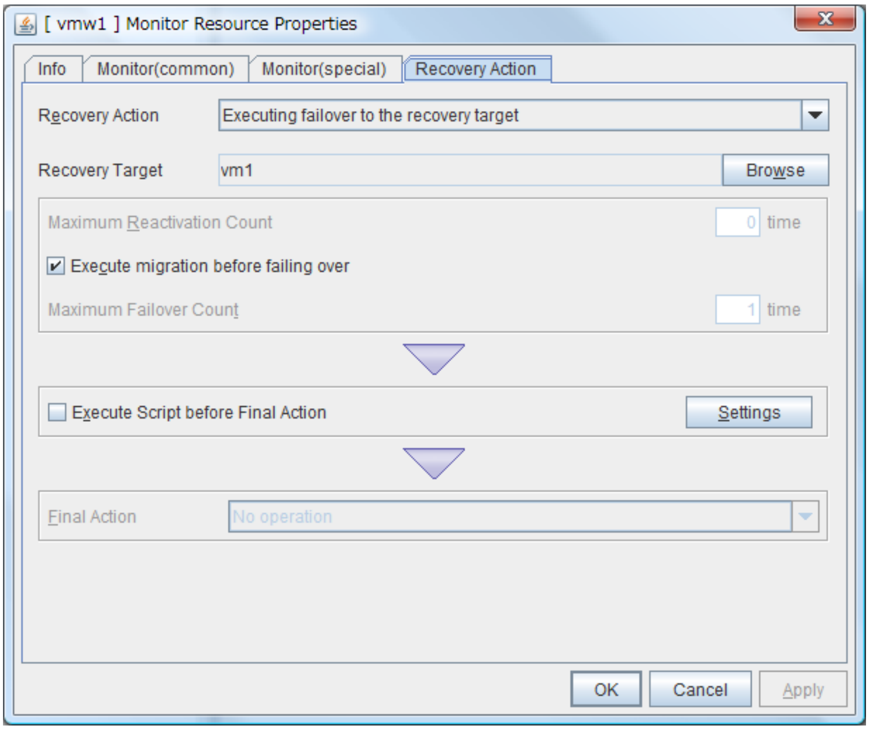
Choose servers that execute monitoring



[vmw1] Monitor Resource Properties

Virtual Machine Resource: vm1

Wait Time When External Migration Occurs: 15 sec



[vmw1] Monitor Resource Properties

Recovery Action: Executing failover to the recovery target

Recovery Target: vm1

Maximum Reactivation Count: 0 time

☑ Execute migration before failing over

Maximum Failover Count: 1 time

□ Execute Script before Final Action

Final Action: No operation

Associating with VMware HA

Associating an instance of ExpressCluster X SingleServerSafe on a host with VMware HA

(1) Create as many virtual machines as necessary.

(2) Set up VMware HA from vCenter.

A) Right-click the data center in the inventory, and then select **New cluster** to create a cluster.

B) Drag and drop a host in the inventory to the created cluster to make the host participate in the cluster. Make all the hosts that are to be failover targets for the guest OS participate in the cluster.

C) Right-click the created cluster, and then select **Edit Settings** to display the cluster configuration window.

D) Select the left pane **Cluster Function** in the cluster configuration window, and then select **Enable VMware HA**.

E) If you require virtual machine monitoring by VMware HA, select the left pane **VMware HA** followed by **Monitor Virtual Machine** in the cluster configuration window, and then select **Enable Virtual Machine Monitoring** in the **Virtual Machine Monitoring Status** field.[1]

(3) Install ExpressCluster X SingleServerSafe (SSS) on the Service Console of VMware ESX as described in the *ExpressCluster X Installation and Configuration Guide*.

(4) Specify diskw for the installed SSS so that the shared disk in which the guest OS is stored is monitored. Select **READ(O_DIRECT)** as the diskw monitor type, and select **OS Shutdown** as the final action. If using the user space monitor resource and shutdown stalling monitoring, set the monitoring method to **keepalive**.

(5) Set up other monitors as appropriate.

[1] For this setting to take effect, you must install VMware Tools on the guest and operate them.

Associating an instance of ExpressCluster X on a guest and an instance of ExpressCluster X SingleServerSafe on a host with VMware HA

(1) Perform the steps up to (2) in the previous section, “Associating an instance of ExpressCluster X SingleServerSafe on a host with VMware HA” (on Page 68).

(2) Install ExpressCluster in the guest OS as described in the *ExpressCluster X Installation and Configuration Guide*.

(3) Set up a cluster by using the ExpressCluster Builder and following the *ExpressCluster X Installation and Configuration Guide*.

(4) Perform step (3) and the subsequent steps in the previous section, “Associating an instance of ExpressCluster X SingleServerSafe on a host with VMware HA” (on Page 68).

Appendix A Sample Scripts

vmpower.start.pl

This is a script for starting a virtual machine. Edit the underlined portion in accordance with your environment.

```
#!/usr/bin/perl -w

#
# Script for power on the Virtual Machine
#

use strict;

#-------------------------------------------------------------------------------
# Configuration
#-------------------------------------------------------------------------------
# The path to VM configuration file. This must be absolute UUID-based path.
my $cfg_path = "/vmfs/volumes/datastore-uuid/vm1/vm1.vmx";

# The interval to check the vm status. (second)
my $interval = 1;
# The maximum count to check the vm status.
my $max_cnt = 100;
# The timeout to start the vm. (second)
my $start_to = 10;
#-------------------------------------------------------------------------------
my $vmname = $cfg_path; # VMname to be outputted on log.
$vmname =~ s/^(.*\/)(.*)(\.vmx)/$2/;

# VM operation command path
my $vmcmd = "/usr/bin/vmware-cmd";

# VM execution state map
my %state = (
   "VM_EXECUTION_STATE_ON" => "on",
   "VM_EXECUTION_STATE_OFF" => "off",
   "VM_EXECUTION_STATE_SUSPENDED" => "suspended",
   "VM_EXECUTION_STATE_STUCK" => "stuck",
   "VM_EXECUTION_STATE_UNKNOWN" => "unknown"
);

#-------------------------------------------------------------------------------
# Main
#-------------------------------------------------------------------------------
exit 1 if (!&RegisterVm());

if (&IsPoweredOn()){
   exit 0;
}
else{
   if (&PowerOn()){
      if (&WaitPoweredOnDone()){
         exit 0;
      }
      else{
         exit 1;
      }
   }
   else{
      exit 1;
   }
```

```
}

#-------------------------------------------------------------------------------
# Functions
#-------------------------------------------------------------------------------
sub RegisterVm{
   my $svop = "-s register";
   my $vmcmd_list = $vmcmd . " -l";
   my @vmlist = `$vmcmd_list`;
   my $ret = 0;
   my $opn_ret;
   my $line;

   foreach (@vmlist){
      if (/$cfg_path/){
         &Log("[I] [$vmname] at localhost already registered.\n");
         return 1;
      }
   }

   $opn_ret = open(my $fh, $vmcmd . " " . $svop . " " . $cfg_path . " 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $svop could not be executed.\n");
      return 0;
   }

   $line = <$fh>;
   if (defined($line)){
      &Log("[E] [$vmname] at localhost: Could not register VM: $line\n");
   }else{
      $ret = 1;
      &Log("[I] [$vmname] at localhost: Registered.\n");
   }

   close($fh);

   return $ret;
}
#-------------------------------------------------------------------------------
sub IsPoweredOn{

   if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
      return 1;
   }else{
      return 0;
   }
}
#-------------------------------------------------------------------------------
sub IsEqualState{
   my $vmop = "getstate";
   my $state = shift;
   my $ret = 0;
   my $opn_ret;
   my $line;

   $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.\n");
      return 0;
   }

   $line = <$fh>;
   if (defined($line)){
      chomp($line);
      if ($line =~ /^$vmop\(\)\s=\s(.+)$/){
         $ret = 1 if ($1 eq $state);
         &Log("[D] [$vmname] at localhost: VM execution state is $1.\n");
         }else{
            &Log("[E] [$vmname] at localhost: Could not get VM execution state: $line\n");
         }
      }

      close($fh);

      return $ret;
   }
#-----------------------------------------------------------------------------
sub PowerOn{
   my $vmop = "start";
   my $ret = 0;
   my $opn_ret;
   my $line;

   $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.\n");
      return 0;
   }

   eval{
      local $SIG{ALRM} = sub { die "timeout" };
      alarm($start_to);
      $line = <$fh>;
      alarm(0);
   };
   alarm(0);

   if ($@){
      if($@ =~ /timeout/){
         &Log("[E] [$vmname] at localhost: Cound not start VM: timeout($start_to
second)\n");

         if (&IsEqualState($state{"VM_EXECUTION_STATE_STUCK"})){
            $ret = 1 if (&ResolveVmStuck());
         }
      }
   }
   else{
      if (defined($line)){
         chomp($line);
         if ($line =~ /^$vmop\(\)\s=\s(.+)$/){
            if ($1 == 1){
               $ret = 1;
               &Log("[I] [$vmname] at localhost: Started.\n");
            }else{
               &Log("[E] [$vmname] at localhost: Cound not start VM: $1\n");
            }
         }else{
            &Log("[E] [$vmname] at localhost: Cound not start VM: $line\n");

            if (&IsEqualState($state{"VM_EXECUTION_STATE_STUCK"})){
               $ret = 1 if (&ResolveVmStuck());
            }
         }
      }

      close($fh);
   }

   return $ret;
}
#-----------------------------------------------------------------------------
sub ResolveVmStuck{
   my $vmop = "answer";
   my $ret = 0;
   my $opn_ret;
   my $line;

   $opn_ret = open(my $fh, "| ". $vmcmd . " " . $cfg_path . " " . $vmop);
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.\n");
      return 0;
   }

   # Answering "1) I _moved it" to keep vm config.
   print($fh "1\n");

   close($fh);

   if (&IsEqualState($state{"VM_EXECUTION_STATE_STUCK"})){
      &Log("[E] [$vmname] at localhost: VM stuck could not be resolved.\n");
   }else{
      $ret = 1;
      &Log("[I] [$vmname] at localhost: VM stuck is resolved.\n");
   }

   return $ret;
}
#-------------------------------------------------------------------------------
sub WaitPoweredOnDone{

   for (my $i = 0; $i < $max_cnt; $i++){
      if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
         &Log("[I] [$vmname] at localhost: Powered on done. ($i)\n");
         return 1;
      }
      sleep $interval;
   }

   &Log("[E] [$vmname] at localhost: Not powered on done. ($max_cnt)\n");

   return 0;
}
#-------------------------------------------------------------------------------
sub Log{
   my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
   $year += 1900;
   $mon += 1;
   my $date = sprintf "%d/%02d/%02d %02d:%02d:%02d", $year, $mon, $mday, $hour, $min,
$sec;
   print "$date $_[0]";
   return 0;
}
```

vmpower.stop.pl

This is a script for stopping a virtual machine. Edit the underlined portion in accordance with your environment.

```
#!/usr/bin/perl -w

#
# Script for power off the Virtual Machine
#

use strict;

#-------------------------------------------------------------------------------
# Configuration
#-------------------------------------------------------------------------------
# The path to VM configuration file. This must be absolute UUID-based path.
my $cfg_path = "/vmfs/volumes/datastore-uuid/vm1/vm1.vmx";

# The interval to check the vm status. (second)
my $interval = 5;
# The maximum count to check the vm status.
my $max_cnt = 100;
#-------------------------------------------------------------------------------
my $vmname = $cfg_path; # VMname to be outputted on log.
$vmname =~ s/^(.*\/)(.*)(\.vmx)/$2/;

# VM operation command path
my $vmcmd = "/usr/bin/vmware-cmd";

# VM execution state map
my %state = (
   "VM_EXECUTION_STATE_ON" => "on",
   "VM_EXECUTION_STATE_OFF" => "off",
   "VM_EXECUTION_STATE_SUSPENDED" => "suspended",
   "VM_EXECUTION_STATE_STUCK" => "stuck",
   "VM_EXECUTION_STATE_UNKNOWN" => "unknown"
);

#-------------------------------------------------------------------------------
# Main
#-------------------------------------------------------------------------------
if (&IsPoweredOn()){
   if (&PowerOff()){
      if (!&WaitPoweredOffDone()){
         exit 1;
      }
   }else{
      exit 1;
   }
}

if (&UnRegisterVm()){
   exit 0;
}else{
   exit 1;
}

#-------------------------------------------------------------------------------
# Functions
#-------------------------------------------------------------------------------
sub IsPoweredOn{
   if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
      return 1;
   }else{
      return 0;
   }
}
#-------------------------------------------------------------------------------
sub IsEqualState{
   my $vmop = "getstate";
   my $state = shift;
   my $ret = 0;
   my $opn_ret;
   my $line;

   $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.\n");
      return 0;
   }

   $line = <$fh>;
   if (defined($line)){
      chomp($line);
      if ($line =~ /^$vmop\(\)\s=\s(.+)$/){
         $ret = 1 if ($1 eq $state);
         &Log("[D] [$vmname] at localhost: VM execution state is $1.\n");
      }else{
         &Log("[E] [$vmname] at localhost: Could not get VM execution state: $line\n");
      }
   }

   close($fh);

   return $ret;
}
#-------------------------------------------------------------------------------
sub PowerOff{
   my $ret;

   # Soft stop.
   $ret = &PowerOffOpMode("soft");

   # Hard stop if Soft stop failed.
   if (!$ret){
      $ret = &PowerOffOpMode("hard");
   }

   return $ret;
}
#-------------------------------------------------------------------------------
sub PowerOffOpMode{
   my $vmop = "stop";
   my $powerop_mode = shift;
   my $ret = 0;
   my $opn_ret;
   my $line;

   return 0 if ($powerop_mode !~ /^hard|soft$/);

   $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " " . $powerop_mode .
" 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop $powerop_mode could not be
executed.\n");
      return 0;
   }
   $line = <$fh>;
   if (defined($line)){
      chomp($line);
      if ($line =~ /^$vmop\($powerop_mode\)\s=\s(.+)$/){
         if ($1 == 1){
            $ret = 1;
            &Log("[I] [$vmname] at localhost: Stopped. ($powerop_mode)\n");
         }else{
            &Log("[E] [$vmname] at localhost: Cound not stop ($powerop_mode) VM: $1\n");
         }
      }else{
         &Log("[E] [$vmname] at localhost: Cound not stop ($powerop_mode) VM: $line\n");
      }
   }else{
      if ($powerop_mode eq "soft"){
         $ret = 1;
         &Log("[I] [$vmname] at localhost: Stopped. ($powerop_mode)\n");
      }
   }

   close($fh);

   return $ret;
}
#-----------------------------------------------------------------------------
sub WaitPoweredOffDone{

   for (my $i = 0; $i < $max_cnt; $i++){
      if (&IsEqualState($state{"VM_EXECUTION_STATE_OFF"})){
         &Log("[I] [$vmname] at localhost: Powered off done. ($i)\n");
         return 1;
      }
      sleep $interval;
   }

   &Log("[E] [$vmname] at localhost: Not powered off done. ($max_cnt)\n");

   return 0;
}
#-----------------------------------------------------------------------------
sub UnRegisterVm{
   my $svop = "-s unregister";
   my $vmcmd_list = $vmcmd . " -l";
   my @vmlist = `$vmcmd_list`;
   my $ret = 0;
   my $opn_ret;
   my $flag = 0;
   my $line;

   foreach (@vmlist){
      if (/$cfg_path/){
         $flag = 1;
      }
   }

   if ($flag == 0){
      &Log("[I] [$vmname] at localhost already unregistered.\n");
      return 1;
   }else{
      $opn_ret = open(my $fh, $vmcmd . " " . $svop . " " . $cfg_path . " 2>&1 |");
      if (!$opn_ret){
         &Log("[E] [$vmname] at localhost: $vmcmd $svop could not be executed.\n");
         return 0;
      }

      $line = <$fh>;
      if (defined($line)){
         &Log("[E] [$vmname] at localhost: Could not unregister VM: $line\n");
      }else{
         $ret = 1;
         &Log("[I] [$vmname] at localhost: Unregistered.\n");
      }

      close($fh);
   }

   return $ret;
}
#-------------------------------------------------------------------------------
sub Log{
   my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
   $year += 1900;
   $mon += 1;
   my $date = sprintf "%d/%02d/%02d %02d:%02d:%02d", $year, $mon, $mday, $hour, $min,
$sec;
   print "$date $_[0]";
   return 0;
}
```

clpvmmon.pl

This is a script for checking the start state of a virtual machine. Edit the underlined portion in accordance with your environment.

```
#!/usr/bin/perl -w

#
# Script for monitoring the Virtual Machine
#

use strict;

#-------------------------------------------------------------------------------
# Configuration
#-------------------------------------------------------------------------------
# The path to VM configuration file. This must be absolute UUID-based path.
my $cfg_path = "/vmfs/volumes/datastore-uuid/vm1/vm1.vmx";

# The interval to check the vm status. (second)
my $interval = 1;
#-------------------------------------------------------------------------------
my $vmname = $cfg_path; # VMname to be outputted on log.
$vmname =~ s/^(.*\/)(.*)(\.vmx)/$2/;

# VM operation command path
my $vmcmd = "/usr/bin/vmware-cmd";

# VM execution state map
my %state = (
   "VM_EXECUTION_STATE_ON" => "on",
   "VM_EXECUTION_STATE_OFF" => "off",
   "VM_EXECUTION_STATE_SUSPENDED" => "suspended",
   "VM_EXECUTION_STATE_STUCK" => "stuck",
   "VM_EXECUTION_STATE_UNKNOWN" => "unknown"
);

#-------------------------------------------------------------------------------
# Main
#-------------------------------------------------------------------------------
system("\"ulimit\" -s unlimited");
while (&IsHbIncrease()){
}
exit 0;

#-------------------------------------------------------------------------------
# Functions
#-------------------------------------------------------------------------------
sub IsHbIncrease{
   my $last_hb = -1;

   for (my $i = 0; $i < 2; $i++){
      if (&IsPoweredOn()){
         my $hb = &GetHb();
         if ($hb == -1){
            return 0;
         }

         if ($hb == $last_hb){
            &Log("[I] [$vmname] is stalled.\n");
            return 0;
         }else{
            $last_hb = $hb;
         }
      }else{
         &Log("[I] [$vmname] is powered off.\n");
         return 0;
      }
      sleep $interval;
   }

   return 1
}
#--------------------------------------------------------------------------------
sub GetHb{
   my $vmop = "getheartbeat";
   my $ret = -1;
   my $opn_ret;
   my $line;

   $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.\n");
      return -1;
   }

   $line = <$fh>;
   if (defined($line)){
      chomp($line);
      if ($line =~ /^$vmop\(\)\s=\s(.+)$/){
         $ret = $1;
#        &Log("[D] [$vmname] at localhost: Got VM heartbeat count $1.\n");
      }else{
         &Log("[E] [$vmname] at localhost: Could not get VM heartbeat count: $line\n");
      }
   }

   close($fh);

   return $ret;
}
#--------------------------------------------------------------------------------
sub IsPoweredOn{

   if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
      return 1;
   }else{
      return 0;
   }
}
#--------------------------------------------------------------------------------
sub IsEqualState{
   my $vmop = "getstate";
   my $state = shift;
   my $ret = 0;
   my $opn_ret;
   my $line;

   $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
   if (!$opn_ret){
      &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.\n");
      return 0;
   }

   $line = <$fh>;
   if (defined($line)){
      chomp($line);
      if ($line =~ /^$vmop\(\)\s=\s(.+)$/){
         $ret = 1 if ($1 eq $state);
#        &Log("[D] [$vmname] at localhost: VM execution state is $1.\n");
      }else{
         &Log("[E] [$vmname] at localhost: Could not get VM execution state: $line\n");
```

```
        }
    }

    close($fh);

    return $ret;
}
#------------------------------------------------------------------------------
sub Log{
    my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
    $year += 1900;
    $mon += 1;
    my $date = sprintf "%d/%02d/%02d %02d:%02d:%02d", $year, $mon, $mday, $hour, $min,
$sec;
    print "$date $_[0]";
    return 0;
}
```

vmpreaction.sh

This is a script for Linux that issues a failover request from ExpressCluster on a guest to ExpressCluster on a host. Edit the underlined portion in accordance with your environment.

1. When the version of ExpressCluster on host OS is X2.1.

```
#! /bin/sh
#***********************************************
#* preaction.sh *
#***********************************************
ulimit -s unlimited

echo START
echo $CLP_MONITORNAME

# Write the name of virtual machine control group resource
CLPRSC="exec vm"
# Write the IP addresses of each host, delimited by commas.
CLPIP="10.0.0.1,10.0.0.2"

/opt/nec/clusterpro/bin/clptrnreq -t GRP_FAILOVER -r $CLPRSC -h $CLPIP
exit 0
```

2. When the version of ExpressCluster on host OS is X3.0 or later.

```
#! /bin/sh
#***********************************************
#* preaction.sh *
#***********************************************
ulimit -s unlimited

echo START
echo $CLP_MONITORNAME

# Write the name of virtual machine control group resource
CLPRSC="vm1"
# Write the IP addresses of each host, delimited by commas.
CLPIP="10.0.0.1,10.0.0.2"

/opt/nec/clusterpro/bin/clprexec --failover -r $CLPRSC -h $CLPIP
exit 0
```

Note: In case that the version of ExpressCluster on guest OS is X2.1, `clprexec` command is not included. Use `clptrnreq` command, or retrieve `clprexec` command from ExpressCluster CD and use the command.

vmpreaction.bat

This is a batch file for Windows that issues a failover request from ExpressCluster on a guest to ExpressCluster on a host. Edit the underlined portion in accordance with your environment.

1. When the version of ExpressCluster on host OS is X2.1.

```
rem ***************************************
rem *          preaction.bat          *
rem ***************************************

echo START
echo %CLP_MONITORNAME%

rem Write the name of virtual machine control group resource
SET CLPRSC="exec-vm"
rem Write the IP addresses of each host, delimited by commas.
SET CLPIP="10.0.0.1,10.0.0.2"

clptrnreq.exe -t GRP_FAILOVER -r %CLPRSC% -h %CLPIP%
echo EXIT
```

2. When the version of ExpressCluster on host OS is X3.0 or later.

```
rem ***************************************
rem *          preaction.bat          *
rem ***************************************

echo START
echo %CLP_MONITORNAME%

rem Write the name of virtual machine control group resource
SET CLPRSC="vm1"
rem Write the IP addresses of each host, delimited by commas.
SET CLPIP="10.0.0.1,10.0.0.2"

clprexec.exe --failover -r %CLPRSC% -h %CLPIP%
echo EXIT
```

Note: In case that the version of ExpressCluster on guest OS is X2.1, `clprexec` command is not included. Use `clptrnreq` command, or retrieve `clprexec` command from ExpressCluster CD and use the command.

clphostmon_wrap.sh

This is a wrapper script for using `clphostmon.pl`. Edit the underlined portion in accordance with your environment.

```
#!/bin/sh

#
# Wrapper script for clphostmon.pl
#

#-------------------------------------------------------------------------------
# Configuration
#-------------------------------------------------------------------------------
# Specify the path of clphostmon.pl. It must be absolute path.
# e,g CLPHOSTMON_PATH="/opt/nec/clusterpro/work/clphostmon/clphostmon.pl"
CLPHOSTMON_PATH="absolute path of clphostmon.pl"
#-------------------------------------------------------------------------------

export PERL_LWP_SSL_VERIFY_HOSTNAME=0
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/vmware/vma/lib64:/opt/vmware/vma/lib
perl $CLPHOSTMON_PATH
```

clphostmon.pl

This is a script for checking the host status. Edit the underlined portion in accordance with your environment.

```
# hostname
hostname … <1>

# vifp listservers
esxi hostname … <2>    ESXi
```

```
#!/usr/bin/perl -w

#
# Script for monitoring the hardware
#


use strict;
use warnings;
use VMware::VILib;
use VMware::VIRuntime;
use VMware::VmaTargetLib;


#--------------------------------------------------------------------------------
# Configuration
#--------------------------------------------------------------------------------
# vMA_and_target_host : vMA hostname and ESXi Host name
# e.g my @vMA_and_target_host1 = ("vMA_host1", "ESXi_host1");
# monitoring_disks_on_host : The path to disk device file. Disk device path
#                        must be absolute path.
#                        We can set plural values in this variable.
#                        If we do not want to monitor disks, we should set
#                        this variable to undef as follows.
# e.g my @monitoring_disks_on_host1 = undef;
# monitoring_pnics_on_host : vmnic name. e.g vmnic0
#                        We can set plural values in this variable.
#                        If we do not want to monitor nics, we should set
#                        this variable to undef as follows.
# e.g my @monitoring_pnics_on_host1 = undef;

# Configuration for server1
my @vMA_and_target_host1 = ("hostname of vma … <1>", "hostname of esxi … <2>");
my @monitoring_disks_on_host1 = ("device file path", …);
my @monitoring_pnics_on_host1 = ("vmnic name",…);

# Configuration for server2
my @vMA_and_target_host2 = ("hostname of vma", "hostname of esxi");
my @monitoring_disks_on_host2 = ("device file path",…);
my @monitoring_pnics_on_host2 = ("vmnic name", …);

my @search_host = ([@vMA_and_target_host1], [@vMA_and_target_host2]);
my @search_disks = ([@monitoring_disks_on_host1], [@monitoring_disks_on_host2]);
my @search_pnics = ([@monitoring_pnics_on_host1],[@monitoring_pnics_on_host2]);
#--------------------------------------------------------------------------------

my $target_host;
my @monitoring_disks;
my @monitoring_pnics;

my $EXIT_CODE = 0;
my $TRACE = (1 << 0);
my $INFO = (1 << 1);
my $WARN = (1 << 2);
my $ERR = (1 << 3);
my $LOG_RELEASE = ($INFO | $WARN | $ERR);

#-------------------------------------------------------------------------------
# Main
#-------------------------------------------------------------------------------
chomp(my $local_host = `hostname`);
log_write($INFO, "Monitoring start on $local_host.\n");

for (my $i= 0; $i < $#search_host + 1; $i++) {
   if($search_host[$i][0] eq $local_host) {
      $target_host = $search_host[$i][1];
      @monitoring_disks = @{$search_disks[$i]};
      @monitoring_pnics = @{$search_pnics[$i]};
      last;
   }
}
if (!defined($target_host)) {
   log_write($ERR,
      "$local_host was not found in specified vMA array. \n");
   exit (1);
}

my $host_exist;
my @esx_lists = VmaTargetLib::enumerate_targets();
foreach my $my_esx (@esx_lists) {
   if ($my_esx->name() eq $target_host) {
      $host_exist = 1;
      last;
   }
}
if (defined($host_exist)) {
   log_write($INFO, "Connecting target server : $target_host\n");
} else {
   log_write($ERR, "Could not find target server in credential store.\n");
   exit (1);
}
my $fast_login = VmaTargetLib::query_target($target_host);
$fast_login->login();
log_write($INFO, "Connected.\n");

my $host = Vim::find_entity_view(view_type => 'HostSystem');
if (!defined($host)) {
   log_write($ERR, "Could not get HostSystem view.\n");
   exit (1);
}

eval {
   if (defined($monitoring_disks[0])) {
      $EXIT_CODE = MonitoringDisk($host, @monitoring_disks);
   }
};
if ($@) {
   log_write($ERR, "Exception error occurred:\n $@\n");
   $EXIT_CODE = 1;
}

eval {
   if (defined($monitoring_pnics[0])) {
      $EXIT_CODE = MonitoringNetwork($host, @monitoring_pnics);
   }
};
if ($@) {
   log_write($ERR, "Exception error occurred:\n $@\n");
   $EXIT_CODE = 1;
}
```

```
$fast_login->logout();
log_write($INFO, "Monitoring completed:$EXIT_CODE\n");
exit ($EXIT_CODE);

#--------------------------------------------------------------------------------
# Functions
#--------------------------------------------------------------------------------
sub MonitoringDisk {
   my ($hview, @target_disks) = @_;
   my $RESULT = 0;

   log_write($INFO, "Monitoring storage start.\n");
   my $storage =
      Vim::get_view(mo_ref => $hview->configManager->storageSystem);

   log_write($INFO, "Refresh storage information.\n");
   $storage->RefreshStorageSystem();
   log_write($INFO, "Rescan HBAs.\n");
   $storage->RescanAllHba();

   foreach my $disk (@target_disks) {
      log_write($INFO, "Access target disk: $disk.\n");
      my $part_info =
         $storage->RetrieveDiskPartitionInfo(devicePath => $disk);
      if ($#$part_info < 0) {
         log_write($ERR, "Could not access target disk.\n");
         $RESULT = 1;
      }
      foreach (@$part_info) {
         log_write($INFO, "Access succeeded.\n");
         my $disk_size =
            ConvirtUnit($_->spec->totalSectors * 512);
         log_write($INFO, "Disk Size is $disk_size \n");
      }
   }
   log_write($INFO, "Monitoring storage completed.\n");
   return $RESULT;
}
#--------------------------------------------------------------------------------
sub MonitoringNetwork {
   my ($hview, @target_pnics) = @_;
   my $RESULT = 0;

   log_write($INFO, "Monitoring network start.\n");
   my $network =
      Vim::get_view(mo_ref => $hview->configManager->networkSystem);

   log_write($INFO, "Refresh network information.\n");
   $network->RefreshNetworkSystem();

   log_write($INFO, "Collect the nic status.\n");
   my $pnics = $network->networkInfo->pnic;
   foreach my $target_pnic (@target_pnics) {
      my $exist = 0;
      foreach my $pnic (@$pnics) {
         if ($target_pnic eq $pnic->device) {
            $exist = 1;
            my $link = $pnic->linkSpeed;
            if (defined($link)) {
               log_write($INFO,
                  "$target_pnic : Link is Up.\n");
            } else {
               log_write($ERR,
                  "$target_pnic : Link is Down.\n");
               $RESULT = 1;
            }
         }
      }
      if ($exist == 0) {
         log_write($ERR, "$target_pnic : Missing device.\n");
         $RESULT = 1;
      }
   }
   log_write($INFO, "Monitoring network completed.\n");
   return $RESULT;
}
#-------------------------------------------------------------------------------
sub ConvirtUnit {
   my ($size) = @_;
   my $convirt_size;

   if ($size <= 0 || $size eq '') {
      return $size;
   }

   if ($size < 1024) {
      $convirt_size = $size . "Bytes";
   } elsif ($size < 1024 * 1024) {
      $convirt_size = sprintf("%.2f", $size / 1024) . "KBytes";
   } elsif ($size < 1024 * 1024 * 1024) {
      $convirt_size = sprintf("%.2f",
         $size / (1024 * 1024)) . "MBytes";
   } elsif ($size < 1024 * 1024 * 1024 * 1024) {
      $convirt_size = sprintf("%.2f",
         $size / (1024 * 1024 * 1024)) . "GBytes";
   } else {
      $convirt_size = sprintf("%.2f",
         $size / (1024 * 1024 * 1024)) . "TBytes";
   }

   return $convirt_size;
}
#-------------------------------------------------------------------------------
sub log_write {
   my ($level, $string) = @_;
   my $pre_level;

   if ($level & $LOG_RELEASE) {
      if ($level == $TRACE) {
         $pre_level = "T";
      } elsif ($level == $INFO) {
         $pre_level = "I";
      } elsif ($level == $WARN) {
         $pre_level = "W";
      } elsif ($level == $ERR) {
         $pre_level = "E";
      } else {
         $pre_level = "U";
      }
   } else {
      # not print log
      return 0;
   }

   my $ctime = get_ctime();
   my $head = sprintf("%s %s : ", $pre_level, $ctime);

   print "$head $string";

   return 0;
}
#-------------------------------------------------------------------------------
sub get_ctime {
   my $time = time;
   my ($sec, $min, $hour, $day, $month, $year) = localtime($time);
   $year += 1900;
   $month += 1;
   my $ctime = sprintf ("%d/%02d/%02d %02d:%02d:%02d",
        $year, $month, $day, $hour, $min, $sec);
   return $ctime;
}
#-------------------------------------------------------------------------------
```

stdnhost_wrap.sh

This is a wrapper script for using shutdownhost.pl. Edit the underlined portion in accordance with your environment.

```
#!/bin/sh

#
# Wrapper script for shutdownhost.pl
#

#--------------------------------------------------------------------------------
# Configuration
#--------------------------------------------------------------------------------
# Specify the path of shutdownhost.pl. It must be absolute path.
# e,g STDNHOST_PATH="/opt/nec/clusterpro/work/shutdownhost/shutdownhost.pl"
STDNHOST_PATH="absolute path of shutdownhost.pl"
#--------------------------------------------------------------------------------

export PERL_LWP_SSL_VERIFY_HOSTNAME=0
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/vmware/vma/lib64:/opt/vmware/vma/lib
perl $STDNHOST_PATH
```

shutdownhost.pl

This is a script for shutting down the host. Edit the underlined portion in accordance with your environment.

# hostname
[hostname] … <1>

# vifp listservers
[esxi hostname] … <2> ESXi

```
#!/usr/bin/perl -w

#
# Script for shutting down ESXi host;
#


use strict;
use warnings;
use VMware::VILib;
use VMware::VIRuntime;
use VMware::VmaTargetLib;

#-------------------------------------------------------------------------------
# Configuration
#-------------------------------------------------------------------------------
# vMA_and_target_host : vMA hostname and ESXi Host name
# e.g my @vMA_and_target_host1 = ("vMA_host1", "ESXi_host1");

# Configuration for server1
my @vMA_and_target_host1 = ("hostname of vma … <1>", "hostname of esxi … <2>");

# Configuration for server2
my @vMA_and_target_host2 = ("hostname of vma", "hostname of esxi");

my @search_host = ([@vMA_and_target_host1], [@vMA_and_target_host2]);

# Specify log path.
# e.g my $log_path = "/opt/nec/clusterpro/log/shutdownhost.log";
my $log_path = "log file path";
#-------------------------------------------------------------------------------

my $target_host;
my $EXIT_CODE = 0;

my $TRACE = (1 << 0);
my $INFO = (1 << 1);
my $WARN = (1 << 2);
my $ERR = (1 << 3);
my $LOG_RELEASE = ($INFO | $WARN | $ERR);

#-------------------------------------------------------------------------------
# Main
#-------------------------------------------------------------------------------

log_write($INFO, "Shutdown the host starting.\n");

eval {
   chomp(my $local_host = `hostname`);
   for (my $i= 0; $i < $#search_host + 1; $i++) {
      if($search_host[$i][0] eq $local_host) {
         $target_host = $search_host[$i][1];
         last;
      }
   }
   if (!defined($target_host)) {
      log_write($ERR,
         "$local_host was not found in specified vMA array. \n");
      exit (1);
   }

   my $host_exist;
   my @esx_lists = VmaTargetLib::enumerate_targets();
   foreach my $my_esx (@esx_lists) {
      if ($my_esx->name() eq $target_host) {
         $host_exist = 1;
         last;
      }
   }
   if (defined($host_exist)) {
      log_write($INFO, "Connecting target server : $target_host\n");
   } else {
      log_write($ERR,
         "Could not find target server in credential store.\n");
      exit (1);
   }

   my $fast_login = VmaTargetLib::query_target($target_host);
   $fast_login->login();
   log_write($INFO, "Connected.\n");

   log_write($INFO,
      "Sending request to shutdown $target_host from $local_host\n");
   my $host = Vim::find_entity_view(view_type => 'HostSystem');
   $host->ShutdownHost(force => 1);

   log_write($INFO, "Done.\n");
   $fast_login->logout();
};
if ($@) {
   log_write($ERR, "Exception error occurred:\n $@\n");
   $EXIT_CODE = 1;
}

exit ($EXIT_CODE);
#----------------------------------------------------------------------------------------
sub log_write {
   my ($level, $string) = @_;
   my $pre_level;

   if ($level & $LOG_RELEASE) {
      if ($level == $TRACE) {
         $pre_level = "T";
      } elsif ($level == $INFO) {
         $pre_level = "I";
      } elsif ($level == $WARN) {
         $pre_level = "W";
      } elsif ($level == $ERR) {
         $pre_level = "E";
      } else {
         $pre_level = "U";
      }
   } else {
      # not print log
      return 0;
   }

   my $ctime = get_ctime();
   my $head = sprintf("%s %s : ", $pre_level, $ctime);

   open(OUT, "+>>$log_path");
   print OUT "$head $string";
   close(OUT);

   return 0;
}
#-------------------------------------------------------------------------------
sub get_ctime {
   my $time = time;
   my ($sec, $min, $hour, $day, $month, $year) = localtime($time);
   $year += 1900;
   $month += 1;
   my $ctime = sprintf ("%d/%02d/%02d %02d:%02d:%02d",
        $year, $month, $day, $hour, $min, $sec);
   return $ctime;
}
#-------------------------------------------------------------------------------
```