

ExpressCluster[®] X *for Windows*

PP Guide (Virtualization Software)

09/30/2012
5th Edition



Revision History

Edition	Revision Date	Description
1	04/14/2009	Created
2	10/20/2009	Corresponds to Hyper-V 2.0. Modified sample scripts. Added notes.
3	10/1/2010	Corresponds to ExpressCluster X 3.0.
4	10/11/2011	Corresponds to ExpressCluster X 3.1.
5	09/30/2012	Corresponds to Windows Server 2012 Hyper-V.

© Copyright NEC Corporation 2009. All rights reserved.

Disclaimer

The contents of this document are subject to change without notice.

NEC Corporation assumes no responsibility for technical or editorial mistakes in or omissions from this document.

To obtain the benefits of the product, it is the customer's responsibility to install and use the product in accordance with this document.

The copyright for the contents of this document belongs to NEC Corporation. Copying, altering, or translating this document, in full or in part, without the permission of NEC Corporation, is prohibited.

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.

Microsoft, Windows, Windows Server, and Hyper-V are registered trademarks of Microsoft Corporation in the United States and other countries.

Other product names and logos are trademarks or registered trademarks of their respective companies.

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

CONTENTS

INTRODUCTION	vii
Target Readers and Purpose	vii
ExpressCluster Manuals.....	viii
Organization of This Manual.....	ix
Conventions	x
Terms Used in This Document	xi
Chapter 1 Configuration	12
Clustering between host OSs	12
Clustering between guest OSs	13
Linking guest OSs and host OSs.....	14
External linkage (for ExpressCluster X 3.0 or later).....	15
Chapter 2 Operating Environment.....	16
Chapter 3 Notes	17
Notes on clustering between host OSs (ExpressCluster X 2.0/2.1).....	17
Notes on clustering between host OSs (when using Hyper-V 2.0 or earlier and ExpressCluster X 3.0 or later)	17
Notes on clustering between host OSs (when using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or later)	18
Notes on clustering between guest OSs.....	19
Chapter 4 Virtualization Environment Setup Procedure	20
Installing Hyper-V	20
Using live migration	21
Chapter 5 Cluster Environment Setup Procedure	22
Setting Up Inter-Host OS Cluster (ExpressCluster X 2.0/2.1).....	22
Checking Operations of Inter-Host OS Clusters (ExpressCluster X 2.0/2.1)	27
Setting Up Inter-Host OS Cluster (When Using Hyper-V 2.0 or Earlier and ExpressCluster X 3.0 or Later)	28
Checking Operations of Inter-Host OS Cluster (When Using Hyper-V 2.0 or Earlier and ExpressCluster X 3.0 or Later).....	32
Setting Up Inter-Host OS Cluster (When Using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or Later).....	33
Checking Operations of Inter-Host OS Clusters (When Using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or Later)	36
Setting Up an Inter-Guest OS Cluster	37
Checking Operations of an Inter-Guest OS Cluster.....	39
Establishing Links between Guest OSs and Host OSs.....	40
Checking the Linkage between Guest OSs and Host OSs.....	41
Setting External Monitor Linkage and Checking Operations	42
Appendix A Sample Scripts.....	43

INTRODUCTION

Target Readers and Purpose

ExpressCluster® PP Guide is intended for administrators who set up systems, system engineers who provide user support, and maintenance personnel for cluster systems. This document introduces the software of which operations have been checked for an ExpressCluster environment. The software descriptions and setup examples in this document are provided only as a reference, and the software operations are not guaranteed.

ExpressCluster Manuals

The following five ExpressCluster manuals are available. The title of each guide and a description of its role follows.

ExpressCluster X Getting Started Guide

This guide is intended for users of ExpressCluster and provides a product overview, a description of the operating environment, update information, and known problems.

ExpressCluster X Install and Configuration Guide

This guide is intended for system engineers who install cluster systems using ExpressCluster and for system administrators who maintain and operate installed cluster systems, and it describes requirements for installing a cluster system using ExpressCluster to for preparing to start operation. This guide follows the actual procedure for installing a cluster system to describe how to design a cluster system using ExpressCluster, how to install and set up ExpressCluster, how to check the system after setting it up, and how to evaluate the system before starting operation.

ExpressCluster X Reference Guide

This guide is intended for system administrators and system engineers who use ExpressCluster to install cluster systems, and it provides information about ExpressCluster operating procedures, the features of each module, maintenance, and troubleshooting. This guide is a companion to the *Install 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.

To obtain the ExpressCluster Manuals, visit the following website.

<http://www.nec.com/expresscluster/>

Organization of This Manual

- Chapter 1. Configuration: This chapter describes cluster systems that can be configured by linking Hyper-V and ExpressCluster.
- Chapter 2. Operating Environment: This chapter describes operating environments supported when Hyper-V and ExpressCluster are linked.
- Chapter 3. Notes: This chapter describes notes on linking Hyper-V and ExpressCluster.
- Chapter 4. Virtualization Environment Setup Procedure: This chapter describes procedures for configuring virtualization environments when Hyper-V and ExpressCluster are linked.
- Chapter 5. Cluster Environment Setup Procedure: This chapter describes procedures for configuring cluster systems when Hyper-V and ExpressCluster are linked.

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.	<code>clpstat -s[-h <i>host_name</i>]</code>
#	Prompt to indicate that a Linux user has logged on as root user.	<code># clpcl -s -a</code>
Monospace (courier)	Indicates path names, commands, system output (message, prompt, etc.), directory, file names, functions and parameters.	<code>/Linux/3.0/eng/server/</code>
Monospace bold (courier)	Indicates the value that a user actually enters from a command line.	Enter the following: <code># clpcl -s -a</code>
Monospace italic (courier)	Indicates that users should replace italicized part with values that they are actually working with.	<code>rpm -i expressclsbuilder -<i>version_number</i>- <i>release_number</i>.i686.rpm</code>

Terms Used in This Document

The meanings of the terms used in this document are as follows.

Term	Meaning
Physical machine	A server where Hyper-V is running.
Host OS	An operating system installed on a physical machine.
Virtual machine	A virtual server or client created on a physical machine.
Guest OS	An operating system installed on a virtual machine.
Integration service	Additional software to access Hypervisor from a virtual machine. When setting up an inter-host OS cluster as described in this chapter, you must install the integration service on the virtual machine.
Failover group	A group for which the type is set to Failover in ExpressCluster X is called a "failover group" in this chapter.
Virtual machine group	A group for which the type is set to Virtual Machine in ExpressCluster X is called a "virtual machine group" in this chapter.

Chapter 1 Configuration

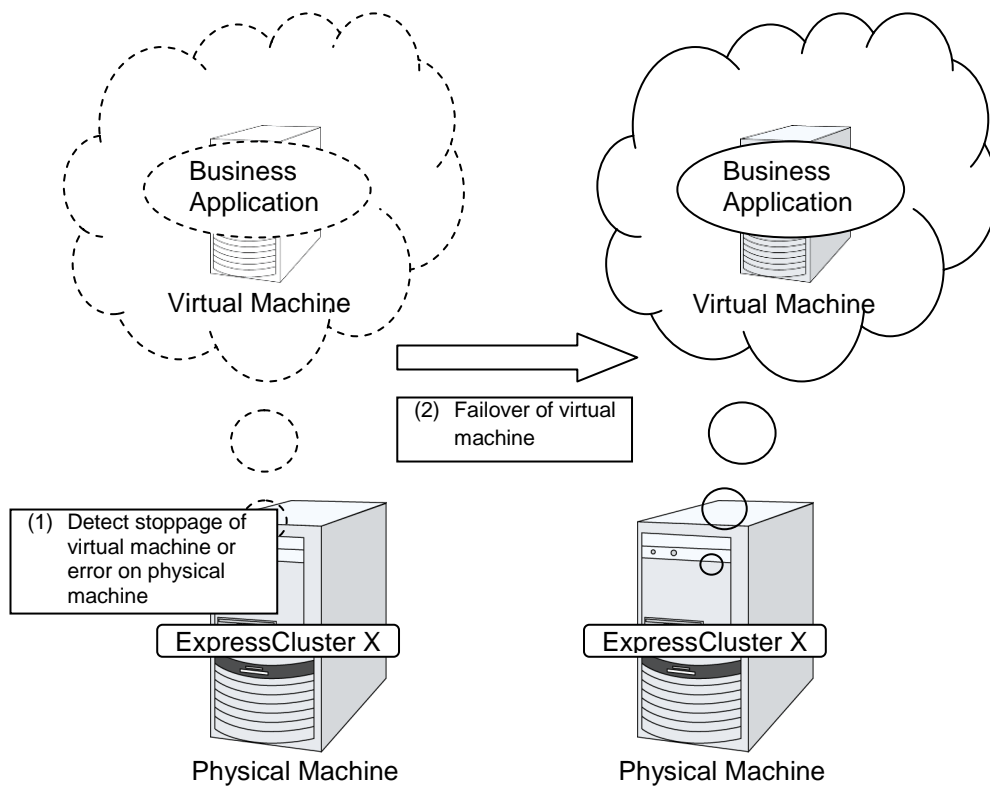
The following cluster configurations can be set up by linking Hyper-V and ExpressCluster.

Clustering between host OSs

Install ExpressCluster X on host OSs, and conduct clustering between physical machines. Not only general business applications but also a guest OS can be failed over.

This chapter describes procedures for setting up a cluster in order to fail over a guest OS. When a guest OS is a target for failover, ExpressCluster starts, stops, and monitors the guest OS. Applications on guest OSs do not need to be aware of the cluster. Store the virtual machine configuration file on a mirror disk or shared disk.

At scheduled shutdown of a host OS, by using quick migration or live migration¹, guest OSs can be migrated to a different physical machine without having the virtual machine shut down.

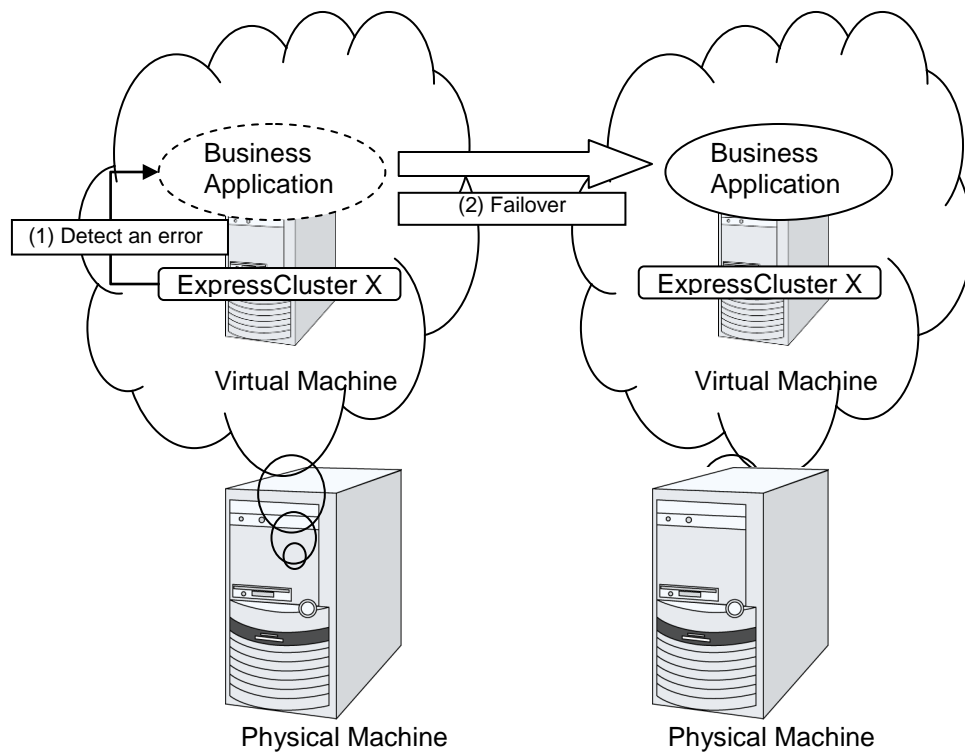


¹ Windows Server 2012 Hyper-V only

Clustering between guest OSs

Install ExpressCluster X on guest OSs, and conduct clustering between virtual machines. By targeting business applications running on guest OSs for failover, the availability of business applications on guest OSs can be enhanced.

The same functions as normal cluster when business applications are failover targets are provided.



Linking guest OSs and host OSs

By installing ExpressCluster X or ExpressCluster X SingleServerSafe on guest and host OSs, it is possible to send a request (such as a failover request), from a cluster of guest OSs to a cluster of host OSs. Conversely, a request can also be sent from a cluster of host OSs to a cluster of guest OSs.

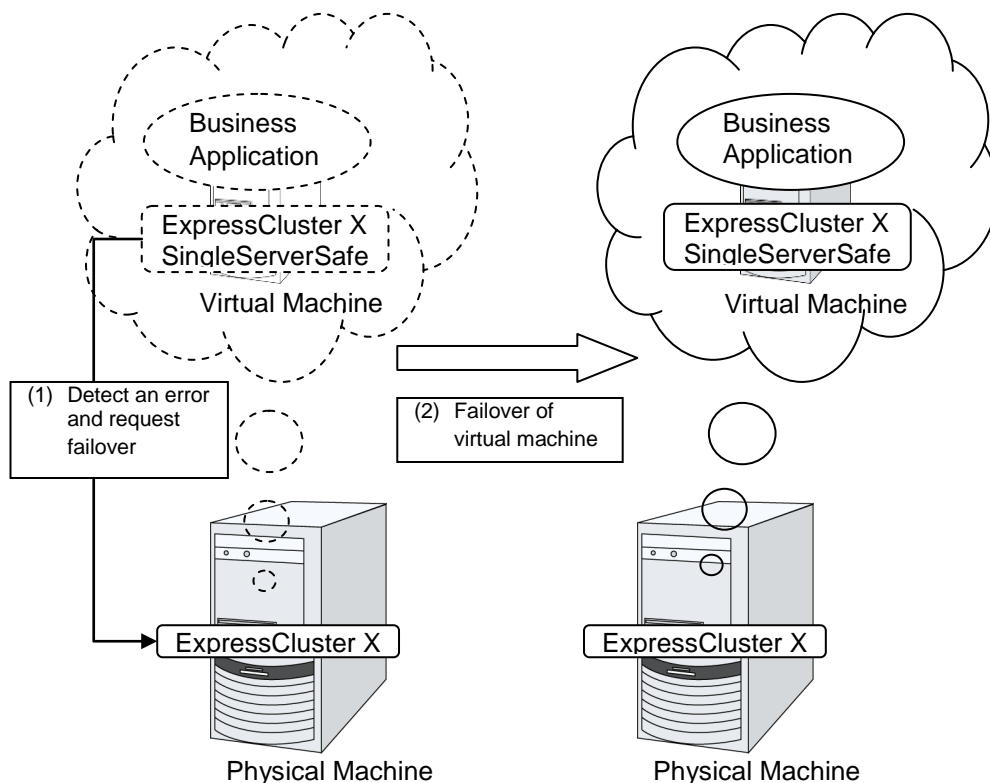
When you are using ExpressCluster X 2.0/2.1, use the `clptrnreq` command to transmit a request.

When you are using ExpressCluster X 3.0 or later, you can use the `clptrnreq` or `clprexec` command to transmit a request.

For example, in the following configuration, error detection by a monitor resource set in ExpressCluster X SingleServerSafe on a guest OS triggers the transmission of a failover request to ExpressCluster X on the host OSs, allowing the guest OS to migrate to the normal physical machine. Even if the guest OS is Linux, it is possible to send a request to ExpressCluster X on the host OSs.

To enable the use of this function, you need to install ExpressCluster X of the following versions or later:

- ExpressCluster X 2.0/2.1 for Windows: Internal version 10.02 or later
- ExpressCluster X 2.0/2.1 for Linux: rpm version 2.0.2-1 or later
- ExpressCluster X 3.0/3.1 for Windows: Internal version 11.00 or later
- ExpressCluster X 3.0/3.1 for Linux: rpm version 3.0.0-1 or later



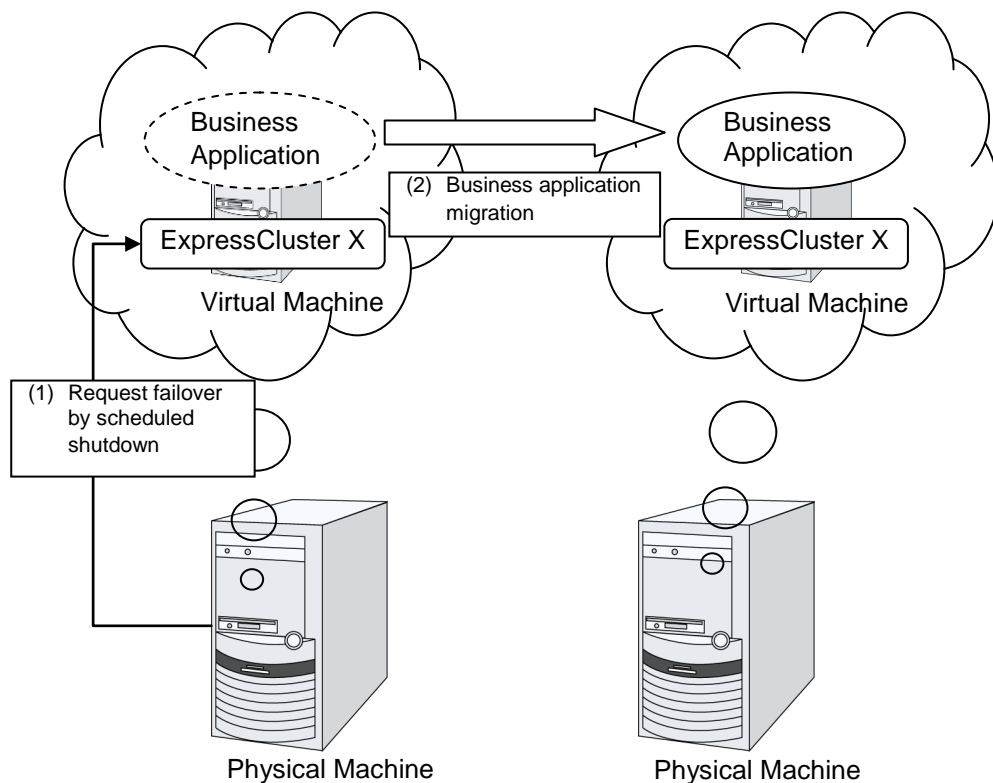
External linkage (for ExpressCluster X 3.0 or later)

With ExpressCluster X 3.0 or later, it is also possible to send a request such as failover, from a server on which ExpressCluster is not installed to a cluster. Use the `clprexec` command for request transmission.

For example, with the following configuration, you can make a group migration request upon events such as scheduled shutdown, from a host OS on which ExpressCluster X is not installed to an inter-guest OS cluster.

To use this function, you must install ExpressCluster of the following versions or later:

- ExpressCluster X 3.0/3.1 for Windows: Internal version 11.00 or later
- ExpressCluster X 3.0/3.1 for Linux: rpm version 3.0.0-1 or later



Chapter 2 Operating Environment

- Hyper-V linkage of ExpressCluster is supported only with x64 Edition of Windows Server 2008 Standard, Enterprise, Datacenter and Windows Server 2008 R2 Standard, Enterprise, Datacenter, Windows Server 2012².

² Windows Server 2012 is supported for ExpressCluster X 3.1 internal version 11.15 or later.

Chapter 3 Notes

Notes on clustering between host OSs (ExpressCluster X 2.0/2.1)

- When setting up an inter-host OS cluster, you must install the integration service on a guest OS.
- If Hyper-V Virtual Machine Management Service (`vmms`) has not started before executing a script (`vmstart.vbs`) to start a virtual machine, the startup of the virtual machine fails. Specify the startup retry count for virtual machine (`W#HyperV6`) in a parameter setting file `W#HyperV.bat` of `vmstart.vbs`.
- Disconnection of disk resource or mirror disk resource may fail when the failover group or cluster is stopped. Specify the disk disconnection error avoidance waiting time (`W#HyperV7`) in `W#HyperV.bat`.
- If you want to shut down the guest OS, rather than suspend it, when the failover group is migrated, change the `W#HyperV8` parameter in `W#HyperV.bat`.
- With Hyper-V 2.0 (Windows Server 2008 R2), it is not possible to perform export of configuration information only. For this reason, the execution of export temporarily requires a capacity equal to double the size of the guest OS disk.
- When a snapshot of the virtual machine has been acquired using Hyper-V Manager, if that server is immediately stopped due to a shutdown or power-off by other than ExpressCluster X, the snapshot will become invalid. You can avoid this problem by moving, migrating, or stopping the virtual machine group after acquiring a snapshot.

Notes on clustering between host OSs (when using Hyper-V 2.0 or earlier and ExpressCluster X 3.0 or later)

- When setting up an inter-host OS cluster, you must install the integration service on a guest OS.
- With Hyper-V 2.0 (Windows Server 2008 R2), it is not possible to perform export of configuration information only. For this reason, the execution of export temporarily requires a capacity equal to twice the size of the guest OS disk.
- **Virtual Machine Stop Waiting Time** for the virtual machine resource is set to 60 seconds by default, but it may take more time to stop the virtual machine depending on the environment. Adjust the time as appropriate, according to your environment.

- When a snapshot of the virtual machine has been acquired using Hyper-V Manager, if that server is immediately stopped due to a shutdown or power-off by other than ExpressCluster, the snapshot will become invalid. You can avoid this problem by moving, migrating, or stopping the virtual machine group after acquiring a snapshot.

Notes on clustering between host OSs (when using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or later)

- When setting up an inter-host OS cluster, you must install the integration service on a guest OS.
- The virtual machine configuration file exported using Hyper-V Manager of Windows Server 2012 Hyper-V cannot be controlled from ExpressCluster. To enable the control of the exported virtual machine, import the configuration file, and then set the path of the file in ExpressCluster.
- Do not use the snapshot function of Hyper-V Manager. ExpressCluster may be unable to normally control the virtual machine for which a snapshot has been captured. If you have created a snapshot by mistake, delete the snapshot.
- When using live migration, set Kerberos for authentication protocol. The number of simultaneous live migrations must be set to more than the number of virtual machines to control.
- When using live migration on Hyper-V, you have to make settings for Active Directory (delegation settings), as well as settings of access privileges for the SMB shared file server.
Also, when creating a virtual machine or when setting up ExpressCluster, specify the path of the SMB shared area using DNS notation of the UNC path (`\\host_name\shared_directory`). If the path is specified using IP address notation (`\\IP_address\shared_directory`), live migration may fail.
- If live migration fails for a reason such as a network disconnection, the virtual machine in question may become unable to start depending on the progress of the live migration at the failure time. If this occurs, restart the virtual machine management service on the destination server.
- In an environment where exclusion control is not performed among simultaneous accesses from different SMB file sharing servers, if a failover due to an event such as a power failure for which normal stop processing is not done on the active system occurs, it may be unable to normally control the virtual machine even when the stopped server recovers. This occurs because the configuration information of the virtual machine is referable from multiple servers. In this case, after confirming that the target virtual machine exists on multiple servers, delete the target virtual machine on the server on which the virtual machine is not running.

- When using live migration, set the request timeout value to a value larger than the time it takes to complete live migration. To set the request timeout, on the **Details** tab displayed for the virtual machine resource, click the **Tuning** button, **Virtual Machine Resource Tuning Properties**, the **Parameter** tab, and then set the time for **Request Timeout**. If the request timeout value is too small, a message indicating live migration failure may be displayed on ExpressCluster even when the live migration is successful.

Notes on clustering between guest OSs

- For setting up a shared disk cluster using clustering between guest OSs, a shared disk device (physical hard disk) is required. A virtual hard disk cannot be used as a shared disk device. In addition, logical drives (LUN) on the shared disk to be used as disk resource must be set to Offline in **Disk Management** on the physical machine.
- Shared disk cluster cannot be set up between guest OSs running on the same physical machine. Because a shared disk is locked during virtual machine startup, an error is output when starting a different virtual machine on the same physical machine, and the startup fails.
- When setting up a shared disk cluster or mirror disk cluster between instances of SUSE Linux Enterprise Server 10 with Service Pack 1, connect a virtual hard disk or physical hard disk to IDE controllers. By default, DVD drive is connected to IDE controller 1. After installing OS, delete the DVD drive, and then connect a virtual hard disk or physical hard disk to IDE controller 1.
- When the cluster configuration is used, do not **Pause** or **Save** a virtual machine. If a virtual machine is paused or saved, ExpressCluster detects a heartbeat timeout, and starts a failover group on a different server. In this situation, both virtual machines where the failover group is running are shut down when executing **Startup** for the virtual machine that was paused or saved, in terms of data protection.
- Clustering between guest OSs for which the virtualization infrastructure is Windows Server 2012 Hyper-V is not verified.

Chapter 4 Virtualization Environment Setup Procedure

Installing Hyper-V

Before setting up an inter-host OS cluster or an inter-guest OS cluster, the role of Hyper-V must be added to the host OS. Add the role of Hyper-V by following the procedure below:

For Windows Server 2008 R2 or earlier:

- (1) Start up **Server Manager** from **Management Tools** of the **Start** menu, and then click **Add a Role**.
- (2) Select **Hyper-V** from Roles, and then click **Next**.
- (3) Notes are displayed. Click **Next** after reading the notes.
- (4) Virtual network creation window appears. Select a network adapter³ to be used for the virtual machine, and then click **Next**.
- (5) Click **Install**.
- (6) Restart the physical machine after the installation.
- (7) If updates for Hyper-V are available, apply them as necessary.

For Windows Server 2012:

- (1) Start up **Server Manager**, and then click **Add Roles and Features**.
- (2) **Before You Begin** appears. Make sure that the tasks have been completed, and then click **Next**.
- (3) Select **Role-based or feature-based installation**, and then click **Next**.
- (4) Choose **Select a server from the server pool**. Select a server on which to install Hyper-V from **Server Pool**, and then click **Next**.
- (5) Select **Hyper-V** from **Roles**.
- (6) A pop-up window appears. Select **Include management tools (if applicable)**, and then click **Add Features**.
- (7) Make sure that **Hyper-V** is selected, and then click **Next**.
- (8) It is not necessary to add **Features**. Click **Next**.
- (9) Notes are displayed. Read those notes, and then click **Next**.
- (10) A virtual switch creation dialog box appears. Select the network adapter³ to be used on the virtual machine, and then click **Next**.
- (11) Live migration setting window appears. For live migration setting, see “Using live migration.” Click **Next**.
- (12) Set the default location in which to store the virtual machine, and then click **Next**.

³ For ExpressCluster use, setting up two LAN systems as a heartbeat path between clusters is recommended. When setting up an inter-guest OS cluster, prepare two network adapters to be used by virtual machines. In addition, when setting up an inter-host OS cluster, assign the same virtual network name to the guest OS.

- (13) Click **Install**.
- (14) Restart the physical machine after the installation.
- (15) If updates for Hyper-V are available, apply them as necessary.

Using live migration

Windows Server 2012 allows you to use live migration. To use live migration, make settings by following the procedure below after installing Hyper-V.

- (1) Start up **Hyper-V Manager**.
- (2) Select the target host OS in the left pane and right-click it.
- (3) Click **Hyper-V Settings** in the shortcut menu.
- (4) Click **Live Migrations** in the left pane.
- (5) Live migration setting window appears. Select **Enable incoming and outgoing live migrations**, and then select **Use Kerberos** for **Authentication protocol**. Also, enter a value larger than the number of virtual machines to control in **Simultaneous live migrations**. Click **OK**.
- (6) Make settings for Active Directory (delegation settings), as well as settings of access privileges for the SMB shared area in which to store the virtual machine.

Chapter 5 Cluster Environment Setup Procedure

Setting Up Inter-Host OS Cluster (ExpressCluster X 2.0/2.1)

The procedure for setting up an inter-host OS cluster is as follows:

1. Install ExpressCluster X
2. Set up a cluster that has the disk type resource
3. Create a virtual machine
4. Export the virtual machine and then remove the exported virtual machine
5. Register a script resource for virtual machine control
6. Upload the configuration information

In this document, the failover group name, resource name, and monitoring resource name are specified as shown in the table below for convenience. Change these names in accordance with your environment as necessary.

Group, Resource, or Monitoring Resource	Name
Failover Group	failover-vm
Disk Resource or Mirror Disk Resource	sd-vm/md-vm
Script Resource	script-vm
Custom Monitoring Resource	genw-vm
Disk RW Monitoring Resource	diskw-vm

Set up an inter-host OS cluster by following the procedure below:

- (1) Install ExpressCluster X on each host OS as described in the *ExpressCluster X Install and Configuration Guide*.
- (2) Create a failover group (failover-vm) with a disk resource (sd-vm) or mirror disk (md-vm) as described in the *ExpressCluster X Install and Configuration Guide*. In addition, check if the failover group can start up normally.
- (3) Start up the failover group on the host OS that creates a virtual machine.
- (4) Create a virtual machine in any location⁴.
 - A) Start up **Hyper-V Manager** from **Management Tools** on the **Start** menu.
 - B) Right-click a host OS name displayed in Hyper-V Manager, and then click **Virtual Machine** on the **New** menu.
 - C) Enter any virtual machine name for **Name**. Specify where to save the virtual machine configuration information, and then click **Next**.

⁴ If Hyper-V 1.0 is used, the export of configuration information only (that is, transition processing in which the copying of the virtual hard disk is not necessary) is possible; we recommend that you create a virtual machine on a shared disk or mirror disk in advance.

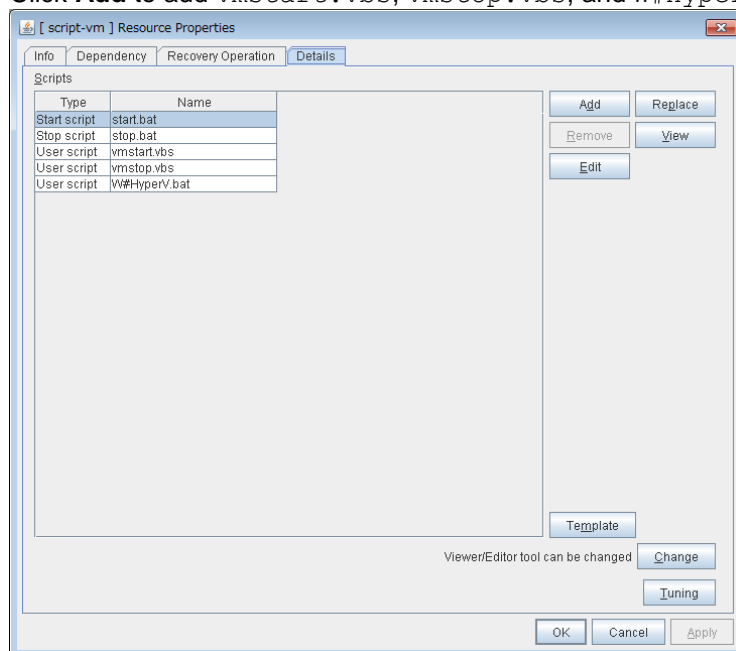
- D) Specify the amount of memory assigned to the virtual machine, and then click **Next**.
 - E) Set Network Adapter to **Not Connect**, and then click **Next**.
 - F) Specify the name of the virtual hard disk (vhd file) and the storage location, and then click **Next**.
 - G) Select the installation media, and then click **Next**.
 - H) Select **Start Virtual Machine for Creation**, and then click **Finish**.
 - I) Guest OS installation starts. Install the guest OS
 - J) After the guest OS is installed, install the integration service⁵.
 - K) Specify an IP address and computer name for the guest OS as necessary.
- (5) Configure a network adapter and auto start action for the virtual machine.
- A) If the virtual machine is running, shut down the virtual machine.
 - B) Right click the failover target virtual machine name displayed on Hyper-V Manager, and then click **Configure**.
 - C) Click **Network Adapter** under **Hardware** in the left pane. Select a network adapter to be assigned to the virtual machine from the pull-down menu of **Network**. Select **Static** for MAC address settings, and then assign a MAC address.
 - D) Click **Auto Start Action** under **Management** in the left pane. Select **Do nothing** as auto start action.
 - E) Click **OK** when the settings above are completed.
- (6) Export a failover target virtual machine to a shared disk or mirror disk.
- A) If the failover target virtual machine is running, shut down the virtual machine.
 - B) Right-click the export target virtual machine name displayed in Hyper-V Manager, and then click **Export**. The export setting differs slightly between Hyper-V 1.0 and Hyper-V 2.0.
 - For Hyper-V 1.0
When the virtual machine configuration information and virtual hard disk are on a shared disk or mirror disk, enter a folder path (for example, Z:\Hyper-V\export), other than the path to the folder where the virtual machine configuration information is saved as an export path. Select **Export only virtual machine configuration**, and then click **Export**.
- When the virtual machine configuration information and virtual hard disk are not on a shared disk or mirror disk
Enter a path to a folder on a shared disk or mirror disk (for example, Z:\Hyper-V) as an export path. Make sure to select **Export only virtual machine configuration**, and then click **Export**.

⁵ **The integration service is required** to manage a virtual machine by using a virtual machine resource.

- For Hyper-V 2.0
Enter any folder (for example, `Z:\Hyper-V`) on a shared disk or mirror disk as the file storage destination, and then click **Export**.
- (7) Remove a virtual machine by using Hyper-V Manager. Right click the exported virtual machine name, and then click **Remove**.⁶
 - (8) With Hyper-V 1.0, only when a virtual machine already saved on a shared disk or mirror disk was exported, the following steps are required. Copy the folder created under `Z:\Hyper-V\export` by exporting with step (6) above to the folder where the virtual machine configuration information was saved (for example, `Z:\Hyper-V\virtual_machine_name`). Check that each file and folder are copied as shown below.
 - `config.xml` file exists under `Z:\Hyper-V\virtual_machine_name`.
 - `guest_OS_ID.exp` file and a folder with the same name as the file exist under `Z:\Hyper-V\virtual_machine_name\Virtual Machines`.
 - (9) Start up ExpressCluster Builder.
 - (10) Create a VBScript to start and stop a virtual machine from ExpressCluster. Create `vmstart.vbs`, `vmstop.vbs`, and `W#HyperV.bat`, as referring to sample scripts described at the end of this chapter, and save them to any location on the machine where ExpressCluster Builder is running.
 - (11) Create a VBScript to confirm virtual machine power status from ExpressCluster. Create `checkstatus.vbs` as referring to sample scripts described at the end of this chapter, and save it to any location on each physical machine (for example, `C:\Program Files\ExpressCluster\bin`).
 - (12) Use ExpressCluster Builder to add the script resource to the failover group and edit the resource.
 - A) In the ExpressCluster Builder window, right click the name of the failover group (`failover-vm`) in the tree view, and then click **Add Resource**.
 - B) Select **Script Resource** from **Type**. Specify a name for the script resource (`script-vm`), and then click **Next**.
 - C) Select **Start script**, and then click **Edit**. Editor starts, and then edit `start.bat` as referring to sample scripts described at the end of this chapter. Save the edited contents, and then close the editor.
 - D) Select **Stop script**, and then click **Edit**. Editor starts, and then edit `stop.bat` as referring to sample scripts described at the end of this chapter. Save the edited contents, and then close the editor.

⁶ When virtual machine configuration information saved on other than a shared disk or mirror disk was exported, the virtual hard disk is no longer needed. Remove it manually.

- E) Click **Add** to add `vmstart.vbs`, `vmstop.vbs`, and `W#HyperV.bat`.



- (13) Use ExpressCluster Builder to add a custom monitoring resource⁷ to a cluster.
- In the ExpressCluster Builder window, right click **Monitors** in the tree view, and then click **Add a Monitor Resource**.
 - Select **Custom Monitoring** from **Type**. Specify a name for the custom monitoring resource (`genw-vm`), and then click **Next**.
 - Click **Edit** to edit `genw.bat`. Editor starts, and then edit `genw.bat` as referring to sample scripts described at the end of this chapter. Save the edited contents and close the editor, and then click **Next**.
 - Select **When Active as Monitoring Timing**, and then click **Browse**. When a list of selectable resources appears, select the script resource (`script-vm`) you have just created.⁸ Check that the script resource is specified for the target resource, and then click **Next**.
 - Select the failover group name (`script-vm`) that includes the script resource (`failover-vm`) as the **Recovery Target**, and then click **OK**. Edit parameters such as **Reactivation Threshold Value** as necessary, and then click **Finish**.
- (14) Add a disk RW monitor resource to the cluster
- In the ExpressCluster Builder window, right-click **Monitors** in the tree view, and then click **Add Monitor Resource**.
 - Select **Disk RW Monitor** from **Type**. Specify a name for the disk RW monitor resource (`diskw-vm`), and then click **Next**.
 - For **File Name**, specify a path on the shared disk or mirror disk. Specify an absolute path for the file. The specified file is automatically created. After making this setting, click **Next**.

⁷ To use custom monitoring resource, updates (CPRO-XW020-01 or later) of EXPERSCLUSTER must be applied.

⁸ By this setting, the guest OS will be monitored immediately after the script is started, in other words, immediately after the guest OS is started. In addition, the guest OS monitoring will be stopped when the script resource stopping process is started, in other words, when the guest OS saving process is started.

- D) Select **Activating** for Monitor Timing, and then specify disk resource or mirror disk resource as the target resource. After making this setting, click **Next**.
 - E) Select Cluster for **Recovery Target**. Select **Stop the cluster service and shutdown OS** for **Final Action**. When this setting is complete, click **Finish**.
- (15) Stop the failover group (*failover-vm*) by using WebManager or the `clpgrp` command.
- (16) Suspend the cluster by using WebManager or the `clpcl` command.
- (17) Upload the configuration information created using ExpressCluster Builder. To upload the information, select **Upload Information File** from the Builder **File** menu.
- (18) After uploading the configuration information, proceed to *Checking Operations of Inter-Host OS Clusters (ExpressCluster X 2.0/2.1)*.

Checking Operations of Inter-Host OS Clusters (ExpressCluster X 2.0/2.1)

- (1) Resume the cluster by using WebManager or the `clpcl` command.
- (2) Start up the failover group by using WebManager or the `clpgrp` command. On the server where the failover group is running, use Hyper-V Manager to check if the guest OS is running.
- (3) Migrate the failover group by using WebManager or the `clpgrp` command. On the server where the failover group was migrated to, use Hyper-V Manager to check if the guest OS is running.
- (4) Shut down or reboot the physical machine where the failover group is running by using WebManager or the `clpdown` command. At this point, use Hyper-V Manager to check that the failover group was migrated to a different server, and the guest OS is running.
- (5) Check if a reactivation of recovery target or a failover is performed when `genw-vm` detects an abnormality caused by a guest OS shutdown using Hyper-V Manager. In addition, use Hyper-V Manager to check if the guest OS is restarted after a failover.
- (6) Use Hyper-V Manager to check if other server detects a shutdown of a different server, starts a failover, and the guest OS is restarted when the physical machine is shut down from other than ExpressCluster.
- (7) In addition to the above, perform items described in *ExpressCluster X Install and Configuration Guide, Chapter 7, Checking Operations, Performing Operation Verification Tests* as necessary.

Setting Up Inter-Host OS Cluster (When Using Hyper-V 2.0 or Earlier and ExpressCluster X 3.0 or Later)

The procedure for setting up an inter-host OS cluster is as follows:

1. Install ExpressCluster X.
2. Set up a cluster that has the disk type resource.
3. Create a virtual machine.
4. Export the virtual machine and then remove the exported virtual machine.
5. Register a virtual machine resource.
6. Apply the configuration information.

In this document, the failover group name, resource name, and monitoring resource name are specified as shown in the table below for convenience. Change these names in accordance with your environment as necessary.

Group, Resource, or Monitoring Resource	Name
Virtual Machine Group	virtualmachine
Virtual Machine Resource	vm
Any of following resources	
Hybrid Disk Resource	hd-vm
Mirror Disk Resource	md-vm
NAS Resource	nas-vm
Disk Resource	sd-vm
Virtual Machine Monitor Resource	vmw1 A virtual machine monitor resource is automatically added when a virtual machine resource has been created.

Set up an inter-host OS cluster by following the procedure below:

- (1) Install ExpressCluster X on each host OS as described in the *ExpressCluster X Install and Configuration Guide*.
- (2) Start up Builder and add each server to a cluster as described in the *ExpressCluster X Install and Configuration Guide*.
- (3) Add a virtual machine group to the cluster.
 - A) Select **Virtual Machine** for group type. After setting a **Name**, click **Next**.
 - B) Set the startup servers, and then click **Next**.
 - C) Set the group attributes, and then click **Next**.
 - D) As the group resource, set any one of the following resources, for storing a virtual machine disk: hybrid disk resource, mirror disk resource, NAS resource, and disk resource. At this time, do not add this virtual machine resource.
- (4) After creating the configuration information, select **Apply the Configuration File** from the Builder **File** menu.

- (5) Start up the cluster by using WebManager or the `clpocl` command.
- (6) Start up the failover group on the host OS that creates a virtual machine.
- (7) Create a virtual machine in any location⁹.
 - A) Start up **Hyper-V Manager** from **Management Tools** on the **Start** menu.
 - B) Right-click a host OS name displayed in Hyper-V Manager, and then click **Virtual Machine** on the **New** menu.
 - C) Enter any virtual machine name for **Name**. Specify where to save the virtual machine configuration information, and then click **Next**.
 - D) Specify the amount of memory assigned to the virtual machine, and then click **Next**.
 - E) Set Network Adapter to **Not Connect**, and then click **Next**.
 - F) Specify the name of the virtual hard disk (`vhd` file) and the storage location, and then click **Next**.
 - G) Select the installation media, and then click **Next**.
 - H) Select **Start Virtual Machine for Creation**, and then click **Finish**.
 - I) Guest OS installation starts. Install the guest OS.
 - J) After the guest OS is installed, install the integration service¹⁰.
 - K) Specify an IP address and computer name for the guest OS as necessary.
- (8) Configure a network adapter and auto start action for the virtual machine.
 - A) If the virtual machine is running, shut down the virtual machine.
 - B) Right click the failover target virtual machine name displayed on Hyper-V Manager, and then click **Configure**.
 - C) Click **Network Adapter** under **Hardware** in the left pane. Select a network adapter to be assigned to the virtual machine from the pull-down menu of **Network**. Select **Static** for MAC address settings, and then assign a MAC address.
 - D) Click **Auto Start Action** under **Management** in the left pane. Select **Do nothing** as auto start action.
 - E) Click **OK** when the settings above are completed.
- (9) Export a failover target virtual machine to a shared disk or mirror disk.
 - A) If the failover target virtual machine is running, shut down the virtual machine.
 - B) Right-click the export target virtual machine name displayed in Hyper-V Manager, and then click **Export**. The export setting differs slightly between Hyper-V 1.0 and Hyper-V 2.0.

⁹ If Hyper-V 1.0 is used, the export of configuration information only (that is, transition processing in which the copying of the virtual hard disk is not necessary) is possible; we recommend that you create a virtual machine on a shared disk or mirror disk in advance.

¹⁰ **The integration service is required** to manage a virtual machine by using a virtual machine resource.

- For Hyper-V 1.0
When the virtual machine configuration information and virtual hard disk are on a shared disk or mirror disk, enter a folder path (for example, `Z:\Hyper-V\export`), other than the path to the folder where the virtual machine configuration information is saved as an export path. Select **Export only virtual machine configuration**, and then click **Export**.

When the virtual machine configuration information and virtual hard disk are not on a shared disk or mirror disk

Enter a path to a folder on a shared disk or mirror disk (for example, `Z:\Hyper-V`) as an export path. Make sure to select **Export only virtual machine configuration**, and then click **Export**.

- For Hyper-V 2.0
Enter any folder (for example, `Z:\Hyper-V`) on a shared disk or mirror disk as the file storage destination, and then click **Export**.

(10) Remove a virtual machine by using Hyper-V Manager. Right click the exported virtual machine name, and then click **Remove**.¹¹

(11) With Hyper-V 1.0, only when a virtual machine already saved on a shared disk or mirror disk was exported, the following steps are required. Copy the folder created under `Z:\Hyper-V\export` by exporting with step (6) above to the folder where the virtual machine configuration information was saved (for example, `Z:\Hyper-V\virtual_machine_name`). Check that each file and folder are copied as shown below.

- `config.xml` file exists under `Z:\Hyper-V\virtual_machine_name`.
- `guest_OS_ID.exp` file and a folder with the same name as the file exist under `Z:\Hyper-V\virtual_machine_name\Virtual Machines`.

(12) Start up ExpressCluster Builder.

¹¹ When virtual machine configuration information saved on other than a shared disk or mirror disk was exported, the virtual hard disk is no longer needed. Remove it manually.

- (13) Use ExpressCluster Builder to add the virtual machine resource to the virtual machine group.
 - A) Right-click the virtual machine group name (`virtualmachine`) and then click **Add Resource**.
 - B) Select **Virtual Machine Resource** from Type, set a name (`vm`), and then click **Next**.
 - C) Set Dependency, and then click **Next**.
 - D) Set Recovery Operation at Activation/Deactivation Failure Detection, and then click **Next**.
 - E) Select **Hyper-V** from **Virtual Machine Type**. For **Virtual Machine Name**, enter a virtual machine name to be displayed on Hyper-V Manager. For **VM Configuration File Path**, enter an absolute path to the exp file exported in step (11) above. Even if the path includes space(s), you do not have to enclose the path in double quotes ("). Click **Finish**.
- (14) After creating the configuration information, select **Apply the Configuration File** from the Builder **File** menu.
- (15) After applying the configuration information, proceed to *Checking Operations of Inter-Host OS Cluster (When Using Hyper-V 2.0 or Earlier and ExpressCluster X 3.0 or Later)*.

Checking Operations of Inter-Host OS Cluster (When Using Hyper-V 2.0 or Earlier and ExpressCluster X 3.0 or Later)

- (1) Resume the cluster by using WebManager or the `clpcl` command.
- (2) Start up the virtual machine group by using WebManager or the `clpgrp` command. On the server where the virtual machine group is running, use Hyper-V Manager to check if the guest OS is running.
- (3) Migrate the virtual machine group by using WebManager or the `clpgrp` command. On the server where the virtual machine group was migrated to, use Hyper-V Manager to check if the guest OS is running.
- (4) Migrate the virtual machine group by using WebManager or the `clpgrp` command. On the server to which the virtual machine group was migrated, use Hyper-V Manager to check whether the guest OS is running.
- (5) Shut down or reboot the physical machine where the virtual machine group is running by using WebManager or the `clpdown` command. At this point, use Hyper-V Manager to check that the failover group was migrated to a different server, and the guest OS is running.
- (6) Check if a reactivation of recovery target or a failover is performed when `vmw1` detects an abnormality caused by a guest OS shutdown using Hyper-V Manager. In addition, use Hyper-V Manager to check if the guest OS is restarted after a failover.
- (7) Use Hyper-V Manager to check if other server detects a shutdown of a different server, starts a failover, and the guest OS is restarted when the physical machine is shut down from other than ExpressCluster.
- (8) In addition to the above, perform items described in *ExpressCluster X Install and Configuration Guide, Chapter 7, Checking Operations, Performing Operation Verification Tests* as necessary.

Setting Up Inter-Host OS Cluster (When Using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or Later)

The procedure for setting up an inter-host OS cluster is as follows:

1. Install ExpressCluster X.
2. Set up a cluster that has the disk type resource.
3. Create a virtual machine.
4. Export the virtual machine and then remove the exported virtual machine.
5. Register a virtual machine resource.
6. Apply the configuration information.

In this document, the failover group name, resource name, and monitoring resource name are specified as shown in the table below for convenience. Change these names in accordance with your environment as necessary.

Group, Resource, or Monitoring Resource	Name
Virtual Machine Group	virtualmachine
Virtual Machine Resource	vm
Any of following resources	
Hybrid Disk Resource	hd-vm
Mirror Disk Resource	md-vm
NAS Resource	nas-vm
Disk Resource	sd-vm
Virtual Machine Monitor Resource	vmw1 A virtual machine monitor resource is automatically added when a virtual machine resource has been created.

Set up an inter-host OS cluster by following the procedure below:

- (1) Install ExpressCluster X on each host OS as described in the *ExpressCluster X Install and Configuration Guide*.
- (2) Start up Builder and add each server to a cluster as described in the *ExpressCluster X Install and Configuration Guide*.
- (3) Add a virtual machine group to the cluster.
 - A) Select **Virtual Machine** for group type. After setting a **Name**, click **Next**.
 - B) Set the startup servers, and then click **Next**.
 - C) Set the group attributes, and then click **Next**.
 - D) As the group resource, set any one of the following resources, for storing a virtual machine disk: hybrid disk resource, mirror disk resource, NAS resource, and disk resource. At this time, do not add this virtual machine resource. Adding the above resources is not necessary when using live migration.
- (4) After creating the configuration information, select **Apply the Configuration File** from the Builder **File** menu.
- (5) Start up the cluster by using WebManager or the `clpctl` command.

- (6) Start up the virtual machine group on the host OS that creates a virtual machine.
- (7) Create a virtual machine on the shared disk or mirror disk that is a target for failover.

When not using live migration:

- A) Start up **Hyper-V Manager**.
- B) Right-click a host OS name displayed in Hyper-V Manager, and then click **Virtual Machine** on the **New** menu.
- C) **Before You Begin** appears. Click **Next**.
- D) Enter any virtual machine name for **Name**. Also, select **Store the virtual machine in a different location**, specify the path to store the virtual machine disk of the resource set in D) of (3), and then click **Next**.
- E) Specify the amount of memory assigned to the virtual machine, and then click **Next**.
- F) Set Network Adapter to **Not Connect**, and then click **Next**.
- G) Specify the name of the virtual hard disk (*vhdx* file) and the path to store the virtual machine disk of the resource set in D) of (3) as the storage location, and then click **Next**.
- H) Select the installation media, and then click **Next**.
- I) Confirm the displayed explanation, and then click **Finish**.
- J) Guest OS installation starts. Install the guest OS.
- K) After the guest OS is installed, install the integration service¹².
- L) Specify an IP address and computer name for the guest OS as necessary.

When using live migration:

- A) Start up **Hyper-V Manager**.
- B) Right-click a host OS name displayed in Hyper-V Manager, and then click **Virtual Machine** on the **New** menu.
- C) **Before You Begin** appears. Click **Next**.
- D) Enter any virtual machine name for **Name**. Also, select **Store the virtual machine in a different location**, specify the SMB shared area to store the virtual machine disk¹³, and then click **Next**.
- E) Specify the amount of memory assigned to the virtual machine, and then click **Next**.
- F) Set Network Adapter to **Not Connect**, and then click **Next**.
- G) Specify the name of the virtual hard disk (*vhdx* file) and the SMB shared area to store the virtual machine disk as the storage location¹³, and then click **Next**.
- H) Select the installation media, and then click **Next**.
- I) Confirm the displayed explanation, and then click **Finish**.
- J) Guest OS installation starts. Install the guest OS.
- K) After the guest OS is installed, install the integration service¹².
- L) Specify an IP address and computer name for the guest OS as necessary.

- (8) Configure a network adapter and auto start action for the virtual machine.
 - A) If the virtual machine is running, shut down the virtual machine.
 - B) Right click the failover target virtual machine name displayed on Hyper-V Manager, and then click **Configure**.
 - C) Click **Network Adapter** under **Hardware** in the left pane. Select a network adapter to be assigned to the virtual machine from the pull-down menu of **Network**. Select **Static** for MAC address settings, and then assign a MAC address.
 - D) Click **Auto Start Action** under **Management** in the left pane. Select **Do nothing** as auto start action.
 - E) Click **OK** when the settings above are completed.

¹² **The integration service is required** to manage a virtual machine by using a virtual machine resource.

¹³ Use DNS notation of the UNC path. See the relevant note in Chapter 3.

- (9) If the virtual machine to be a target for failover is running, shut down the virtual machine.
- (10) Start up ExpressCluster Builder.
- (11) On ExpressCluster Builder, add a virtual machine resource to the virtual machine group.
 - A) Right-click the virtual machine group name (`virtualmachine`) and click **Add Resource**.
 - B) Select **Virtual Machine Resource** for **Type**, set a name (`vm`), and then click **Next**.
 - C) Set Dependency, and then click **Next**.
 - D) Set Recovery Operation at detection of activation/deactivation failure, and then click **Next**.
 - E) Select **Hyper-V** for **Virtual Machine Type**. In **Virtual Machine Name**, enter the virtual machine name to be displayed on Hyper-V Manager. In **VM Configuration File Path**, enter the absolute path¹⁴ of the XML file created in (11) above. Even if the path contains spaces, it is not necessary to input a double quote mark (") at the beginning and end of the line. Click **Finish**.
- (12) Set the type of migration.

When not using live migration:

 - A) On ExpressCluster Builder, right-click the cluster name and select **Properties**.
 - B) Select **Quick Migration** from the pull-down menu of **Migration Type** on the **Migration** tab.

When using live migration:

 - A) On ExpressCluster Builder, right-click the cluster name and select **Properties**.
 - B) Select **Live Migration** from the pull-down menu of **Migration Type** on the **Migration** tab, and then set **Account** and **Password**.
Set an **Account** in the "`domain_name\account_name`" format.
- (13) After creating the configuration information, select **Apply the Configuration File** from the Builder **File** menu.
- (14) Start up the virtual machine group by using WebManager or the `clpgrp` command. If this is the first time to start up the group, you must start it up on the server where the virtual machine has been created.
- (15) If you have checked with Hyper-V Manager that the guest OS is running on the server where the virtual machine group is running, stop the virtual machine group by using WebManager or the `clpgrp` command. If you have not stopped the virtual machine group even once on the server you started it for the first time, the virtual machine group may be unable to be started on the standby server.
- (16) Proceed to *Checking Operations of Inter-Host OS Clusters (When Using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or Later)*.

¹⁴ In the case of the SMB shared area, use DNS notation of the UNC path.

Checking Operations of Inter-Host OS Clusters (When Using Windows Server 2012 Hyper-V and ExpressCluster X 3.1 (11.15) or Later)

- (1) Start up the virtual machine group by using WebManager or the `clpgrp` command. If this is the first time to start up the group, you must start it up on the server where the virtual machine has been created. On the server where the virtual machine group is running, use Hyper-V Manager to check if the guest OS is running.
- (2) Move the virtual machine group by using WebManager or the `clpgrp` command. On the server to which the virtual machine group was moved, use Hyper-V Manager to check if the guest OS is running.
- (3) Migrate the virtual machine group by using WebManager or the `clpgrp` command. On the server to which the virtual machine group was migrated, use Hyper-V Manager to check whether the guest OS is running.
- (4) Shut down or reboot the physical machine where the virtual machine group is running by using WebManager or the `clpdown` command. At this point, use Hyper-V Manager to check that the failover group was migrated to a different server, and the guest OS is running.
- (5) Check if a reactivation of recovery target or a failover is performed when `vmw1` detects an abnormality caused by a guest OS shutdown using Hyper-V Manager. In addition, use Hyper-V Manager to check if the guest OS is restarted after a failover.
- (6) Use Hyper-V Manager to check if other server detects a shutdown of a different server, the virtual machine group starts, and the guest OS is restarted when the physical machine is shut down from other than ExpressCluster.
- (7) In addition to the above, perform items described in *ExpressCluster X Install and Configuration Guide, Chapter 7, Checking Operations, Performing Operation Verification Tests* as necessary.

Setting Up an Inter-Guest OS Cluster

The procedure for setting up an inter-host OS is as follows:

1. Create a virtual machine
2. Install ExpressCluster X on the guest OSs
3. Set up a cluster using ExpressCluster Builder

Set up an inter-guest OS cluster by following the procedure below:

(1) Create a virtual machine in any location.

- A) Start up **Hyper-V Manager** from **Management Tools** on the **Start** menu.
- B) Right-click a host OS name displayed in Hyper-V Manager, and then click **Virtual Machine** on the **New** menu.
- C) Enter any virtual machine name for **Name**. Specify where to save the virtual machine configuration information, and then click **Next**
- D) Specify the amount of memory assigned to the virtual machine, and then click **Next**.
- E) Select **Connect** and specify the network adapter to use for the communication between the clusters and then click **Next**.
- F) Specify the name of the virtual hard disk (`vhd` file) and the storage location, and then click **Next**.
- G) Select the installation media, and then click **Next**.
- H) Select **Start Virtual Machine for Creation**, and then click **Finish**.
- I) Guest OS installation starts. Install the guest OS.
- J) After the guest OS is installed, install the integration service.
- K) Specify an IP address and computer name for the guest OS as necessary.

(2) Add a disk for disk resource or mirror disk resource.

- A) If the virtual machine is running, shut down the virtual machine.
- B) Right click a virtual machine displayed on Hyper-V Manager and then click **Configure**.
- C) Click **Add Hardware** of **Hardware** in the left pane, select **SCSI Controller**, and then click **Add**.
- D) Click **SCSI Controller** added to the list of **Hardware** in the left pane, and then click **Add**.
 - For disk resource

Only physical hard disks¹⁵ can be used as disk resource. Select a logical drive on the shared disk connected to the physical machine. Specify the same logical drive for all virtual machines.
 - For mirror disk resource

Either virtual or physical hard disks¹⁶ can be selected as mirror disk resource.

(3) Add any other hardware as appropriate.

(4) Start up the virtual machine once the virtual machine setting is complete.

¹⁵ To use a physical hard disk, the disk must be offline. Use **Disk Management** on the physical machine to check if the disk is offline.

¹⁶ Check if the disk is offline as described above.

- (5) Install a guest OS supported by ExpressCluster on the virtual machine as described in *ExpressCluster X Install and Configuration Guide*.
- (6) Set up a cluster using ExpressCluster Builder as described in *ExpressCluster X Install and Configuration Guide*.
- (7) Apply the configuration information created using ExpressCluster Builder.

Checking Operations of an Inter-Guest OS Cluster

- (1) Start up the cluster by using WebManager or the `clpcl` command.
- (2) Migrate the failover group by using WebManager or the `clpgrp` command. On the server where the failover group was migrated to, check if the failover group is started by using WebManager or the `clpstat` command.
- (3) Shut down or reboot the virtual machine where the failover group is running by using WebManager or the `clpdown` command. At this point, check if the failover group is started on a different server by using WebManager or the `clpstat` command.
- (4) Use WebManager or the `clpstat` command to check if a server detects a shutdown of a partner server, and starts the failover group when the physical machine is shut down from other than ExpressCluster.
- (5) In addition to the above, perform items described in *ExpressCluster X Install and Configuration Guide, Chapter 7, Checking Operations, Performing Operation Verification Tests* as necessary.

Establishing Links between Guest OSs and Host OSs

This section describes how to make the settings for sending a virtual machine failover request to a host OS when an abnormality is detected by ExpressCluster X SingleServerSafe on a guest OS.

The flow of the procedure for establishing links between the guest OSs and host OSs is as follows:

1. Set up an inter-host OS cluster
2. Install ExpressCluster X SingleServerSafe on the guest OSs
3. Make the settings for sending a failover request to ExpressCluster on the host OSs when an abnormality is detected by the monitor resource set in ExpressCluster X SingleServerSafe on the guest OS

Establish links between the guest OSs and host OSs by following the procedure below:

- (1) Set up an inter-host OS cluster as described in *Setting Up Inter-Host OS Cluster (ExpressCluster X 2.0/2.1)* or *Setting Up Inter-Host OS Cluster (When Using Hyper-V 2.0 or Earlier and ExpressCluster X 3.0 or Later)*.
- (2) Install ExpressCluster X SingleServerSafe on the guest OSs.
- (3) Edit the configuration information of ExpressCluster X SingleServerSafe on the guest OS.
 - A) Start up ExpressCluster Builder.
 - B) In the ExpressCluster Builder window, right-click **Monitors**, and then click **Add Monitor Resource**.
 - C) Select any monitor resource from **Type**.
 - D) Set parameters as per the instructions in the window. Proceed to the screen for setting **Final Action**.
 - E) Select **No operation** for **Final Action**.
 - F) Select **Execute Script before Final Action** and then click **Set**.
 - G) Click **Edit**, and then edit `preaction.bat` by referring to the sample script provided at the end of this guide.
 - H) When the setting is complete, click **Finish**.
- (4) Apply the configuration information created using ExpressCluster Builder.

Checking the Linkage between Guest OSs and Host OSs

- (1) Start up the cluster by using WebManager or the `clpctl` command.
- (2) When using WebManager, check that the status of the monitor resource set in the previous section is **Normal**. Or, check for **Normal** by executing the `clpstat` command.
- (3) Trigger a pseudo error (such as application stop, service stop, or network disconnection) to enable the above monitor resource to detect an abnormality.
- (4) Make sure that, after an error is detected, ExpressCluster X on the host OSs receives a failover request and fails over the virtual machine.

Setting External Monitor Linkage and Checking Operations

This section explains how to make the settings needed to send a failover request from a host OS on which ExpressCluster is not installed to an inter-guest OS cluster.

1. Set up an inter-guest OS cluster.
2. Store the `clprexec` command in any location under the host OS

Set up an inter-host OS cluster by following the procedure below:

- (1) Set up an inter-guest OS cluster as described in *Setting Up an Inter-Guest OS Cluster*.
- (2) Store the `clprexec` command in any location on the host OS. The `clprexec` command can be found in the ExpressCluster installation folder or in the installation CD.

- `ExpressCluster_installation_folder\bin`
- `CD_drive:\Windows\version_number\common\tools\architecture`

- (3) Check that the target group is running on the guest OS, execute the following command on the host OS, and then make sure that the group is migrated

```
clprexec --failover group_name -h  
IP_of_the_guest_OS_on_which_the_group_is_running
```

Appendix A Sample Scripts

The following scripts are required to set up an inter-host OS cluster described in this chapter. If you are using ExpressCluster X 3.0 or later, the following scripts are not necessary because a virtual machine resource is used. Edit the bold text in W#HyperV.bat and genw.bat scripts for use in accordance with your environment.

- vmstart.vbs
- vmstop.vbs
- W#HyperV.bat
- start.bat
- stop.bat
- genw.bat

The following script is required to link guest OSs and host OSs as described in this chapter:

- preaction.bat

vmstart.vbs

A script to start up a virtual machine.

```
'=====
=
' vmstart.vbs
'=====
=
' Title    : Import VM's config file and start the VM
' Date     : 2009/04/07
' Revised  : 2009/10/20
'=====
=

option explicit

dim objWMIService
dim managementService
dim fileSystem
dim objStdOut
const JobStarting = 3
const JobRunning = 4
const JobCompleted = 7
const wmiStarted = 4096
const wmiSuccessful = 0
const Enabled = 2
const Disabled = 3
const Suspended = 32769
' Instance for standard output
Set objStdOut = Wscript.Stdout

Main()
```

```

'-----
---
' Main
'-----
---
Sub Main()

    dim computer, objArgs, vmName, vm, importDirectory,
backupDirectory, _
        vmGUID, vmDefaultRootDirectory
    dim FromPath, DestPath, ConfigXMLPath, BackupPath
    dim objManagementServiceList
    dim retry, conRetryMax, conInterval
    set fileSystem =
Wscript.CreateObject("Scripting.FileSystemObject")
    computer = "."

    set objArgs = WScript.Arguments

    if WScript.Arguments.Count = 6 then
        vmName = objArgs.Unnamed.Item(0)
        importDirectory = objArgs.Unnamed.Item(1)
        backupDirectory = objArgs.Unnamed.Item(2)
        vmGUID = objArgs.Unnamed.Item(3)
        vmDefaultRootDirectory = objArgs.Unnamed.Item(4)
        conRetryMax = CInt(objArgs.Unnamed.Item(5))
    else
objStdOut.WriteLine "usage: cscript vmstart.vbs vmName " & _
        "importDirectoryName backupDirectoryName " & _
        "vmGUID vmDefaultRootDirectoryName " & _
        "vmmsWaitTime(second)"
        WScript.Quit(1)
    end if

    set objWMIService = GetObject("winmgmts:\\\" & computer & _
"\root\virtualization")

    conInterval = 1000
    for retry = 0 To conRetryMax
        set objManagementServiceList = objWMIService.ExecQuery(_
            "select * from Msvm_VirtualSystemManagementService")
        if objManagementServiceList.count = 0 then
            if retry < conRetryMax then
                objStdOut.WriteLine "Error! Virtual Machine
Management " & _
                    "Service (VMMS) is not started, retry..."
            else
                objStdOut.WriteLine "Error! Virtual Machine
Management " & _
                    "Service (VMMS) is not started."
                WScript.Quit(1)
            end If
        else
            objStdOut.WriteLine "Virtual Machine Management
Service " & _
                "(VMMS) has been started."
            Exit For
        end if
        WScript.Sleep(conInterval)
    end for
end sub

```

```

next

set managementService =
objManagementServiceList.ItemIndex(0)
FromPath = backupDirectory & "\" & vmName
DestPath = importDirectory

'check if GuestOS(vmName) exist ,if exist destroy it
if CheckVMIfExist(vmName, importDirectory, vmGUID, _
vmDefaultRootDirectory, backupDirectory) then
    if fileSystem.FolderExists(FromPath) then
        set vm = GetComputerSystem(vmName)
        if DestroyVirtualSystem(vm) then
            objStdOut.WriteLine "Destroy previous vm done."
        else
            objStdOut.WriteLine "DestroyVirtualSystem
Failed."
            WScript.Quit(1)
        end if
        'copy previous vm's config file to importDirectory
        fileSystem.CopyFolder FromPath, DestPath, true
    else
        objStdOut.WriteLine "error! not first time, " & _
"but previous backupd vm's config file doesn't exist!"
        objStdOut.WriteLine "Maybe backup file was deleted by
user, " & _
"please check it."
        WScript.Quit(1)
    end if
end if

ConfigXMLPath = importDirectory & "\" & "config.xml"
if not fileSystem.FileExists(ConfigXMLPath) then
    'copy previous vm's config file to importDirectory
    fileSystem.CopyFolder FromPath, DestPath, true
end if

'edit config.xml
UpdateConfigFile ConfigXMLPath
objStdOut.WriteLine "UpdateConfigFile done."

'clear previous vm that have the same GUID.
ClearVmGuid vmDefaultRootDirectory, vmGUID

'import VM's config file
if ImportVirtualSystem(importDirectory, vmGUID, "false") then
    objStdOut.WriteLine "Import done."
else
    objStdOut.WriteLine "ImportVirtualSystem Failed."
    WScript.Quit(1)
end if

set vm = GetComputerSystem(vmName)

'objStdOut.WriteLine "importDirectory :" & importDirectory
'objStdOut.WriteLine "backupDirectory :" & backupDirectory

BackupPath = backupDirectory & "\" & vmName
'objStdOut.WriteLine "BackupPath :" & BackupPath

```

```

        if not fileSystem.FolderExists(BackupPath) then
            objStdOut.WriteLine "first time start, vm's configuration
data " & _
                "is backuping"
                'Clear VM's Scopes.
                ClearVMScopes(vm)
                'Backup vm's config file
                if BackupVMConfigFile(vm, backupDirectory,
importDirectory) then
                    objStdOut.WriteLine "BackupVMConfigFile done."
                else
                    objStdOut.WriteLine "BackupVMConfigFile failed"
                    WScript.Quit(1)
                end if
            end if

            'start VM
            if RequestStateChange(vm, "start") then
                objStdOut.WriteLine " start done."
            WScript.Quit(0)
            else
                objStdOut.WriteLine "RequestStateChange failed"
                WScript.Quit(1)
            end if

End Sub

'-----
'---
' Edit config.xml, set 'VmStateCopied' to 'true'
'-----
'---
Sub UpdateConfigFile(ConfigFilePath)
    dim objDOM
    dim subNode
    dim rtResult

    Set objDOM = WScript.CreateObject("MSXML2.DOMDocument")

    objDOM.async = False
    ' Load copy source file
    rtResult = objDOM.load(ConfigFilePath)
    If rtResult = False Then
        objStdOut.WriteLine "Failed to load XML file " &
ConfigFilePath & " ."
        Set objDOM = Nothing
        WScript.Quit(1)
    End If

    set subNode =
objDOM.selectSingleNode("/configuration/VmStateCopied" & _
                        "[@type = 'bool']")
    objStdOut.WriteLine Format1("VmStateCopied = {0}",
subNode.text)
    subNode.text = "true"
    objStdOut.WriteLine Format1(" change VmStateCopied to '{0}'",

```

```

-      subNode.text)

      'Save the config.xml file
      objDOM.Save(ConfigFilePath)
      Set objDOM = Nothing

End Sub

'-----
---
' Retrieve Msvm_VirtualComputerSystem from base on its ElementName
'-----
---
Function GetComputerSystem(vmElementName)
    On Error Resume Next
    dim query
    query = Format1("select * from Msvm_ComputerSystem where " &
-           "ElementName = '{0}'", vmElementName)
    set GetComputerSystem =
objWMIService.ExecQuery(query).ItemIndex(0)
    if (Err.Number <> 0) then
        objStdOut.WriteLine Format1("Err.Number: {0}",
Err.Number)
        objStdOut.WriteLine
Format1("Err.Description:{0}",Err.Description)
        WScript.Quit(1)
    end if
End Function

'-----
---
' check if GuestOS vmName exist ,if exist destroy it
'-----
---
Function CheckVMIfExist(vmElementName, importDirectory, vmGUID, _
                        vmDefaultRootDirectory,
backupDirectory)
    'On Error Resume Next
    dim query, query2
    dim vmList, vmList2, vm
    dim count
    dim ConfigXMLPath, vmXMLPath, vmDefaultRootPath, BinFilePath,
VSVFilePath
    dim objFolder, objFiles, objFile

    CheckVMIfExist = false
    count = 0
    ConfigXMLPath = importDirectory & "\" & "config.xml"
    vmXMLPath = importDirectory & "\" & "Virtual Machines" & "\"
& vmGUID & ".xml"
    vmDefaultRootPath = vmDefaultRootDirectory & "\" & "Virtual
Machines" & _
                        "\" & vmGUID & ".xml"
    objStdOut.WriteLine "vmXMLPath: " & vmXMLPath
    'set objFolder=filesystem.GetFolder(vmXMLDirectroy)
    'set objFiles=objFolder.Files
    'objStdOut.WriteLine "objFiles.count =" & objFiles.count

```

```

Do
    query = Format1("select * from Msvm_ComputerSystem where
" & _
        "ElementName = '{0}'", vmElementName)
    set vmList = objWMIService.ExecQuery(query)

    objStdOut.WriteLine "vmList.count =" & vmList.count

    if vmList.count <> 0 then
        if (Err.Number <> 0) then
            objStdOut.WriteLine Format1(_
                "CheckVMIfExist Err.Number: {0}", Err.Number)
            objStdOut.WriteLine Format1(_
                "CheckVMIfExist
Err.Description:{0}",Err.Description)
            WScript.Quit(1)
        end if
        CheckVMIfExist = true
        Exit Do
    elseif (not fileSystem.FileExists(ConfigXMLPath)) and _
        fileSystem.FileExists(vmXMLPath) then
        query2 = Format1("select * from Msvm_ComputerSystem
where " & _
            "ElementName = '{0}'", vmGUID)
        set vmList2 = objWMIService.ExecQuery(query2)
        objStdOut.WriteLine "vmList2.count =" & vmList2.count
        if vmList2.count <> 0 then
            objStdOut.WriteLine "Loop count: " & count & _
                ", per loop sleep 10 seconds."
            WScript.Sleep(10000)
        else
            Exit Do
        end if
    else
        objStdOut.WriteLine "CheckVMIfExist check OK " & _
            "(previous VM doesn't exist)."
        Exit Do
    end if
Loop

    if CheckVMIfExist = false and
fileSystem.FileExists(vmXMLPath) then
        fileSystem.DeleteFile vmXMLPath, true
        if fileSystem.FileExists(vmDefaultRootPath) then
            fileSystem.DeleteFile vmDefaultRootPath, true
        end if

        'delete previous guid.bin and guid.vsv file in backup
folder
        BinFilePath = backupDirectory & "\" & vmElementName & "\"
& _
            "Virtual Machines" & "\" & vmGUID & "\" & vmGUID & ".bin"
VSVFilePath = backupDirectory & "\" & vmElementName & "\"
& _
            "Virtual Machines" & "\" & vmGUID & "\" & vmGUID & ".vsv"
        'objStdOut.WriteLine "BinFilePath : " & BinFilePath
        'objStdOut.WriteLine "VSVFilePath : " & VSVFilePath
        if fileSystem.FileExists(BinFilePath) then
            fileSystem.DeleteFile BinFilePath, true

```



```

        objStdOut.WriteLine "delete previous vm's "& vmGUID &
-
        ".bin file in backupDirectory success."
        end if
        if fileSystem.FileExists(VSVFilePath) then
            fileSystem.DeleteFile VSVFilePath, true
        objStdOut.WriteLine "delete previous vm's "& vmGUID & _
            ".vsv file in backupDirectory success."
        end if

        'delete previous guid.bin and guid.vsv file in import
folder
        BinFilePath = importDirectory & "\" & "Virtual Machines"
& "\" & _
        vmGUID & "\" & vmGUID & ".bin"
        VSVFilePath = importDirectory & "\" & "Virtual Machines"
& "\" & _
        vmGUID & "\" & vmGUID & ".vsv"
        if fileSystem.FileExists(BinFilePath) then
            fileSystem.DeleteFile BinFilePath, true
            objStdOut.WriteLine "delete previous vm's "& vmGUID &
-
            ".bin file in importDirectory success."
        end if
        if fileSystem.FileExists(VSVFilePath) then
            fileSystem.DeleteFile VSVFilePath, true
        objStdOut.WriteLine "delete previous vm's "& vmGUID & _
            ".vsv file in importDirectory success."
        end if
    end if
End Function

'-----
---
' Change status of a virtual machine
'-----
---
Function RequestStateChange(computerSystem, action)
    dim objInParam, objOutParams

    objStdOut.WriteLine Format2("RequestStateChange({0}, {1})",
-
    computerSystem.ElementName, action)

    RequestStateChange = false
    set objInParam =
computerSystem.Methods_("RequestStateChange")_
        .InParameters.SpawnInstance_()

    if action = "Suspended" then
        objInParam.RequestedState = Suspended
    elseif action = "start" then
        objInParam.RequestedState = Enabled
    elseif action = "stop" then
        objInParam.RequestedState = Disabled
    else
        objStdOut.WriteLine Format1("change state to {0} is not
support!", _

```

```

        action)
        WScript.Quit(1)
    end if

    set objOutParams =
computerSystem.ExecMethod_("RequestStateChange", _
                            objInParam)

    if (WMIMethodStarted(objOutParams)) then
        if (WMIJobCompleted(objOutParams)) then
            objStdOut.WriteLine Format2("VM {0} was {1}
successfully", _
computerSystem.ElementName, action)
            RequestStateChange = true
        end if
    end if

End Function

'-----
---
' Define a virtual system
'-----
---
Function ImportVirtualSystem(importDirectory, vmGUID,
generateNewID)

    dim objInParam, objOutParams, vmXMLPath, binPath, vsvPath
    ImportVirtualSystem = false

    'if xml, bin, vsv file exists, delete it to import
    vmXMLPath = importDirectory & "\" & "Virtual Machines" & "\"
& vmGUID & ".xml"
    if fileSystem.FileExists(vmXMLPath) then
        objStdOut.WriteLine vmXMLPath & "exists."
        fileSystem.DeleteFile vmXMLPath, true

        ' When xml file exists, VM was terminated by not responding
to cluster.
        ' So, remove bin and vsv file to start VM without bin nor
vsv file.
        binPath = importDirectory & "\" & "Virtual Machines" & "\"
& _
            vmGUID & "\" & vmGUID & "\" & ".bin"
        if fileSystem.FileExists(binPath) then
            objStdOut.WriteLine binPath & "exists."
            fileSystem.DeleteFile binPath, true
        end if
        vsvPath = importDirectory & "\" & "Virtual Machines" & "\"
& _
            vmGUID & "\" & vmGUID & "\" & ".vsv"
        if fileSystem.FileExists(vsvPath) then
            objStdOut.WriteLine vsvPath & "exists."
            fileSystem.DeleteFile vsvPath, true
        end if
    end if

    if Not fileSystem.FolderExists(importDirectory) then
        objStdOut.WriteLine Format1("importDirectory({0})
doesn't exists.", _

```

```

        importDirectory)
    end if

    set objInParam =
managementService.Methods_("ImportVirtualSystem")._
        InParameters.SpawnInstance_()
    objInParam.GenerateNewID = generateNewID
    objInParam.ImportDirectory = importDirectory

    set objOutParams = managementService.ExecMethod_(
        "ImportVirtualSystem", objInParam)

    if objOutParams.ReturnValue = wmiStarted then
        if (WMIJobCompleted(objOutParams)) then
            ImportVirtualSystem = true
        end if
    elseif (objOutParams.ReturnValue = wmiSuccessful) then
        ImportVirtualSystem = true
    else
        objStdOut.WriteLine Format1("DefineVirtualSystem failed
with " & _
        "ReturnValue {0}", wmiStatus)
    end if

End Function

'-----
---
' Destroy a virtual system
'-----
---
Function DestroyVirtualSystem(computerSystem)

    dim objInParam, objOutParams

    DestroyVirtualSystem = false
    set objInParam =
managementService.Methods_("DestroyVirtualSystem")._
        InParameters.SpawnInstance_()
    objInParam.ComputerSystem = computerSystem.Path_.Path

    set objOutParams = managementService.ExecMethod_(
        "DestroyVirtualSystem", objInParam)

    if (objOutParams.ReturnValue = wmiStarted) then
        if (WMIJobCompleted(objOutParams)) then
            DestroyVirtualSystem = true
        end if
    elseif (objOutParams.ReturnValue = wmiSuccessful) then
        DestroyVirtualSystem = true
    else
        objStdOut.WriteLine Format1("DestroyVirtualSystem failed
with " & _
        "ReturnValue {0}", objOutParams.ReturnValue)
    end if

End Function

'-----

```

```

---
' Export a virtual system
'-----
---
Function ExportVirtualSystem(computerSystem, exportDirectory)
    dim vmPath, vmTmpPath, tmpExportDirectory
    dim objInParam, objOutParams

    ExportVirtualSystem = false
    tmpExportDirectory = exportDirectory & "\" & "tmp"
    vmPath = exportDirectory & "\" & computerSystem.ElementName
    vmTmpPath = tmpExportDirectory & "\" &
computerSystem.ElementName

    if Not FileSystem.FolderExists(exportDirectory) then
        FileSystem.CreateFolder(exportDirectory)
        objStdOut.WriteLine "Warn, previous backedup vm's config
file " & _
        "doesn't exist!"
    else
        if FileSystem.FolderExists(vmTmpPath) then
            FileSystem.DeleteFolder vmTmpPath, true
        end if

    end if

    set objInParam =
managementService.Methods_("ExportVirtualSystem")._
InParameters.SpawnInstance_(
objInParam.ComputerSystem = computerSystem.Path_.Path
objInParam.CopyVmState = false
objInParam.ExportDirectory = tmpExportDirectory

    set objOutParams = managementService.ExecMethod_(
"ExportVirtualSystem", objInParam)

    if objOutParams.ReturnValue = wmiStarted then
        if (WMIJobCompleted(objOutParams)) then
            ExportVirtualSystem = true
            objStdOut.WriteLine "vmPath: " & vmPath
            if FileSystem.FolderExists(vmPath) then
                FileSystem.DeleteFolder vmPath, true
            end if
            FileSystem.MoveFolder vmTmpPath, vmPath
        end if
    elseif (objOutParams.ReturnValue = wmiSuccessful) then
        ExportVirtualSystem = true

        if FileSystem.FolderExists(vmPath) then
            FileSystem.DeleteFolder vmPath, true
        end if
        FileSystem.MoveFolder vmTmpPath, vmPath
    else
        objStdOut.WriteLine Format1("DefineVirtualSystem failed
with " & _
        "ReturnValue {0}", wmiStatus)
    end if

End Function

```

```

'-----
---
' Backup a virtual system's config file
'-----
---
Function BackupVMConfigFile(computerSystem, exportDirectory,
vmrootDirectory)
    Dim tmpExportDirectory
    Dim FromPath, DestPath, BinFilePath

    BackupVMConfigFile = false

    'export VM's config file
    if ExportVirtualSystem(computerSystem, exportDirectory) then
        objStdOut.WriteLine "Export done."
    else
        objStdOut.WriteLine "ExportVirtualSystem Failed."
        Exit Function
    end if

    'move status file to temp folder

    FromPath = vmrootDirectory & "\" & "Virtual Machines" & "\" &
-
        computerSystem.Name & "\" & "*"
    DestPath = exportDirectory & "\" & computerSystem.ElementName
& "\" & -
        "Virtual Machines" & "\" & computerSystem.Name
    BinFilePath = vmrootDirectory & "\" & "Virtual Machines" & "\"
& -
        computerSystem.Name & "\" & computerSystem.Name
& ".bin"

    if fileSystem.FileExists(BinFilePath) then
        fileSystem.CopyFile FromPath, DestPath, true
        objStdOut.WriteLine "move status file done"
    else
        objStdOut.WriteLine "vm's status file doesn't exist."
    end if

    BackupVMConfigFile = true

End Function

'-----
---
' Handle wmi return values
'-----
---
Function WMIMethodStarted(outParam)

    dim wmiStatus

    WMIMethodStarted = false

    if Not IsNull(outParam) then
        wmiStatus = outParam.ReturnValue

        if wmiStatus = wmiStarted then

```

```

        WMIMethodStarted = true
    end if

    end if

End Function

'-----
'---
' Handle wmi Job object
'-----
'---
Function WMIJobCompleted(outParam)

    dim WMIJob, jobState

    set WMIJob = objWMIService.Get(outParam.Job)

    WMIJobCompleted = true

    jobState = WMIJob.JobState

    while jobState = JobRunning or jobState = JobStarting
        objStdOut.WriteLine Format1("In progress... {0}%%
completed.", _
        WMIJob.PercentComplete)
        WScript.Sleep(1000)
        set WMIJob = objWMIService.Get(outParam.Job)
        jobState = WMIJob.JobState
    wend

    if (jobState <> JobCompleted) then
        objStdOut.WriteLine Format1("ErrorCode:{0}",
WMIJob.ErrorCode)
        objStdOut.WriteLine Format1("ErrorDescription:{0}", _
        WMIJob.ErrorDescription)
        WMIJobCompleted = false
    end if

End Function

'-----
'---
' The string formating functions to avoid string concatenation.
'-----
'---
Function Format2(myString, arg0, arg1)
    Format2 = Format1(myString, arg0)
    Format2 = Replace(Format2, "{1}", arg1)
End Function

'-----
'---
' The string formating functions to avoid string concatenation.
'-----
'---
Function Format1(myString, arg0)
    Format1 = Replace(myString, "{0}", arg0)
End Function

```

```

'-----
---
' clear vm's ScopeOfResidence value
' vm import failer be due to SVCMM installed.
'-----
---
Function ClearVMScopes(computerSystem)

    Dim VMSystemGlobalSettingData
    Dim Result

    Set VMSystemGlobalSettingData = _
        (computerSystem.Associators_("MSVM_ElementSettingData",
-
        "Msvm_VirtualSystemGlobalSettingData")).ItemIndex(0)
    VMSystemGlobalSettingData.ScopeOfResidence = ""
    Result = _

managementService.ModifyVirtualSystem(computerSystem.Path_.Path
' - VMSystemGlobalSettingData.GetText_(1))

    objStdOut.WriteLine "ClearVMScopes Result:" & Result
    if Result <> 0 then
        objStdOut.WriteLine "ClearVMScopes Error : " & Result
    end if

End Function

'-----
---
' clear previous vm that have the same GUID.
'-----
---
Sub ClearVmGuid(vmDefaultRootDirectory, vmGUID)
    Dim vmDefaultRootPath
    vmDefaultRootPath = vmDefaultRootDirectory & "\" & "Virtual
Machines" & _
        "\" & vmGUID & ".xml"

    if (fileSystem.FileExists(vmDefaultRootPath)) then
        fileSystem.DeleteFile vmDefaultRootPath, true
        objStdOut.WriteLine "previous vm that have the same GUID
is deleted."
    end if

End Sub

```

vmstop.vbs

A script to save a virtual machine.

```

'=====
=
' vmstop.vbs
'=====
=
' Title : Save VM's state, export VM's config file then destroy
the VM

```

```
' Date      : 2009/04/10
' Revised   : 2009/10/20
'=====
=

option explicit

dim objWMIService
dim managementService
dim fileSystem
dim objStdOut
const JobStarting = 3
const JobRunning = 4
const JobCompleted = 7
const wmiStarted = 4096
const wmiSuccessful = 0
const Enabled = 2
const Disabled = 3
const Suspended = 32769
' Instance for standard output
Set objStdOut = Wscript.StdOut

Main()

'-----
---
' Main
'-----
---
Sub Main()

    dim computer, objArgs, vmName, vm, backupDirectory,
vmrootDirectory, _
    OffAction
    dim DestPath, FromPath

    set fileSystem =
Wscript.CreateObject("Scripting.FileSystemObject")
    computer = "."
    set objWMIService = GetObject("winmgmts:\\." & computer & _
"\root\virtualization")
    set managementService = objWMIService.ExecQuery("select *
from " & _
"Msvm_VirtualSystemManagementService").ItemIndex(0)

    set objArgs = WScript.Arguments
    if WScript.Arguments.Count = 4 then
        vmName = objArgs.Unnamed.Item(0)
        backupDirectory = objArgs.Unnamed.Item(1)
        vmrootDirectory = objArgs.Unnamed.Item(2)
        OffAction = objArgs.Unnamed.Item(3)
    else
        objStdOut.WriteLine "usage: cscript vmstop.vbs vmName " &
-
        "backupDirectoryName vmrootDirectoryName"
        WScript.Quit(1)
    end if

    set vm = GetComputerSystem(vmName)
```



```

objStdOut.WriteLine "GetComputerSystem complete"

if IsVMStarted(vm) then
    if OffAction = "shutdown" then
        if RequestShutdown(vm, vmName) then
            objStdOut.WriteLine "RequestShutdown success."
        else
            objStdOut.WriteLine "RequestShutdown failed."
        end if
    else
        'change VM's status
        if RequestStateChange(vm, OffAction) then
            objStdOut.WriteLine "Save done."
        else
            objStdOut.WriteLine "RequestStateChange failed"
            WScript.Quit(1)
        end if
    end if
else
    objStdOut.WriteLine "warn! vm isn't started, " & _
    Don't backup vm then destroy it."
end if

'Clear VM's Scopes.
ClearVMScopes(vm)

'Backup vm's config file
if BackupVMConfigFile(vm, backupDirectory, vmrootDirectory)
then
    objStdOut.WriteLine "BackupVMConfigFile done."
else
    objStdOut.WriteLine "BackupVMConfigFile failed"
    WScript.Quit(1)
end if

'destroy the VM
if DestroyVirtualSystem(vm) then
    objStdOut.WriteLine "Destroy done."
    'WScript.Quit(0)
else
    objStdOut.WriteLine "DestroyVirtualSystem Failed."
    WScript.Quit(1)

end if

'copy exported vm's config file to up folder
FromPath = backupDirectory & "\" & vm.ElementName
DestPath = vmrootDirectory
fileSystem.CopyFolder FromPath, DestPath, true
'fileSystem.DeleteFolder backupDirectory, true
objStdOut.WriteLine "copy exported vm's config file done"

End Sub

'-----
---
' Retrieve Msvm_VirtualComputerSystem from base on its ElementName
'-----
---
```

```

Function GetComputerSystem(vmElementName)
    On Error Resume Next
    dim query
    query = Format1("select * from Msvm_ComputerSystem where " &
        "ElementName = '{0}'", vmElementName)
    set GetComputerSystem =
objWMIService.ExecQuery(query).ItemIndex(0)
    if (Err.Number <> 0) then
        objStdOut.WriteLine Format1("Err.Number: {0}",
Err.Number)
        objStdOut.WriteLine
Format1("Err.Description:{0}",Err.Description)
        WScript.Quit(1)
    end if
End Function

'-----
---
' Change status of a virtual machine
'-----
---
Function RequestStateChange(computerSystem, action)
    dim objInParam, objOutParams

    objStdOut.WriteLine Format2("RequestStateChange({0}, {1})",
        computerSystem.ElementName, action)

    RequestStateChange = false
    set objInParam =
computerSystem.Methods_("RequestStateChange")._
        InParameters.SpawnInstance_()

    if action = "suspended" then
        objInParam.RequestedState = Suspended
    elseif action = "start" then
        objInParam.RequestedState = Enabled
    elseif action = "stop" then
        objInParam.RequestedState = Disabled
    else
        objStdOut.WriteLine Format1("change state to {0} is not
support!", _
            action)
        WScript.Quit(1)
    end if

    set objOutParams =
computerSystem.ExecMethod_("RequestStateChange", _
        objInParam)

    if (WMIMethodStarted(objOutParams)) then
        if (WMIJobCompleted(objOutParams)) then
            objStdOut.WriteLine Format2("VM {0} was {1}
successfully", _
                computerSystem.ElementName, action)
            RequestStateChange = true
        end if
    end if
end if

```

```

End Function

'-----
---
' Export a virtual system
'-----
---
Function ExportVirtualSystem(computerSystem, exportDirectory)
    dim vmPath, vmTmpPath, tmpExportDirectory
    dim objInParam, objOutParams

    ExportVirtualSystem = false
    tmpExportDirectory = exportDirectory & "\" & "tmp"
    vmPath = exportDirectory & "\" & computerSystem.ElementName
    vmTmpPath = tmpExportDirectory & "\" &
computerSystem.ElementName

    if Not fileSystem.FolderExists(exportDirectory) then
        fileSystem.CreateFolder(exportDirectory)
        objStdOut.WriteLine "Warn, previous backed up vm's config
file " & _
        "doesn't exist!"
    else
        if fileSystem.FolderExists(vmTmpPath) then
            fileSystem.DeleteFolder vmTmpPath, true
        end if

    end if

    set objInParam =
managementService.Methods_("ExportVirtualSystem")._
        InParameters.SpawnInstance_()
    objInParam.ComputerSystem = computerSystem.Path_.Path
    objInParam.CopyVmState = false
    objInParam.ExportDirectory = tmpExportDirectory

    set objOutParams =
managementService.ExecMethod_("ExportVirtualSystem", _
        objInParam)

    if objOutParams.ReturnValue = wmiStarted then
        if (WMIJobCompleted(objOutParams)) then
            ExportVirtualSystem = true

            if fileSystem.FolderExists(vmPath) then
                fileSystem.DeleteFolder vmPath, true
            end if
            fileSystem.MoveFolder vmTmpPath, vmPath
        end if
    elseif (objOutParams.ReturnValue = wmiSuccessful) then
        ExportVirtualSystem = true

        if fileSystem.FolderExists(vmPath) then
            fileSystem.DeleteFolder vmPath, true
        end if
        fileSystem.MoveFolder vmTmpPath, vmPath
    else
        objStdOut.WriteLine Format1("DefineVirtualSystem failed
with " & _

```

```

        "ReturnValue {0}", objOutParams.ReturnValue)
    end if

End Function

'-----
'---
' Backup a virtual system's config file
'-----
'---
Function BackupVMConfigFile(computerSystem, exportDirectory,
vmrootDirectory)
    Dim tmpExportDirectory
    Dim FromPath, DestPath, BinFilePath

    BackupVMConfigFile = false

    'export VM's config file
    if ExportVirtualSystem(computerSystem, exportDirectory) then
        objStdOut.WriteLine "Export done."
    else
        objStdOut.WriteLine "ExportVirtualSystem Failed."
        Exit Function
    end if

    'move status file to temp folder

    FromPath = vmrootDirectory & "\" & "Virtual Machines" & "\" &
-
        computerSystem.Name & "\" & "*"
    DestPath = exportDirectory & "\" & computerSystem.ElementName
& "\" & -
        "Virtual Machines" & "\" & computerSystem.Name
    BinFilePath = vmrootDirectory & "\" & "Virtual Machines" & "\"
& -
        computerSystem.Name & "\" & computerSystem.Name
& ".bin"
    objStdOut.WriteLine "FromPath: " & FromPath
    objStdOut.WriteLine "DestPath : " & DestPath
    objStdOut.WriteLine "BinFilePath: " & BinFilePath
    if fileSystem.FileExists(BinFilePath) then
        if not fileSystem.FolderExists(DestPath) then
            fileSystem.CreateFolder(DestPath)
        end if
        fileSystem.CopyFile FromPath, DestPath, true
        objStdOut.WriteLine "move status file done"
    else
        objStdOut.WriteLine "vm's status file doesn't exists."
    end if

    BackupVMConfigFile = true

End Function

'-----
'---
' Destroy a virtual system
'-----
'---
Function DestroyVirtualSystem(computerSystem)

```

```

dim objInParam, objOutParams

DestroyVirtualSystem = false
set objInParam = _
    managementService.Methods_("DestroyVirtualSystem")._
    InParameters.SpawnInstance_()
objInParam.ComputerSystem = computerSystem.Path_.Path

set objOutParams = _
    managementService.ExecMethod_("DestroyVirtualSystem",
objInParam)

if (objOutParams.ReturnValue = wmiStarted) then
    if (WMIJobCompleted(objOutParams)) then
        DestroyVirtualSystem = true
    end if
elseif (objOutParams.ReturnValue = wmiSuccessful) then
    DestroyVirtualSystem = true
else
    objStdOut.WriteLine Format1("DestroyVirtualSystem failed
with " & _
    "ReturnValue {0}", objOutParams.ReturnValue)
end if

End Function

'-----
---
' Handle wmi return values
'-----
---
Function WMIMethodStarted(outParam)

    dim wmiStatus

    WMIMethodStarted = false

    if Not IsNull(outParam) then
        wmiStatus = outParam.ReturnValue

        if wmiStatus = wmiStarted then
            WMIMethodStarted = true
        end if

    end if

End Function

'-----
---
' Handle wmi Job object
'-----
---
Function WMIJobCompleted(outParam)

    dim WMIJob, jobState

    set WMIJob = objWMIService.Get(outParam.Job)

```

```

    WMIJobCompleted = true

    jobState = WMIJob.JobState

    while jobState = JobRunning or jobState = JobStarting
        objStdOut.WriteLine Format1("In progress... {0}%%
completed.", _
    WMIJob.PercentComplete)
        WScript.Sleep(1000)
        set WMIJob = objWMIService.Get(outParam.Job)
        jobState = WMIJob.JobState
    wend

    if (jobState <> JobCompleted) then
        objStdOut.WriteLine Format1("ErrorCode:{0}", _
WMIJob.ErrorCode)
        objStdOut.WriteLine Format1("ErrorDescription:{0}", _
    WMIJob.ErrorDescription)
        WMIJobCompleted = false
    end if

End Function

'-----
'---
' The string formatting functions to avoid string concatenation.
'-----
'---
Function Format2(myString, arg0, arg1)
    Format2 = Format1(myString, arg0)
    Format2 = Replace(Format2, "{1}", arg1)
End Function

'-----
'---
' The string formatting functions to avoid string concatenation.
'-----
'---
Function Format1(myString, arg0)
    Format1 = Replace(myString, "{0}", arg0)
End Function

'-----
'---
' Check VM if is started
'-----
'---
Function IsVMStarted(computerSystem)

    if computerSystem.EnabledState <> Enabled then
        IsVMStarted = false
    else
        IsVMStarted = true
    end if

End Function

'-----
'---

```

```

' clear vm's ScopeOfResidence value
' vm import failer be due to SVCMM installed.
'-----
---
Function ClearVMScopes(computerSystem)

    Dim VMSystemGlobalSettingData
    Dim Result

    Set VMSystemGlobalSettingData = _
        (computerSystem.Associators_("MSVM_ElementSettingData",
-
        "Msvm_VirtualSystemGlobalSettingData")).ItemIndex(0)
    VMSystemGlobalSettingData.ScopeOfResidence = ""
    Result = _

managementService.ModifyVirtualSystem(computerSystem.Path_.Path
' - VMSystemGlobalSettingData.GetText_(1))

    objStdOut.WriteLine "ClearVMScopes Result:" & Result
    if Result <> 0 then
        objStdOut.WriteLine "ClearVMScopes Error : " & Result
    end if

End Function

'-----
---
' Shutdown vm
'-----
---
Function RequestShutdown(vm, vmName)
    Dim vmShut
    Dim vmReturn
    Dim objWMI
    Dim vmList
    Dim vmState

    Set vmshut = objWMIService.ExecQuery(_
        "SELECT * FROM Msvm_ShutdownComponent WHERE SystemName='" &
vm.Name & "'")
    vmReturn =
vmShut.ItemIndex(0).InitiateShutdown(True, "Scripted Shutdown")
    if vmReturn = 0 then
        RequestShutdown = true
    else
        RequestShutdown = false
    end if

    Do
        Set objWMI =
GetObject("winmgmts:\\.\root\virtualization")
        Set vmList = objWMI.ExecQuery(_
            "SELECT * FROM Msvm_ComputerSystem Where ElementName='" &
vmName & "'")
        vmState = vmList.ItemIndex(0).EnabledState
        ' For debug
        ' objStdOut.WriteLine "Status: " & vmState

```

```

        if vmState = 3 then
            Exit Do
        end if
        WScript.Sleep(1000)
    Loop
End Function

```

W#HyperV.bat

A parameter file for `vmstart.vbs` and `vmstop.vbs`. Edit the bold text for use as necessary.

```

rem *****
rem *           W#HyperV.BAT           *
rem *           *                       *
rem * Title    : Hyper-V Parameters    *
rem * Date     : 2008/04/14            *
rem * Revised  : 2009/10/16            *
rem *           *                       *
rem *****

rem -----
rem Environment variable list set in W#HyperV.BAT
rem W#HyperV1 : Virtual machine name
rem W#HyperV2 : Directory name where virtual machine configuration
rem file is saved
rem W#HyperV3 : Backup directory name of virtual machine
rem W#HyperV4 : GUID name of virtual machine
rem W#HyperV5 : Default directory name where virtual machine
rem configuration file is saved
rem W#HyperV6 : Startup retry count for virtual machine
rem W#HyperV7 : Disk disconnection error avoidance waiting time
rem [sec]
rem W#HyperV8 : Action while shutdown
rem             suspended: Save startup status of the guest OS
rem             shutdown  : Shut down the guest OS
rem -----

SET W#HyperV1=w2k3-x64jp-vm
SET W#HyperV2="Z:\Hyper-V\w2k3-x64jp-vm"
SET W#HyperV3="Z:\Hyper-V\w2k3-x64jp-vm\backup"
SET W#HyperV4=1DE5E97B-F1C7-4699-96C2-40FAAE271ECA
SET W#HyperV5="C:\ProgramData\Microsoft\Windows\Hyper-V"
SET W#HyperV6=30
SET W#HyperV7=30
SET W#HyperV8="suspended"

```

checkstatus.vbs

A script to check virtual machine startup status.

```

'=====
'
' Title    : Check virtual machine status
' Date     : 2009/04/07
' Revised  : 2009/10/20
'=====
'
' VM State
'       0: Unknown

```



```
'      2: Enabled (Running)
'      3: Disabled
'     10: Rebooted
'     11: Reset
' 32768: Paused
' 32769: Suspended (Save completed)

Option Explicit

'-----
-
' Variables
'-----
-
Public objWMI
Public VM
Public VMList
Public VMName
Public VMEnabledState

Main()

'-----
-
' Main()
'-----
-
Sub Main()
    dim objArgs

    set objArgs = WScript.Arguments

    VMName = objArgs.Unnamed.Item(0)

    checkVMState()
    For Each VM In VMList
        VMEnabledState = VMList.ItemIndex(0).EnabledState

        WScript.Stdout.WriteLine VMEnabledState

        If VMEnabledState=0 then
            WScript.Stdout.WriteLine "Unknown"
            WScript.Quit (1)
        ElseIf VMEnabledState=3 then
            WScript.Stdout.WriteLine "Disabled"
            WScript.Quit (2)
        Else
            WScript.Quit (0)
        End If
    Next
End Sub

'-----
-
' checkVMState()
'-----
-
```

```
Sub checkVMState()
  set objWMI = GetObject("winmgmts:\\.\root\virtualization")
  set VMList = objWMI.ExecQuery(
    "SELECT * FROM Msvm_ComputerSystem Where ElementName='" & VMName &
    & "'")
  if VMList.count = 0 then
    WScript.Stdout.WriteLine "There is no such VM '" & VMName &
    "'" in this system."
    WScript.Quit (3)
  end if
End Sub
```

start.bat

A script to start up vmstart.vbs.

```
rem *****
rem *                start.bat                *
rem *                *                        *
rem * Title   : Start script file            *
rem * Date    : 2009/04/07                   *
rem * Revised : 2009/10/20                   *
rem *****

rem -----
rem Environment variable settings used for batch
rem -----

CALL W#HyperV.BAT

rem *****
rem Startup factor check
rem *****
IF "%CLP_EVENT%" == "START" GOTO NORMAL
IF "%CLP_EVENT%" == "FAILOVER" GOTO FAILOVER
IF "%CLP_EVENT%" == "RECOVER" GOTO RECOVER

rem ExpressCluster Server not running
GOTO no_arm

rem *****
rem Normal startup response process
rem *****
:NORMAL

rem Disk check
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK

rem Move to script path
cd %CLP_SCRIPT_PATH%

rem *****
rem Normal business process
rem *****
cscript
```

```

vmstart.vbs %W#HyperV1% %W#HyperV2% %W#HyperV3% %W#HyperV4% %W#
HyperV5% %W#HyperV6%

rem Priority check
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *****
rem Process at top priority
rem (Ex.) ARMBCAST /MSG "Starting on top priority server" /A
rem *****
GOTO EXIT

:ON_OTHER1
rem *****
rem Process at other than top priority
rem (Ex.) ARMBCAST /MSG "Starting on a server other than top
priority server" /A
rem *****
GOTO EXIT

rem *****
rem Recovery response process
rem *****
:RECOVER

rem *****
rem Recovery process after cluster restored
rem *****

GOTO EXIT

rem *****
rem Failover response process
rem *****
:FAILOVER

rem Disk check
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK

rem Move to script path
cd %CLP_SCRIPT_PATH%

rem *****
rem Business application startup and recovery process after
failover
rem *****
cscript
vmstart.vbs %W#HyperV1% %W#HyperV2% %W#HyperV3% %W#HyperV4% %W#
HyperV5% %W#HyperV6%

rem Priority check

```

```

IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER2

rem *****
rem Process at top priority
rem (Ex.) ARMBICAST /MSG "Starting on top priority server (after
failover)" /A
rem *****
GOTO EXIT

:ON_OTHER2
rem *****
rem Process at other than top priority
rem (Ex.) ARMBICAST /MSG "Starting on a server other than top
priority server (after failover)" /A
rem *****
GOTO EXIT

rem *****
rem Exception process
rem *****

rem Disk related error process
:ERROR_DISK
ARMBICAST /MSG "Connecting switching partition failed" /A
GOTO EXIT

rem ARM not running
:no_arm
ARMBICAST /MSG "ExpressCluster Server is not running" /A

:EXIT

```

stop.bat

A script to start up vmstop.vbs.

```

rem *****
rem *           stop.bat           *
rem *           *                   *
rem * Title    : Stop script file   *
rem * Date     : 2009/04/07         *
rem * Revised  : 2009/10/16         *
rem *           *                   *
rem *****

rem -----
rem Environment variable settings used for batch
rem -----

CALL W#HyperV.BAT

rem *****
rem Startup factor check
rem *****
IF "%CLP_EVENT%" == "START" GOTO NORMAL
IF "%CLP_EVENT%" == "FAILOVER" GOTO FAILOVER

```

```

rem ExpressCluster Server not running
GOTO no_arm

rem *****
rem Normal end response process
rem *****
:NORMAL

rem Disk check
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK

rem Move to script path
cd %CLP_SCRIPT_PATH%

rem *****
rem Normal business process
rem *****
cscript
vmstop.vbs %W#HyperV1% %W#HyperV3% %W#HyperV2% %W#HyperV8%

rem To avoid error when disconnecting disk
armsleep %W#HyperV7%

rem Priority check
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *****
rem Process at top priority
rem (Ex.) ARMBCAST /MSG "Shutting down on top priority server" /A
rem *****
GOTO EXIT

:ON_OTHER1
rem *****
rem Process at other than top priority
rem (Ex.) ARMBCAST /MSG "Shutting down on a server other than top
priority server" /A
rem *****
GOTO EXIT

rem *****
rem Failover response process
rem *****
:FAILOVER

rem Disk check
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK

rem Move to script path

```

```

cd %CLP_SCRIPT_PATH%

rem *****
rem Business application startup and recovery process after
failover
rem *****
cscript
vmstop.vbs %W#HyperV1% %W#HyperV3% %W#HyperV2% %W#HyperV8%

rem To avoid error when disconnecting disk
armsleep %W#HyperV7%

rem Priority check
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER2

rem *****
rem Process at top priority
rem (Ex.) ARMBCAST /MSG "Shutting down on top priority server (after
failover)" /A
rem *****
GOTO EXIT

:ON_OTHER2
rem *****
rem Process at other than top priority
rem (Ex.) ARMBCAST /MSG "Shutting down on a server other than top
priority server (after failover)" /A
rem *****
GOTO EXIT

rem *****
rem Exception process
rem *****

rem Disk related error process
:ERROR_DISK
ARMBCAST /MSG "Connecting switching partition failed" /A
GOTO EXIT

rem ARM not running
:no_arm
ARMBCAST /MSG "ExpressCluster Server is not running" /A

:EXIT

```

genw.bat

A script to start up checkstatus.vbs. Edit the bold text for use as necessary.

```

rem *****
rem *           genw.bat           *
rem * Title    : Start up checkstatus.vbs *
rem * Date     : 2009/04/07          *
rem * Revised  : 2009/04/07          *
rem *****

```

```

set vmname=w2k8-x86jp-vm
set vbs_path=C:\Program Files\ExpressCluster\bin\checkstatus.vbs

echo START

cscript "%vbs_path%" %vmname%
if errorlevel 1 goto ERROR

:NORMAL
exit 0

:ERROR
exit 1

```

preaction.bat

This script is used to send a failover request to another cluster (for example, from a guest OS cluster to a host OS cluster) before executing the final action upon error detection by a monitor resource. Edit the **bold** text in accordance with your environment.

The `clptrnreq` command is used for request transmission. For an explanation of the use of the `clptrnreq` command, see the *ExpressCluster X Reference Guide*.

```

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

echo START
echo %CLP_MONITORNAME%

rem Specify the name of the resource to be failed over.
rem ex. set CLPRSC=script
set CLPRSC=script-vm

rem Specify each server IP address, separated by commas.
rem ex. set CLPIP=10.0.0.1,10.0.0.2
set CLPIP=192.168.0.1,192.168.0.2

rem Send a failover request.
clptrnreq -t GRP_FAILOVER -r %CLPRSC% -h %CLPIP%

echo EXIT

```

preaction.bat (ExpressCluster X 3.0 or later)

This script is used to send a failover request to another cluster (for example, from a guest OS cluster to a host OS cluster) before executing the final action upon error detection by a monitor resource. Edit the **bold text** in accordance with your environment.

When you are using ExpressCluster X 3.0 or later, you can use the `clptrnreq` or `clprexec` command to send a request. The following is a sample script using the `clprexec` command.

```

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

echo START
echo %CLP_MONITORNAME%

```

Appendix A Sample Scripts

```
rem Specify the failover group name.
rem ex. set CLPGRP=failover
set CLPGRP=failover

rem Specify each server IP address, separated by commas.
rem ex. set CLPIP=10.0.0.1,10.0.0.2
set CLPIP=192.168.0.1,192.168.0.2

rem Send a failover request
clprexec --failover %CLPGRP% -h %CLPIP%

echo EXIT
```